ARM Cortex™-M0

32-bit Microcontroller

NuMicro™ Famly
Mini51 DE 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 10
2 FEATURES 11
3 ABBREVIATIONS 15
4 PARTS INFORMATION LIST AND PIN CONFIGURATION 16
4.1 NuMicro™ Mini51 Series Selection Code 16
4.2 NuMicro™ Mini51 Series Product Selection Guide 17
4.3 PIN CONFIGURATION 18
4.3.1 LQFP 48-pin 18
4.3.2 QFN 33-pin 19
4.3.3 TSSOP 20-pin 20
4.3.4 Mini54FHC (TSSOP20-pin) 20
4.4 Pin Description 21
5 BLOCK DIAGRAM 25
5.1 NuMicro™ Mini51 Block Diagram 25
6 Functional Description 26
6.1 ARM® Cortex™-M0 core 26
6.2 System Manager 28
6.2.1 Overview 28
6.2.2 System Reset 28
6.2.3 System Power Architecture 28
6.2.4 Whole System Memory Mapping 30
6.2.5 Register Protection 31
6.2.6 System Memory Map 32
6.2.7 Register Map 34
6.2.8 Register Description 35
6.2.9 System Timer (SysTick) 59
6.2.10 Nested Vectored Interrupt Controller (NVIC) 64
6.2.11 System Control Registers (SCB) 87
6.3 Clock Controller 95
6.3.1 Overview 95
6.3.2 System Clock and SysTick Clock 96
6.3.3 ISP Clock Source Selection 97
6.3.4 Module Clock Source Selection 97

6.3.5 Power-down Mode Clock 98
6.3.6 Frequency Divider Output 98
6.3.7 Register Map 100
6.3.8 Register Description 101
6.4 Flash Memory Controller (FMC) 114
6.4.1 Overview 114
6.4.2 Features 114
6.4.3 Block Diagram 115
6.4.4 Flash Memory Organization 115
6.4.5 User Configuration 118
6.4.6 *Boot* Selection 123
6.4.7 In-Application-Programming (IAP) 124
6.4.8 In System Programming (ISP) 125
6.4.9 ISP Registers Control Procedure 126
6.4.10 Multi-booting by Vector Remapping 128
6.4.11 Flash Control Register Map 130
6.4.12 Flash Control Register Description 131
6.5 General Purpose I/O (GPIO) 139
6.5.1 Overview 139
6.5.2 Features 139
6.5.3 Basic Configuration 139
6.5.4 Functional Description 139
6.5.5 GPIO Interrupt and Wake-up Function 141
6.5.6 Register Map 142
6.5.7 Register Description 146
6.6 Timer Controller (TMR) 160
6.6.1 Overview 160
6.6.2 Features 160
6.6.3 Block Diagram 160
6.6.4 Basic Configuration 161
6.6.5 Functional Description 162
6.6.6 Register Map 166
6.6.7 Register Description 167
6.7 Enhanced PWM Generator 177
6.7.1 Overview 177
6.7.2 Features 177

6.7.3 Block Diagram 179
6.7.4 Basic Configuration 181
6.7.5 Functional Description 181
6.7.6 PWM for Motor Control Interrupt Architecture 191
6.7.7 PWM Timer Start Procedure 192
6.7.8 PWM Timer Stop Procedure 193
6.7.9 Register Map 194
6.7.10 Register Description 195
6.8 Watchdog Timer (WDT) 230
6.8.1 Overview 230
6.8.2 Features 230
6.8.3 Block Diagram 230
6.8.4 Basic Configuration 231
6.8.5 Functional Description 231
6.8.6 Registers Map 233
6.8.7 Register Description 234
6.9 UART Controller (UART) 236
6.9.1 Overview 236
6.9.2 Features 236
6.9.3 Block Diagram 237
6.9.4 Basic Configuration 239
6.9.5 Functional Description 240
6.9.6 Register Map 250
6.9.7 Register Description 251
6.10 I^2C Serial Interface Controller (I^2C) 271
6.10.1 Overview 271
6.10.2 Features 271
6.10.3 Basic Configuration 271
6.10.4 Functional Description 271
6.10.5 I^2C Protocol 272
6.10.6 I^2C Protocol Registers 276
6.10.7 Operation Modes 280
6.10.8 Register Map 288
6.10.9 Register Description 289
6.11 Serial Peripheral Interface (SPI) 299
6.11.1 Overview 299

6.11.2 Features 299
6.11.3 SPI Block Diagram 299
6.11.4 Basic Configuration 300
6.11.5 Functional Description 300
6.11.6 Timing Diagram 307
6.11.7 Programming Examples 309
6.11.8 Register Map 312
6.11.9 Register Description 313
6.12 Analog-to-Digital Converter (ADC) 327
6.12.1 Overview 327
6.12.2 Features 327
6.12.3 Block Diagram 328
6.12.4 Basic Configuration 329
6.12.5 Functional Description 329
6.12.6 Register Map 333
6.12.7 Register Description 334
6.13 Analog Comparator (ACMP) 346
6.13.1 Overview 346
6.13.2 Features 346
6.13.3 Block Diagram 347
6.13.4 Basic Configuration 348
6.13.5 Functional Description 348
6.13.6 Comparator Reference Voltage (CRV) 349
6.13.7 Register Map 350
6.13.8 Register Description 351
7 ELECTRICAL CHARACTERISTICS 357
8 PACKAGE DIMENSIONS 358
8.1 48-pin LQFP 358
8.2 33-pin QFN (4 mm x 4 mm) 359
8.3 33-pin QFN (5 mm x 5 mm) 360
8.4 20-pin TSSOP 361
9 REVISION HISTORY 362

LIST OF FIGURES

Figure 4.1-1 NuMicro™ Mini51 Series Selection Code 16
Figure 4.3-1 NuMicro™ Mini51 Series LQFP 48-pin Diagram 18
Figure 4.3-2 NuMicro™ Mini51 Series QFN 33-pin Diagram 19
Figure 4.3-3 NuMicro™ Mini51 Series TSSOP 20-pin Diagram 20
Figure 4.3-4 NuMicro™ Mini51 Series TSSOP 20-pin Diagram 20
Figure 5.1-1 NuMicro™ Mini51 Series Block Diagram 25
Figure 6.1-1 Functional Block Diagram 26
Figure 6.2-1 NuMicro™ Mini51 Series Power Architecture Diagram 29
Figure 6.3-1 Clock Generator Block Diagram 95
Figure 6.3-2 System Clock Block Diagram 96
Figure 6.3-3 SysTick Clock Control Block Diagram 96
Figure 6.3-4 AHB Clock Source for HCLK 97
Figure 6.3-5 Peripherals Clock Source Selection for PCLK 97
Figure 6.3-6 Clock Source of Frequency Divider 99
Figure 6.3-7 Block Diagram of Frequency Divider 99
Figure 6.4-1 Flash Memory Control Block Diagram 115
Figure 6.4-2 Flash Memory Organization 117
Figure 6.4-3 Flash Memory Structure 122
Figure 6.4-4 Boot Select (BS) for Power-on Action 123
Figure 6.4-5 Program Executing Range for boot from APROM and boot from LDROM 123
Figure 6.4-6 Executable Range of Code with IAP Function Enabled 125
Figure 6.4-7 Example Flow of Boot Selection by BS Bit when CBS[0] = 1 126
Figure 6.4-8 ISP Flow Example 127
Figure 6.4-9 Multi-booting by Vector Page Remapping 129
Figure 6.5-1 Push-Pull Output 140
Figure 6.5-2 Open-Drain Output 140
Figure 6.5-3 Quasi-Bidirectional I/O Mode 141
Figure 6.6-1 Timer Controller Block Diagram 161
Figure 6.6-2 Clock Source of Timer Controller 161
Figure 6.6-3 Continuous Counting Mode 163
Figure 6.7-1 Application Circuit Diagram 178
Figure 6.7-2 PWM Block Diagram 179
Figure 6.7-3 PWM Generator 0 Architecture Diagram 179
Figure 6.7-4 PWM Generator 2 Architecture Diagram 180
Figure 6.7-5 PWM Generator 4 Architecture Diagram 180
Figure 6.7-6 Edge-aligned PWM 182

Figure 6.7-7 PWM Edge-aligned Waveform Output 182
Figure 6.7-8 Edge-aligned Flow Diagram 183
Figure 6.7-9 Legend of Internal Comparator Output of PWM-Timer 184
Figure 6.7-10 PWM-Timer Operation Timing 184
Figure 6.7-11 Center-aligned Mode 185
Figure 6.7-12 Center-aligned Mode Operation Timing 186
Figure 6.7-13 PWM Center-aligned Waveform Output 186
Figure 6.7-14 Center-aligned Flow Diagram (INT_TYPE = 0) 187
Figure 6.7-15 PWM Double Buffering Illustration 188
Figure 6.7-16 PWM Controller Output Duty Ratio 189
Figure 6.7-17 Dead-zone Insertion 190
Figure 6.7-18 Initial State and Polarity Control with Rising Edge Dead-zone Insertion 191
Figure 6.7-19 Motor Control PWM Architecture 192
Figure 6.8-1 Watchdog Timer Clock Control 230
Figure 6.8-2 Watchdog Timer Block Diagram 231
Figure 6.8-3 Watchdog Timer Time-out Interval and Reset Period Timing 232
Figure 6.9-1 UART Controller Clock Control 237
Figure 6.9-2 UART Controller Block Diagram 237
Figure 6.9-3 UART Auto Flow Control Block Diagram 239
Figure 6.9-4 UART Controller Baud Rate Equation Table 240
Figure 6.9-5 UART Controller Baud Rate Parameter And Register Setting Table 240
Figure 6.9-6 UART Controller Interrupt Source and Flag 242
Figure 6.9-7 UART Line Control of Word and Stop Length Setting 242
Figure 6.9-8 UART Line Control of Parity Bit Setting 243
Figure 6.9-9 CTS Auto Flow Control Enabled 243
Figure 6.9-10 RTS Auto Flow Control Enabled 244
Figure 6.9-11 RTS Flow with Software Control 244
Figure 6.9-12 IrDA Control Block Diagram 245
Figure 6.9-13 IrDA TX/RX Timing Diagram 246
Figure 6.9-14 RS-485 RTS Driving Level in Auto Direction Mode 247
Figure 6.9-15 RS-485 RTS Driving Level with Software Control 248
Figure 6.9-16 Structure of RS-485 Frame 249
Figure 6.10-1 I^2C Bus Timing 272
Figure 6.10-2 I^2C Protocol 273
Figure 6.10-3 START and STOP Condition 273
Figure 6.10-4 Bit Transfer on the I^2C Bus 274
Figure 6.10-5 Acknowledge on the I^2C Bus 274

Figure 6.10-6 Acknowledge on the I^2C Bus 274
Figure 6.10-7 Master Reads Data from Slave 275
Figure 6.10-8 Timing of Two Level FIFO Transmit in Master Write 275
Figure 6.10-9 Timing of Two Level FIFO Transmit in Slave Read 275
Figure 6.10-10 I^2C Data Shifting Direction 277
Figure 6.10-11 I^2C Time-out Count Block Diagram 279
Figure 6.10-12 Control I^2C Bus according to Current I^2C Status 280
Figure 6.10-13 Master Transmitter Mode Control Flow 281
Figure 6.10-14 Master Receiver Mode Control Flow 282
Figure 6.10-15 Slave Mode Control Flow 283
Figure 6.10-16 GC Mode 285
Figure 6.10-17 EEPROM Random Read 286
Figure 6.10-18 Protocol of EEPROM Random Read 287
Figure 6.11-1 SPI Block Diagram 299
Figure 6.11-2 SPI Master Mode Application Block Diagram 300
Figure 6.11-3 SPI Slave Mode Application Block Diagram 301
Figure 6.11-4 32-bit in One Transaction 302
Figure 6.11-5 Byte Reorder 304
Figure 6.11-6 Timing Waveform for Byte Suspend 304
Figure 6.11-7 FIFO mode Block Diagram 305
Figure 6.11-8 SPI Timing in Master Mode 307
Figure 6.11-9 SPI Timing in Master Mode (Alternate Phase of SPICLK) 308
Figure 6.11-10 SPI Timing in Slave Mode 308
Figure 6.11-11 SPI Timing in Slave Mode (Alternate Phase of SPICLK) 309
Figure 6.12-1 A/D Controller Block Diagram 328
Figure 6.12-2 ADC Peripheral Clock Control 329
Figure 6.12-3 Single Mode Conversion Timing Diagram 330
Figure 6.12-4 V_{BG} for Measuring AV_{DD} Application Block Diagram 331
Figure 6.12-5 A/D Conversion Result Monitor Logics Diagram 332
Figure 6.12-6 A/D Controller Interrupt 333
Figure 6.12-7 Conversion Result Mapping Diagram of ADC Single-end Input 335
Figure 6.13-1 Analog Comparator Block Diagram 347
Figure 6.13-2 Analog Comparator Controller Interrupt Sources 348
Figure 6.13-3 Comparator Hysteresis Function 348
Figure 6.13-4 Comparator Reference Voltage Block Diagram 349

LIST OF TABLES

Table 4.1-1 List of Abbreviations 15
Table 4.2-1NuMicro™ Mini51 Series Product Selection Guide 17
Table 6.2-1 Memory Mapping Table 30
Table 6.2-2 Address Space Assignments for On-Chip Modules 33
Table 6.2-3 Exception Model 65
Table 6.2-4 System Interrupt Map Vector Table 66
Table 6.2-5 Vector Table Format 66
Table 6.3-1 Peripheral Clock Source Selection Table 98
Table 6.3-2 Power-down Mode Control 103
Table 6.4-1 Flash Memory Address Map 116
Table 6.4-2 Data Flash Table 121
Table 6.4-3 Supported Boot Selection Options 124
Table 6.4-4 ISP Command List 128
Table 6.6-1 Input Capture Mode Operation 165
Table 6.8-1 Watchdog Timer Time-out Interval Period Selection 232
Table 6.10-1 I^2C Status Code Description 278

1 GENERAL DESCRIPTION

The NuMicro™ Mini51 series 32-bit microcontroller is embedded with ARM® Cortex™-M0 core for industrial control and applications which require high performance, high integration, and low cost. The Cortex™-M0 is the newest ARM® embedded processor with 32-bit performance at a cost equivalent to the traditional 8-bit microcontroller.

The NuMicro™ Mini51 series can run up to 24 MHz and operate at 2.5V ~ 5.5V, -40℃ ~ 105℃, and thus can afford to support a variety of industrial control and applications which need high CPU performance. The NuMicro™ Mini51 series offers 4K/8K/16Kbytes embedded program flash, size configurable Data Flash (shared with program flash), 2Kbytes flash for the ISP, and 2Kbytes SRAM.

Many system level peripheral functions, such as I/O Port, Timer, UART, SPI, I^2C, PWM, ADC, Watchdog Timer, Analog Comparator and Brown-out Detector, have been incorporated into the NuMicro™ Mini51 series in order to reduce component count, board space and system cost. These useful functions make the NuMicro™ Mini51 series powerful for a wide range of applications.

Additionally, the NuMicro™ Mini51 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.

2 FEATURES

- Core
  - ARM® Cortex™-M0 core running up to 24 MHz
  - One 24-bit system timer
  - Supports Low Power Sleep 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 watch points/four breakpoints
- Built-in LDO for wide operating voltage ranged: 2.5 V to 5.5 V
- Memory
  - 4 KB/ 8 KB/ 16 KB Flash memory for program memory (APROM)
  - Configurable Flash memory for data memory (Data Flash)
  - 2 KB Flash for loader (LDROM)
  - 2 KB SRAM for internal scratch-pad RAM (SRAM)
- Clock Control
  - Programmable system clock source
    - Switch clock sources on-the-fly
  - 4 ~ 24 MHz external crystal input (HXT)
  - 32.768 kHz external crystal input (LXT) for Power-down wake-up and system operation clock
  - 22.1184 MHz internal oscillator (HIRC) (1% accuracy at 25°C, 5V)
    - Dynamically calibrating the HIRC OSC to 22.1184 MHz ±1% from -40°C to 105°C by external 32.768K crystal oscillator (LXT)
  - 10 kHz internal low-power oscillator (LIRC) for Watchdog Timer and Power-down wake-up
- I/O Port
  - Up to 30 general-purpose I/O (GPIO) pins for LQFP-48 package
  - Four I/O modes:
    - Input-only with high impendence
    - Push-pull output
    - Open-drain output
    - Quasi-bidirectional
  - TTL/Schmitt trigger input selectable
  - I/O pin can be configured as interrupt source with edge/level setting
  - Supports high driver and high sink I/O mode
  - Configurable default I/O mode of all pins after POR
- Timer
  - Provides two channel 32-bit timers. One 8-bit pre-scale counter with 24-bit up

counter for each timer
  - Independent clock source for each timer
  - Provides One-shot, Periodic, Toggle and Continuous operation modes
  - 24-bit up counter value is readable through TDR (Timer Data Register)
  - Provides trigger counting/free counting/counter reset function triggered by external capture pin or internal comparator signal
  - Provides event counter function
  - Supports wake-up from Idle or Power-down mode
- WDT (Watchdog Timer)
  - Multiple clock sources
  - Supports wake-up from Idle or Power-down mode
  - Interrupt or reset selectable on watchdog time-out
- PWM
  - Independent 16-bit PWM duty control units with maximum six outputs
  - Supports group/synchronous/independent/ complementary modes
  - Supports One-shot or Auto-reload mode
  - Supports Edge-aligned and Center-aligned type
  - Programmable dead-zone insertion between complementary channels
  - Each output has independent polarity setting control
  - Hardware fault brake protections
  - Supports duty, period, and fault break interrupts
  - Supports duty/period trigger ADC conversion
  - Timer comparing matching event trigger PWM to do phase change
  - Supports comparator event trigger PWM to force PWM output low for current period
  - Provides interrupt accumulation function
- UART (Universal Asynchronous Receiver/Transmitters)
  - One UART device
  - 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 devices
  - Supports Master/Slave mode
  - Full-duplex synchronous serial data transfer
  - Provides 3-wire function

- Variable length of transfer data from 8 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
- 4-level depth FIFO buffer

- I^2C
  - 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 allows devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
  - Programmable clocks allow for versatile rate control
  - Supports 7-bit addressing mode
  - Supports multiple address recognition (four slave addresses with mask option)
  - Supports Power-down wake-up function
  - Support FIFO function
- ADC (Analog-to-Digital Converter)
  - 10-bit SAR ADC with 300K SPS
  - Up to 8-ch single-end input and one internal input from band-gap
  - Conversion started either by software trigger, PWM trigger, or external pin trigger
  - Supports conversion value monitoring (or comparison) for threshold voltage detection
- Analog Comparator
  - Two analog comparators with programmable 16-level internal voltage reference
  - Build-in CRV (comparator reference voltage)
  - Supports Hysteresis function
  - Interrupt when compared results changed
- ISP (In-System Programming) and ICP (In-Circuit Programming)
- 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
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

Table 4.1-1 List of Abbreviations

4 PARTS INFORMATION LIST AND PIN CONFIGURATION

4.1 NuMicro™ Mini51 Series Selection Code

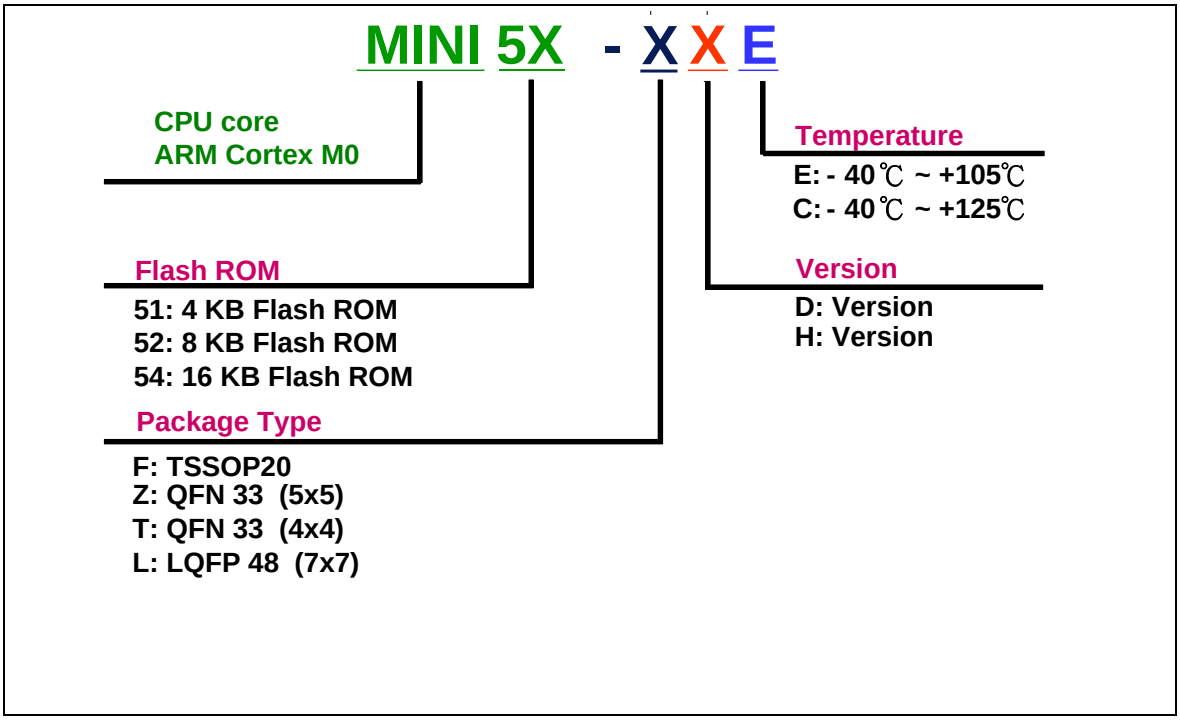


Figure 4.1-1 NuMicro™ Mini51 Series Selection Code

4.2 NuMicro™ Mini51 Series Product Selection Guide

Part No.	APROM	RAM	Data Flash	ISP Loader ROM	I/O	Timer	Connectivity			Comp.	PWM	ADC	ISP ICP IAP	IRC 22.1184 MHz	Package
							UART	SPI	I^2C						
MINI51FDE	4 KB	2 KB	Configurable	2 KB	up to 17	2x 32-bit	1	1	1	-	3	4x10-bit	v	v	TSSOP20
MINI51LDE	4 KB	2 KB	Configurable	2 KB	up to 30	2x 32-bit	1	1	1	2	6	8x10-bit	v	v	LQFP48
MINI51ZDE	4 KB	2 KB	Configurable	2 KB	up to 29	2x 32-bit	1	1	1	2	6	8x10-bit	v	v	QFN33 (5x5)
MINI51TDE	4 KB	2 KB	Configurable	2 KB	up to 29	2x 32-bit	1	1	1	2	6	8x10-bit	v	v	QFN33 (4x4)
MINI52FDE	8 KB	2 KB	Configurable	2 KB	up to 17	2x 32-bit	1	1	1	-	3	4x10-bit	v	v	TSSOP20
MINI52LDE	8 KB	2 KB	Configurable	2 KB	up to 30	2x 32-bit	1	1	1	2	6	8x10-bit	v	v	LQFP48
MINI52ZDE	8 KB	2 KB	Configurable	2 KB	up to 29	2x 32-bit	1	1	1	2	6	8x10-bit	v	v	QFN33 (5x5)
MINI52TDE	8 KB	2 KB	Configurable	2 KB	up to 29	2x 32-bit	1	1	1	2	6	8x10-bit	v	v	QFN33 (4x4)
MINI54FDE	16 KB	2 KB	Configurable	2 KB	up to 17	2x 32-bit	1	1	1	-	3	4x10-bit	v	v	TSSOP20
MINI54LDE	16 KB	2 KB	Configurable	2 KB	up to 30	2x 32-bit	1	1	1	2	6	8x10-bit	v	v	LQFP48
MINI54ZDE	16 KB	2 KB	Configurable	2 KB	up to 29	2x 32-bit	1	1	1	2	6	8x10-bit	v	v	QFN33 (5x5)
MINI54TDE	16 KB	2 KB	Configurable	2 KB	up to 29	2x 32-bit	1	1	1	2	6	8x10-bit	v	v	QFN33 (4x4)
*MINI54FHC	16 KB	2 KB	Configurable	2 KB	up to 17	2x 32-bit	1	1	1	-	6	3x10-bit	v	v	TSSOP20

Table 4.2-1NuMicro™ Mini51 Series Product Selection Guide

* Mini54FHC is a special part number, not pin to pin compatible to others Mini51series part number.

4.3 PIN CONFIGURATION

4.3.1 LQFP 48-pin

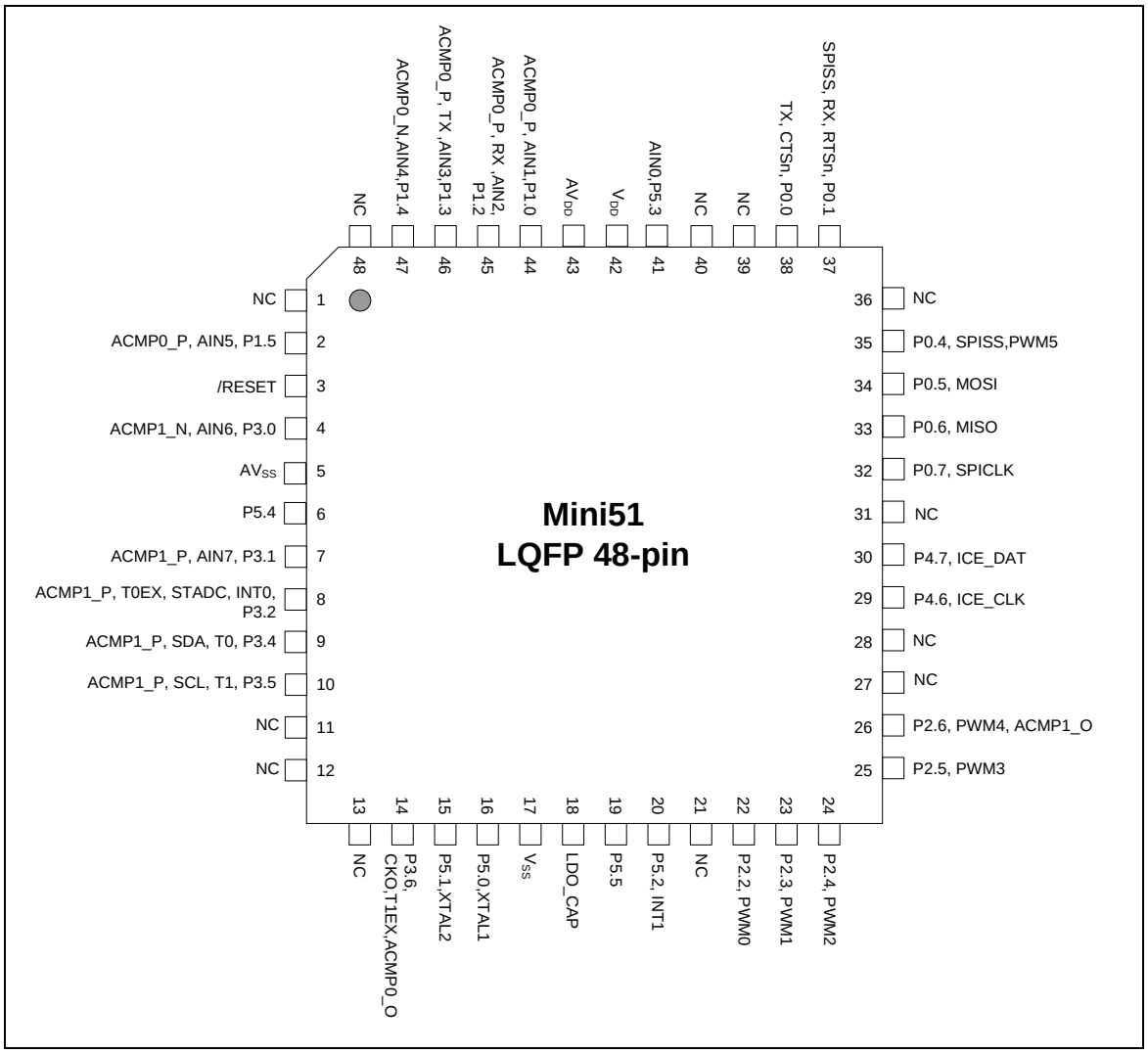


Figure 4.3-1 NuMicro™ Mini51 Series LQFP 48-pin Diagram

4.3.2 QFN 33-pin

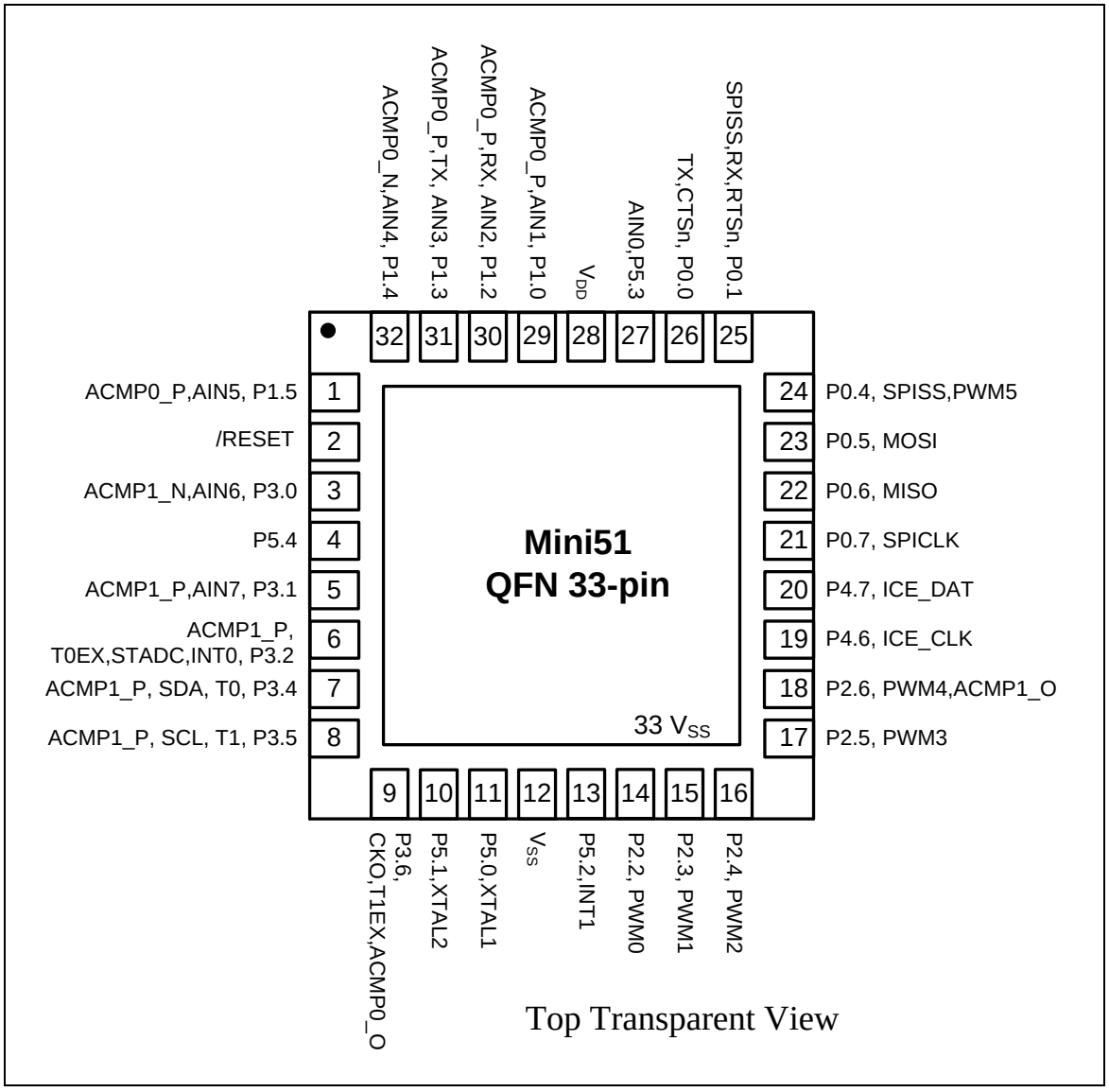


Figure 4.3-2 NuMicro™ Mini51 Series QFN 33-pin Diagram

4.3.3 TSSOP 20-pin

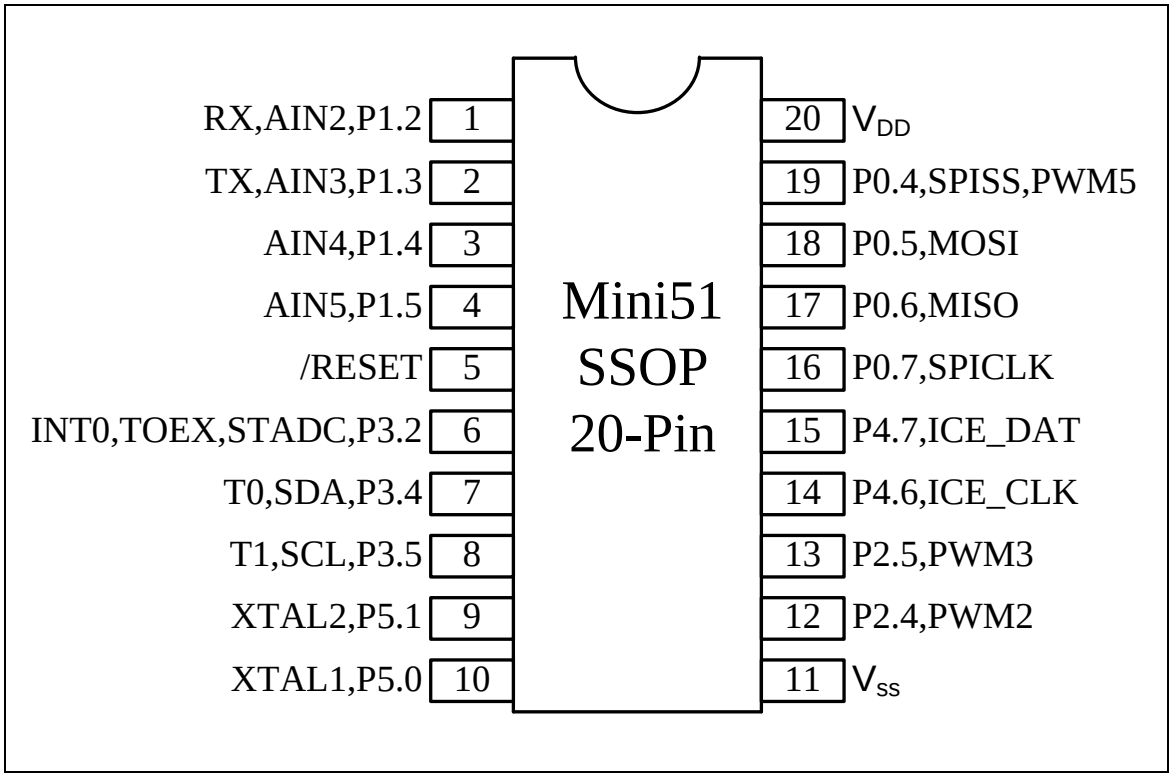


Figure 4.3-3 NuMicro™ Mini51 Series TSSOP 20-pin Diagram

4.3.4 Mini54FHC (TSSOP20-pin)

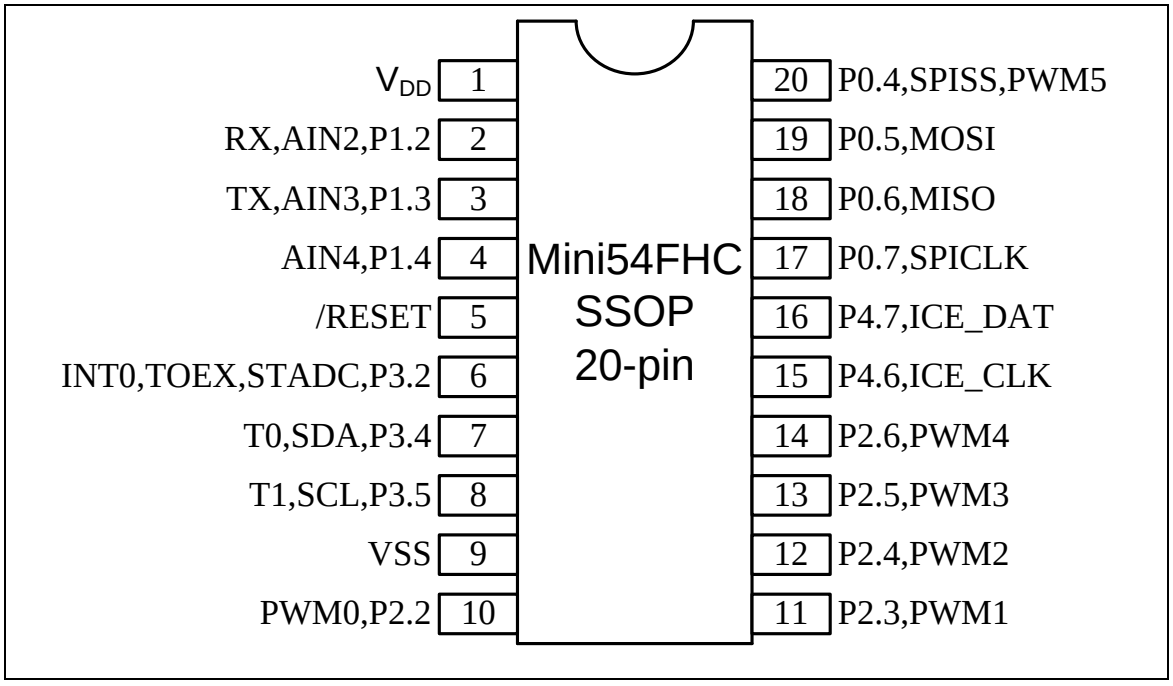


Figure 4.3-4 NuMicro™ Mini51 Series TSSOP 20-pin Diagram

4.4 Pin Description

Pin Number				Pin Name	Pin Type	Description
LQFP 48-pin	QFN 33-pin	TSSOP 20-pin	Mini54FHC TSSOP20-pin			
1	---	---	---	NC	---	Not connected
2	1	4	---	P1.5	I/O	General purpose digital I/O pin
				AIN5	AI	ADC analog input pin
				ACMP0_P	AI	Analog comparator positive input pin
3	2	5	5	/RESET	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. /RESET 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
				AIN6	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
				AIN7	AI	ADC analog input pin
				ACMP1_P	AI	Analog comparator positive input pin
8	6	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
				T0EX	I	Timer 0 external capture/reset trigger input pin
				ACMP1_P	AI	Analog comparator positive input pin
9	7	7	7	P3.4	I/O	General purpose digital I/O pin
				T0	I/O	Timer 0 external event counter input pin
				SDA	I/O	I^2C data I/O pin
				ACMP1_P	AI	Analog comparator positive input pin
10	8	8	8	P3.5	I/O	General purpose digital I/O pin
				T1	I/O	Timer 1 external event counter input pin
				SCL	I/O	I^2C clock I/O pin
				ACMP1_P	AI	Analog comparator positive input pin
11	---	---	---	NC	---	Not connected.

Pin Number				Pin Name	Pin Type	Description
LQFP 48-pin	QFN 33-pin	TSSOP 20-pin	Mini54FHC TSSOP20-pin			
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.
				CKO	O	Frequency divider output pin.
				T1EX	I	Timer 1 external capture/reset trigger input pin.
15	10	9	---	P5.1	I/O	General purpose digital I/O pin.
				XTAL2	O	The output pin from the internal inverting amplifier. It emits the inverted signal of XTAL1.
16	11	10	---	P5.0	I/O	General purpose digital I/O pin.
				XTAL1	I	The input pin to the internal inverting amplifier. The system clock could be from external crystal or resonator.
17	12 33	11	9	V_{SS}	P	Ground pin for digital circuit.
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	---	10	P2.2	I/O	General purpose digital I/O pin.
				PWM0	O	PWM0 output of PWM unit.
23	15	---	11	P2.3	I/O	General purpose digital I/O pin.
				PWM1	O	PWM1 output of PWM unit.
24	16	12	12	P2.4	I/O	General purpose input/output digital pin.
				PWM2	O	PWM2 output of PWM unit.
25	17	13	13	P2.5	I/O	General purpose digital I/O pin.
				PWM3	O	PWM3 output of PWM unit.
26	18	---	14	P2.6	I/O	General purpose digital I/O pin.
				PWM4	O	PWM4 output of PWM unit.
				ACMP1_O	O	Analog comparator output pin.
27	---	---	---	NC	---	Not connected.
28	---	---	---	NC	---	Not connected.

Pin Number				Pin Name	Pin Type	Description
LQFP 48-pin	QFN 33-pin	TSSOP 20-pin	Mini54FHC TSSOP20-pin			
29	19	14	15	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.
30	20	15	16	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.
31	---	---	---	NC	**---**	Not connected.
32	21	16	17	P0.7	**I/O**	General purpose digital I/O pin.
				SPICLK	**I/O**	SPI serial clock pin.
33	22	17	18	P0.6	**I/O**	General purpose digital I/O pin.
				MISO	**I/O**	SPI MISO (master in/slave out) pin.
34	23	18	19	P0.5	**I/O**	General purpose digital I/O pin.
				MOSI	**O**	SPI MOSI (master out/slave in) pin.
35	24	19	20	P0.4	**I/O**	General purpose digital I/O pin.
				SPISS	**I/O**	SPI slave select pin.
				PWM5	**O**	PWM5 output of PWM unit.
36	---	---	---	NC	**---**	Not connected.
37	25	---	---	P0.1	**I/O**	General purpose digital I/O pin.
				RTSn	**O**	UART RTS pin.
				RX	**I**	UART data receiver input pin.
				SPISS	**I/O**	SPI slave select pin.
38	26	---	---	P0.0	**I/O**	General purpose digital I/O pin.
				CTSn	**I**	UART CTS pin.
				TX	**O**	UART transmitter output pin.
39	---	---	---	NC	**---**	Not connected.
40	---	---	---	NC	**---**	Not connected.
41	27	---	---	P5.3	**I/O**	General purpose digital I/O pin.
				AIN0	**AI**	ADC analog input pin.
42	28	20	1	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.
				AIN1	**AI**	ADC analog input pin.

Pin Number				Pin Name	Pin Type	Description
LQFP 48-pin	QFN 33-pin	TSSOP 20-pin	Mini54FHC TSSOP20-pin			
				ACMP0_P	**AI**	Analog comparator positive input pin.
45	30	1	2	P1.2	**I/O**	General purpose digital I/O pin.
				AIN2	**AI**	ADC analog input pin.
				RX	**I**	UART data receiver input pin.
				ACMP0_P	**AI**	Analog comparator positive input pin.
46	31	2	3	P1.3	**I/O**	General purpose digital I/O pin.
				AIN3	**AI**	ADC analog input pin.
				TX	**O**	UART transmitter output pin.
				ACMP0_P	**AI**	Analog comparator positive input pin.
47	32	3	4	P1.4	**I/O**	General purpose digital I/O pin.
				AIN4	**I/O**	PWM5: PWM output/Capture input.
				ACMP0_N	**AI**	Analog comparator negative input pin.
48	---	--	--	NC	**---**	Not connected.

[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™ Mini51 Block Diagram

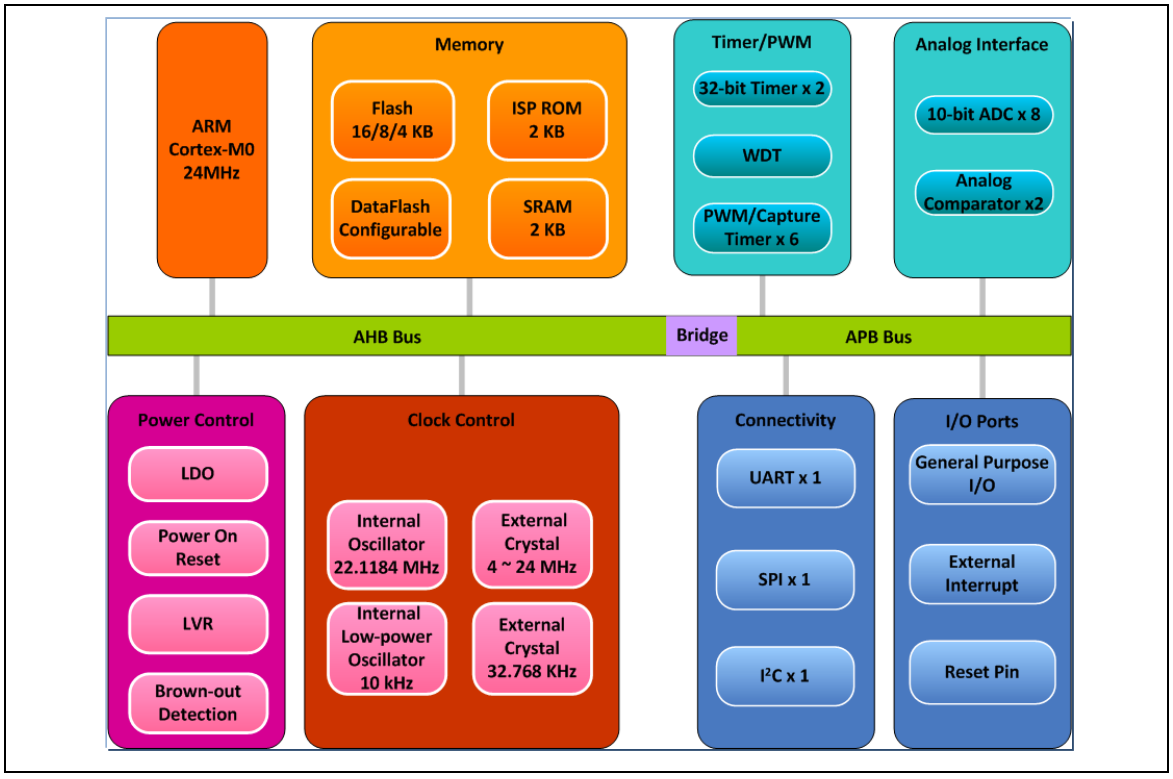


Figure 5.1-1 NuMicro™ Mini51 Series Block Diagram

6 FUNCTIONAL DESCRIPTION

6.1 ARM® Cortex™-M0 core

The Cortex™-M0 processor is a configurable, multistage, 32-bit RISC processor which has an AMBA AHB-Lite interface and includes an NVIC component. It also has optional hardware debug functionality. The processor can execute Thumb code and is compatible with other Cortex™-M profile 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. The following figure shows the functional controller of the processor.

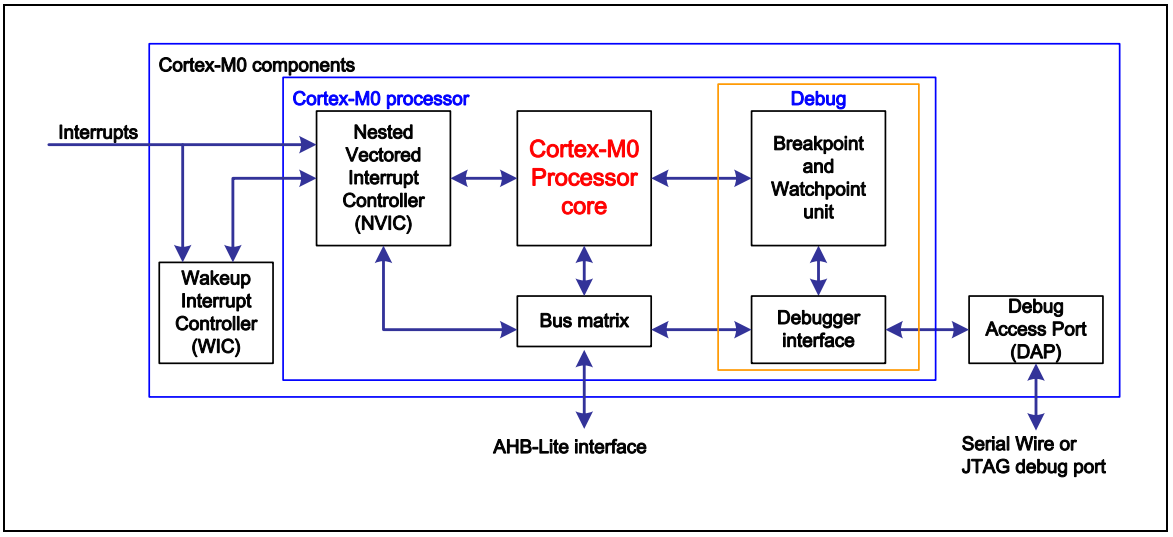


Figure 6.1-1 Functional Block Diagram

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 included by one of the following listed events. For these reset events flags can be read by RSTSRC register.

- Power-On Reset (POR)
- Low level on the Reset Pin (/RESET)
- Watchdog Timer Time-out Reset (WDT)
- Brown-out Detector Reset (BOD)
- Cortex™-M0 MCU Reset
- CPU Reset

6.2.3 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.
- Build-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}). The following figure shows the power distribution of the Mini51™DE series.

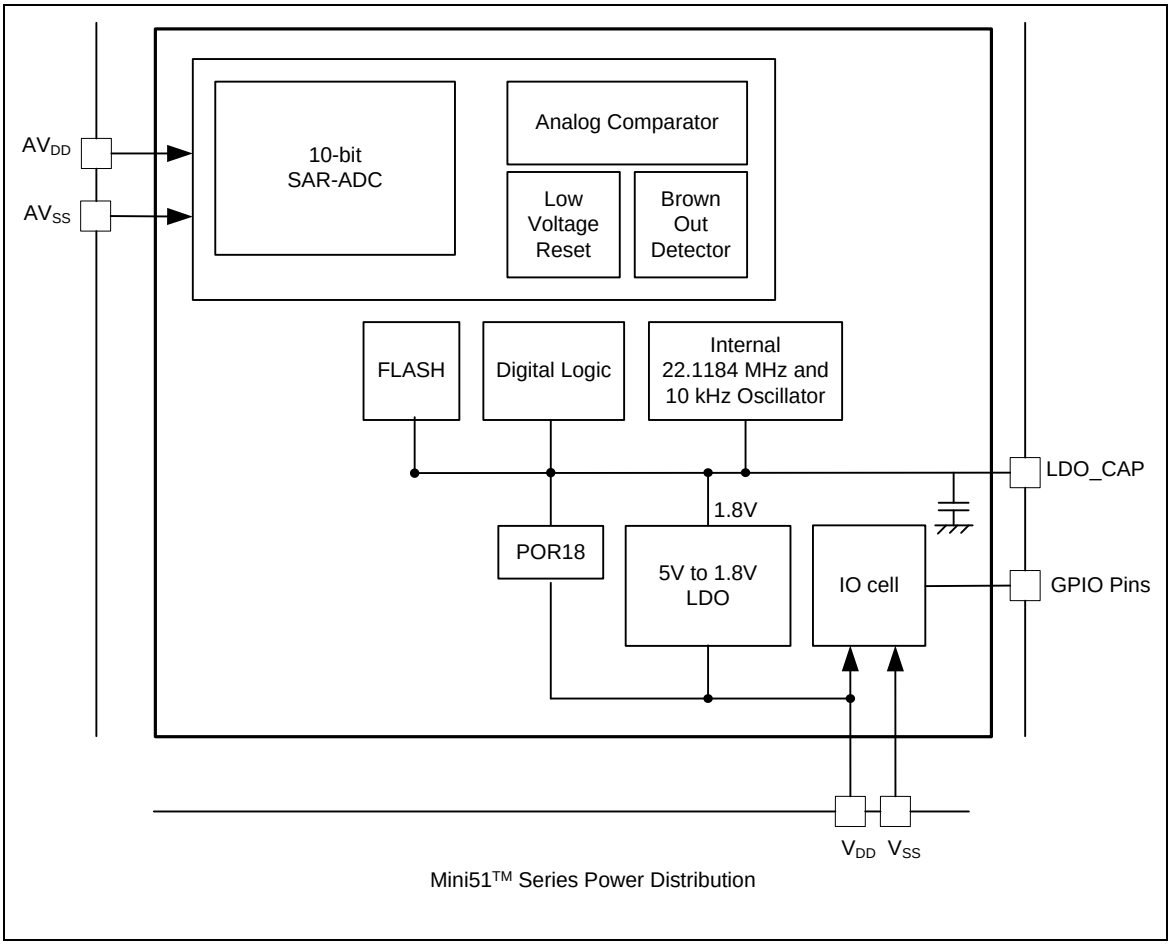


Figure 6.2-1 NuMicro™ Mini51 Series Power Architecture Diagram

6.2.4 Whole System Memory Mapping

Mini51/52/54

4 GB	Reserved	0xFFFF_FFFF | 0xE000_F000
	System Control	0xE000_EFFF 0xE000_E000
	Reserved	0xE000_E00F | 0x6002_0000
	Reserved	0x6001_FFFF 0x6000_0000
	Reserved	0x5FFF_FFFF | 0x5020_0000
	AHB	0x501F_FFFF 0x5000_0000
	Reserved	0x4FFF_FFFF | 0x4020_0000
	APB	0x401F_FFFF | 0x4000_0000
1 GB	Reserved	0x3FFF_FFFF | 0x2000_0800
	2 KB SRAM	0x2000_07FF 0x2000_0000
0.5 GB	Reserved	0x1FFF_FFFF | 0x0000_4000
	16 KB on-chip Flash (Mini54)	0x0000_3FFF
	8 KB on-chip Flash (Mini52)	0x0000_1FFF
	4 KB on-chip Flash (Mini51)	0x0000_0FFF 0x0000_0000
0 GB		

System Control

System Control	0xE000_ED00	SCS_BA
External Interrupt Control	0xE000_E100	SCS_BA
System Timer Control	0xE000_E010	SCS_BA

AHB peripherals

FMC	0x5000_C000	FMC_BA
GPIO Control	0x5000_4000	GP_BA
Interrupt Multiplexer Control	0x5000_0300	INT_BA
Clock Control	0x5000_0200	CLK_BA
System Global Control	0x5000_0000	GCR_BA

APB peripherals

ADC Control	0x400E_0000	ADC_BA
ACMP Control	0x400D_0000	CMP_BA
UART Control	0x4005_0000	UART_BA
PWM Control	0x4004_0000	PWM_BA
SPI Control	0x4003_0000	SPI_BA
I2C Control	0x4002_0000	I2C_BA
Timer0/Timer1 Control	0x4001_0000	TMR_BA
WDT Control	0x4000_4000	WDT_BA

Table 6.2-1 Memory Mapping Table

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

The protected registers are listed as following table.

Register	Bit	Description
IPRSTC1	[0] CHIP_RST	CHIP One-shot Reset (Write Protect)
BODCR	[5] BOD_LPM	Brown-out Detector Low Power Mode (Write Protect)
NMI_CON	[8] NMI_SEL_EN	NMI Interrupt Enable Control (Write Protect)
PWRCON	[7] PWR_DOWN_EN	System Power-down Enable Bit (Write Protect)
	[5] PD_WU_INT_EN	Power-down Mode Wake-up Interrupt Enable Control (Write Protect)
	[4] PD_WU_DLY	Wake-up Delay Counter Enable Control (Write Protect)
	[3] OSC10K_EN	10 KHz Internal Low Speed RC Oscillator (LIRC) Enable Control (Write Protect)
	[2] OSC22M_EN	22.1184 MHz Internal High Speed RC Oscillator (HIRC) Enable Control (Write Protect)
	[1:0] XTLCLK_EN[1:0]	External Crystal HXT Or LXT Enable Control (Write Protect)
APBCLK	[0] WDT_EN	Watchdog Timer Clock Enable Control (Write Protect)
CLKSEL0	[5:3] STCLK_S[2:0]	Cortex™-M0 SysTick Clock Source Selection From Reference Clock (Write Protect)
CLKSEL0	[2:0] HCLK_S[2:0]	HCLK Clock Source Selection (Write Protect)
CLKSEL1	[1:0] WDT_S[1:0]	WDT CLK Clock Source Selection (Write Protect)
ISPCON	[6] ISPFF	ISP Fail Flag (Write Protect)
	[5] LDUEN	LDROM Update Enable Control (Write Protect)
	[4] CFGUEN	CONFIG Update Enable Control (Write Protect)
	[3] APUEN	APROM Update Enable Control (Write Protect)
	[1] BS	Boot Select (Write Protect)
	[0] ISPEN	ISP Enable Control (Write Protect)
ISPTRG	[0] ISPGO	ISP Start Trigger (Write Protect)
ISPSTA	[6] ISPFF	ISP Fail Flag (Write Protect)
TCSR0	[31] DBGACK_TMR	ICE Debug Mode Acknowledge Disable Control (Write Protect)

TCSR1	[31] DBGACK_TMR	ICE Debug Mode Acknowledge Disable Control (Write Protect)
WTCR	[31] DBGACK_WDT	ICE Debug Mode Acknowledge Disable Control (Write Protect)
	[7] WTE	Watchdog Timer Enable Control (Write Protect)
	[6] WTIE	Watchdog Timer Time-out Interrupt Enable Control (Write Protect)
	[4] WTWKE	Watchdog Timer Time-out Wake-up Function Control (Write Protect)
	[0] WTR	Reset Watchdog Timer Up Counter (Write Protect)

6.2.6 System Memory Map

The NuMicro™ Mini51 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 NuMicro™ Mini51 series only supports little-endian data format.

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_3FFF	FLASH_BA	Flash Memory Space (16 KB)
0x2000_0000 – 0x2000_07FF	SRAM_BA	SRAM Memory Space (2 KB)
AHB Modules Space (0x5000_0000 – 0x501F_FFFF)		
0x5000_0000 – 0x5000_01FF	GCR_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_7FFF	WDT_BA	Watchdog Timer Control Registers
0x4001_0000 – 0x4001_3FFF	TMR_BA	Timer0/Timer1 Control Registers
0x4002_0000 – 0x4002_3FFF	I2C_BA	I^2C 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	UART_BA	UART Control Registers
0x400D_0000 – 0x400D_3FFF	ACMP_BA	Analog Comparator Control Registers
0x400E_0000 – 0x400E_3FFF	ADC_BA	Analog-Digital-Converter (ADC) 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

Addressing Space	Token	Modules
0xE000_ED00 – 0xE000_ED8F	SCB_BA	System Control Block Registers

Table 6.2-2 Address Space Assignments for On-Chip Modules

6.2.7 Register Map

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

Register	Offset	R/W	Description	Reset Value
GCR Base Address: **GCR_BA = 0x5000_0000**				
PDID	GCR_BA+0x00	R	Part Device Identification Number Register	0xXXXX_XXXX[1]
RSTSRC	GCR_BA+0x04	R/W	System Reset Source Register	0x0000_00XX
IPRSTC1	GCR_BA+0x08	R/W	Peripheral Reset Control Resister 1	0x0000_0000
IPRSTC2	GCR_BA+0x0C	R/W	Peripheral Reset Control Resister 2	0x0000_0000
BODCR	GCR_BA+0x18	R/W	Brown-out Detector Control Register	0x0000_000X
P0_MFP	GCR_BA+0x30	R/W	P0 Multiple Function and Input Type Control Register	0x0000_0000
P1_MFP	GCR_BA+0x34	R/W	P1 Multiple Function and Input Type Control Register	0x0000_0000
P2_MFP	GCR_BA+0x38	R/W	P2 Multiple Function and Input Type Control Register	0x0000_0000
P3_MFP	GCR_BA+0x3C	R/W	P3 Multiple Function and Input Type Control Register	0x0000_0000
P4_MFP	GCR_BA+0x40	R/W	P4 Multiple Function and Input Type Control Register	0x0000_00C0
P5_MFP	GCR_BA+0x44	R/W	P5 Multiple Function and Input Type Control Register	0x0000_0000
IRCTRIMCTL	GCR_BA+0x80	R/W	HIRC Trim Control Register	0x0000_0000
IRCTRIMIEN	GCR_BA+0x84	R/W	HIRC Trim Interrupt Enable Control Register	0x0000_0000
IRCTRIMINT	GCR_BA+0x88	R/W	HIRC Trim Interrupt Status Register	0x0000_0000
REGWRPROT	GCR_BA+0x100	R/W	Register Write-Protection Control Register	0x0000_0000

6.2.8 Register Description

Part Device Identification Number Register (PDID)

Register	Offset	R/W	Description	Reset Value
PDID	GCR_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[31:0]**	**Product Device Identification Number** This register reflects the device part number code. Software can read this register to identify which device is used. For example, the MINI51LDE PDID code is “0x20205100”.

NuMicro™ Mini51 Series	Part Device Identification Number
MINI51LDE	0x20205100
MINI51ZDE	0x20205103
MINI51TDE	0x20205104
MINI51FDE	0x20205105
MINI52LDE	0x20205200
MINI52ZDE	0x20205203
MINI52TDE	0x20205204
MINI52FDE	0x20205205
MINI54LDE	0x20205400
MINI54ZDE	0x20205403
MINI54TDE	0x20205404
MINI54FDE	0x20205405

System Reset Source Register (RSTSRC)

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

Register	Offset	R/W	Description	Reset Value
RSTSRC	GCR_BA+0x04	R/W	System Reset Source Register	0x0000_00XX

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

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**RSTS_CPU**	**CPU Reset Flag** The RSTS_CPU flag is set by hardware if software writes CPU_RST (IPRSTC1[1]) 1 to reset Cortex™-M0 core and Flash memory controller (FMC). 0 = No reset from CPU. 1 = Cortex™-M0 core and FMC are reset by software setting CPU_RST to 1. **Note:** Software can write 1 to clear this bit to 0.
[6]	**Reserved**	Reserved.
[5]	**RSTS_MCU**	**MCU Reset Flag** The RSTS_MCU 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 bit SYSRESETREQ (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 0.
[4]	**RSTS_BOD**	**Brown-out Detector Reset Flag** The RSTS_BOD 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 0.
[3]	**Reserved**	Reserved.

Bits	Description	
[2]	**RSTS_WDT**	**Watchdog Reset Flag** The RSTS_WDT flag is set by the “reset signal” from the Watchdog timer to indicate the previous reset source. 0 = No reset from Watchdog timer. 1 = The Watchdog timer had issued the reset signal to reset the system. **Note:** Software can write 1 to clear this bit to 0.
[1]	**RSTS_RESET**	**Reset Pin Reset Flag** The RSTS_RESET flag is set by the “reset signal” from the /RESET pin to indicate the previous reset source. 0 = No reset from pin /RESET pin. 1 = The /RESET pin had issued the reset signal to reset the system. **Note:** Software can write 1 to clear this bit to 0.
[0]	**RSTS_POR**	**Power-on Reset Flag** The RSTS_POR flag is set by the “reset signal”, which is from the Power-On Reset (POR) controller or bit CHIP_RST (IPRSTC1[0]), to indicate the previous reset source. 0 = No reset from POR or CHIP_RST. 1 = Power-on-Reset (POR) or CHIP_RST had issued the reset signal to reset the system. **Note:** Software can write 1 to clear this bit to 0.

Peripheral Reset Control Register 1 (IPRSTC1)

Register	Offset	R/W	Description	Reset Value
IPRSTC1	GCR_BA+0x08	R/W	Peripheral Reset Control Resister 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						**CPU_RST**	**CHIP_RST**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**CPU_RST**	**CPU Kernel Reset** Setting this bit will reset the CPU kernel, and this bit will automatically return to 0 after the 2 clock cycles. 0 = CPU normal operation. 1 = Reset CPU Kernel. **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 REGWRPROT at address GCR_BA + 0x100.
[0]	**CHIP_RST**	**CHIP One-shot Reset (Write Protect)** Setting this bit will reset the CHIP, including CPU kernel and all peripherals, and this bit will automatically return to 0 after the 2 clock cycles. The CHIP_RST is the same as the POR reset, and all the chip module is reset and the chip settings from flash are also reloaded. 0 = Chip normal operation. 1 = CHIP one-shot reset. **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 REGWRPROT at address GCR_BA + 0x100.

Peripheral Reset Control Register 2 (IPRSTC2)

Setting the bit to 1 will generate the asynchronous reset signal to the corresponding module. User needs to set the bit to 0 to release module from the reset state.

Register	Offset	R/W	Description	Reset Value
IPRSTC2	GCR_BA+0x0C	R/W	Peripheral Reset Control Resister 2	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			**ADC_RST**	**Reserved**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**ACMP_RST**	**Reserved**	**PWM_RST**	**Reserved**			**UART_RST**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**SPI_RST**	**Reserved**			**I2C_RST**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**TMR1_RST**	**TMR0_RST**	**GPIO_RST**	**Reserved**

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**ADC_RST**	**ADC Controller Reset** 0 = ADC module normal operation. 1 = ADC module reset.
[27:23]	**Reserved**	Reserved.
[22]	**ACMP_RST**	**ACMP Controller Reset** 0 = ACMP module normal operation. 1 = ACMP module reset.
[21]	**Reserved**	Reserved.
[20]	**PWM_RST**	**PWM Controller Reset** 0 = PWM module normal operation. 1 = PWM module reset.
[19:17]	**Reserved**	Reserved.
[16]	**UART_RST**	**UART Controller Reset** 0 = UART module normal operation. 1 = UART module reset.
[15:13]	**Reserved**	Reserved.
[12]	**SPI_ RST**	**SPI Controller Reset** 0 = SPI module normal operation. 1 = SPI module reset.
[11:9]	**Reserved**	Reserved.

Bits	Description	
[8]	**I^2C _RST**	**I^2C Controller Reset** 0 = I^2C module normal operation. 1 = I^2C module reset.
[7:4]	**Reserved**	Reserved.
[3]	**TMR1_RST**	**Timer1 Controller Reset** 0 = Timer1 module normal operation. 1 = Timer1 module reset.
[2]	**TMR0_RST**	**Timer0 Controller Reset** 0 = Timer0 module normal operation. 1 = Timer0 module reset.
[1]	**GPIO_RST**	**GPIO (P0~P5) Controller Reset** 0 = GPIO module normal operation. 1 = GPIO module reset.
[0]	**Reserved**	Reserved.

Brown-out Detector Control Register (BODCR)

Partial of the BODCR 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
BODCR	GCR_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	BOD_OUT	BOD_LPM	BOD_INTF	BOD_RSTEN	BOD_VL		BOD_VL_EXT

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**BOD_OUT**	**Brown-out Detector Output State** 0 = Brown-out Detector status output is 0, the detected voltage is higher than BOD_VL setting. 1 = Brown-out Detector status output is 1, the detected voltage is lower than BOD_VL setting.
[5]	**BOD_LPM**	**Brown-out Detector Low Power Mode (Write Protect)** 0 = BOD operate in normal mode **(default).** 1 = Enable the BOD low power mode. The BOD consumes about 100uA in normal mode, the low power mode can reduce the current to about 1uA but slow the BOD response.
[4]	**BOD_INTF**	**Brown-out Detector Interrupt Flag** 0 = Brown-out Detector does not detect any voltage dropped at AV_{DD} down through or up through the voltage of BOD_VL setting. 1 = When Brown-out Detector detects the AV_{DD} is dropped through the voltage of BOD_VL setting or the AV_{DD} is raised up through the voltage of BOD_VL 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>BOD_RSTEN</td><td>Brown-out Reset Enable Control (Initiated And Write-protected 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.
The default value is set by flash controller user configuration register config0 bit[20].
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>BOD_VL[1:0]</td><td>Brown-out Detector Threshold Voltage Selection (Initiated & Write-protected Bit)
The default value is set by flash controller user configuration CBOV bit (config0 [22:21]).
<table>
<tr><th>BOD_VL[1]</th><th>BOD_VL[0]</th><th>Brown-out voltage</th></tr>
<tr><td>1</td><td>1</td><td>Disable 2.7V and 3.7V</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>Reserved</td></tr>
</table></td></tr>
<tr><td>[0]</td><td>BOD_VL_EXT</td><td>Brown-out Detector Selection Extension (Initiated & Write-protected Bit)
The default value is set by flash controller user configuration CBOVEXT bit (config0 [23]).
If config0 bit[23] is set to 1, default value of BOD_VL_EXT is 0.
If config0 bit[23] is set to 0, default value of BOD_VL_EXT is 1.
0 = Brown-out detector threshold voltage is selected by the table defined in BOD_VL.
1 = Brown-out detector threshold voltage is selected by the table defined below.
<table>
<tr><th>BOD_VL[1]</th><th>BOD_VL[0]</th><th>Brown-out voltage</th></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 (P0_MFP)

Register	Offset	R/W	Description	Reset Value
P0_MFP	GCR_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**
P0_TYPE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
P0_ALT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P0_MFP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**P0_TYPE[n]**	**P0[7:0] TTL Or Schmitt Trigger Function Enable Control** 0 = P0[7:0]Select I/O input as TTL function. 1 = P0[7:0] Select I/O input as Schmitt Trigger function .
[15]	**P0_ ALT[7]**	**P0.7 Alternate Function Selection** The pin function of P0.7 depends on P0_MFP[7] and P0_ALT[7].

P0_ALT[7]	P0_MFP[7]	P0.7 function
0	0	P0.7
0	1	Reserved
1	0	SPICLK (SPI)
1	1	Reserved

Bits	Description	
[14]	**P0_ ALT[6]**	**P0.6 Alternate Function Selection** The pin function of P0.6 depends on P0_MFP[6] and P0_ALT[6].

P0_ALT[6]	P0_MFP[6]	P0.6 function
0	0	P0.6
0	1	Reserved
1	0	MISO (SPI)
1	1	Reserved

<table>
<tr><th>Bits</th><th>Description</th><th></th></tr>
<tr><td>[13]</td><td>P0_ ALT[5]</td><td>P0.5 Alternate Function Selection
The pin function of P0.5 depends on P0_MFP[5] and P0_ALT[5].
<table>
<tr><th>P0_ALT[5]</th><th>P0_MFP[5]</th><th>P0.5 function</th></tr>
<tr><td>0</td><td>0</td><td>P0.5</td></tr>
<tr><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>MOSI (SPI)</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>[12]</td><td>P0_ ALT[4]</td><td>P0.4 Alternate Function Selection
The pin function of P0.4 depends on P0_MFP[4] and P0_ALT[4].
<table>
<tr><th>P0_ALT[4]</th><th>P0_MFP[4]</th><th>P0.4 function</th></tr>
<tr><td>0</td><td>0</td><td>P0.4</td></tr>
<tr><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>SPISS (SPI)</td></tr>
<tr><td>1</td><td>1</td><td>PWM5 (PWM)</td></tr>
</table></td></tr>
<tr><td>[11:10]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[9]</td><td>P0_ ALT[1]</td><td>P0.1 Alternate Function Selection
The pin function of P0.1 depends on P0_MFP[1] and P0_ALT[1].
<table>
<tr><th>P0_ALT[1]</th><th>P0_MFP[1]</th><th>P0.1 function</th></tr>
<tr><td>0</td><td>0</td><td>P0.1</td></tr>
<tr><td>0</td><td>1</td><td>SPISS (SPI)</td></tr>
<tr><td>1</td><td>0</td><td>RTSn (UART)</td></tr>
<tr><td>1</td><td>1</td><td>RX (UART)</td></tr>
</table></td></tr>
<tr><td>[8]</td><td>P0_ ALT[0]</td><td>P0.0 Alternate Function Selection
The pin function of P0.0 depends on P0_MFP[0] and P0_ALT[0].
<table>
<tr><th>P0_ALT[0]</th><th>P0_MFP[0]</th><th>P0.0 function</th></tr>
<tr><td>0</td><td>0</td><td>P0.0</td></tr>
<tr><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>CTSn (UART)</td></tr>
<tr><td>1</td><td>1</td><td>TX (UART)</td></tr>
</table></td></tr>
<tr><td>[7:0]</td><td>P0_MFP[7:0]</td><td>P0 Multiple Function Selection
The pin function of P0 depends on P0_MFP and P0_ALT.
Refer to P0_ALT Description for details.</td></tr>
</table>

Multiple Function Port1 Control Register (P1_MFP)

Register	Offset	R/W	Description	Reset Value
P1_MFP	GCR_BA+0x34	R/W	P1 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**
P1_TYPE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
P1_ALT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P1_MFP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**P1_TYPE[n]**	**P1[7:0] TTL Or Schmitt Trigger Function Enable Control** 0 = P1[7:0]Select I/O input as TTL function. 1 = P1[7:0] Select I/O input as Schmitt Trigger function .
[15:14]	**Reserved**	Reserved.
[13]	**P1_ ALT[5]**	**P1.5 Alternate Function Selection** The pin function of P1.5 depends on P1_MFP[5] and P1_ALT[5].

P1_ALT[5]	P1_MFP[5]	P1.5 function
0	0	P1.5
0	1	AIN5 (ADC)
1	0	Reserved
1	1	ACMP0_P (ACMP)

Bits	Description	
[12]	**P1_ ALT[4]**	**P1.4 Alternate Function Selection** The pin function of P1.4 depends on P1_MFP[4] and P1_ALT[4].

P1_ALT[4]	P1_MFP[4]	P1.4 function
0	0	P1.4
0	1	AIN4 (ADC)
1	0	Reserved
1	1	ACMP0_N (ACMP)

Bits	Description	
[11]	**P1_ ALT[3]**	**P1.3 Alternate Function Selection** The pin function of P1.3 depends on P1_MFP[3] and P1_ALT[3].

P1_ALT[3]	P1_MFP[3]	P1.3 function
0	0	P1.3
0	1	AIN3 (ADC)
1	0	TX (UART)
1	1	ACMP0_P (ACMP)

Bits	Description	
[10]	**P1_ ALT[2]**	**P1.2 Alternate Function Selection** The pin function of P1.2 depends on P1_MFP[2] and P1_ALT[2].

P1_ALT[2]	P1_MFP[2]	P1.2 function
0	0	P1.2
0	1	AIN2 (ADC)
1	0	RX (UART)
1	1	ACMP0_P (ACMP)

Bits	Description	
[9]	**Reserved**	Reserved.
[8]	**P1_ ALT[0]**	**P1.0 Alternate Function Selection** The pin function of P1.0 depends on P1_MFP[0] and P1_ALT[0].

P1_ALT[0]	P1_MFP[0]	P1.0 function
0	0	P1.0
0	1	AIN1 (ADC)
1	0	Reserved
1	1	ACMP0_P (ACMP)

Bits	Description	
[7:0]	**P1_MFP[7:0]**	**P1 Multiple Function Selection** The pin function of P1 depends on P1_MFP and P1_ALT. Refer to P1_ALT Description for details.

Multiple Function Port2 Control Register (P2_MFP)

Register	Offset	R/W	Description	Reset Value
P2_MFP	GCR_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**
P2_TYPE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
P2_ALT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P2_MFP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**P2_TYPE[n]**	**P2[7:0] TTL Or Schmitt Trigger Function Enable Control** 0 = P2[7:0]Select I/O input as TTL function. 1 = P2[7:0] Select I/O input as Schmitt Trigger function .
[15]	**Reserved**	Reserved.
[14]	**P2_ ALT[6]**	**P2.6 Alternate Function Selection** The pin function of P2.6 depends on P2_MFP[6] and P2_ALT[6].

P2_ALT[6]	P2_MFP[6]	P2.6 function
0	0	P2.6
0	1	Reserved
1	0	PWM4 (PWM)
1	1	ACMP1_PO (ACMP)

Bits	Description	
[13]	**P2_ ALT[5]**	**P2.5 Alternate Function Selection** The pin function of P2.5 depends on P2_MFP[5] and P2_ALT[5].

P2_ALT[5]	P2_MFP[5]	P2.5 function
0	0	P2.5
0	1	Reserved
1	0	PWM3 (PWM)
1	1	Reserved

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>[12]</td><td>P2_ ALT[4]</td><td>
P2.4 Alternate Function Selection
The pin function of P2.4 depends on P2_MFP[4] and P2_ALT[4].
<table>
<tr><th>P2_ALT[4]</th><th>P2_MFP[4]</th><th>P2.4 function</th></tr>
<tr><td>0</td><td>0</td><td>P2.4</td></tr>
<tr><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>PWM2 (PWM)</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>[11]</td><td>P2_ ALT[3]</td><td>
P2.3 Alternate Function Selection
The pin function of P2.3 depends on P2_MFP[3] and P2_ALT[3].
<table>
<tr><th>P2_ALT[3]</th><th>P2_MFP[3]</th><th>P2.3 function</th></tr>
<tr><td>0</td><td>0</td><td>P2.3</td></tr>
<tr><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>PWM1 (PWM)</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>[10]</td><td>P2_ ALT[2]</td><td>
P2.2 Alternate Function Selection
The pin function of P2.2 depends on P2_MFP[2] and P2_ALT[2].
<table>
<tr><th>P2_ALT[2]</th><th>P2_MFP[2]</th><th>P2.2 function</th></tr>
<tr><td>0</td><td>0</td><td>P2.2</td></tr>
<tr><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>PWM0 (PWM)</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>[9:8]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[7:0]</td><td>P2_MFP[7:0]</td><td>
P2 Multiple Function Selection
The pin function of P2 depends on P2_MFP and P2_ALT.
Refer to P2_ALT Description for details.
</td></tr>
</table>

Multiple Function Port3 Control Register (P3_MFP)

Register	Offset	R/W	Description	Reset Value
P3_MFP	GCR_BA+0x3C	R/W	P3 Multiple Function and Input Type Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							**P32CPP1**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
P3_TYPE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
P3_ALT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P3_MFP							

Bits	Description	
[31:25]	**Reserved**	Reserved.
[24]	**P32CPP1**	**P3.2 Alternate Function Selection Extension** 0 = P3.2 is set by P3_ALT[2] and P3_MFP[2]. 1 = P3.2 is set to CPP1 of ACMP1.
[23:16]	**P3_TYPE[n]**	**P3[7:0] TTL Or Schmitt Trigger Function Enable Control** 0 = P3[7:0]Select I/O input as TTL function. 1 = P3[7:0] Select I/O input as Schmitt Trigger function .
[15]	**Reserved**	Reserved.
[14]	**P3_ ALT[6]**	**P3.6 Alternate Function Selection** The pin function of P3.6 depends on P3_MFP[6] and P3_ALT[6].

P3_ALT[6]	P3_MFP[6]	P3.6 function
0	0	P3.6
0	1	T1EX(Time External Capture)
1	0	CKO (Clock Driver Output)
1	1	ACMP0_O (ACMP)

[13]	**P3_ ALT[5]**	**P3.5 Alternate Function Selection** The pin function of P3.5 depends on P3_MFP[5] and P3_ALT[5].

P3_ALT[5]	P3_MFP[5]	P3.5 function
0	0	P3.5
0	1	T1 (Timer External Event Counter)
1	0	SCL (I^2C)
1	1	ACMP1_P (ACMP)

<table>
<tr><th>Bits</th><th>Description</th><th></th></tr>
<tr><td>[12]</td><td>P3_ ALT[4]</td><td>P3.4 Alternate Function Selection
The pin function of P3.4 depends on P3_MFP[4] and P3_ALT[4].
<table>
<tr><th>P3_ALT[4]</th><th>P3_MFP[4]</th><th>P3.4 function</th></tr>
<tr><td>0</td><td>0</td><td>P3.4</td></tr>
<tr><td>0</td><td>1</td><td>T0 (Timer0)</td></tr>
<tr><td>1</td><td>0</td><td>SDA (I^2C)</td></tr>
<tr><td>1</td><td>1</td><td>ACMP1_P (ACMP)</td></tr>
</table></td></tr>
<tr><td>[11]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[10]</td><td>P3_ ALT[2]</td><td>P3.2 Alternate Function Selection
The pin function of P3.2 depends on P3_MFP[2] and P3_ALT[2].
Note: If P32CPP1 is set to 1, P3_ALT[2] and P3_MFP[2] settings will be ignored and pin P3.2 function will be CPP1 of ACMP1.
<table>
<tr><th>P3_ALT[2]</th><th>P3_MFP[2]</th><th>P3.2 function</th></tr>
<tr><td>0</td><td>0</td><td>P3.2</td></tr>
<tr><td>0</td><td>1</td><td>INT0</td></tr>
<tr><td>1</td><td>0</td><td>T0EX</td></tr>
<tr><td>1</td><td>1</td><td>STADC (ADC)</td></tr>
</table></td></tr>
<tr><td>[9]</td><td>P3_ ALT[1]</td><td>P3.1 Alternate Function Selection
The pin function of P3.1 depends on P3_MFP[1] and P3_ALT[1].
<table>
<tr><th>P3_ALT[1]</th><th>P3_MFP[1]</th><th>P3.1 function</th></tr>
<tr><td>0</td><td>0</td><td>P3.1</td></tr>
<tr><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>ACMP1_P (ACMP)</td></tr>
<tr><td>1</td><td>1</td><td>AIN7 (ADC)</td></tr>
</table></td></tr>
<tr><td>[8]</td><td>P3_ ALT[0]</td><td>P3.0 Alternate Function Selection
The pin function of P3.0 depends on P3_MFP[0] and P3_ALT[0].
<table>
<tr><th>P3_ALT[0]</th><th>P3_MFP[0]</th><th>P3.0 function</th></tr>
<tr><td>0</td><td>0</td><td>P3.0</td></tr>
<tr><td>0</td><td>1</td><td>Reserved</td></tr>
<tr><td>1</td><td>0</td><td>ACMP1_N (ACMP)</td></tr>
<tr><td>1</td><td>1</td><td>AIN6 (ADC)</td></tr>
</table></td></tr>
<tr><td>[7:0]</td><td>P3_MFP[7:0]</td><td>P3 Multiple Function Selection
The pin function of P3 depends on P3_MFP and P3_ALT.
Refer to P3_ALT Description for details.</td></tr>
</table>

Multiple Function Port4 Control Register (P4_MFP)

Register	Offset	R/W	Description	Reset Value
P4_MFP	GCR_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**
P4_TYPE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
P4_ALT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P4_MFP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**P4_TYPE[n]**	**P4[7:0] TTL Or Schmitt Trigger Function Enable Control** 0 = P4[7:0]Select I/O input as TTL function. 1 = P4[7:0] Select I/O input as Schmitt Trigger function .
[15]	**P4_ ALT[7]**	**P4.7 Alternate Function Selection** The pin function of P4.7 depends on P4_MFP[7] and P4_ALT[7]. **P4_ALT[7]** / **P4_MFP[7]** / **P4.7 function** 0 / 0 / P4.7 0 / 1 / ICE_DAT (ICE) 1 / x / Reserved **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_DAT pin.
[14]	**P4_ ALT[6]**	**P4.6 Alternate Function Selection** The pin function of P4.6 depends on P4_MFP[6] and P4_ALT[6]. **P4_ALT[6]** / **P4_MFP[6]** / **P4.6 function** 0 / 0 / P4.6 0 / 1 / ICE_CLK (ICE) 1 / x / Reserved **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_CLK pin.
[13:8]	**Reserved**	Reserved.
[7:0]	**P4_MFP[7:0]**	**P4 Multiple Function Selection** The pin function of P4 depends on P4_MFP and P4_ALT. Refer to P4_ALT Description for details.

Multiple Function Port5 Control Register (P5_MFP)

Register	Offset	R/W	Description	Reset Value
P5_MFP	GCR_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**
P5_TYPE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
P5_ALT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P5_MFP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**P5_TYPE[n]**	**P5[7:0] TTL Or Schmitt Trigger Function Enable Control** 0 = P5[7:0]Select I/O input as TTL function. 1 = P5[7:0] Select I/O input as Schmitt Trigger function .
[15:14]	**Reserved**	Reserved.
[13]	**P5_ ALT[5]**	**P5.5 Alternate Function Selection** The pin function of P5.5 depends on P5_MFP[5] and P5_ALT[5].

P5_ALT[5]	P5_MFP[5]	P5.5 function
0	0	P5.5
0	1	Reserved
1	x	Reserved

[12]	**P5_ ALT[4]**	**P5.4 Alternate Function Selection** The pin function of P5.4 depends on P5_MFP[4] and P5_ALT[4].

P5_ALT[4]	P5_MFP[4]	P5.4 function
0	0	P5.4
0	1	Reserved
1	x	Reserved

<table>
<tr><th>Bits</th><th>Description</th><th></th></tr>
<tr><td>[11]</td><td>P5_ ALT[3]</td><td>P5.3 Alternate Function Selection
The pin function of P5.3 depends on P5_MFP[3] and P5_ALT[3].
<table>
<tr><th>P5_ALT[3]</th><th>P5_MFP[3]</th><th>P5.3 function</th></tr>
<tr><td>0</td><td>0</td><td>P5.3</td></tr>
<tr><td>0</td><td>1</td><td>AIN0 (ADC)</td></tr>
<tr><td>1</td><td>x</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>[10]</td><td>P5_ ALT[2]</td><td>P5.2 Alternate Function Selection
The pin function of P5.2 depends on P5_MFP[2] and P5_ALT[2].
<table>
<tr><th>P5_ALT[2]</th><th>P5_MFP[2]</th><th>P5.2 function</th></tr>
<tr><td>0</td><td>0</td><td>P5.2</td></tr>
<tr><td>0</td><td>1</td><td>INT1</td></tr>
<tr><td>1</td><td>x</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>[9]</td><td>P5_ ALT[1]</td><td>P5.1 Alternate Function Selection
The pin function of P5.1 depends on P5_MFP[1] and P5_ALT[1].
<table>
<tr><th>P5_ALT[1]</th><th>P5_MFP[1]</th><th>P5.1 function</th></tr>
<tr><td>0</td><td>0</td><td>P5.1</td></tr>
<tr><td>0</td><td>1</td><td>XTAL2 (Output pin)</td></tr>
<tr><td>1</td><td>x</td><td>Reserved</td></tr>
</table>
Note: To enable external XTAL function, XTLCLK_EN (PWRCON[1:0]) bit, the external Crystal HXT or LXT enable control register must also be set.</td></tr>
<tr><td>[8]</td><td>P5_ ALT[0]</td><td>P5.0 Alternate Function Selection
The pin function of P5.0 depends on P5_MFP[0] and P5_ALT[0].
<table>
<tr><th>P5_ALT[0]</th><th>P5_MFP[0]</th><th>P5.0 function</th></tr>
<tr><td>0</td><td>0</td><td>P5.0</td></tr>
<tr><td>0</td><td>1</td><td>XTAL1 (Input pin)</td></tr>
<tr><td>1</td><td>x</td><td>Reserved</td></tr>
</table>
Note: To enable external XTAL function, XTLCLK_EN (PWRCON[1:0]) bit, the external Crystal HXT or LXT enable register must also be set.</td></tr>
<tr><td>[7:0]</td><td>P5_MFP[7:0]</td><td>P5 Multiple Function Selection
The pin function of P5 depends on P5_MFP and P5_ALT.
Refer to P5_ALT Description for details.</td></tr>
</table>

HIRC Trim Control Register (IRCTRIMCTL)

Register	Offset	R/W	Description	Reset Value
IRCTRIMCTL	GCR_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**
TRIM_RETRY_CNT		**TRIM_LOOP**		**Reserved**			**TRIM_SEL**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:6]	**TRIM_RETRY_CNT[1:0]**	**Trim Value Update Limitation Count** This field defines that how many times the auto trim circuit will try to update the HIRC trim value before the frequency of HIRC locked. Once the HIRC locked, the internal trim value update counter will be reset. If the trim value update counter reached this limitation value and frequency of HIRC still doesn't lock, the auto trim operation will be disabled and TRIM_SEL will be cleared to 00. **TRIM_RETRY_CNT** — **Trim Retry Count Limitation** 00 — Trim retry count limitation is 64 01 — Trim retry count limitation is 128 10 — Trim retry count limitation is 256 11 — Trim retry count limitation is 512
[5:4]	**TRIM_LOOP[1:0]**	**Trim Calculation Loop** This field defines that trim value calculation is based on how many LXT clocks in. For example, if TRIM_LOOP is set as 00, auto trim circuit will calculate trim value based on the average frequency difference in 4 LXT clock. 00 = Trim value calculation is based on average difference in 4 LXT clocks. 01 = Trim value calculation is based on average difference in 8 LXT clocks. 10 = Trim value calculation is based on average difference in 16 LXT clocks. 11 = Trim value calculation is based on average difference in 32 LXT clocks.
[3:1]	**Reserved**	Reserved.

Bits	Description	
[0]	**TRIM_SEL**	**Trim Frequency Selection** 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 Control Register (IRCTRIMIEN)

Register	Offset	R/W	Description	Reset Value
IRCTRIMIEN	GCR_BA+0x84	R/W	HIRC Trim Interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**32K_ERR_IEN**	**TRIM_FAIL_IEN**	**Reserved**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**32K_ERR_IEN**	**LXT Clock Error Interrupt Enable Control** This bit controls if CPU could get an interrupt while LXT clock is inaccurate during auto trim operation. If this bit is high, and 32K_ERR_INT is set during auto trim operation, an interrupt will be triggered to notify the LXT clock frequency is inaccurate. 0 = 32K_ERR_INT status Disabled to trigger an interrupt to CPU. 1 = 32K_ERR_INT status Enabled to trigger an interrupt to CPU.
[1]	**TRIM_FAIL_IEN**	**Trim Failure Interrupt Enable Control** 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 TRIM_SEL. If this bit is high and TRIM_FAIL_INT is set during auto trim operation, an interrupt will be triggered to notify that HIRC trim value update limitation count is reached. 0 = TRIM_FAIL_INT status Disabled to trigger an interrupt to CPU. 1 = TRIM_FAIL_INT status Enabled to trigger an interrupt to CPU.
[0]	**Reserved**	Reserved.

HIRC Trim Interrupt Status Register (IRCTRIMINT)

Register	Offset	R/W	Description	Reset Value
IRCTRIMINT	GCR_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					**32K_ERR_INT**	**TRIM_FAIL_INT**	**FREQ_LOCK**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**32K_ERR_INT**	**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 TRIM_SEL will be cleared to 0 by hardware automatically. If this bit is set and 32K_ERR_IEN 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]	**TRIM_FAIL_INT**	**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 TRIM_SEL will be cleared to 0 by hardware automatically. If this bit is set and TRIM_FAIL_IEN 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]	**FREQ_LOCK**	**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 (REGWRPROT)

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 REGWRPROT 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 REGPROTDIS status.

Register	Offset	R/W	Description	Reset Value
REGWRPROT	GCR_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**
REGWRPROT							**REGPROTDIS**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**REGWRPROT[7:0]**	**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 REGPROTDIS bit will be set to 1 and write-protection registers can be normal write.
[0]	**REGPROTDIS**	**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 6.2.5 Register Protection.

6.2.9 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".

6.2.9.1 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

6.2.9.2 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 Selection** 0 = Clock source is optional, refer to STCLK_S. 1 = Core clock used for SysTick timer.
[1]	**TICKINT**	**System Tick Interrupt Enabled** 0 = Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to 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 Control** 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[23:0]**	**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[23:0]**	**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).

6.2.10 Nested Vectored Interrupt Controller (NVIC)

6.2.10.1 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.

6.2.10.2 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".

6.2.10.3 Exception Model and System Interrupt Map

The following table lists the exception model supported by NuMicro™ Mini51 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 6.2-3 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	**BOD_OUT**	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 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	**UART_INT**	UART	UART interrupt	Yes
29	13	**-**	-	-	
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	**I2C_INT**	I^2C	I^2C interrupt	Yes

Exception Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Source Module	Interrupt Description	Power-Down Wake-Up
35 ~ 40	19 ~ 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 6.2-4 System Interrupt Map Vector Table

6.2.10.4 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 6.2-5 Vector Table Format

6.2.10.5 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.

6.2.10.6 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

6.2.10.7 NVIC Control Register Description

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[31:0]**	**Interrupt Enable Control** Enable one or more interrupts. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Write: 0 = No effect. 1 = Write 1 to enable associated interrupt. Read: 0 = Associated interrupt status Disabled. 1 = Associated interrupt status Enabled. 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[31:0]**	**Interrupt Disable Control** Disable one or more interrupts. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Write: 0 = No effect. 1 = Write 1 to disable associated interrupt. Read: 0 = Associated interrupt status Disabled. 1 = Associated interrupt status Enabled. 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[31:0]**	**Set Interrupt Pending Bits** Write: 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: 0 = Associated interrupt in not in pending status. 1 = Associated interrupt is in pending status. 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[31:0]**	**Clear Interrupt Pending Bits** Write: 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: 0 = Associated interrupt in not in pending status. 1 = Associated interrupt is in pending status. 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[1:0]**	**Priority Of IRQ3** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_2[1:0]**	**Priority Of IRQ2** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_1[1:0]**	**Priority Of IRQ1** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_0[1:0]**	**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[1:0]**	**Priority Of IRQ7** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_6[1:0]**	**Priority Of IRQ6** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_5[1:0]**	**Priority Of IRQ5** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_4[1:0]**	**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[1:0]**	**Priority Of IRQ11** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_10[1:0]**	**Priority Of IRQ10** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_9[1:0]**	**Priority Of IRQ9** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_8[1:0]**	**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[1:0]**	**Priority Of IRQ15** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_14[1:0]**	**Priority Of IRQ14** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_13[1:0]**	**Priority Of IRQ13** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_12[1:0]**	**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[1:0]**	**Priority Of IRQ19** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_18[1:0]**	**Priority Of IRQ18** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_17[1:0]**	**Priority Of IRQ17** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_16[1:0]**	**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[1:0]**	**Priority Of IRQ23** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_22[1:0]**	**Priority Of IRQ22** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_21[1:0]**	**Priority Of IRQ21** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_20[1:0]**	**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[1:0]**	**Priority Of IRQ27** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_26[1:0]**	**Priority Of IRQ26** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_25[1:0]**	**Priority Of IRQ25** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_24[1:0]**	**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[1:0]**	**Priority Of IRQ31** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	**Reserved**	Reserved.
[23:22]	**PRI_30[1:0]**	**Priority Of IRQ30** 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	**Reserved**	Reserved.
[15:14]	**PRI_29[1:0]**	**Priority Of IRQ29** 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	**Reserved**	Reserved.
[7:6]	**PRI_28[1:0]**	**Priority Of IRQ28** 0 denotes the highest priority and 3 denotes the lowest priority.
[5:0]	**Reserved**	Reserved.

6.2.10.8 Interrupt Source Control Registers

Besides the interrupt control registers associated with the NVIC, the NuMicro™ Mini51 series also implements some specific control registers to facilitate the interrupt functions, including "interrupt source identify", "NMI source selection", 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 (UART) Interrupt Source Identity	0xXXXX_XXXX
IRQ13_SRC	INT_BA+0x34	-	Reserved	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^2C) Interrupt Source Identity	0xXXXX_XXXX
IRQ19_SRC	INT_BA+0x4C	-	Reserved	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

Interrupt Source Identify Register (IRQn_SRC)

Register	Offset	R/W	Description	Reset Value
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 (UART) Interrupt Source Identity	0xXXXX_XXXX
IRQ13_SRC	INT_BA+0x34	-	Reserved	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^2C) Interrupt Source Identity	0xXXXX_XXXX
IRQ19_SRC	INT_BA+0x4C	-	Reserved	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
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

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC[2:0]**	**Interrupt Source** Define the interrupt sources for interrupt event.

Bits	Address	IRQ No.	Description
[31:3]	-	-	**Reserved**
[2:0]	INT_BA+0x00	0	Bit1~2: Reserved Bit0: BOD_INT
[2:0]	INT_BA+0x04	1	Bit1~2: Reserved Bit0: WDT_INT
[2:0]	INT_BA+0x08	2	Bit1~2: Reserved Bit0: EINT0 – external interrupt 0 from P3.2.
[2:0]	INT_BA+0x0C	3	Bit1~2: Reserved Bit0: EINT1 – external interrupt 1 from P5.2.
[2:0]	INT_BA+0x10	4	Bit2: Reserved Bit1: P1_INT Bit0: P0_INT
[2:0]	INT_BA+0x14	5	Bit2: P4_INT Bit1: P3_INT Bit0: P2_INT
[2:0]	INT_BA+0x18	6	Bit1~2: Reserved Bit0: PWM_INT

Bits	Address	IRQ No.	Description
[2:0]	INT_BA+0x1C	7	Bit1~2: Reserved Bit0: BRAKE_INT
[2:0]	INT_BA+0x20	8	Bit1~2: Reserved Bit0: TMR0_INT
[2:0]	INT_BA+0x24	9	Bit1~2: Reserved Bit0: TMR1_INT
[2:0]	INT_BA+0x28	10	**Reserved**
[2:0]	INT_BA+0x2C	11	**Reserved**
[2:0]	INT_BA+0x30	12	Bit1~2: Reserved Bit0: UART_INT
[2:0]	INT_BA+0x34	13	**Reserved**
[2:0]	INT_BA+0x38	14	Bit1~2: Reserved Bit0: SPI_INT
[2:0]	INT_BA+0x3C	15	**Reserved**
[2:0]	INT_BA+0x40	16	Bit1~2: Reserved Bit0: P5_INT
[2:0]	INT_BA+0x44	17	Bit1~2: Reserved Bit0: HIRC_TRIM_INT
[2:0]	INT_BA+0x48	18	Bit1~2: Reserved Bit0: I2C_INT
[2:0]	INT_BA+0x4C	19	**Reserved**
[2:0]	INT_BA+0x50	20	**Reserved**
[2:0]	INT_BA+0x54	21	**Reserved**
[2:0]	INT_BA+0x58	22	**Reserved**
[2:0]	INT_BA+0x5C	23	**Reserved**
[2:0]	INT_BA+0x60	24	**Reserved**
[2:0]	INT_BA+0x64	25	Bit1~2: Reserved Bit0: ACMP_INT
[2:0]	INT_BA+0x68	26	**Reserved**
[2:0]	INT_BA+0x6C	27	**Reserved**
[2:0]	INT_BA+0x70	28	Bit1~2: Reserved Bit0: PWRWU_INT
[2:0]	INT_BA+0x74	29	Bit1~2: Reserved Bit0: ADC_INT
[2:0]	INT_BA+0x78	30	**Reserved**
[2:0]	INT_BA+0x7C	31	**Reserved**

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 Control (Write Protect)** 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 REGWRPROT at address GCR_BA+0x100.
[7:5]	**Reserved**	Reserved.
[4:0]	**NMI_SEL[4:0]**	**NMI Interrupt Source Selection** 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[31:0]**	**MCU IRQ Source** The MCU_IRQ collects all the interrupts from the peripherals and generates the synchronous interrupt to Cortex™-M0 core. This modes to generate interrupt to Cortex™-M0 - the normal 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.

6.2.11 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".

6.2.11.1 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**				
CPUID	SCS_BA+0xD00	R	CPUID Register	0x410C_C200
ICSR	SCS_BA+0xD04	R/W	Interrupt Control State Register	0x0000_0000
AIRCR	SCS_BA+0xD0C	R/W	Application Interrupt and Reset Control Register	0xFA05_0000
SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000
SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000
SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

6.2.11.2 System Control Register Description

CPUID Register (CPUID)

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

31	30	29	28	27	26	25	24
IMPLEMENTER							
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 [7:0]**	**Implementer Code** Implementer code assigned by ARM®. (ARM® = 0x41).
[23:20]	**Reserved**	Reserved.
[19:16]	**PART[3:0]**	**Architecture Of The Processor** Read as 0xC for ARMv6-M parts.
[15:4]	**PARTNO[11:0]**	**Part Number Of The Processor** Read as 0xC20.
[3:0]	**REVISION[3:0]**	**Revision Number** Read as 0x0.

Interrupt Control State Register (ICSR)

Register	Offset	R/W	Description	Reset Value
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: 0 = No effect. 1 = Changes NMI exception state to pending. Read: 0 = NMI exception not pending. 1 = NMI exception 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: 0 = No effect. 1 = Changes PendSV exception state to pending. Read: 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: 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: 0 = No effect. 1 = Changes SysTick exception state to pending. Read: 0 = SysTick exception is not pending. 1 = SysTick exception is pending.
[25]	**PENDSTCLR**	**SysTick Exception Clear-pending Bit** Write: 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. This bit is read only.
[22]	**ISRPENDING**	**Interrupt Pending Flag, Excluding NMI And Faults** 0 = Interrupt not pending. 1 = Interrupt pending. This bit is read only.
[21]	**Reserved**	Reserved.
[20:12]	**VECTPENDING [8:0]**	**Exception Number Of The Highest Priority Pending Enabled Exception** 0 = No pending exceptions. Non-zero = Exception number of the highest priority pending enabled exception. This bit is read only.
[11:9]	**Reserved**	Reserved.
[8:0]	**VECTACTIVE [8:0]**	**Contains The Active Exception Number** 0 = Thread mode. Non-zero = Exception number of the currently active exception. This bit is read only.

Application Interrupt and Reset Control Register (AIRCR)

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

31	30	29	28	27	26	25	24
VECTORKEY							
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**	**VECTCLKACTIVE**	**Reserved**

Bits	Description	
[31:16]	**VECTORKEY [15:0]**	**Register Access Key** Write: 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: 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 (SCR)

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

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

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**SEVONPEND**	**Send Event On Pending 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 (SHPR2)

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

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

Bits	Description	
[31:30]	**PRI_11[1:0]**	**Priority Of System Handler 11 – SVCall** 0 denotes the highest priority and 3 denotes the lowest priority.
[29:0]	**Reserved**	Reserved.

System Handler Priority Register 3 (SHPR3)

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

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

Bits	Description	
[31:30]	**PRI_15[1:0]**	**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[1:0]**	**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 PWR_DOWN_EN (PWRCON[7]) bit and PD_WAIT_CPU (PWRCON[8]) bit are both 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
- 22.1184 MHz internal high speed RC oscillator (HIRC)
- 10 kHz internal low speed RC oscillator (LIRC)

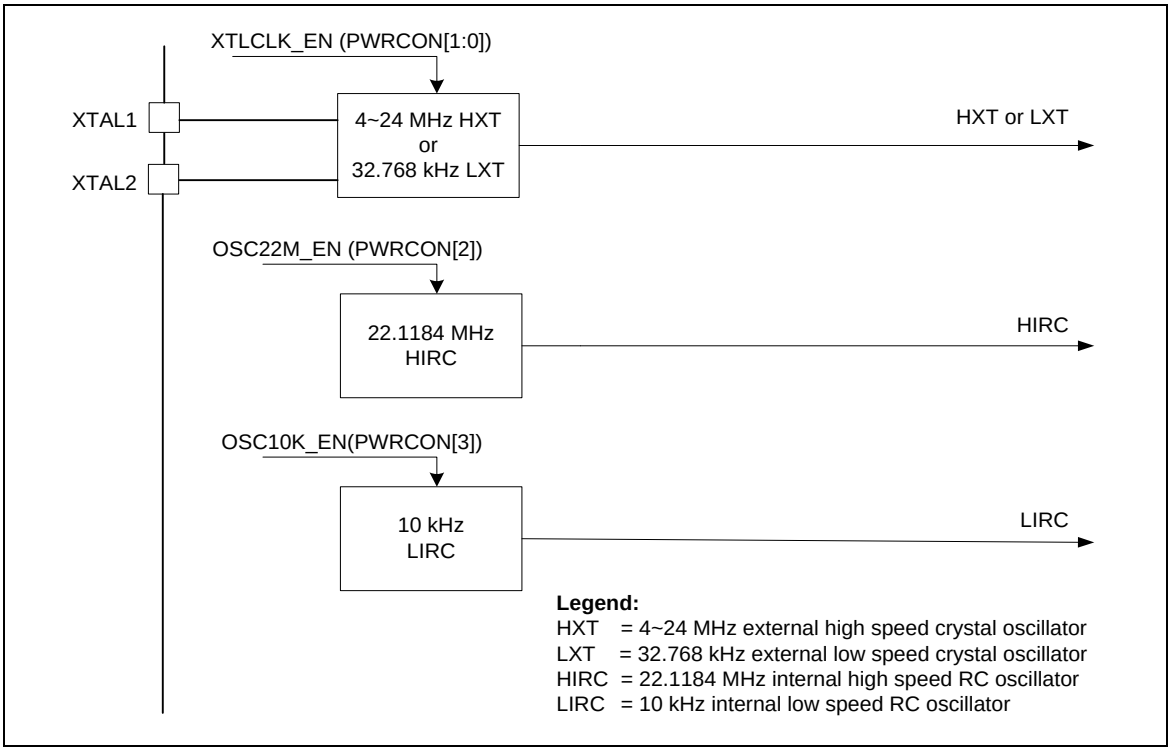
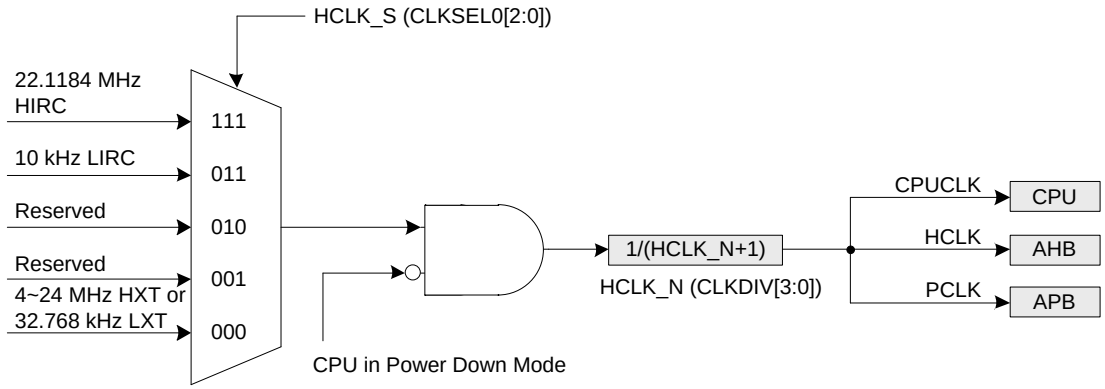


Figure 6.3-1 Clock Generator Block Diagram

6.3.2 System Clock and SysTick Clock

The system clock has three clock sources which are generated from clock generator block. The clock source switches depending on the register HCLK_S (CLKSEL0[2:0]). The block diagram is shown below.

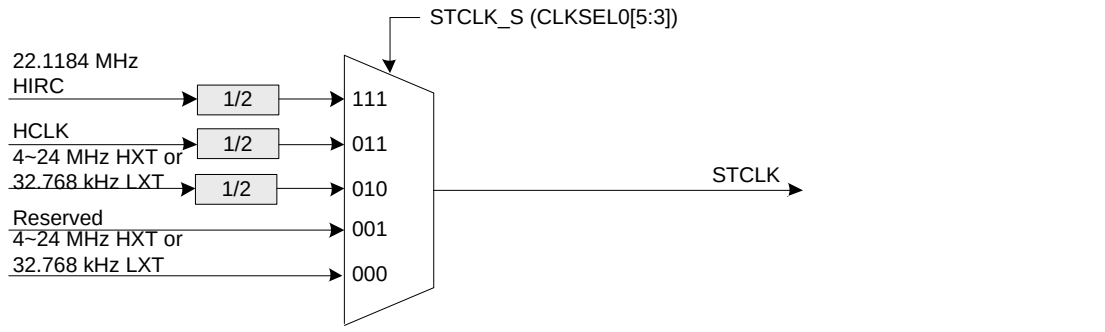


Legend:
HXT = 4~24 MHz external high speed crystal oscillator
HIRC = 22.1184 MHz internal high speed RC oscillator
LIRC = 10 kHz internal low speed RC oscillator

Note: Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6.3-2 System Clock Block Diagram

The clock source of SysTick in Cortex™-M0 core can use CPU clock or external clock (SYST_CSR[2]). If using external clock, the SysTick clock (STCLK) has 4 clock sources. The clock source switches depending on the setting of the register STCLK_S (CLKSEL0[5:3]). The block diagram is shown below.



Legend:
HXT = 4~24 MHz external high speed crystal oscillator
HIRC = 22.1184 MHz internal high speed RC oscillator
LIRC = 10 kHz internal low speed RC oscillator

Note: Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6.3-3 SysTick Clock Control Block Diagram

6.3.3 ISP Clock Source Selection

The clock source of ISP is from AHB clock (HCLK). Please refer to the register AHBCLK.

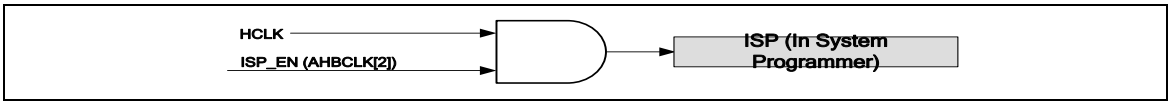


Figure 6.3-4 AHB Clock Source for HCLK

6.3.4 Module Clock Source Selection

The peripheral clock has different clock source switch settings depending on different peripherals. Please refer to the CLKSEL1 and APBCLK register description in section 6.3.8.

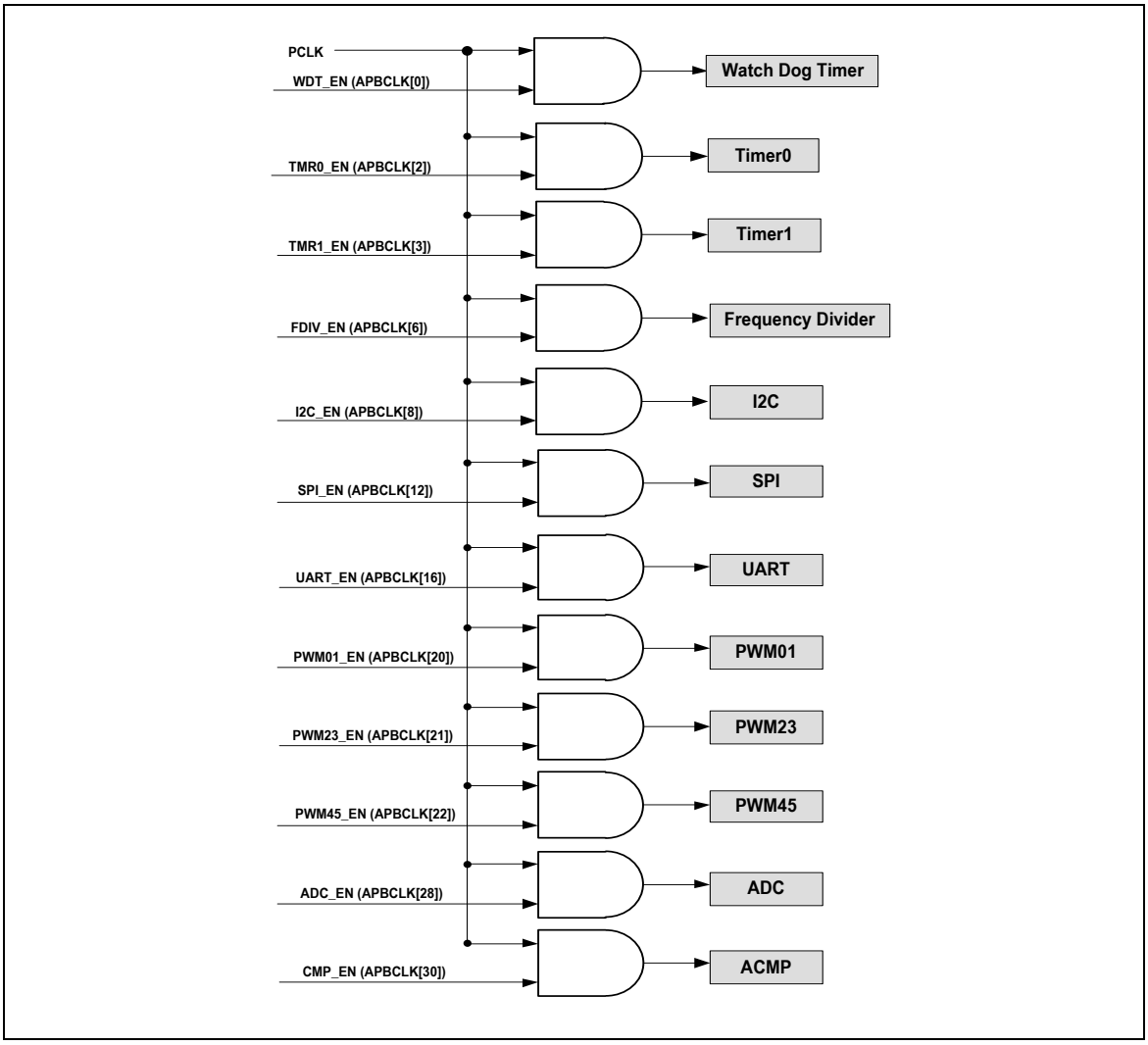


Figure 6.3-5 Peripherals Clock Source Selection for PCLK

	Ext. CLK (HXT Or LXT)	**HIRC**	**LIRC**	**PCLK**

WDT	Yes	No	Yes	Yes
Timer0	Yes	Yes	Yes	Yes
Timer1	Yes	Yes	Yes	Yes
I^2C	No	No	No	Yes
SPI	No	No	No	Yes
UART	Yes	Yes	No	No
PWM	No	No	No	Yes
ADC	Yes	Yes	No	Yes
ACMP	No	No	No	Yes

Table 6.3-1 Peripheral Clock Source Selection Table

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 PD_32K = 1 and XTLCLK_EN[1:0] = 10)
- Peripherals Clock (When 10 kHz low speed oscillator is adopted as clock source)
  - Watchdog Clock
  - Timer 0/1 Clock

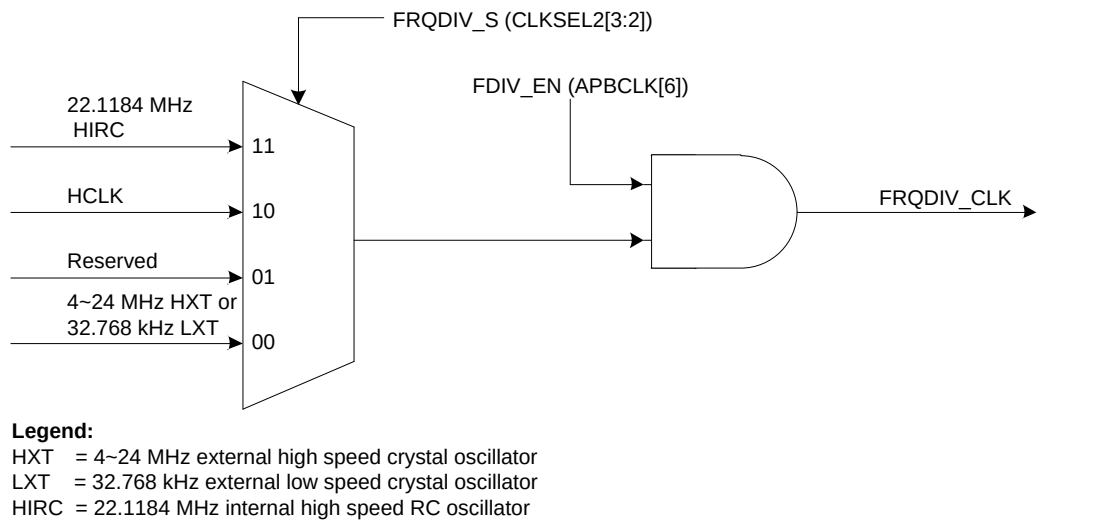
6.3.6 Frequency Divider Output

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

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

if DIVIDER1(FRQDIV[5]) is set to 1, the frequency divider clock (FRQDIV_CLK) will bypass power-of-2 frequency divider. The frequency divider clock will be output to CKO pin directly.



Legend:
HXT = 4~24 MHz external high speed crystal oscillator
LXT = 32.768 kHz external low speed crystal oscillator
HIRC = 22.1184 MHz internal high speed RC oscillator

Note: Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6.3-6 Clock Source of Frequency Divider

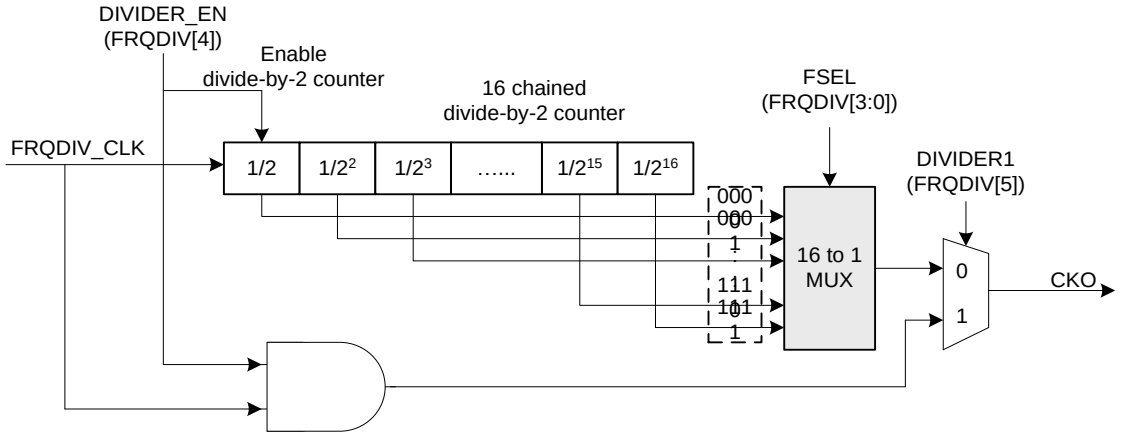


Figure 6.3-7 Block Diagram of Frequency Divider

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**				
PWRCON	CLK_BA+0x00	R/W	System Power-down Control Register	0x0000_001C
AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_0005
APBCLK	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register	0x0000_0001
CLKSTATUS	CLK_BA+0x0C	R/W	Clock Status Monitor Register	0x0000_0018
CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0000_003F
CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0xAFFF_FFFF
CLKDIV	CLK_BA+0x18	R/W	Clock Divider Number Register	0x0000_0000
CLKSEL2	CLK_BA+0x1C	R/W	Clock Source Select Control Register 2	0x0000_00EF
FRQDIV	CLK_BA+0x24	R/W	Frequency Divider Control Register	0x0000_0000

6.3.8 Register Description

Power-down Control Register (PWRCON)

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 REGWRPROT register at address GCR_BA + 0x100.

Register	Offset	R/W	Description	Reset Value
PWRCON	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						PD_32K	Reserved
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PWR_DOWN_EN	PD_WU_STS	PD_WU_INT_EN	PD_WU_DLY	OSC10K_EN	OSC22M_EN	XTLCLK_EN	

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**PD_32K**	**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 XTLCLK_EN[1:0] = 10, LXT is still active in Power-down mode.
[8]	**Reserved**	Reserved.
[7]	**PWR_DOWN_EN**	**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 system clock is 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]	**PD_WU_STS**	**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, I^2C, ACMP, Timer or BOD wake-up occurred. **Note:** This bit works only if PD_WU_INT_EN (PWRCON[5]) set to 1. Write 1 to clear the bit to 0.
[5]	**PD_WU_INT_EN**	**Power-down Mode Wake-up Interrupt Enable Control (Write Protect)** 0 = Disabled. 1 = Enabled. **Note:** The interrupt will occur when both PD_WU_STS and PD_WU_INT_EN are high.
[4]	**PD_WU_DLY**	**Wake-up Delay Counter Enable Control (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]	**OSC10K_EN**	**10 KHz Internal Low Speed RC Oscillator (LIRC) Enable Control (Write Protect)** 0 = 10 kHz internal low speed RC oscillator (LIRC) Disabled. 1 = 10 kHz internal low speed RC oscillator (LIRC) Enabled.
[2]	**OSC22M_EN**	**22.1184 MHz Internal High Speed RC Oscillator (HIRC) Enable Control (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 OSC22M_EN bit is 1.
[1:0]	**XTLCLK_EN[1:0]**	**External Crystal HXT Or LXT Enable Control (Write Protect)** The default clock source is from HIRC. These two bits are default set to "00" and the XTAL1 and XTAL2 pins are GPIO. 00 = XTAL1 and XTAL2 are GPIO, disable both LXT & HXT (default). 01 = HXT Enabled. 10 = LXT Enabled. 11 = XTAL1 is external clock input pin, XTAL2 is GPIO. **Note:** To enable the external XTAL function, the P5_ALT[1:0] and P5_MFP[1:0] bits must also be set in P5_MFP.

Register Or Instruction / Mode	SLEEPDEEP (SCR[2])	PWR_DOWN_EN (PWRCON[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 periperal 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 PWR_DOWN_EN bit in PWRCON[7] to ensure chip can enter Power-down and wake-up successfully.

AHB Devices Clock Enable Control Register (AHBCLK)

The bit in this register is used to enable/disable clock for system clock.

Register	Offset	R/W	Description	Reset Value
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					**ISP_EN**	**Reserved**	**Reserved**

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

APB Devices Clock Enable Control Register (APBCLK)

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

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

31	30	29	28	27	26	25	24
Reserved	**ACMP_EN**	**Reserved**	**ADC_EN**	**Reserved**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**PWM45_EN**	**PWM23_EN**	**PWM01_EN**	**Reserved**			**UART_EN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**SPI_EN**	**Reserved**			**I2C_EN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**FDIV_EN**	**Reserved**		**TMR1_EN**	**TMR0_EN**	**Reserved**	**WDT_EN**

Bits	Description	
[31]	**Reserved**	Reserved.
[30]	**ACMP_EN**	**Analog Comparator Clock Enable Control** 0 = Analog Comparator clock Disabled. 1 = Analog Comparator clock Enabled.
[29]	**Reserved**	Reserved.
[28]	**ADC_EN**	**Analog-digital-converter (ADC) Clock Enable Control** 0 = ADC peripheral clock Disabled. 1 = ADC peripheral clock Enabled.
[27:23]	**Reserved**	Reserved.
[22]	**PWM45_EN**	**PWM_45 Clock Enable Control** 0 = PWM45 clock Disabled. 1 = PWM45 clock Enabled.
[21]	**PWM23_EN**	**PWM_23 Clock Enable Control** 0 = PWM23 clock Disabled. 1 = PWM23 clock Enabled.
[20]	**PWM01_EN**	**PWM_01 Clock Enable Control** 0 = PWM01 clock Disabled. 1 = PWM01 clock Enabled.
[19:17]	**Reserved**	Reserved.
[16]	**UART_EN**	**UART Clock Enable Control** 0 = UART clock Disabled. 1 = UART clock Enabled.
[15:13]	**Reserved**	Reserved.

Bits	Description	
[12]	**SPI_EN**	**SPI Peripheral Clock Enable Control** 0 = SPI peripheral clock Disabled. 1 = SPI peripheral clock Enabled.
[11:9]	**Reserved**	Reserved.
[8]	**I2C_EN**	**I^2C Clock Enable Control** 0 = I^2C clock Disabled. 1 = I^2C clock Enabled.
[7]	**Reserved**	Reserved.
[6]	**FDIV_EN**	**Frequency Divider Output Clock Enable Control** 0 = FDIV clock Disabled. 1 = FDIV clock Enabled.
[5:4]	**Reserved**	Reserved.
[3]	**TMR1_EN**	**Timer1 Clock Enable Control** 0 = Timer1 clock Disabled. 1 = Timer1 clock Enabled.
[2]	**TMR0_EN**	**Timer0 Clock Enable Control** 0 = Timer0 clock Disabled. 1 = Timer0 clock Enabled.
[1]	**Reserved**	Reserved.
[0]	**WDT_EN**	**Watchdog Timer Clock Enable Control (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 REGWRPROT at address GCR_BA + 0x100.

Clock Status Register (CLKSTATUS)

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
CLKSTATUS	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**
CLK_SW_FAIL	**Reserved**		**OSC22M_STB**	**OSC10K_STB**	**Reserved**		**XTL_STB**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**CLK_SW_FAIL**	**Clock Switch Fail Flag** 0 = Clock switching success. 1 = Clock switching failed. **Note1:** 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. **Note2:** This bit is read only. After selected clock source is stable, hardware will switch system clock to selected clock automatically, and CLK_SE_FAIL will be cleared automatically by hardware.
[6:5]	**Reserved**	Reserved.
[4]	**OSC22M_STB**	**HIRC Clock Source Stable Flag (Read Only)** 0 = HIRC clock is not stable or disabled. 1 = HIRC clock is stable.
[3]	**OSC10K_STB**	**LIRC Clock Source Stable Flag (Read Only)** 0 = LIRC clock is not stable or disabled. 1 = LIRC clock is stable.
[2:1]	**Reserved**	Reserved.
[0]	**XTL_STB**	**HXT Or LXT Clock Source Stable Flag** 0 = HXT or LXT clock is not stable or disabled. 1 = HXT or LXT clock is stable.

Clock Source Select Control Register 0 (CLKSEL0)

Register	Offset	R/W	Description	Reset Value
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		**STCLK_S**			**HCLK_S**		

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:3]	**STCLK_S[2:0]**	**Cortex™-M0 SysTick Clock Source Selection From Reference Clock (Write Protect)** If SYST_CSR[2] = 1, SysTick clock source is from HCLK. If 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 REGWRPROT at address GCR_BA + 0x100. **Note2:** If the SysTick clock source is not from HCLK (i.e. SYST_CSR[2] = 0), SysTick clock source must less than or equal to HCLK/2. **Note3:** To set PWRCON[1:0], select HXT or LXT crystal clock.
[2:0]	**HCLK_S[2:0]**	**HCLK Clock Source Selection (Write Protect)** 000 = Clock source is from HXT or LXT. 001 = Reserved. 010 = Reserved. 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 REGWRPROT at address GCR_BA + 0x100. **Note3:** To set PWRCON[1:0], select HXT or LXT crystal clock.

Clock Source Select Control Register 1 (CLKSEL1)

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

Register	Offset	R/W	Description	Reset Value
CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0xAFFF_FFFF

31	30	29	28	27	26	25	24
Reserved						**UART_S**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**TMR1_S**			**Reserved**	**TMR0_S**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**SPI_S**	**ADC_S**		**WDT_S**	

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25:24]	**UART_S[1:0]**	**UART Clock Source Selection** 00 = Clock source is from HXT or LXT. 01 = Reserved. 10 = Clock source is from HIRC. 11 = Clock source is from HIRC . **Note:** To set PWRCON[1:0], select HXT or LXT crystal clock.
[23:15]	**Reserved**	Reserved.
[14:12]	**TMR1_S[2:0]**	**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 PWRCON[1:0], select HXT or LXT crystal clock.
[11]	**Reserved**	Reserved.
[10:8]	**TMR0_S[2:0]**	**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 PWRCON[1:0], select HXT or LXT crystal clock.

Bits	Description	
[7:5]	**Reserved**	Reserved.
[4]	**SPI_S**	**SPI Clock Source Selection** 0 = Clock source is from HXT or LXT. 1 = Clock source is from HCLK. **Note:** To set PWRCON[1:0], select HXT or LXT crystal clock.
[3:2]	**ADC_S[1:0]**	**ADC Peripheral Clock Source Selection** 00 = Clock source is from HXT or LXT. 01 = Reserved. 10 = Clock source is from HCLK. 11 = Clock source is from HIRC. **Note:** To set PWRCON[1:0], select HXT or LXT crystal clock.
[1:0]	**WDT_S[1:0]**	**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 REGWRPROT at address GCR_BA + 0x100. **Note2:** To set PWRCON[1:0], select HXT or LXT crystal clock.

Clock Source Select Control Register (CLKSEL2)

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

Register	Offset	R/W	Description	Reset Value
CLKSEL2	CLK_BA+0x1C	R/W	Clock Source Select Control Register 2	0x0000_00EF

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

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3:2]	**FRQDIV_S[1:0]**	**Clock Divider Clock Source Selection** 00 = Clock source is from HXT or LXT. 01 = Reserved. 10 = Clock source is from HCLK. 11 = Clock source is from HIRC. **Note:** To set PWRCON[1:0], select HXT or LXT crystal clock.
[1:0]	**Reserved**	Reserved.

Clock Divider Register (CLKDIV)

Register	Offset	R/W	Description	Reset Value
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**
ADC_N							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**UART_N**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**HCLK_N**			

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

Frequency Divider Control Register (FRQDIV)

Register	Offset	R/W	Description	Reset Value
FRQDIV	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		**DIVIDER1**	**DIVIDER_EN**	**FSEL**			

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**DIVIDER1**	**Frequency Divider 1 Enable Control** 0 = Divider output frequency is depended on FSEL value. 1 = Divider output frequency is the same as input clock frequency.
[4]	**DIVIDER_EN**	**Frequency Divider Enable Control** 0 = Frequency Divider Disabled. 1 = Frequency Divider Enabled.
[3:0]	**FSEL[3:0]**	**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 FSEL[3:0].

6.4 Flash Memory Controller (FMC)

6.4.1 Overview

The NuMicro Mini51™ series is equipped with 4K/8K/16K bytes on chip embedded flash memory for application program (APROM) that can be updated through ISP procedure. In-System-Programming (ISP) and In-Application-Programming (IAP) enable user to update program memory when chip is soldered on PCB. After chip power on Cortex™-M0 CPU fetches code from APROM or LDROM decided by boot select (CBS) in CONFIG0. By the way, the NuMicro Mini51™ series also provides Data Flash region that is shared with APROM and its start address is configurable and defined by user in CONFIG1.

6.4.2 Features

- Running up to 24 MHz with zero wait state for discontinuous address read access
- 4/8/16 Kbytes application program memory (APROM)
- 2 Kbytes in system programming (ISP) loader program memory (LDROM)
- Programmable Data Flash start address
- All embedded flash memory supports 512 bytes page erase
- In System Program (ISP)/In Application Program (IAP) to update on chip flash memory

6.4.3 Block Diagram

The flash memory controller consists of ISP control logic, writer interface and flash macro interface timing control logic. The block diagram of flash memory controller is shown in the following figure:

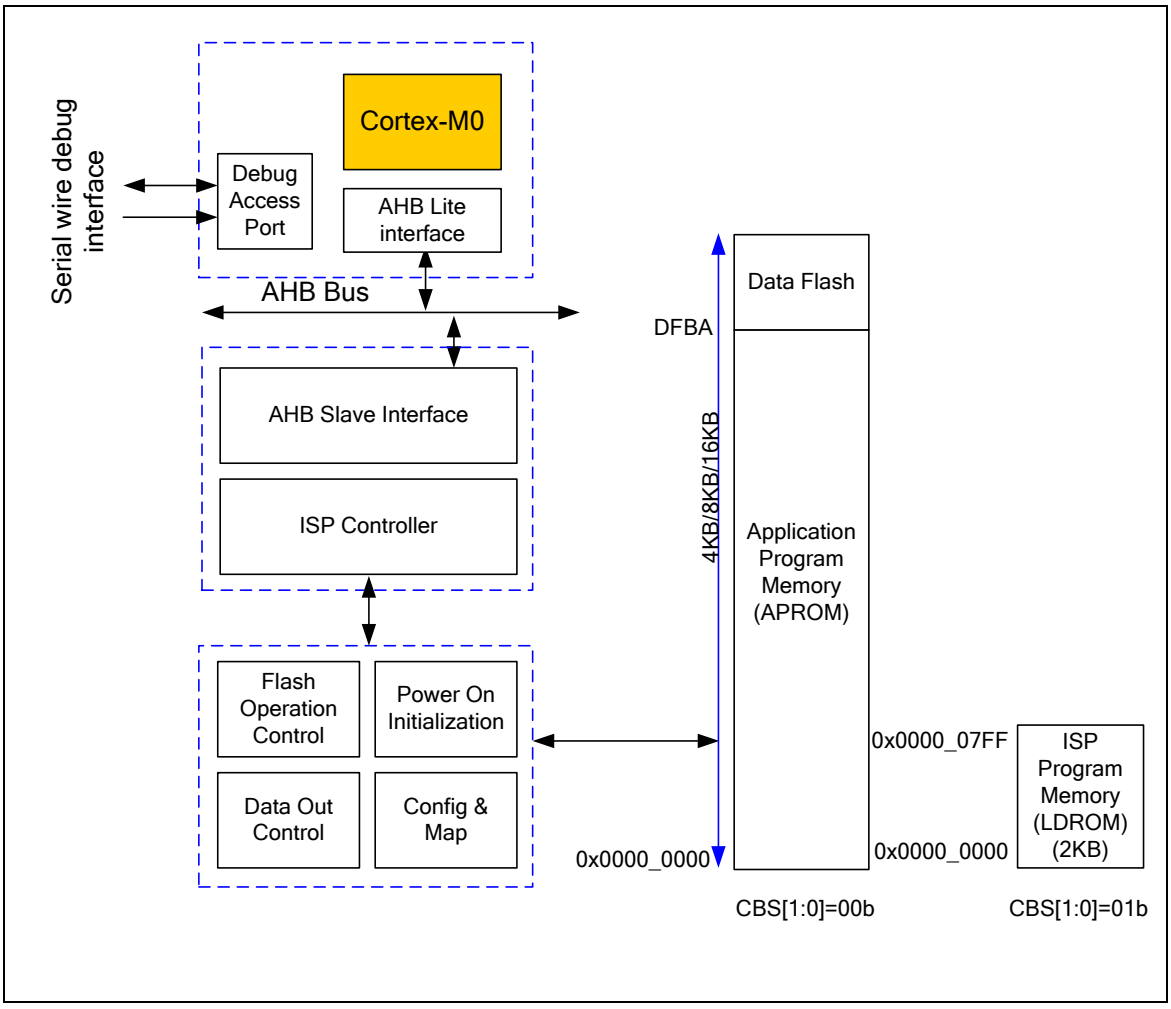


Figure 6.4-1 Flash Memory Control Block Diagram

6.4.4 Flash Memory Organization

The NuMicro Mini51™ flash memory consists of Application program memory (APROM), Data Flash, ISP loader program memory (LDROM), and user configuration.

APROM is main memory for user applications. User can write their application to APROM and set system to boot from APROM.

LDROM is designed for a loader to implement In-System-Programming function. LDROM is independent to APROM and system can also be set to boot from LDROM. Therefore, user can use LDROM to avoid system boot fail when code of APROM was corrupted.

Data Flash is used for user to store data. It can be read by ISP read or memory read and

programmed through ISP register. The size of each erase unit is 512 bytes. Data Flash is shared with APROM. The size and start address are defined CONFIG1.

User configuration provides two words to control system logic, such as flash security lock, boot select, Brown-out voltage level, Data Flash base address, etc…. User configuration works like a fuse for power on setting and loaded from flash memory to its corresponding control register during chip powers on.

In the NuMicro™ Family, the flash memory organization is different to system memory map. Flash memory organization is used when user using ISP command to read, program or erase flash memory. System memory map is used when CPU access flash memory to fetch code or data. For example, When system is set to boot from LDROM by CBS[1:0] = 01b, CPU will be able to fetch code of LDROM from 0x0 ~ 0x7FF. However, if user want to read LDROM by ISP, they still need to read the address of LDROM as 0x0010_0000 ~ 0x0010_07FF.

The following table and figure show the address mapping information of APROM, LDROM, Data Flash and user configuration.

Block Name	DFEN	Size	Start Address	End Address
AP-ROM	0	(4-0.5*N) Kbytes / (8-0.5*N) Kbytes / (16-0.5*N) Kbytes	0x0000_0000	DFBA-1
AP-ROM	1	4 Kbytes / 8 Kbytes / 16 Kbytes	0x0000_0000	0x0000_0FFF / 0x0000_1FFF / 0x0000_3FFF
Data Flash	0	0.5*N Kbytes	DFBA	0x0000_0FFF / 0x0000_1FFF / 0x0000_3FFF
Data Flash	1	0 Kbytes	N/A	N/A
LD-ROM	x	2 Kbytes	0x0010_0000	0x0010_07FF
User Configuration	x	2 words	0x0030_0000	0x0030_0007

Table 6.4-1 Flash Memory Address Map

The Flash memory organization is shown below:

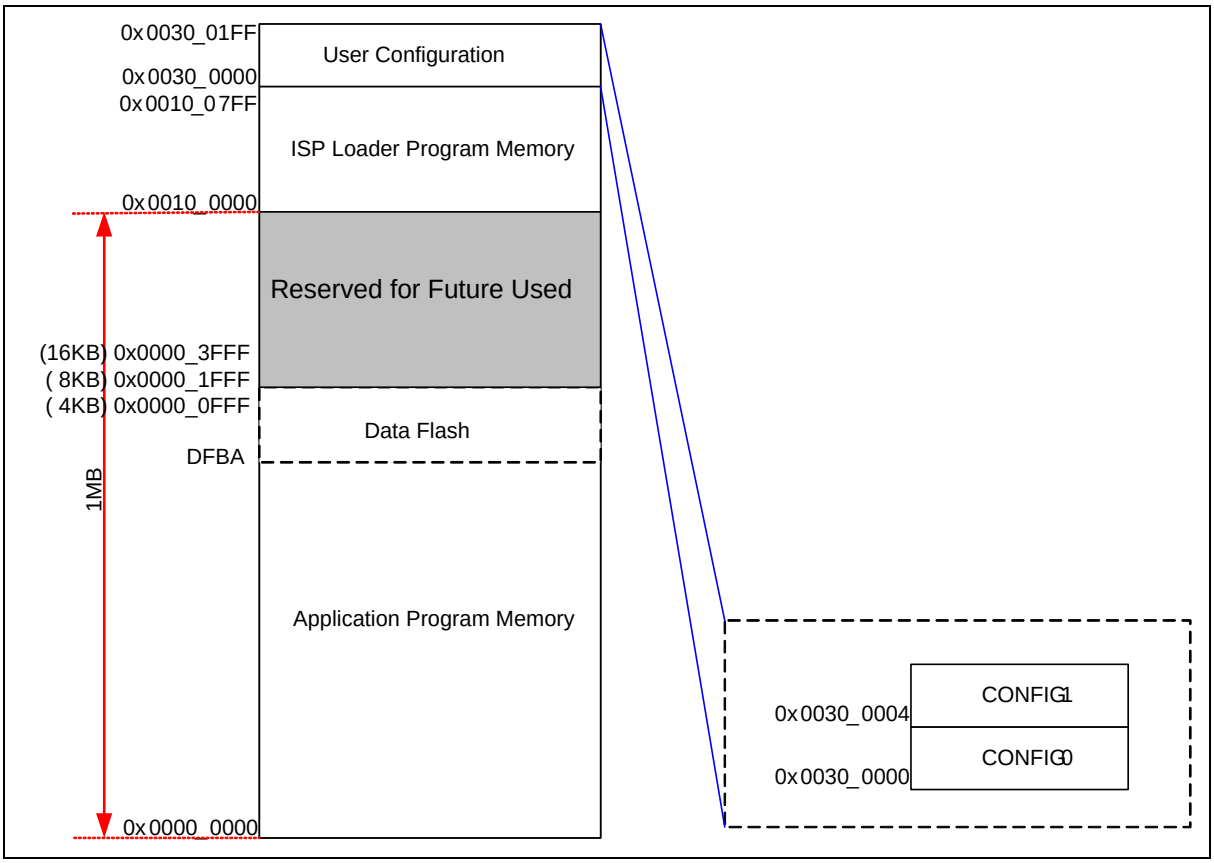


Figure 6.4-2 Flash Memory Organization

6.4.5 User Configuration

User configuration is internal programmable configuration area for boot options. The user configuration is located at 0x300000 of Flash Memory Organization and they are two words. Any change on user configuration will take effect after system reboot.

CONFIG0 (Address = 0x0030_0000)

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

Config0	Address = 0x0030_0000	
Bits	**Description**	
[31:24]	**Reserved**	Reserved
[23]	**CBOVEXT**	**Brown-out Voltage Selection Extension** 0 = Brown-out voltage selection includes 2.2V, 2.7V, 3.7V and 4.4V 1 = Brown-out voltage selection includes BOD disable mode. To disable BOD function, CBOV[1:0] must be 11b. See table in CBOV.
[22:21]	**CBOV[1:0]**	**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	11	Disable BOD function

[20]	**CBORST**	**Brown-out Reset Enable Control** 0 = Brown-out reset Enabled after power on. 1 = Brown-out reset Disabled after power on.
[19:11]	**Reserved**	Reserved
[10]	**CIOINI**	**IO Initial State Selection** 0 = Quasi bi-direction mode. All GPIO default to be Quasi bi-direction mode after chip power on. 1 = Input tri-state mode. All GPIO default to be input tri-state mode after power on. **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 Boot Selection** 00 = LDROM with IAP function. 01 = LDROM without IAP function.

Config0	Address = 0x0030_0000	
Bits	**Description**	
		10 = APROM with IAP function. 11 = APROM without IAP function. The NuMicro Mini51™, user can set CBS[0] = 0 to support IAP function. When CBS[0] = 0, the LDROM is mapping to address 0x100000 and APROM is mapping to address 0x0. User could access them by their address without boot switching. In other words, if IAP function is supported, the code in LDROM and APROM can be called by each other. **Note1:** BS bit of ISPCON is only can be used to control boot switching when CBS[0] = 1. CBS[0] means bit 6 of this register. **Note2:** VECMAP is only can be used to remap page 0 of APROM or LDROM to 0x0~0x1ff when CBS[0] = 0.
[5:2]	**Reserved**	Reserved
[1]	**LOCK**	**Security Lock** 0 = Flash data locked. 1 = Flash data unlocked. When flash data is locked, only device ID, unique ID, user configuration can be read by writer and ICP through serial debug interface. Others data is locked as 0xFFFFFFFF. ISP can read data anywhere regardless of LOCK bit value.
[0]	**DFEN**	**Data Flash Enable Control** 0 = Data Flash Enabled. 1 = Data Flash Disabled.

Note: The reserved bits of user configuration should be kept as '1'.

CONFIG1 (Address = 0x0030_0004)

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

Config1	Address = 0x0030_0004	
Bits	**Description**	
[31:14]	**Reserved**	Reserved
[13:0]	**DFBA[13:0]**	**Data Flash Base Address** 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.

6.4.5.1 Data Flash

The NuMicro Mini51™ provides Data Flash for user to store data which is read/write thru ISP registers. The Data Flash base address is defined by DFBA(CONFIG0[0]) is enabled. For example for 4K/2K/1K/0KB Data Flash, the DFBA setting value is listed in the following table. The erase unit is 512 bytes. When a word will be changed, all 128 words need to be copied to another page or SRAM in advance.

Data Flash	4KB (DFEN=0)	2KB (DFEN=0)	1KB (DFEN=0)	0KB (DFEN=1)
16K Flash	DFBA=0x0000_3000	DFBA=0x0000_3800	DFBA=0x0000_3C00	DFEN=1
8K Flash	DFBA=0x0000_1000	DFBA=0x0000_1800	DFBA=0x0000_1C00	DFEN=1
4K Flash	Forbidden	DFBA=0x0000_0800	DFBA=0x0000_0C00	DFEN=1

Table 6.4-2 Data Flash Table

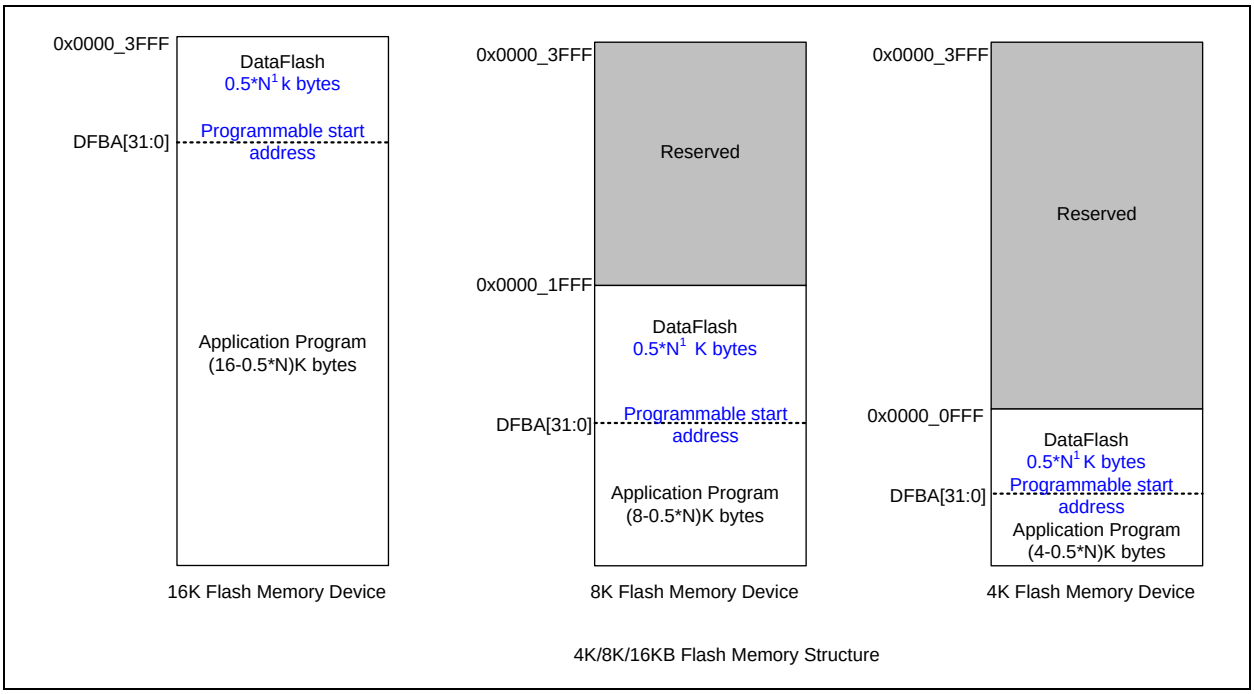


Figure 6.4-3 Flash Memory Structure

6.4.5.2 Brown-out Detection

The NuMicro Mini51™ includes Brown-Out detection function for monitoring the voltage on AV_{DD} pin. If AV_{DD} voltage falls below level setting of both CBOVEXT and CBOV, the BOD event will be triggered when BOD enabled. When BOD event is detected, user can decide to trigger BOD reset by enabling CBORST or just raise BOD interrupt by NVIC. Because BOD reset is issued whenever AV_{DD} voltage falls below the level setting of CBOV, user must make sure the CBOV setting to avoid BOD reset right after BOD reset enabled. For example, if the AV_{DD} is 3.3V and CBOVEXT is 0b, CBOV could only be 00b or 01b. Otherwise, the system will be halted in BOD reset state when BOD reset is enabled and CBOV is 10b or 11b.

6.4.6 *Boot* Selection

The NuMicro Mini51™ provides in system programming (ISP) feature to update APROM when chip is mounted on PCB. A dedicated 2 Kbytes LDROM is used to store ISP firmware. Users can select to start program fetch from APROM or LDROM by CBS[1](CONFIG0[7]).

In addition to setting boot from APROM or LDROM, CBS(CONFIG0[7:6]) is also used to control system memory map after booting. The value of CBS[1](CONFIG0[7]) will be loaded to BS(ISPCON[1]) after booting.

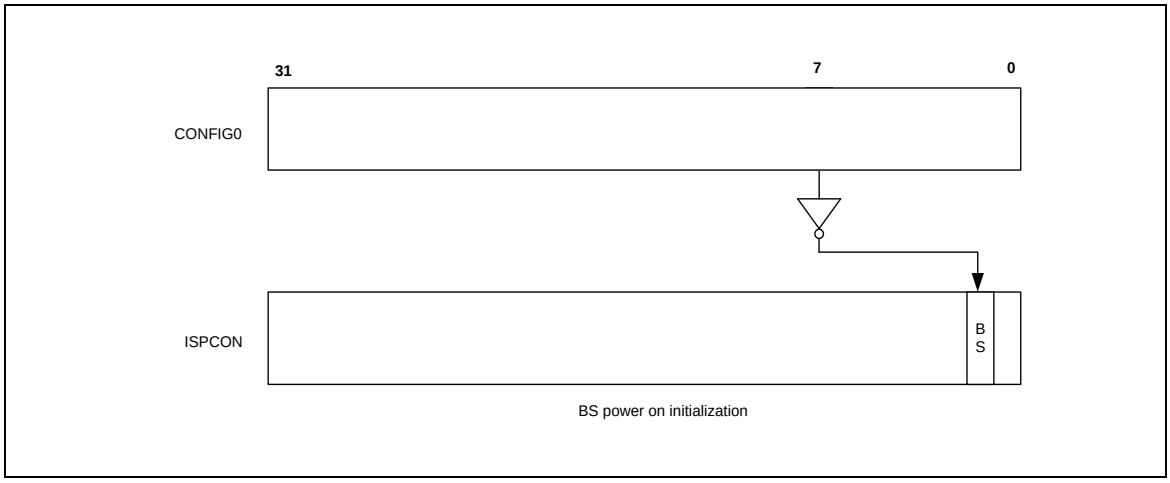


Figure 6.4-4 Boot Select (BS) for Power-on Action

When CBS[0] = 1 and set CBS[1] = 1 to boot from APROM, the application in APROM will not be able to access LDROM by CPU read. In other words, when CBS[0] = 1 and CBS[1] = 0 are set to boot from LDROM, the software executed in LDROM will not be able to access APROM by memory read. The following figure shows the memory map when booting from APROM and LDROM.

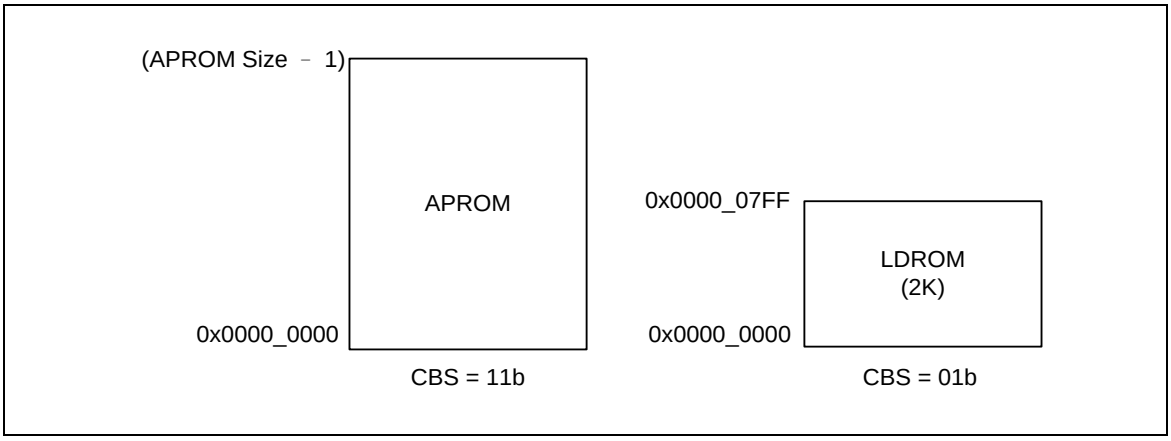


Figure 6.4-5 Program Executing Range for boot from APROM and boot from LDROM

For the application that software needs to execute code in APROM and call the functions in LDROM or to execute code in LDROM and call the APROM function without changing boot mode, CBS[0] needs to be set as 0 and this is called In-Application-Programming(IAP).

CBS[1:0]	Boot Selection
00	**LDROM with IAP function** Chip booting from LDROM, program executing range including LDROM and most of APROM (all except first 512 bytes as the first 512 bytes is mapped from LDROM). LDROM address is mapping to 0x0010_0000 ~ 0x0010_07FF, and also the first 512 bytes of LDROM is mapping to the address 0x0000_0000 ~ 0x0000_01FF. The address 0x0000_0000 ~ 0x0000_01FF can be re-mapping to any other page within executing range though ISP command. Both APROM and LDROM are programmable in this mode no matter the code is currently running on LDROM or APROM. Data Flash is meaningless in this mode, because any area of APROM and LDROM can just be used as the Data Flash and DFBA is not functioned in this mode.
01	**LDROM without IAP function** Chip booting from LDROM, program executing range only including LDROM; APROM cannot be access by program directly, except by through ISP. LDROM is write-protected in this mode.
10	**APROM with IAP function** Chip booting from APROM, program executing range including LDROM and APROM LDROM address is mapping to 0x0010_0000~0x0010_07FF The address 0x0000_0000 ~ 0x0000_01FF can be re-mapping to any other page within executing range though ISP command. Both APROM and LDROM are programmable in this mode no matter the code is currently running on LDROM or APROM. Data Flash is meaningless in this mode, because any area of APROM and LDROM can just be used as the Data Flash and DFBA is not functioned in this mode.
11	**APROM without IAP function** Chip booting from APROM and program executing range only including APROM. LDROM cannot be access by program directly, except by through ISP. APROM is write-protected in this mode.

Table 6.4-3 Supported Boot Selection Options

6.4.7 In-Application-Programming (IAP)

The NuMicro Mini51™ Series provides In-application-programming (IAP) function for user to switch the code executing between APROM and LDROM without reset. User can enable the IAP function by re-booting chip and setting the chip boot selection bits in CBS[1:0](CONFIG0[7:6]) as 10b or 00b.

In the case that the chip boots from APROM with the IAP function enabled (CBS[1:0] = 10b), the executable range of code includes all of APROM and LDROM. The address space of APROM is kept as the original size but the address space of the 2 KB LDROM is mapped to 0x0010_0000~ 0x0010_07FF.

In the case that the chip boots from LDROM with the IAP function enabled (CBS[1:0] = 00b), the executable range of code includes all of LDROM and almost all of APROM except for its first page. User cannot access the first page of APROM because the first page of executable code range becomes the mirror of the first page of LDROM as set by default. Meanwhile, the address space of 2 KB LDROM is mapped to 0x0010_0000~0x0010_07FF, and the first page of LDROM is mapped to 0x0000_0000~0x0000_01FF.

Please refer to the following figure for the address map while IAP is activating.

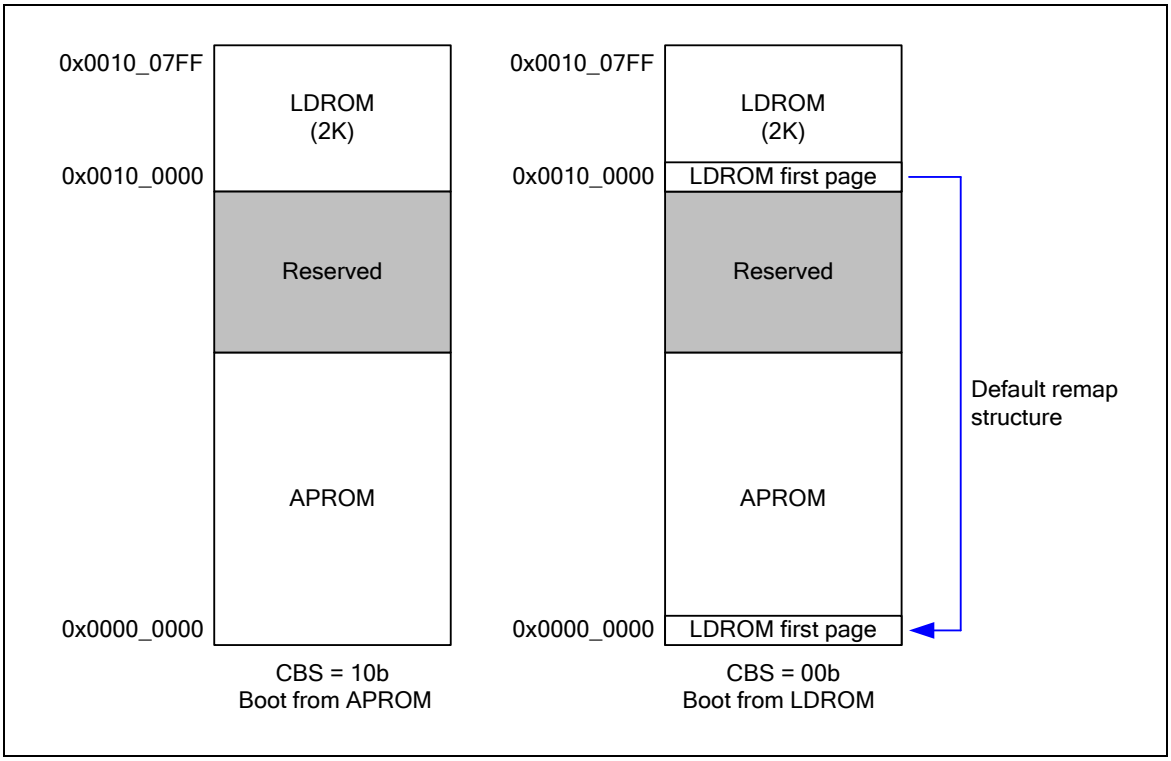


Figure 6.4-6 Executable Range of Code with IAP Function Enabled

When chip boots with the IAP function enabled, any other page within the executable range of code can be mirrored to the first page of executable code (0x0000_0000~0x0000_01FF) any time. User can change the remap address of the first executing page by filling the target remap address to ISPADR and then go through ISP procedure with the Vector Page Re-map command. After changing the remap address, user can check if the change is successful by reading the VECMAP(ISPSTA[20:9]).

6.4.8 In System Programming (ISP)

The NuMicro Mini51™ supports In-System-Programming which allows a device to be reprogrammed under software control and avoids system fail risk when download or programming fail. Furthermore, the capability to update the application firmware makes a wide range of applications possible.

To supports In-System-Programming, NuMicro Mini51™ includes LDROM and ISP controller. User can implement their ISP loader programming in LDROM and this loader can programming user application code (APROM) through ISP register. In other words, the loader could provide the ability to update system firmware on board. By ISP loader, various hardware peripheral interfaces make it be easier to receive new program code. The most common method to perform ISP is via UART along with the ISP loader in LDROM. General speaking, PC transfers the new APROM code through serial port. Then ISP loader receives it and re-programs into APROM through ISP commands.

6.4.9 ISP Registers Control Procedure

The NuMicro Mini51™ supports booting from APROM or LDROM initially defined by user configuration. The change of user configuration needs to reboot system to make it take effect. If user wants to switch boot mode from APROM or LDROM without changing user configuration with CBS[0], he needs to control BS(ISPCON[1]) and then resets CPU by CPU_RST(IPRSTC1[1]) (Not reset I/O and peripherals) or SYSRESETREQ(AIRCR[2])(Reset I/O and peripherals). The software boot switching flow is shown in the following figure. Boot switching function by BS bit is only valid when CBS[0] = 1.

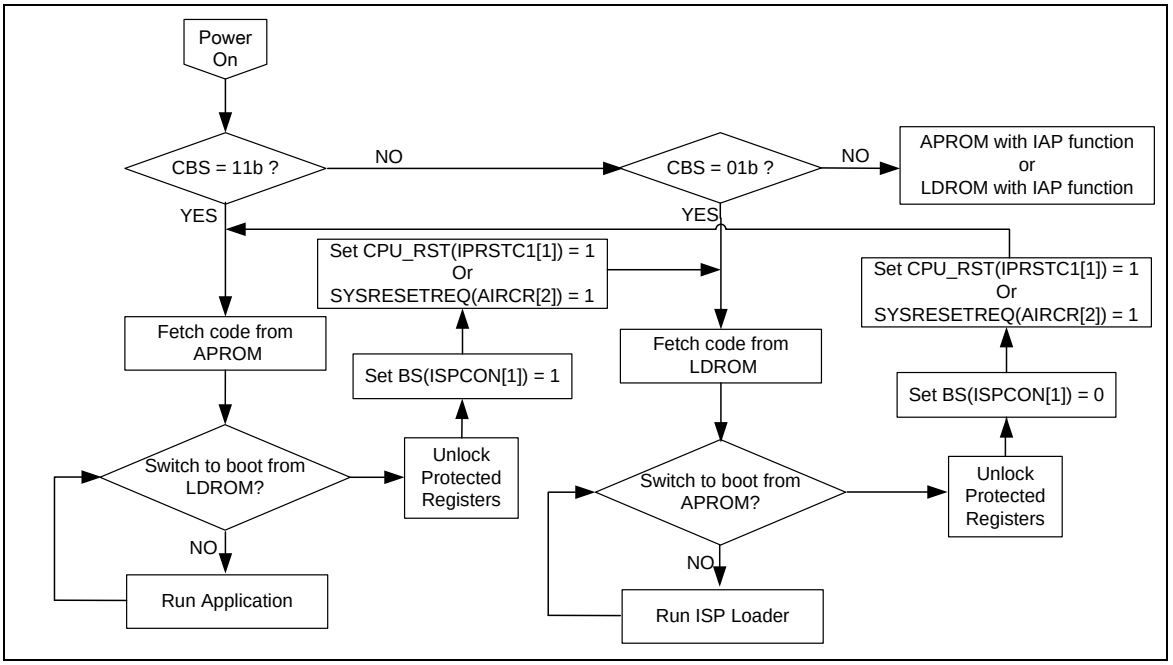


Figure 6.4-7 Example Flow of Boot Selection by BS Bit when CBS[0] = 1

Updating APROM by software in LDROM or updating LDROM by software in APROM can avoid a system failure when update fails.

The ISP controller supports to read, erase and program embedded flash memory. Several control bits of ISP controller are write-protected, thus it is necessary to unlock before we can set them. To unlock the protected register bits, software needs to write 0x59, 0x16 and 0x88 sequentially to REGWRPROT. If register is unlocked successfully, the value of REGWRPROT will be 1. The unlock sequence must not be interrupted by other access; otherwise it may fail to unlock.

After unlocking the protected register bits, user needs to set the ISPCON control register to decide to update LDROM, User Configuration, APROM and enable ISP controller.

Once the ISPCON register is set properly, user can set ISPCMD for erase, read or programming. Set ISPADR for target flash memory based on flash memory origination. ISPDAT can be used to set the data to program or used to return the read data according to ISPCMD.

Finally, set ISPGO(ISPTRG[0]) to perform the relative ISP register function. The ISPGO(ISPTRG[0]) is self-cleared when ISP register function has been done.

Several error conditions are checked after ISP register function is completed. If an error condition

occurs, ISP register operation is not started and the ISP fail flag will be set instead. ISPFF(ISPSTA[6]) can only be cleared by software. The next ISP register control procedure can be started even ISPFF(ISPSTA[6]) is kept as 1. Therefore, it is recommended to check the ISPFF(ISPSTA[6]) and clear it after each ISP register operation if it is set to 1.

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

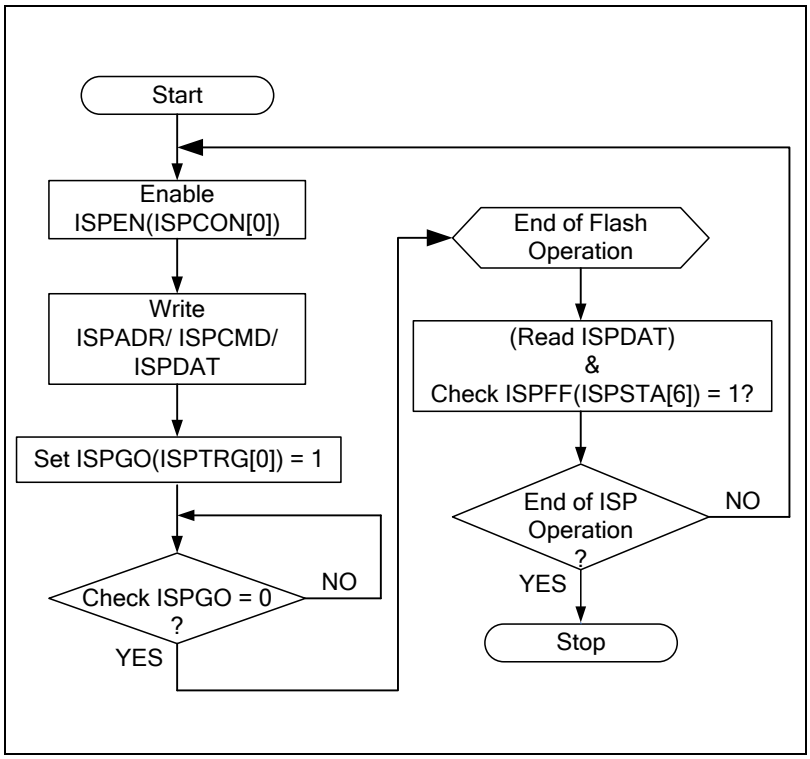


Figure 6.4-8 ISP Flow Example

The following table lists ISP commands supported by NuMicro Mini51™.

ISP Command	ISPCMD	ISPADR	ISPDAT
FLASH Page Erase	0x22	Valid address of flash memory origination. It must be 512 bytes page alignment.	Don't care
FLASH Program	0x21	Valid address of flash memory origination	Programming Data
FLASH Read	0x00	Valid address of flash memory origination	Return Data
Read Unique ID	0x04	0x0000_0000	Unique ID Word 0
		0x0000_0004	Unique ID Word 1
		0x0000_0008	Unique ID Word 2
		0x0000_0010	Unique CID Word 0
		0x0000_0014	Unique CID Word 1
		0x0000_0018	Unique CID Word 2
		0x0000_001C	Unique CID Word 3
Read Company ID	0x0B	Don't care	Company ID (0xDA)
Vector Page Re-Map	0x2E	Page in APROM or LDROM It must be 512 bytes page alignment	Don't care

Table 6.4-4 ISP Command List

6.4.10 Multi-booting by Vector Remapping

The NuMicro Mini51™ can support to boot from different address by vector page remapping function. When CBS[0] = 0, All pages of LDROM and APROM can be remap to vector address 0x0. The remapping address can be got by VECMAP(ISPSTA[20:9]). When CBS[1:0] = 10b, the remapping address is default to 0x0 when powered on. This means the vector page is mapping from first page of APROM to boot from APROM at power on. When CBS[1:0] = 00b, the remapping address is 0x100000. This means the vector page is mapping from first page of LDROM to boot from LDROM at power on.

The remapping address can be changed by Vector Page Re-Map command. User may remap specified page to vector page by Vector Page Re-Map command, than using CPU_RST(IPRSTC1[1]) (Not reset I/O and peripherals) or SYSRESETREQ(AIRCR[2])(Reset I/O and peripherals) to reset system to reboot. The CPU will fetch the stack and reset handler pointer from new vector page, than boot to the specified application.

For example, if user has two independent applications in APROM called App0 and App1. App0 is located at 0x0, and App1 is located at 0x2000. The CBS[1:0] was set to 00b to boot from LDROM. When power on, the system will execute the code in LDROM. The code in LDROM will decide to boot to App0 or App1. For boot to App0, the code in LDROM will enable ISP and set vector page remapping to 0x0, than reset CPU by CPU_RST(IPRSTC1[1]) (Not reset I/O and peripherals) or SYSRESETREQ(AIRCR[2])(Reset I/O and peripherals) to boot to App0. For boot to App1, the code in LDROM will enable ISP and set vector page remapping to 0x2000, than reset CPU by CPU_RST or SYSRESETREQ to boot to App1. The following figure shows how to use vector page remapping to boot to different applications.

To set vector page remapping, user needs to set new page address to ISPADR, set remap

command code 0x2E to ISPCMD. Than trigger ISP command by set ISPGO(ISPTRG[0]) to 1. User can confirm the new vector page mapping address by VECMAP(ISPSTA[20:9]).

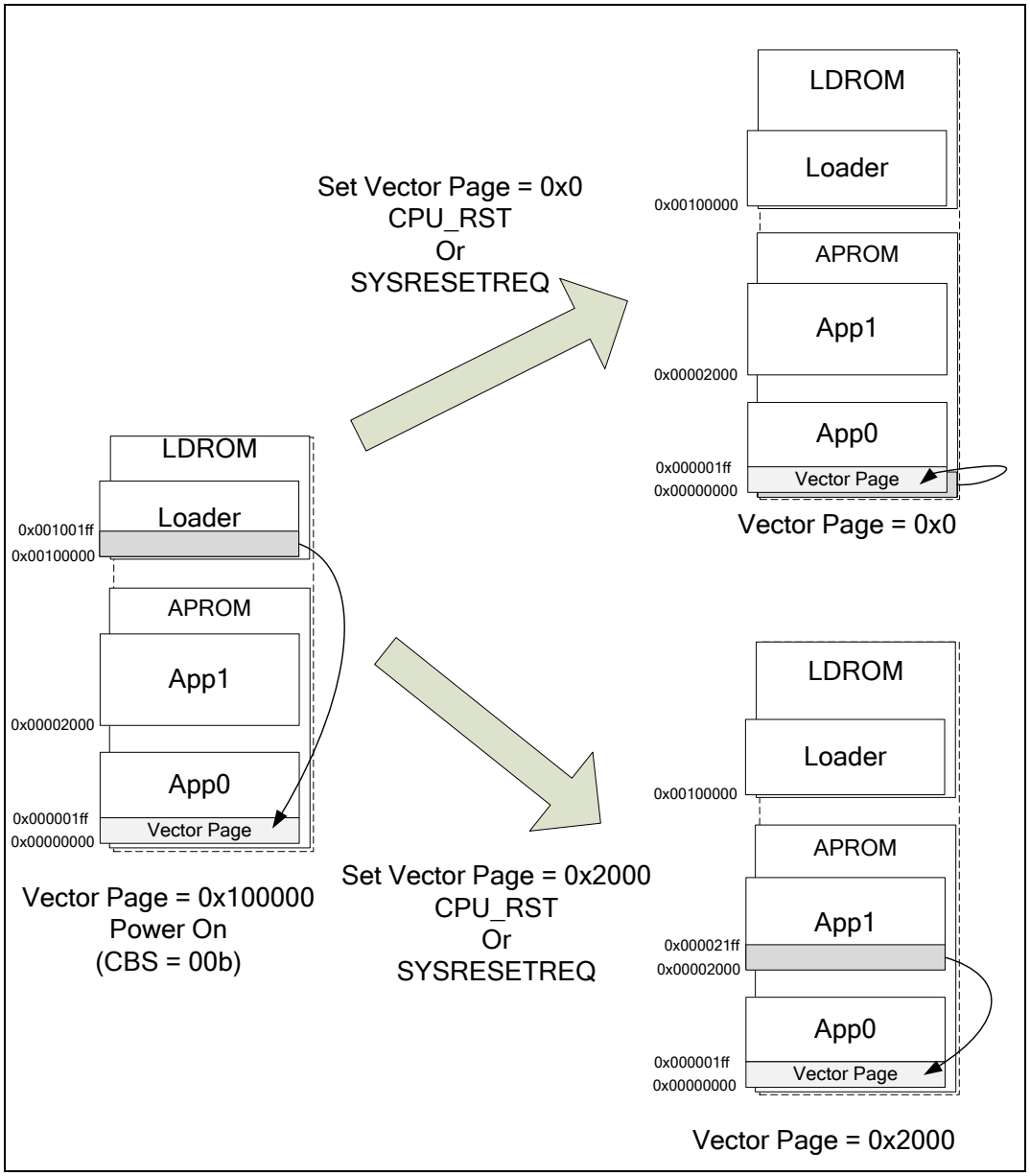


Figure 6.4-9 Multi-booting by Vector Page Remapping

6.4.11 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**				
ISPCON	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000
ISPADR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000
ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000
ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000
ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Register	0x0000_0000
DFBA	FMC_BA+0x14	R	Data Flash Start Address	0x0000_3800
ISPSTA	FMC_BA+0x40	R/W	ISP Status Register	0x0000_0000

6.4.12 Flash Control Register Description

ISP Control Register (ISPCON)

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

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

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**ISPFF**	**ISP Fail Flag (Write Protect)** This bit is set by hardware when a triggered ISP meets any of the following conditions: (1) APROM writes to itself if APUEN is set to 0 or CBS[0]=1. (2) LDROM writes to itself if LDUEN is set to 0 or CBS[0]=1. (3) User Configuration is erased/programmed when CFGUEN is 0. (4) Destination address is illegal, such as over an available range. **Note:** Write 1 to clear this bit to 0.
[5]	**LDUEN**	**LDROM Update Enable Control (Write Protect)** 0 = LDROM cannot be updated. 1 = LDROM can be updated when the MCU runs in APROM.
[4]	**CFGUEN**	**CONFIG Update Enable Control (Write Protect)** Writing this bit to 1 enables software to update CONFIG value by ISP register control procedure regardless of program code is running in APROM or LDROM. 0 = ISP update User Configuration Disabled. 1 = ISP update User Configuration Enabled.
[3]	**APUEN**	**APROM Update Enable Control (Write Protect)** 0 = APROM cannot be updated when chip runs in APROM. 1 = APROM can be updated when chip runs in APROM.
[2]	**Reserved**	Reserved.

Bits	Description	
[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 in CONFIG0 after any reset is happened except CPU reset (RSTS_CPU is 1) or system reset (RSTS_SYS) is happened. 0 = Boot from APROM. 1 = Boot from LDROM.
[0]	**ISPEN**	**ISP Enable Control (Write Protect)** Set this bit to enable ISP function. 0 = ISP function Disabled. 1 = ISP function Enabled.

ISP Address Register (ISPADR)

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

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

Bits	Description	
[31:0]	**ISPADR**	**ISP Address** The NuMicro Mini51™ series supports word program only. ISPADR[1:0] must be kept 00 for ISP operation.

ISP Data Register (ISPDAT)

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

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

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.

ISP Command Register (ISPCMD)

Register	Offset	R/W	Description	Reset Value
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		**ISPCMD**					

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**ISPCMD**	**ISP Command** ISP commands are shown below: 0x00 = Read. 0x04 = Read Unique ID. 0x0B = Read Company ID (0xDA). 0x21 = Program. 0x22 = Page Erase. 0x2E = Set Vector Page Re-Map.

ISP Trigger Control Register (ISPTRG)

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

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

Data Flash Base Address Register (DFBA)

Register	Offset	R/W	Description	Reset Value
DFBA	FMC_BA+0x14	R	Data Flash Start Address	0x0000_3800

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

Bits	Description	
[31:0]	**DFBA[31:0]**	**Data Flash Base Address** This register indicates Data Flash start address. It is a read only register. The Data Flash start address is defined by user. Since on chip flash erase unit is 512 bytes, it is mandatory to keep bit 8-0 as 0.

Example:

Data Flash	4KB (DFEN=0)	2KB (DFEN=0)	1KB (DFEN=0)	0KB (DFEN=1)
16K Flash	DFBA=0x0000_3000	DFBA=0x0000_3800	DFBA=0x0000_3C00	DFEN=1
8K Flash	DFBA=0x0000_1000	DFBA=0x0000_1800	DFBA=0x0000_1C00	DFEN=1
4K Flash	Forbidden	DFBA=0x0000_0800	DFBA=0x0000_0C00	DFEN=1

ISP Status Register (ISPSTA)

Register	Offset	R/W	Description	Reset Value
ISPSTA	FMC_BA+0x40	R/W	ISP Status Register	0x0000_0000

31	30	29	28	27	26	25	24
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**		**ISPGO**

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20:9]	**VECMAP[11:0]**	**Vector Page Mapping Address (Read Only)** The current flash address space 0x0000_0000~0x0000_01FF is mapping to 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 set by hardware when a triggered ISP meets any of the following conditions: (1) APROM writes to itself if APUEN is set to 0 or CBS[0]=1. (2) LDROM writes to itself if LDUEN is set to 0 or CBS[0]=1. (3) User Configuration is erased/programmed when CFGUEN is 0. (4) Destination address is illegal, such as over an available range. Write 1 to clear. **Note:** This bit functions the same as ISPCON bit 6.
[5:3]	**Reserved**	Reserved.
[2:1]	**CBS[1:0]**	**Config Boot Selection (Read Only)** This is a mirror of CBS in CONFIG0.
[0]	**ISPGO**	**ISP Start Trigger (Read Only)** 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 operation is progressed. **Note:** This bit is the same with ISPTRG bit 0.

6.5 General Purpose I/O (GPIO)

6.5.1 Overview

The NuMicro Mini51™ series have 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. 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:
  - Input-only with high impendence
  - Push-pull output
  - Open-drain output
  - Quasi-bidirectional
- TTL/Schmitt trigger input mode selected by 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 (default)

6.5.3 Basic Configuration

The GPIO pin functions are configured in P0_MFP, P1_MFP, P2_MFP, P3_MFP, P4_MFP and P5_MFP registers.

6.5.4 Functional Description

6.5.4.1 Input Mode

Set Px_PMD (PMDn[1:0]) to 00b 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 Px_PIN value reflects the status of the corresponding port pins.

6.5.4.2 Push-pull Output Mode

Set Px_PMD (PMDn[1:0]) to 01b as the 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 Px_DOUT[n] bit is driven on the pin.

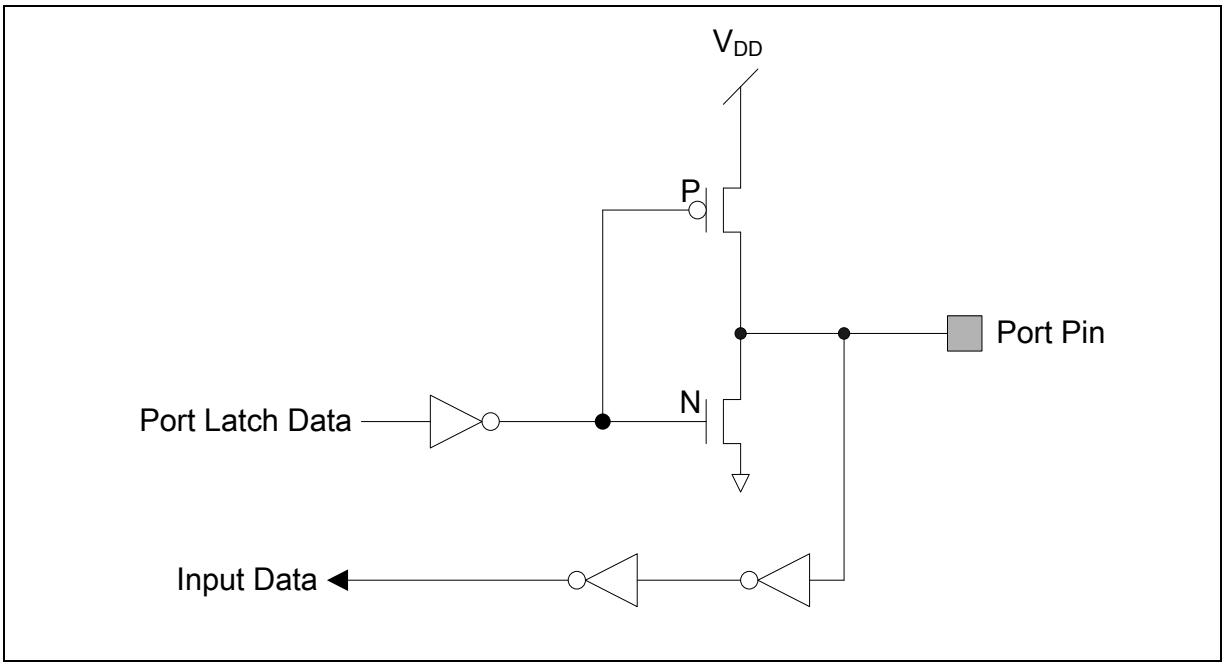


Figure 6.5-1 Push-Pull Output

6.5.4.3 Open-drain Output Mode

Set Px_PMD (PMDn[1:0]) to 10b as the Px.n pin is in Open-drain mode and the digital output function of I/O pin only supports sink current capability, an external pull-up register is needed for driving high state. If the bit value in the corresponding Px_DOUT[n] bit is 0, the pin drives a low output on the pin. If the bit value in the corresponding Px_DOUT[n] bit is 1, the pin output drives high that is controlled by external pull-up resistor.

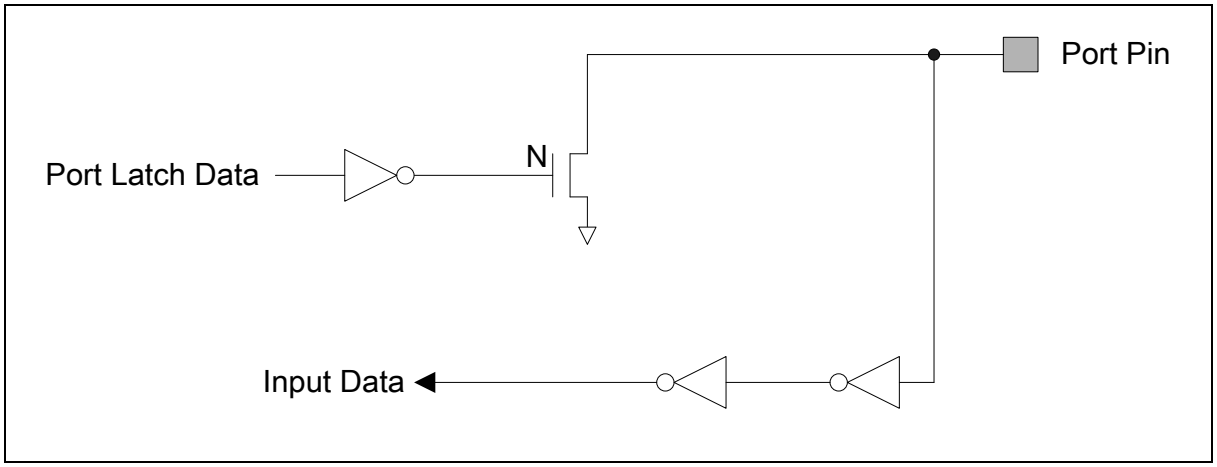


Figure 6.5-2 Open-Drain Output

6.5.4.4 Quasi-bidirectional Mode

Set Px_PMD (PMDn[1:0]) to 11b 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 of uA. Before the digital input function is performed the corresponding Px_DOUT[n] bit must be set to 1. If the bit value in the corresponding Px_DOUT[n] bit is 0, the pin drive a low

output on the pin. If the bit value in the corresponding 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, then 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 200uA to 30uA for V_{DD} from 5.0 V to 2.5 V.

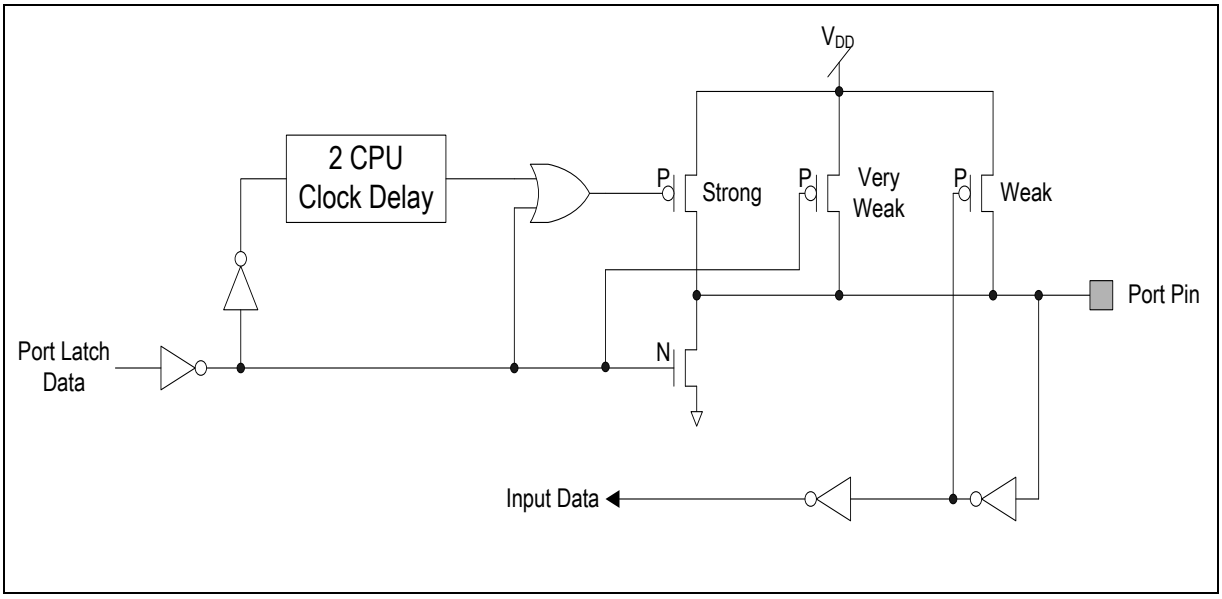


Figure 6.5-3 Quasi-Bidirectional I/O Mode

6.5.5 GPIO Interrupt and Wake-up Function

Each GPIO pin can be set as chip interrupt source by setting correlative Px_IEN bit and Px_IMD. 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 DBNCECON register.

6.5.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
GPIO Base Address: **GP_BA = 0x5000_4000**				
P0_PMD	GP_BA+0x000	R/W	P0 I/O Mode Control	0x0000_XXXX
P0_OFFD	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_DMASK	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_IMD	GP_BA+0x018	R/W	P0 Interrupt Mode Control	0x0000_0000
P0_IEN	GP_BA+0x01C	R/W	P0 Interrupt Enable Control	0x0000_0000
P0_ISRC	GP_BA+0x020	R/W	P0 Interrupt Source Flag	0x0000_0000
P1_PMD	GP_BA+0x040	R/W	P1 I/O Mode Control	0x0000_XXXX
P1_OFFD	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_DMASK	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_IMD	GP_BA+0x058	R/W	P1 Interrupt Mode Control	0x0000_0000
P1_IEN	GP_BA+0x05C	R/W	P1 Interrupt Enable Control	0x0000_0000
P1_ISRC	GP_BA+0x060	R/W	P1 Interrupt Source Flag	0x0000_0000
P2_PMD	GP_BA+0x080	R/W	P2 I/O Mode Control	0x0000_XXXX
P2_OFFD	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_DMASK	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_IMD	GP_BA+0x098	R/W	P2 Interrupt Mode Control	0x0000_0000
P2_IEN	GP_BA+0x09C	R/W	P2 Interrupt Enable Control	0x0000_0000

Register	Offset	R/W	Description	Reset Value
GPIO Base Address: **GP_BA = 0x5000_4000**				
P2_ISRC	GP_BA+0x0A0	R/W	P2 Interrupt Source Flag	0x0000_0000
P3_PMD	GP_BA+0x0C0	R/W	P3 I/O Mode Control	0x0000_XXXX
P3_OFFD	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_DMASK	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_IMD	GP_BA+0x0D8	R/W	P3 Interrupt Mode Control	0x0000_0000
P3_IEN	GP_BA+0x0DC	R/W	P3 Interrupt Enable Control	0x0000_0000
P3_ISRC	GP_BA+0x0E0	R/W	P3 Interrupt Source Flag	0x0000_0000
P4_PMD	GP_BA+0x100	R/W	P4 I/O Mode Control	0x0000_XXXX
P4_OFFD	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_DMASK	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_IMD	GP_BA+0x118	R/W	P4 Interrupt Mode Control	0x0000_0000
P4_IEN	GP_BA+0x11C	R/W	P4 Interrupt Enable Control	0x0000_0000
P4_ISRC	GP_BA+0x120	R/W	P4 Interrupt Source Flag	0x0000_0000
P5_PMD	GP_BA+0x140	R/W	P5 I/O Mode Control	0x0000_XXXX
P5_OFFD	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_DMASK	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_IMD	GP_BA+0x158	R/W	P5 Interrupt Mode Control	0x0000_0000
P5_IEN	GP_BA+0x15C	R/W	P5 Interrupt Enable Control	0x0000_0000
P5_ISRC	GP_BA+0x160	R/W	P5 Interrupt Source Flag	0x0000_0000
DBNCECON	GP_BA+0x180	R/W	Interrupt De-bounce Control	0x0000_0020
P00_PDIO	GP_BA+0x200	R/W	GPIO P0.0 Pin Data Input/Output	0x0000_0001

Register	Offset	R/W	Description	Reset Value
GPIO Base Address: **GP_BA = 0x5000_4000**				
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

Register	Offset	R/W	Description	Reset Value
GPIO Base Address: **GP_BA = 0x5000_4000**				
P55_PDIO	GP_BA+0x2B4	R/W	GPIO P5.5 Pin Data Input/Output	0x0000_0001

Note: Software must set the un-bonding out pin P5.5 to output mode when using QFN-33 package IC for minimize the power-down consumption.

6.5.7 Register Description

Port 0-5 I/O Mode Control Register (Px_PMD)

Register	Offset	R/W	Description	Reset Value
P0_PMD	GP_BA+0x000	R/W	P0 I/O Mode Control Register	0x0000_XXXX
P1_PMD	GP_BA+0x040	R/W	P1 I/O Mode Control Register	0x0000_XXXX
P2_PMD	GP_BA+0x080	R/W	P2 I/O Mode Control Register	0x0000_XXXX
P3_PMD	GP_BA+0x0C0	R/W	P3 I/O Mode Control Register	0x0000_XXXX
P4_PMD	GP_BA+0x100	R/W	P4 I/O Mode Control Register	0x0000_XXXX
P5_PMD	GP_BA+0x140	R/W	P5 I/O Mode Control Register	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**
PMD7		**PMD6**		**PMD5**		**PMD4**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PMD3		**PMD2**		**PMD1**		**PMD0**	

Bits	Description	
[31:16]	**Reserved**	Reserved.
[2n+1:2n] n=0,1..7	**PMDn**	**Port 0-5 I/O Pin [N] Mode Control** Determine each I/O mode of Px.n pin. Default mode is controlled by CIOINI (CONFIG0[10]). 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:** x = 0~4, n = 0~7. **Note2:** P0_PMD[7:4] are reserved. P1_PMD[15:12], [3:2] are reserved. P2_PMD[15:14], [3:0] are reserved. P3_PMD[15:14], [7:6] are reserved. P4_PMD[11:0] are reserved. P5_PMD[15:12] are reserved.

Port 0-5 Digital Input Path Disable Control Register (Px_OFFD)

Register	Offset	R/W	Description	Reset Value
P0_OFFD	GP_BA+0x004	R/W	P0 Digital Input Path Disable Control Register	0x0000_0000
P1_OFFD	GP_BA+0x044	R/W	P1 Digital Input Path Disable Control Register	0x0000_0000
P2_OFFD	GP_BA+0x084	R/W	P2 Digital Input Path Disable Control Register	0x0000_0000
P3_OFFD	GP_BA+0x0C4	R/W	P3 Digital Input Path Disable Control Register	0x0000_0000
P4_OFFD	GP_BA+0x104	R/W	P4 Digital Input Path Disable Control Register	0x0000_0000
P5_OFFD	GP_BA+0x144	R/W	P5 Digital Input Path Disable Control Register	0x0000_0000

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**OFFD[n]**	**Port 0-5 Pin [N] Digital Input Path Disable Control** 0 = Px.n digital input path Enabled. 1 = Px.n digital input path Disabled (digital input tied to low). **Note:** x = 0~5, n = 0~7.
[15:0]	**Reserved**	Reserved.

Note:
P0_OFFD[19:18] are reserved.
P1_OFFD[23:22], [17] are reserved.
P2_OFFD[23], [17:16] are reserved.
P3_OFFD[23], [19] are reserved.
P4_OFFD[21:16] are reserved.
P5_OFFD[23:22] are reserved.

Port 0-5 Data Output Value Register (Px_DOUT)

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

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**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 and 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. **Note1:** x = 0~5, n = 0~7. **Note2:** P0_DOUT[3:2] are reserved. P1_DOUT[7:6], [1] are reserved. P2_DOUT[7], [1:0] are reserved. P3_DOUT[7], [3] are reserved. P4_DOUT[5:0] are reserved. P5_DOUT[7:6] are reserved.

Port 0-5 Data Output Write Mask Register (Px_DMASK)

Register	Offset	R/W	Description	Reset Value
P0_DMASK	GP_BA+0x00C	R/W	P0 Data Output Write Mask Register	0x0000_0000
P1_DMASK	GP_BA+0x04C	R/W	P1 Data Output Write Mask Register	0x0000_0000
P2_DMASK	GP_BA+0x08C	R/W	P2 Data Output Write Mask Register	0x0000_0000
P3_DMASK	GP_BA+0x0CC	R/W	P3 Data Output Write Mask Register	0x0000_0000
P4_DMASK	GP_BA+0x10C	R/W	P4 Data Output Write Mask Register	0x0000_0000
P5_DMASK	GP_BA+0x14C	R/W	P5 Data Output Write Mask Register	0x0000_0000

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

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DMASK[n]**	**Port 0-5 Pin [N] Data Output Write Mask** These bits are used to protect the corresponding Px_DOUT[n] bit. When the DMASK[n] bit is set to 1, the corresponding Px_DOUT[n] bit is protected. If the write signal is masked, writing data to the protect bit is ignored. 0 = Corresponding Px_DOUT[n] bit can be updated. 1 = Corresponding Px_DOUT[n] bit is protected. **Note1:** x = 0~5, n = 0~7. **Note2:** This function only protects the corresponding Px_DOUT[n] bit, and will not protect the corresponding Pxn_PDIO bit. **Note3:** P0_DMASK[3:2] are reserved. P1_DMASK[7:6], [1] are reserved. P2_DMASK[7], [1:0] are reserved. P3_DMASK[7], [3] are reserved. P4_DMASK[5:0] are reserved. P5_DMASK[7:6] are reserved.

Port 0-5 Pin Value Register (Px_PIN)

Register	Offset	R/W	Description	Reset Value
P0_PIN	GP_BA+0x010	R	P0 Pin Value Register	0x0000_00XX
P1_PIN	GP_BA+0x050	R	P1 Pin Value Register	0x0000_00XX
P2_PIN	GP_BA+0x090	R	P2 Pin Value Register	0x0000_00XX
P3_PIN	GP_BA+0x0D0	R	P3 Pin Value Register	0x0000_00XX
P4_PIN	GP_BA+0x110	R	P4 Pin Value Register	0x0000_00XX
P5_PIN	GP_BA+0x150	R	P5 Pin Value Register	0x0000_00XX

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

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**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. **Note1:** x = 0~5, n = 0~7. **Note2:** P0_PIN[3:2] are reserved. P1_PIN[7:6], [1] are reserved. P2_PIN[7], [1:0] are reserved. P3_PIN[7], [3] are reserved. P4_PIN[5:0] are reserved. P5_PIN[7:6] are reserved.

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

Register	Offset	R/W	Description	Reset Value
P0_DBEN	GP_BA+0x014	R/W	P0 De-bounce Enable Control Register	0x0000_0000
P1_DBEN	GP_BA+0x054	R/W	P1 De-bounce Enable Control Register	0x0000_0000
P2_DBEN	GP_BA+0x094	R/W	P2 De-bounce Enable Control Register	0x0000_0000
P3_DBEN	GP_BA+0x0D4	R/W	P3 De-bounce Enable Control Register	0x0000_0000
P4_DBEN	GP_BA+0x114	R/W	P4 De-bounce Enable Control Register	0x0000_0000
P5_DBEN	GP_BA+0x154	R/W	P5 De-bounce Enable Control Register	0x0000_0000

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

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DBEN[n]**	**Port 0-5 Pin [N] Input Signal De-bounce Enable Control** 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 DBNCECON[4], one de-bounce sample cycle period is controlled by DBNCECON[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:** x = 0~5, n = 0~7. **Note2:** 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 woken up caused by the Px.n de-bounce function. **Note3:** P0_DBEN[3:2] are reserved. P1_DBEN[7:6], [1] are reserved. P2_DBEN[7], [1:0] are reserved. P3_DBEN[7], [3] are reserved. P4_DBEN[5:0] are reserved. P5_DBEN[7:6] are reserved.

Port 0-5 Interrupt Mode Control Register (Px_IMD)

Register	Offset	R/W	Description	Reset Value
P0_IMD	GP_BA+0x018	R/W	P0 Interrupt Mode Control Register	0x0000_0000
P1_IMD	GP_BA+0x058	R/W	P1 Interrupt Mode Control Register	0x0000_0000
P2_IMD	GP_BA+0x098	R/W	P2 Interrupt Mode Control Register	0x0000_0000
P3_IMD	GP_BA+0x0D8	R/W	P3 Interrupt Mode Control Register	0x0000_0000
P4_IMD	GP_BA+0x118	R/W	P4 Interrupt Mode Control Register	0x0000_0000
P5_IMD	GP_BA+0x158	R/W	P5 Interrupt Mode 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**
IMD							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**IMD[n]**	**Port 0-5 Pin [N] Edge Or Level Detection Interrupt Mode Control** IMD[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 pin is set as the level trigger interrupt, only one level can be set on the registers Px_IEN. 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. **Note1:** x = 0~5, n = 0~7. **Note2:** P0_IMD[3:2] are reserved. P1_IMD[7:6], [1] are reserved. P2_IMD[7], [1:0] are reserved. P3_IMD[7], [3] are reserved. P4_IMD[5:0] are reserved. P5_IMD[7:6] are reserved.

Port 0-5 Interrupt Enable Control Register (Px_IEN)

Register	Offset	R/W	Description	Reset Value
P0_IEN	GP_BA+0x01C	R/W	P0 Interrupt Enable Control Register	0x0000_0000
P1_IEN	GP_BA+0x05C	R/W	P1 Interrupt Enable Control Register	0x0000_0000
P2_IEN	GP_BA+0x09C	R/W	P2 Interrupt Enable Control Register	0x0000_0000
P3_IEN	GP_BA+0x0DC	R/W	P3 Interrupt Enable Control Register	0x0000_0000
P4_IEN	GP_BA+0x11C	R/W	P4 Interrupt Enable Control Register	0x0000_0000
P5_IEN	GP_BA+0x15C	R/W	P5 Interrupt Enable Control Register	0x0000_0000

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**IR_EN[n]**	**Port 0-5 Pin [N] Interrupt Enabled By Input Rising Edge Or Input Level High** IR_EN[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 IR_EN[n] bit to 1: If the interrupt is level trigger (IMD[n] is 1), the input Px.n pin will generate the interrupt while this pin state is at high level. If the interrupt is edge trigger (IMD[n] is 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:** x = 0~5, n = 0~7.
[15:8]	**Reserved**	Reserved.

[n]	**IF_EN[n]**	**Port 0-5 Pin [N] Interrupt Enabled By Input Falling Edge Or Input Level Low** IF_EN[n] 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 IF_EB[n] bit to 1: If the interrupt is level trigger (IMD[n] is 1), the input Px.n pin will generate the interrupt while this pin state is at low level. If the interrupt is edge mode trigger (IMD[n] is 0), the input Px.n pin will generate the interrupt while this pin state changed from high to low. 0 = Px.n low level or high to low interrupt Disabled. 1 = Px.n low level or high to low interrupt Enabled. **Note1:** x = 0~5, n = 0~7. **Note2:** P0_IEN[19:18], [3:2] are reserved. P1_IEN[23:22], [17], [7:6], [1] are reserved. P2_IEN[23], [17:16], [7], [1:0] are reserved. P3_IEN[23], [19], [7], [3] are reserved. P4_IEN[21:16], [5:0] are reserved. P5_IEN[23:22], [7:6] are reserved.

Port 0-5 Interrupt Source Flag Register (Px_ISRC)

Register	Offset	R/W	Description	Reset Value
P0_ISRC	GP_BA+0x020	R/W	P0 Interrupt Source Flag Register	0x0000_0000
P1_ISRC	GP_BA+0x060	R/W	P1 Interrupt Source Flag Register	0x0000_0000
P2_ISRC	GP_BA+0x0A0	R/W	P2 Interrupt Source Flag Register	0x0000_0000
P3_ISRC	GP_BA+0x0E0	R/W	P3 Interrupt Source Flag Register	0x0000_0000
P4_ISRC	GP_BA+0x120	R/W	P4 Interrupt Source Flag Register	0x0000_0000
P5_ISRC	GP_BA+0x160	R/W	P5 Interrupt Source Flag Register	0x0000_0000

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

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**ISRC[n]**	**Port 0-5 Pin [N] Interrupt Source Flag** Write : 0 = No action. 1 = Clear the corresponding pending interrupt. Read : 0 = No interrupt at Px.n. 1 = Px.n generates an interrupt. **Note1:** x = 0~5, n = 0~7. **Note2:** P0_ISRC[3:2] are reserved. P1_ISRC[7:6], [1] are reserved. P2_ISRC[7], [1:0] are reserved. P3_ISRC[7], [3] are reserved. P4_ISRC[5:0] are reserved. P5_ISRC[7:6] are reserved.

Interrupt De-bounce Cycle Control Register (DBNCECON)

Register	Offset	R/W	Description	Reset Value
DBNCECON	GP_BA+0x180	R/W	Interrupt De-bounce Control Register	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		**ICLK_ON**	**DBCLKSRC**	**DBCLKSEL**			

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**ICLK_ON**	**Interrupt Clock On Mode** 0 = Edge detection circuit is active only if I/O pin corresponding Px_IEN bit is set to 1. 1 = All I/O pins edge detection circuit is always active after reset. **Note:** It is recommended to turn off 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 the HCLK. 1 = De-bounce counter clock source is the 10 kHz internal low speed oscillator.

[3:0]	**DBCLKSEL [3:0]**	**De-bounce Sampling Cycle Selection**

DBCLKSEL	Description
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 Pin Data Input/Output Register (P[x][n]_PDIO)

P[x][n]_PDIO: x = 0~5, n = 0~7

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

P54_PDIO	GP_BA+0x2B0	R/W	GPIO P5.4 Pin Data Input/Output Register	0x0000_0001
P55_PDIO	GP_BA+0x2B4	R/W	GPIO P5.5 Pin Data Input/Output Register	0x0000_0001

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

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**P[x][n]_PDIO**	**GPIO Px.N Pin Data Iutput/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 I/O pin status. 0 = Corresponding GPIO pin status is low. 1 = Corresponding GPIO pin status is high. For example, writing P01_PDIO will reflect the written value tobit P0_DOUT[1], reading P01_PDIO will return the value of P0_PIN[1]. **Note1:** x = 0~5, n = 0~7. **Note2:** The writing operation will not be affected by register Px_DMASK[n].

6.6 Timer Controller (TMR)

6.6.1 Overview

The Timer Controller includes two 32-bit timers, TIMER0 ~ TIMER1, allowing user to easily implement a timer control for applications. The timer can perform functions, such as frequency measurement, delay timing, clock generation, and event counting by external input pins, and interval measurement by external capture pins.

6.6.2 Features

- Two sets of 32-bit timers with 24-bit up-timer and one 8-bit pre-scale counter
- Independent clock source for each channel (TMR0_CLK, TMR1_CLK)
- Provides four timer counting modes: one-shot, periodic, toggle and continuous counting
- Time-out period = (period of timer clock input) * (8-bit pre-scale counter + 1) * (24-bit TCMP)
- Maximum counting cycle time = (1 / T MHz) * (2^8) * (2^{24}); T is the period of timer clock
- 24-bit up counter value is readable through TDR (Timer Data Register)
- Supports event counting function to count the event from external pin (T0, T1)
- 24-bit capture value is readable through TCAP (Timer Capture Data Register)
- Supports external capture pin (T0EX, T1EX) for interval measurement
- Supports internal signal (CPO0, CPO1) for interval measurement
- Supports external capture pin (T0EX, T1EX) to reset 24-bit up counter
- Supports chip wake-up from Idle/Power-down mode if a timer interrupt signal is generated

6.6.3 Block Diagram

Each channel is equipped with an 8-bit pre-scale counter, a 24-bit up-timer, a 24-bit compare register and an interrupt request signal. Refer to . There are five options of clock sources for each channel. illustrates the clock source control function.

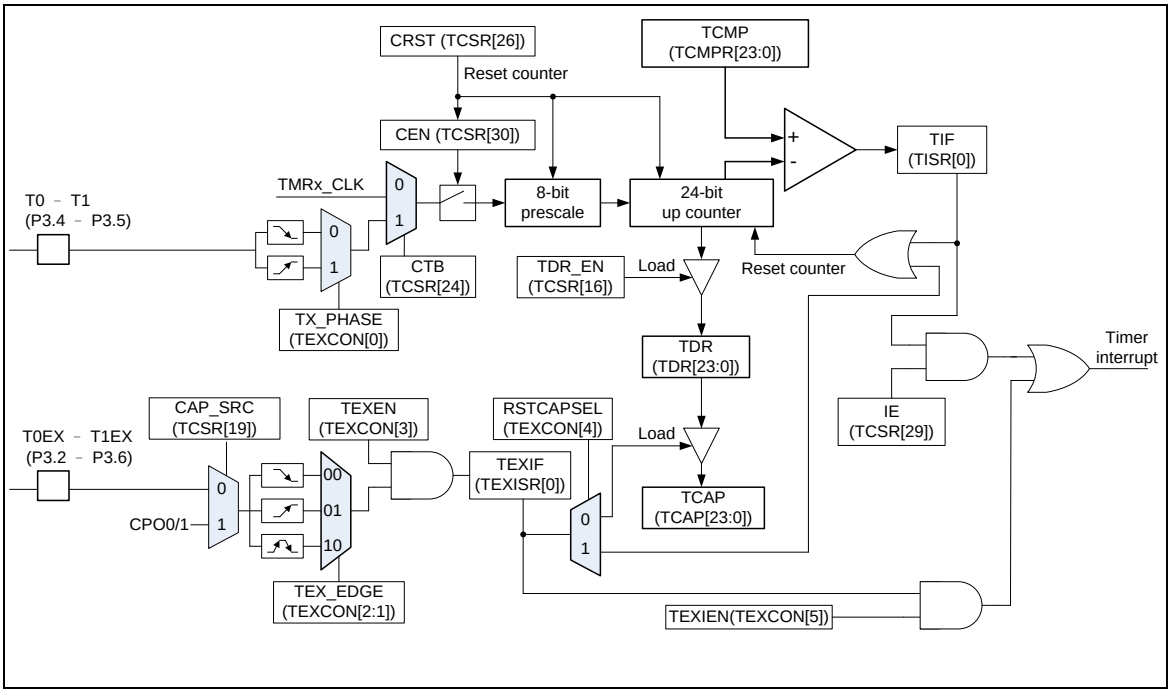


Figure 6.6-1 Timer Controller Block Diagram

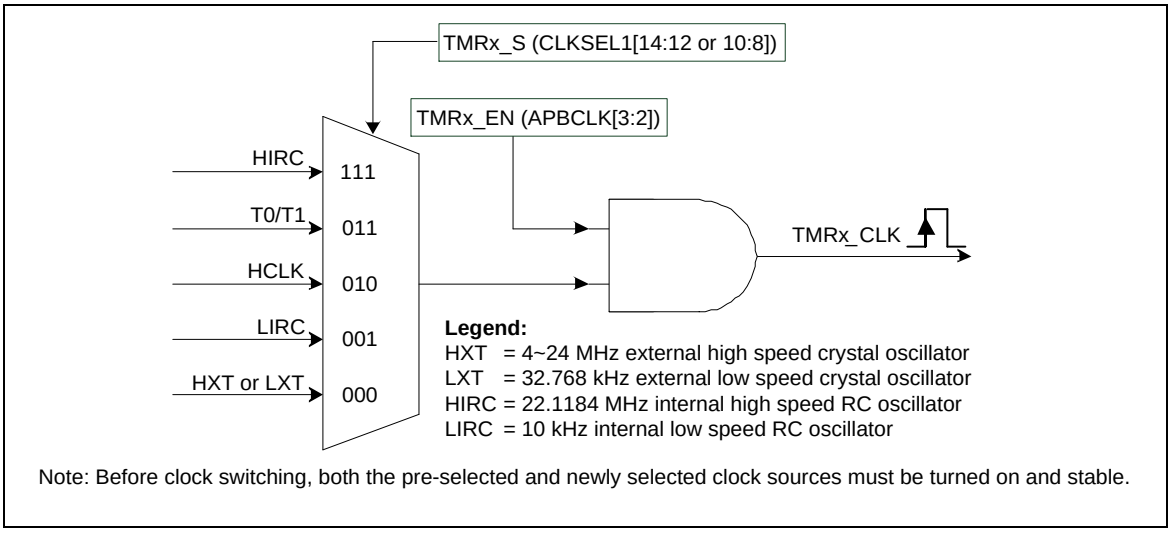


Figure 6.6-2 Clock Source of Timer Controller

6.6.4 Basic Configuration

The peripheral clock source of Tiimer0 ~ Timer1 can be enabled in TMRx_EN (APBCLK[3:2]) and selected as different frequency in TMR0_S (CLKSEL1[10:8]) for Timer0, TMR1_S (CLKSEL1[14:12]) for Timer1.

6.6.5 Functional Description

6.6.5.1 Timer Interrupt Flag

Timer controller supports two interrupt flags; one is TIF (I2CTOC[0]) flag and its set while timer counter value (TDR) matches the timer compared value TCMP (TCMPR[23:0]), the other is TEXIF (TEXISR[0]) flag and its set when the transition on the TxEX pin associated TEX_EDGE (TEXCON[2:1]) setting.

6.6.5.2 Timer Counting Operation Mode

Timer controller provides four timer counting modes: one-shot, periodic, toggle-output and continuous counting operation modes, which are described below.

- **One-shot Mode**

If the timer controller is configured at one-shot mode (MODE (TCSRx[28:27]) is 00b) and CEN (TCSR[30]) bit is set, the timer counter starts up counting. Once the TDR value reaches TCMP value, the TIF flag will be set to 1, TDR value and CEN bit is cleared by timer controller then timer counting operation stops. In the meantime, if the IE (TCSR[29]) bit is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also.

- **Periodic Mode**

If the timer controller is configured at periodic mode (MODE (TCSRx[28:27]) is 01b) and CEN bit is set, the timer counter starts up counting. Once the TDR value reaches TCMP value, the TIF flag will be set to 1, TDR value will be cleared by timer controller and timer counter operates counting again. In the meantime, if the IE bit is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also. In this mode, timer controller operates counting and compares with TCMP value periodically until the CEN bit is cleared by software.

- **Toggle-output Mode**

If timer controller is configured at toggle-out mode (MODE (TCSRx[28:27]) is 10b) and CEN bit is set, the timer counter starts up counting. The counting operation of toggle-out mode is almost the same as periodic mode, except toggle-out mode has associated output pin to output signal while specify TIF bit is set. Thus, the toggle-output signal on output pin is changing back and forth with 50% duty cycle. The output pin could be ether T0/T1 or T0EX/T1EX depending on the TOUT_PIN (TCSR[18]) bit of TCSR register.

- **Continuous Counting Mode**

If timer controller is configured at continuous counting mode (MODE (TCSRx[28:27]) is 11b) and CEN bit is set, the timer counter starts up counting. Once the TDR value reaches TCMP value, the TIF flag will be set to 1 and TDR value keeps up counting. In the meantime, if the IE bit is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also. User can change different TCMP value immediately without disabling timer counting and restarting timer counting in this mode.

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

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

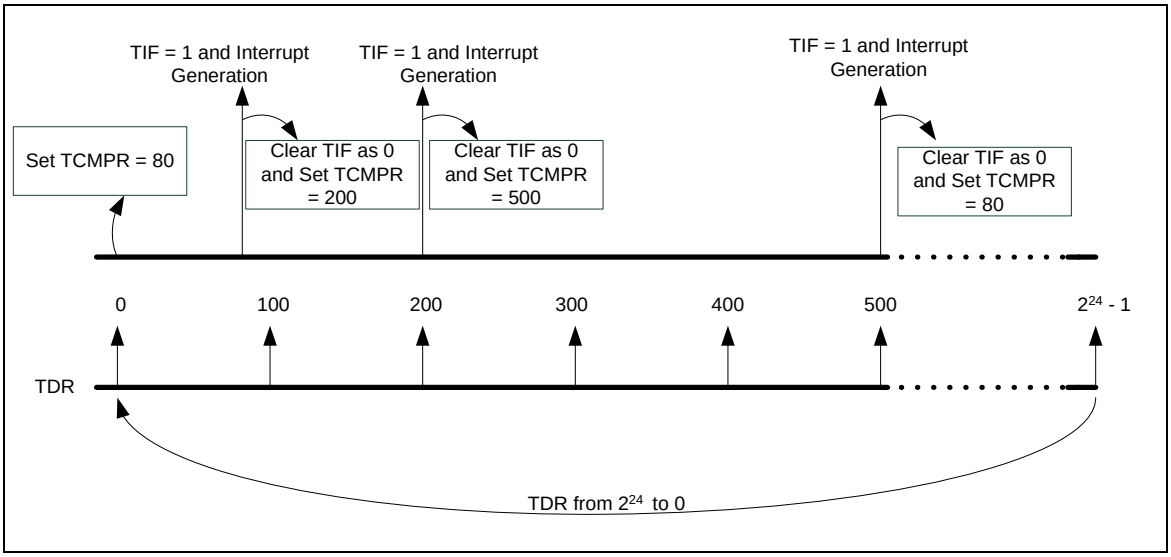


Figure 6.6-3 Continuous Counting Mode

6.6.5.3 *Event Counting Mode*

Timer controller also provides an application which can count the input event from Tx pin (x= 0~1) and the number of event will reflect to TDR value. It is also called as event counting function. In this function, CTB (TCSR[24]) bit should be set and the timer peripheral clock source should be set as HCLK.

Software can enable or disable Tx pin de-bounce circuit by TCDB (TEXCON[7]) bit. The input event frequency should be less than 1/3 HCLK if Tx pin de-bounce disabled or less than 1/8 HCLK if Tx pin de-bounce enabled to assure the returned TDR value is incorrect, and software can also select edge detection phase of Tx pin by TX_PHASE (TEXCON[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 TDR value by input event from Tx pin.

6.6.5.4 *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 CAP_MODE (TEXCON[8]). The free-counting capture mode, reset mode, trigger-counting capture mode are described as follows: The input source could be either TxEX or comparator output CPOx depending on CAP_SRC (TCSR[19]) bit.

6.6.5.5 *Free-Counting Capture Mode*

If CAP_MODE is cleared to 0, TEXEN (TEXCON[3]) is set to 1 and RSTCAPSEL (TEXCON[4]) is set to 0, the TDR will be captured into TCAP register when TEX (Timer External Pin) pin trigger condition occurred. The TEX trigger edge can be chosen by TEX_EDGE. The detail operation method is described in .

6.6.5.6 *Reset Mode*

If CAP_MODE is cleared to 0, TEXEN (TEXCON[3]) is set to 1 and RSTCAPSEL is set to 1, the TDR will be reset to 0 when TEX pin trigger condition happened. The TEX trigger edge can be

chosen by TEX_EDGE (TEXCON[2:1]). The detail operation method is described in .

6.6.5.7 *Trigger-Counting Capture Mode*

If CAP_MODE is set to 1, TEXEN (TEXCON[3]) is set to 1 and RSTCAPSEL is set to 0, the TDR will be reset to 0 then captured into TCAP register when TEX (Timer External Pin) pin trigger condition occurred. The TEX trigger edge can be chosen by TEX_EDGE. The detailed operation method is described in .

When TEX trigger occurred, TEXIF (Timer External Interrupt Flag) is set to 1, and the interrupt signal is generated, then, sent to NVIC to inform CPU if TEXIEN (Timer External Interrupt Enable Bit) is 1. And, the TEX source operating frequency should be less than 1/3 HCLK frequency if disable TEX de-bounce or less than 1/8 HCLK frequency if enabling TEX de-bounce. It also provides T0EX~T1EX enabled or disabled capture de-bounce function by TEXDB (TEXCON[6]).

Function	CAP_MODE (TEXCON[8])	RSTCAPN (TEXCON[4])	TEX_EDGE (TEXCON[2:1])	Operation Description
Free-counting Capture Mode	0	0	00	The high to zero transition on Timer External Input Pin is detected. TDR is captured to TCAP.
	0	0	01	The zero to high transition on Timer External Input Pin is detected. TDR is captured to TCAP.
	0	0	10	Either high to zero or zero to high transition on Timer External Input Pin is detected. TDR is captured to TCAP.
	0	0	11	Reserved
Reset Mode	0	1	00	The high to zero transition on Timer External Input Pin is detected. TDR is reset to 0.
	0	1	01	The zero to high transition on Timer External Input Pin is detected. TDR is reset to 0.
	0	1	10	Either high to zero or zero to high transition on Timer External Input Pin is detected. TDR is reset to 0.
	0	1	11	Reserved
Trigger-Counting Capture Mode	1	0	00	Falling Edge Trigger: The 1st high to zero transition on Timer External Input Pin is detected to reset TDR as 0 and then starts counting, while the 2nd high to zero transition stops counting.
	1	0	01	Rising Edge Trigger: The 1st zero to high transition to on Timer External Input Pin is detected to reset TDR as 0 and then starts counting, while the 2nd zero to high transition stops counting.
	1	0	10	Level Change Trigger: The high to zero transition on Timer External Input Pin is detected to reset TDR as 0 and then starts counting, while zero to high transition stops counting.
	1	0	11	Level Change Trigger:

				The zero to high transition on Timer External Input Pin is detected to reset TDR as 0 and then starts counting, while high to zero transition stops counting.

Table 6.6-1 Input Capture Mode Operation

6.6.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
TMR Base Address **TMR_BA = 0x4001_0000**				
TCSR0	TMR_BA+0x00	R/W	Timer0 Control and Status Register	0x0000_0005
TCMPR0	TMR_BA+0x04	R/W	Timer0 Compare Register	0x0000_0000
TISR0	TMR_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TDR0	TMR_BA+0x0C	R	Timer0 Data Register	0x0000_0000
TCAP0	TMR_BA+0x10	R	Timer0 Capture Data Register	0x0000_0000
TEXCON0	TMR_BA+0x14	R/W	Timer0 External Control Register	0x0000_0000
TEXISR0	TMR_BA+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TCSR1	TMR_BA+0x20	R/W	Timer1 Control and Status Register	0x0000_0005
TCMPR1	TMR_BA+0x24	R/W	Timer1 Compare Register	0x0000_0000
TISR1	TMR_BA+0x28	R/W	Timer1 Interrupt Status Register	0x0000_0000
TDR1	TMR_BA+0x2C	R	Timer1 Data Register	0x0000_0000
TCAP1	TMR_BA+0x30	R	Timer1 Capture Data Register	0x0000_0000
TEXCON1	TMR_BA+0x34	R/W	Timer1 External Control Register	0x0000_0000
TEXISR1	TMR_BA+0x38	R/W	Timer1 External Interrupt Status Register	0x0000_0000

6.6.7 Register Description

Timer Control Register (TCSR)

Register	Offset	R/W	Description	Reset Value
TCSR0	TMR_BA+0x00	R/W	Timer0 Control and Status Register	0x0000_0005
TCSR1	TMR_BA+0x20	R/W	Timer1 Control and Status Register	0x0000_0005

31	30	29	28	27	26	25	24
DBGACK_TMR	CEN	IE	MODE		CRST	CACT	CTB
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WAKE_EN	Reserved			CAP_SRC	TOUT_PIN	PERIODIC_SEL	TDR_EN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRESCALE							

Bits	Description	
[31]	**DBGACK_TMR**	**ICE Debug Mode Acknowledge Disable Control (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.
[30]	**CEN**	**Timer Enable Control** 0 = Stops/Suspends counting. 1 = Starts counting. **Note1:** In stop status, and then set CEN 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 (MODE (TCSRx[28:27]) = 00) when the timer interrupt flag (TIF) is generated.
[29]	**IE**	**Interrupt Enable Control** 0 = Timer Interrupt function Disabled. 1 = Timer Interrupt function Enabled. If this bit is enabled, when the timer interrupt flag (TIF) is set to 1, the timer interrupt signal is generated and inform to CPU.
[28:27]	**MODE[1:0]**	**Timer Operating Mode** **MODE** — **Timer Operating Mode** 00 — The timer is operating in the One-shot mode. The associated interrupt signal is generated once (if IE is enabled) and CEN is automatically cleared by hardware.

Bits	Description	
		01: The timer is operating in Periodic mode. The associated interrupt signal is generated periodically (if IE is enabled). 10: The timer is operating in Toggle mode. The interrupt signal is generated periodically (if IE is enabled). The associated signal (tout) is changing back and forth with 50% duty cycle. 11: The timer is operating in Continuous Counting mode. The associated interrupt signal is generated when TDR = TCMPR (if IE is enabled). However, the 24-bit up-timer counts continuously. Please refer to 6.12.5.2 for detailed description about Continuous Counting mode operation.
[26]	**CRST**	**Timer Reset** 0 = No effect. 1 = Reset 8-bit prescale counter, 24-bit up counter value and CEN bit if CACT is 1.
[25]	**CACT**	**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]	**CTB**	**Counter Mode Enable Control** This bit is for external counting pin function enabled. When timer is used as an event counter, this bit should be set to 1 and select HCLK as timer clock source. Please refer to section 6.12.5.3 for detail description. 0 = External event counter mode Disabled. 1 = External event counter mode Enabled.
[23]	**WAKE_EN**	**Wake-up Enable Control** When WAKE_EN (UA_IER[6]) is set and the TIF or TEXIF (TEXISR[0]) is set, the timer controller will generator a wake-up trigger event to CPU. 0 = Wake-up trigger event Disabled. 1 = Wake-up trigger event Enabled.
[22:20]	**Reserved**	Reserved.
[19]	**CAP_SRC**	**Capture Pin Source Selection** 0 = Capture Function source is from TxEX pin. 1 = Capture Function source is from ACMPx output signal.
[18]	**TOUT_PIN**	**Toggle Out Pin Selection** When Timer is set to toggle mode, 0 = Time0/1 toggle output pin is T0/T1 pin. 1 = Time0/1 toggle output pin is T0EX/T1EX pin.
[17]	**PERIODIC_SEL**	**Periodic Mode Behavior Selection** 0 = In One-shot or Periodic mode, when write new TCMP, timer counter will reset. 1 = In One-shot or Periodic mode, when write new TCMP if new TCMP > TDR(current counter) , timer counter keep counting and will not reset. If new TCMP <= TDR(current counter) , timer counter will reset.
[16]	**TDR_EN**	**Data Load Enable Control** When TDR_EN is set, TDR (Timer Data Register) will be updated continuously with the 24-bit up-timer value as the timer is counting. 0 = Timer Data Register update Disabled. 1 = Timer Data Register update Enabled while Timer counter is active.
[15:8]	**Reserved**	Reserved.

Bits	Description	
[7:0]	**PRESCALE [7:0]**	**Prescale Counter** Timer input clock source is divided by (PRESCALE+1) before it is fed to the Timer up counter. If this field is 0 (PRESCALE = 0), then there is no scaling.

Timer Compare Register (TCMPR)

Register	Offset	R/W	Description	Reset Value
TCMPR0	TMR_BA+0x04	R/W	Timer0 Compare Register	0x0000_0000
TCMPR1	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**
TCMP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TCMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TCMP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**TCMP[23:0]**	**Timer Compared Value** TCMP is a 24-bit compared value register. When the internal 24-bit up counter value is equal to TCMP value, the TIF flag will set to 1. Time-out period = (Period of Timer clock source) * (8-bit PRESCALE + 1) * (24-bit TCMP). **Note1:** Never write 0x0 or 0x1 in TCMP field, or the core will run into unknown state.

Timer Interrupt Status Register (TISR)

Register	Offset	R/W	Description	Reset Value
TISR0	TMR_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TISR1	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						**TWF**	**TIF**

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**TWF**	**Timer Wake-up Flag** This bit indicates the interrupt wake-up flag status of Time. 0 = Timer does not cause chip wake-up. 1 = Chip 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 TDR value reaches to TCMP value. 0 = No effect. 1 = TDR value matches the TCMP value. **Note:** This bit is cleared by writing 1 to it.

Timer Data Register (TDR)

Register	Offset	R/W	Description	Reset Value
TDR0	TMR_BA+0x0C	R	Timer0 Data Register	0x0000_0000
TDR1	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**
TDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDR							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**TDR[23:0]**	**Timer Data Register** If TDR_EN (TCSRx[16]) is set to 1, TDR register value will be updated continuously to monitor 24-bit up counter value.

Timer Capture Data Register (TCAP)

Register	Offset	R/W	Description	Reset Value
TCAP0	TMR_BA+0x10	R	Timer0 Capture Data Register	0x0000_0000
TCAP1	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**
TCAP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TCAP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TCAP							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**TCAP[23:0]**	**Timer Capture Data Register** When TEXIF flag is set to 1, the current TDR value will be auto-loaded into this TCAP filed immediately.

Timer External Control Register (TEXCON)

Register	Offset	R/W	Description	Reset Value
TEXCON0	TMR_BA+0x14	R/W	Timer0 External Control Register	0x0000_0000
TEXCON1	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							**CAP_MODE**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TCDB	**TEXDB**	**TEXIEN**	**RSTCAPSEL**	**TEXEN**	**TEX_EDGE**		**TX_PHASE**

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**CAP_MODE**	**Capture Mode Selection** 0 = Timer counter reset function or free-counting mode of timer capture function. 1 = Trigger-counting mode of timer capture function.
[7]	**TCDB**	**Timer External Counter Input Pin De-bounce Enable Control** 0 = Tx (x = 0~1) pin de-bounce Disabled. 1 = Tx (x = 0~1) pin de-bounce Enabled. If this bit is enabled, the edge detection of Tx (x = 0~1) pin is detected with de-bounce circuit.
[6]	**TEXDB**	**Timer External Capture Input Pin De-bounce Enable Control** 0 = TxEX (x = 0~1) pin de-bounce Disabled. 1 = TxEX (x = 0~1) pin de-bounce Enabled. If this bit is enabled, the edge detection of TxEX (x = 0~1) pin is detected with de-bounce circuit.
[5]	**TEXIEN**	**Timer External Capture Interrupt Enable Control** 0 = TxEX (x = 0~1) pin detection Interrupt Disabled. 1 = TxEX (x = 0~1) pin detection Interrupt Enabled. If TEXIEN enabled, Timer will raise an external capture interrupt signal and inform to CPU while TEXIF flag is set to 1.
[4]	**RSTCAPSEL**	**Timer External Reset Counter / Timer External Capture Mode Selection** 0 = Transition on TxEX (x = 0~1) pin is using to save the TDR value into TCAP value if TEXIF flag is set to 1. 1 = Transition on TxEX (x = 0~1) pin is using to reset the 24-bit up counter.
[3]	**TEXEN**	**Timer External Pin Function Enable Control** This bit enables the RSTCAPSEL function on the TxEX (x = 0~1) pin.

Bits	Description	
		0 = RSTCAPSEL function of TxEX (x = 0~1) pin will be ignored. 1 = RSTCAPSEL function of TxEX (x = 0~1) pin is active.
[2:1]	**TEX_EDGE [1:0]**	**Timer External Pin Edge Detection** **00 =** A 1 to 0 transition on TxEX (x = 0~1) will be detected. 01 = A 0 to 1 transition on TxEX (x = 0~1) will be detected. 10 = Either 1 to 0 or 0 to 1 transition on TxEX (x = 0~1) will be detected. 11 = Reserved.
[0]	**TX_PHASE**	**Timer External Count Pin Phase Detect Selection** This bit indicates the detection phase of Tx (x = 0~1) pin. 0 = A falling edge of Tx (x = 0~1) pin will be counted. 1 = A rising edge of Tx (x = 0~1) pin will be counted.

Timer External Interrupt Status Register (TEXISR)

Register	Offset	R/W	Description	Reset Value
TEXISR0	TMR_BA+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TEXISR1	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							**TEXIF**

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**TEXIF**	**Timer External Interrupt Flag** This bit indicates the external capture interrupt flag status When TEXEN enabled, TxEX (x = 0, 1) pin selected as external capture function, and a transition on TxEX (x = 0, 1) pin matched the TEX_EDGE setting, this flag will set to 1 by hardware. 0 = TxEX (x = 0, 1) pin interrupt did not occur. 1 = TxEX (x = 0, 1) pin interrupt occurred. **Note:** This bit is cleared by writing 1 to it

6.7 Enhanced PWM Generator

6.7.1 Overview

The NuMicro™ Mini51 series has built one PWM unit which is specially designed for motor driving control applications. The PWM unit supports six PWM generators which can be configured as six independent PWM outputs, PWM0~PWM5, or as three complementary PWM pairs, (PWM0, PWM1), (PWM2, PWM3) and (PWM4, PWM5) with three programmable dead-zone 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 PWM unit supports the following features:

- Independent 16-bit PWM duty control units with maximum six port pins:
  - Six independent PWM outputs – PWM0, PWM1, PWM2, PWM3, PWM4, and PWM5
  - Three complementary PWM pairs, with each pin in a pair mutually complement to each other and capable of programmable dead-zone insertion – (PWM0, PWM1), (PWM2, PWM3) and (PWM4, PWM5)
  - Three synchronous PWM pairs, with each pin in a pair in-phase – (PWM0, PWM1), (PWM2, PWM3) and (PWM4, PWM5)
- Group control bit – PWM2 and PWM4 are synchronized with PWM0, PWM3 and PWM5 are synchronized with PWM1
- One-shot (only support edge alignment mode) or Auto-reload mode PWM
- Up to 16-bit resolution
- Supports Edge-aligned and Center-aligned mode
- Programmable dead-zone insertion between complementary paired PWMs
- Each pin of PWM0 to PWM5 has independent polarity setting control
- Hardware fault brake protections
  - Two Interrupt source types:

- Synchronously requested at PWM frequency when down counter comparison matched (edge- and center-aligned mode) or underflow (edge-aligned mode)
- 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 independently rising CMR matching (in Center-aligned mode), CNR matching (in Center-aligned mode), falling CMR matching, 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

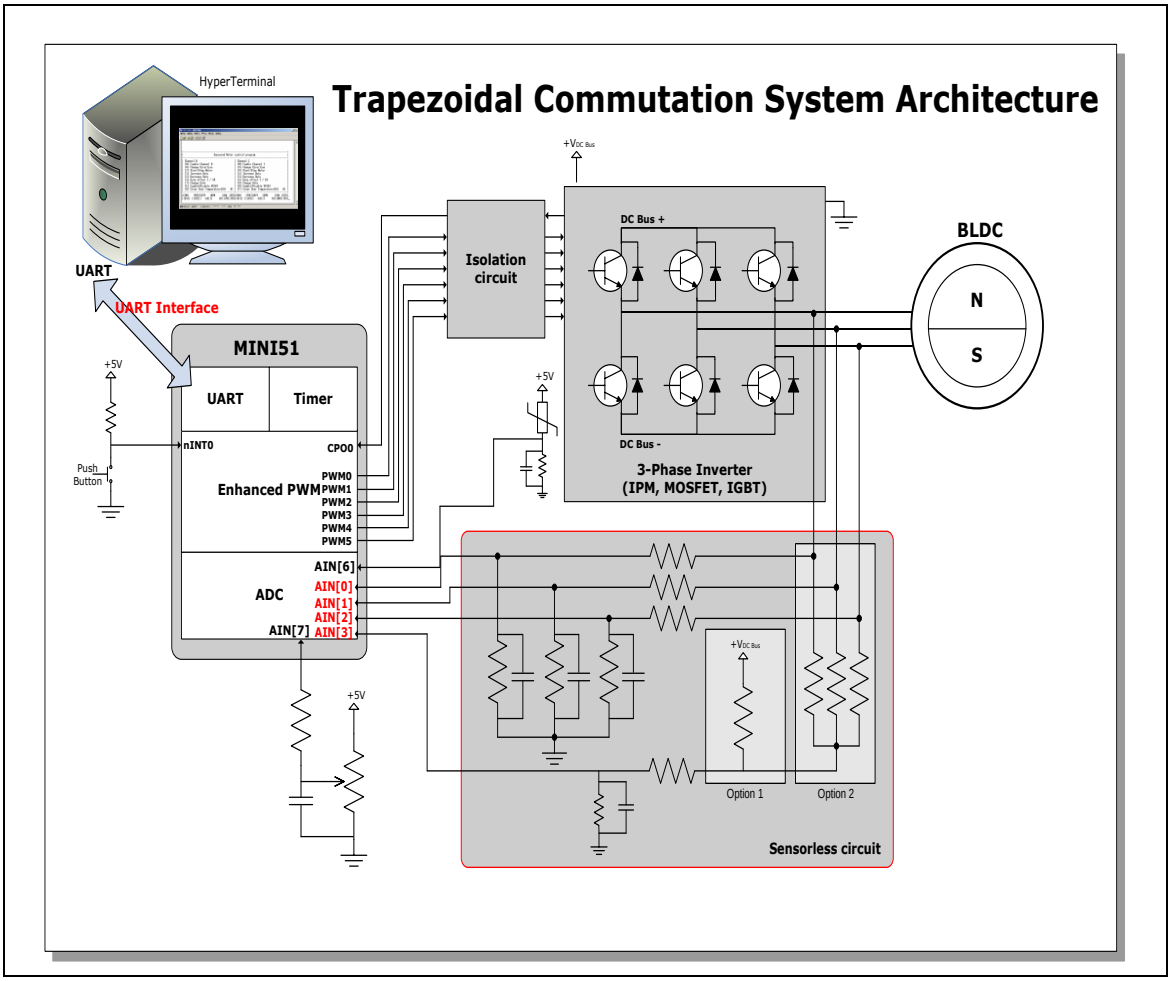


Figure 6.7-1 Application Circuit Diagram

6.7.3 Block Diagram

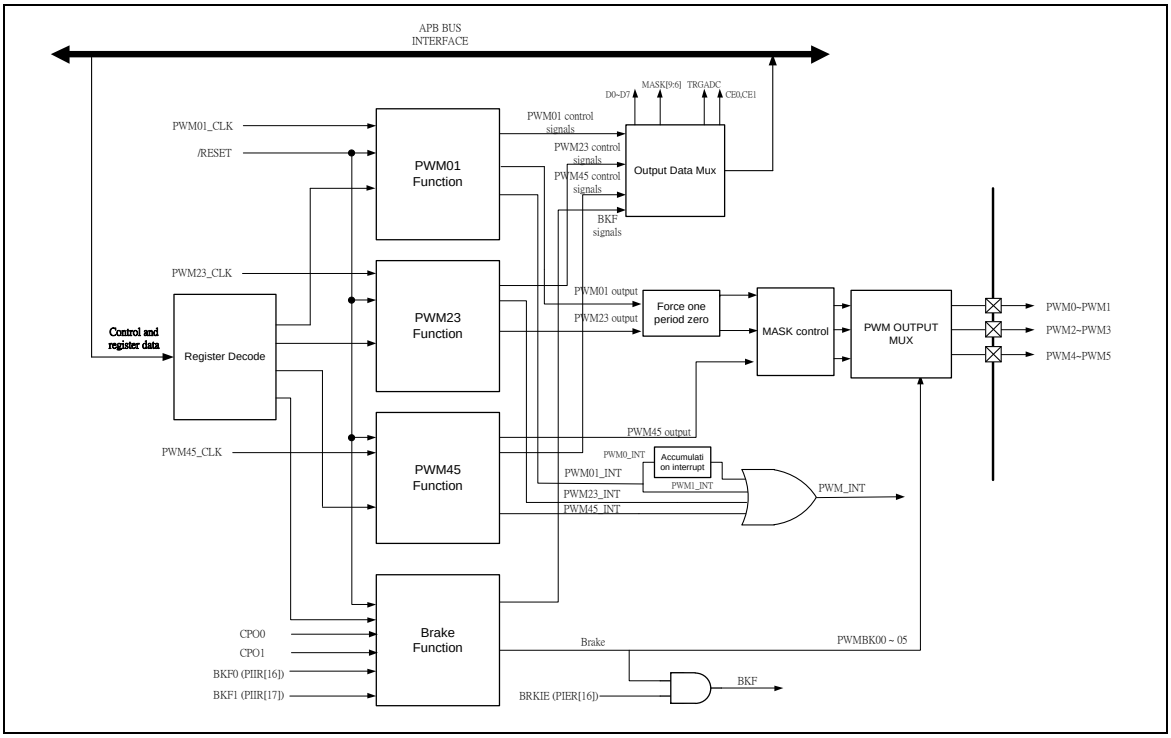


Figure 6.7-2 PWM Block Diagram

The following figures illustrate the architecture of PWM in pair (e.g. PWM-Timer 0/1 are in one pair and PWM-Timer 2/3 are in another one, and so on.).

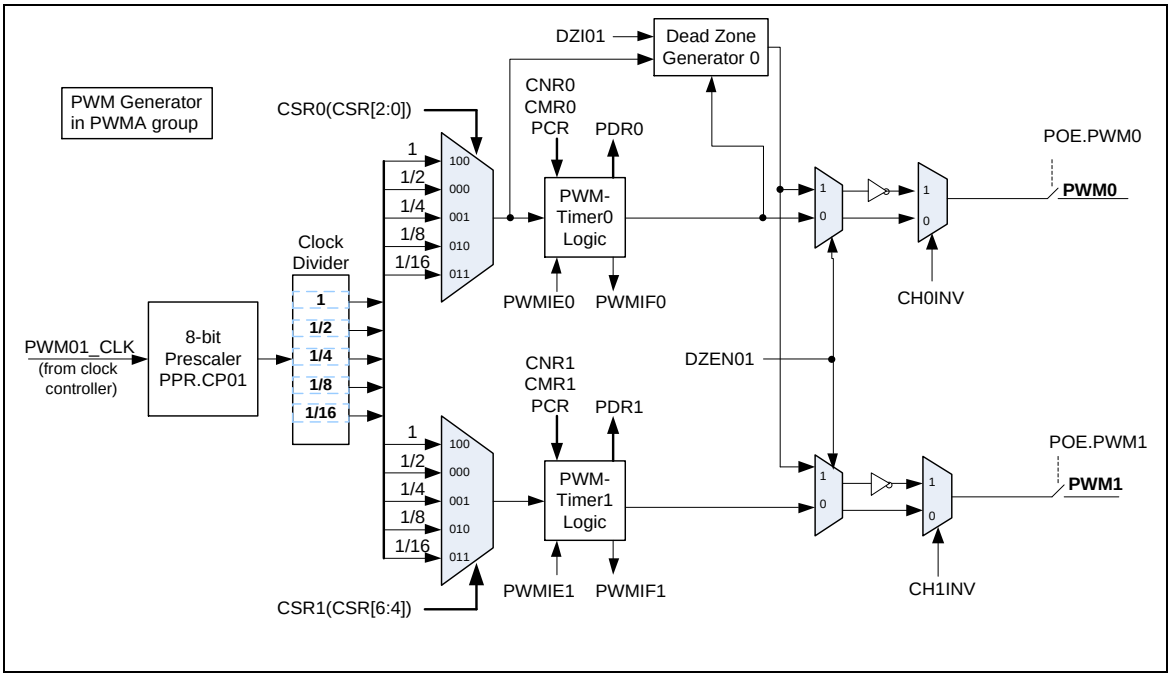


Figure 6.7-3 PWM Generator 0 Architecture Diagram

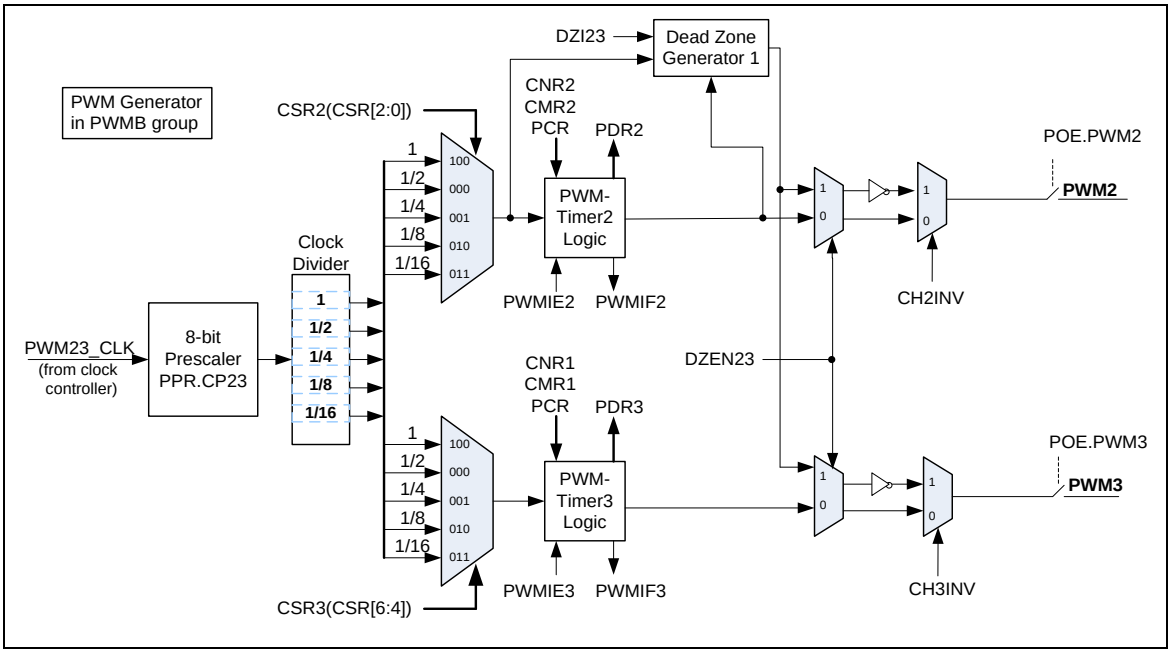


Figure 6.7-4 PWM Generator 2 Architecture Diagram

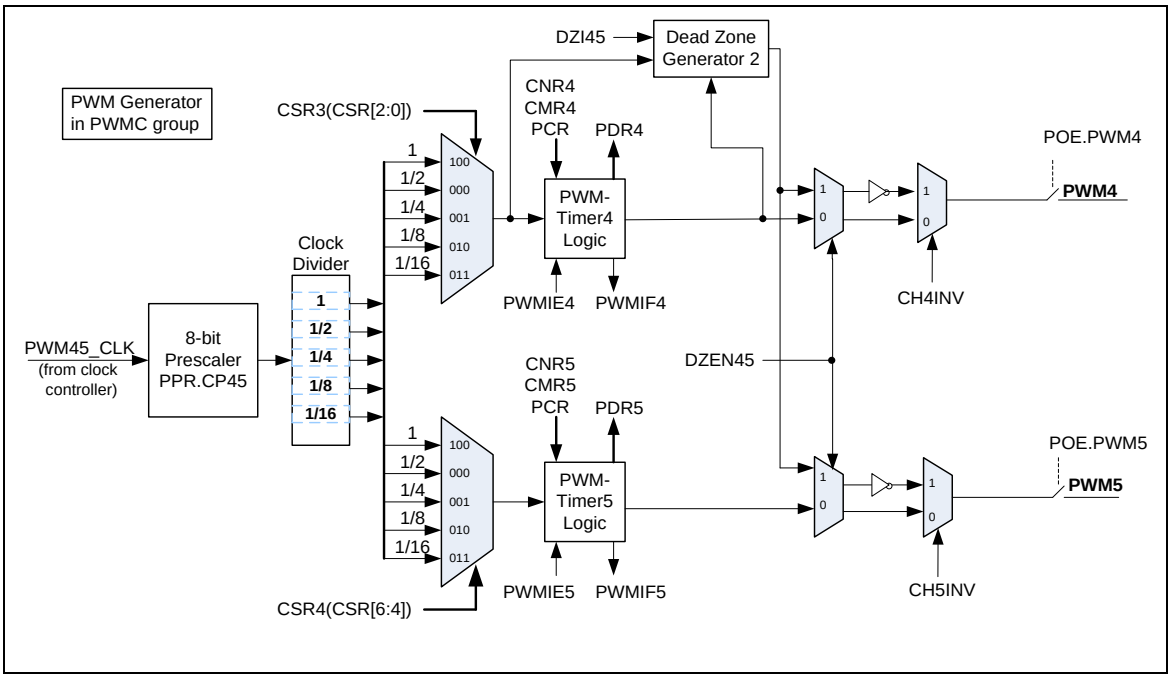


Figure 6.7-5 PWM Generator 4 Architecture Diagram

6.7.4 Basic Configuration

The PWM pin functions are configured in P0_MFP and P2_MFP registers.

The PWM clock can be enabled in APBCLK[22:20]. The PWM clock source must be HCLK.

6.7.5 Functional Description

6.7.5.1 PWM-Timer Operation

This device supports two operation modes: Edge-aligned and Center-aligned mode.

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

Edge aligned (Down counter)

Duty ratio = (CMR+1) / (CNR+1)

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

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

Center aligned (Up and Down Counter):

Duty ratio = (CNR - CMR) / (CNR+1)

Duty = (CNR - CMR) * 2 * (clock period)

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

Edge aligned PWM (Down-counter)

In Edge-aligned PWM Output mode, the 16-bit PWM counter will start counting-down from CNRn to match with the value of the duty cycle CMRn (old); when this happens it will toggle the PWMn generator output to high. The counter will continue counting-down to 0; at this moment, it toggles the PWMn generator output to low and CMRn (new) and CNRn (new) are updated with CHnMODE=1 and requests the PWM interrupt if PWM interrupt is enabled (PIER.n=1).

and describe the Edge-aligned PWM timing and operation flow.

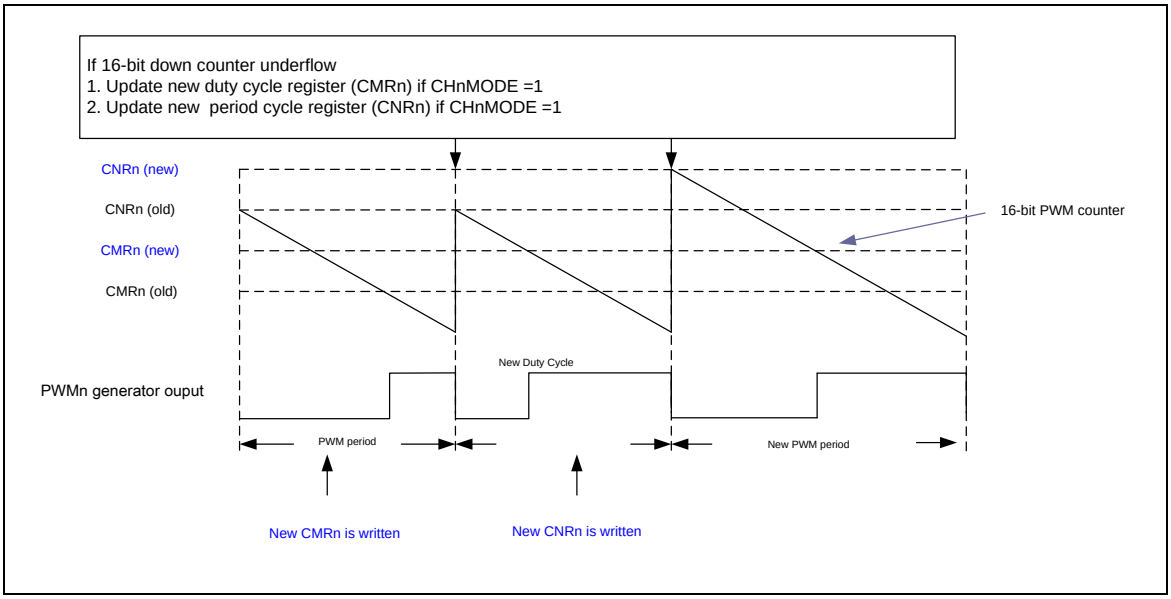


Figure 6.7-6 Edge-aligned PWM

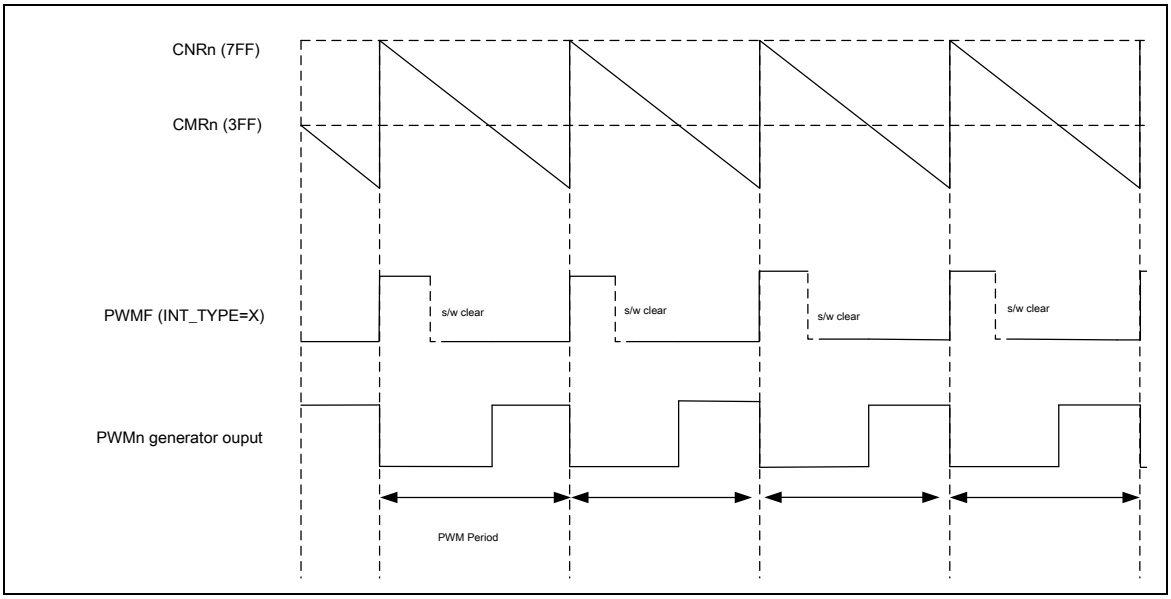


Figure 6.7-7 PWM Edge-aligned Waveform Output

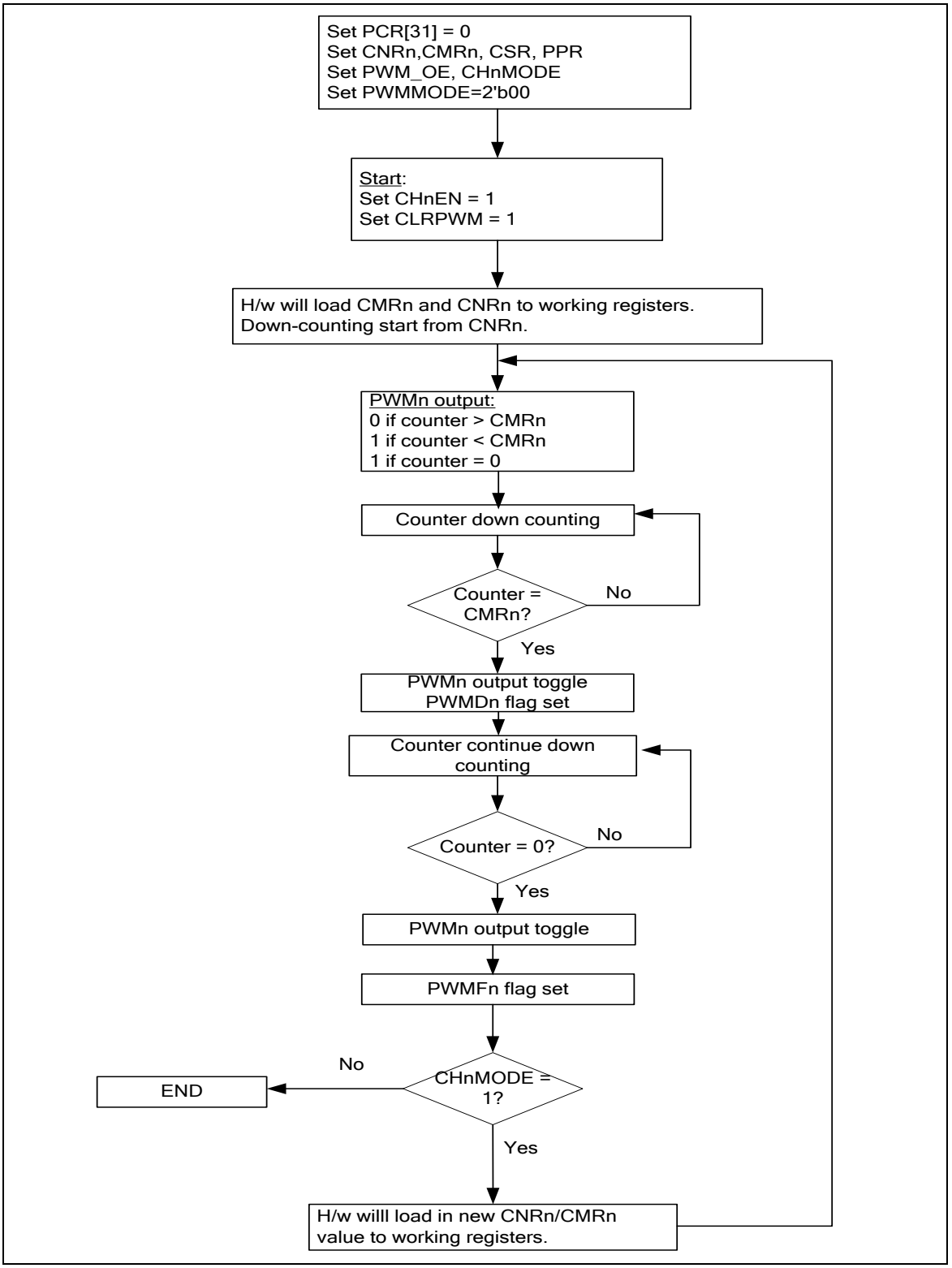


Figure 6.7-8 Edge-aligned Flow Diagram

The PWM period and duty control are decided by PWM down-counter register (CNRn) and PWM comparator register (CMRn). The PWM-Timer timing operation is shown in . The pulse width modulation follows the formula below and the legend of PWM-Timer Comparator is shown in . Note that the corresponding GPIO pins must be configured as PWM function (enable PWMPOE)

for the corresponding PWM channel.

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

Duty ratio = (CMR+1)/(CNR+1)

CMR >= CNR: PWM output is always high

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

CMR = 0: PWM low width = (CNR) unit; PWM high width = 1 unit

Note: 1. Unit = one PWM clock cycle.

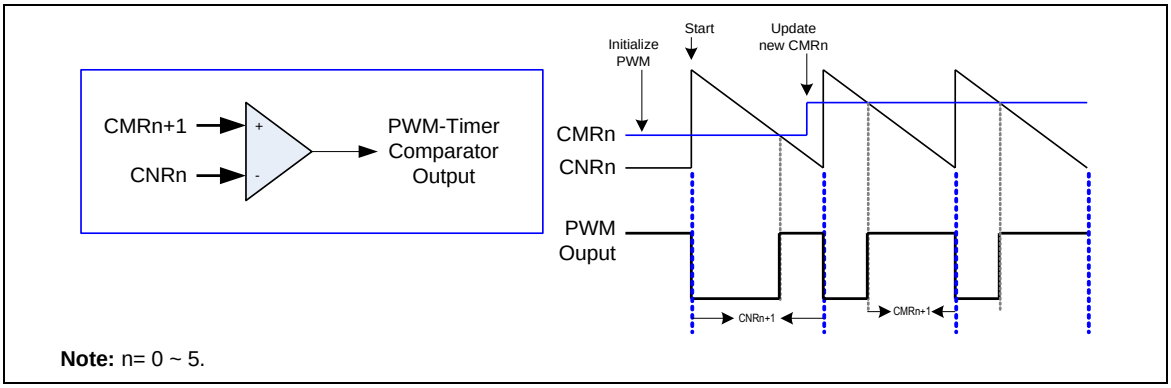


Figure 6.7-9 Legend of Internal Comparator Output of PWM-Timer

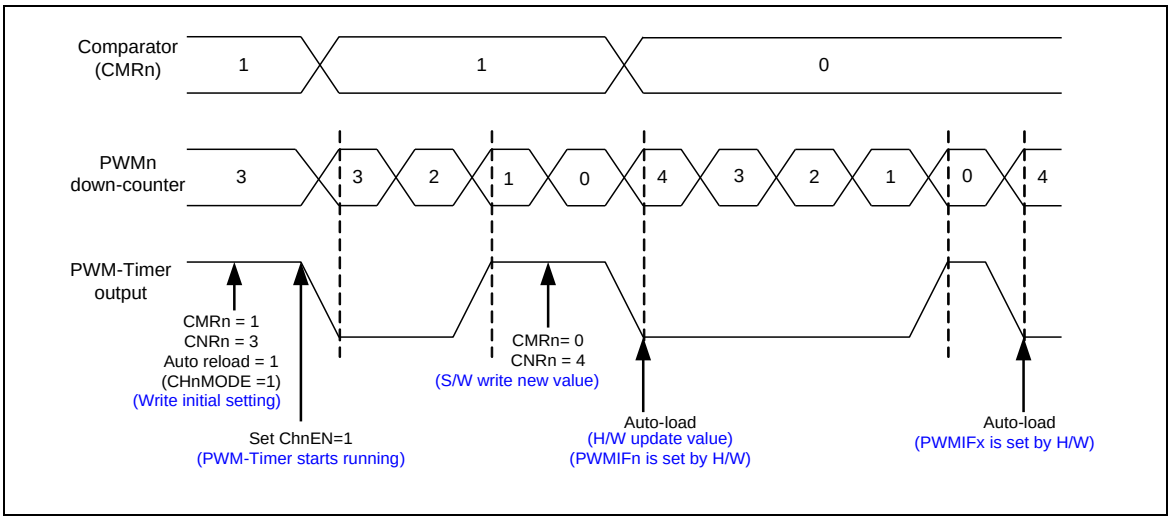


Figure 6.7-10 PWM-Timer Operation Timing

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 mode. The PWM counter will start counting-up from 0 to match the value of CMRn (old); this will cause the toggling of the PWMn generator output to high. The counter will continue counting to match with the CNRn (old). Upon reaching this state counter is configured automatically to down counting, when PWM counter matches the CMRn (old) value again the PWMn generator output toggles to low. Once the PWM counter underflows it will update the PWM period register CNRn (new) and duty cycle register CMRn (new) with CHnMODE = 1.

In Center-aligned mode, the PWM period interrupt is requested at down-counter underflow if INT_TYPE (PIER[17]) =0, i.e. at start (end) of each PWM cycle or at up-counter matching with CNRn if INT_TYPE (PIER[17]) =1, i.e. at center point of PWM cycle.

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

Duty ratio = (CNR - CMR) / (CNR+1)

CMR >= CNR: PWM output is always low

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

CMR = 0: PWM high width = CNR * 2 units; PWM low width = 2 units

Note: 1. Unit = one PWM clock cycle.

, and descript the Center-aligned PWM timing and operation flow.

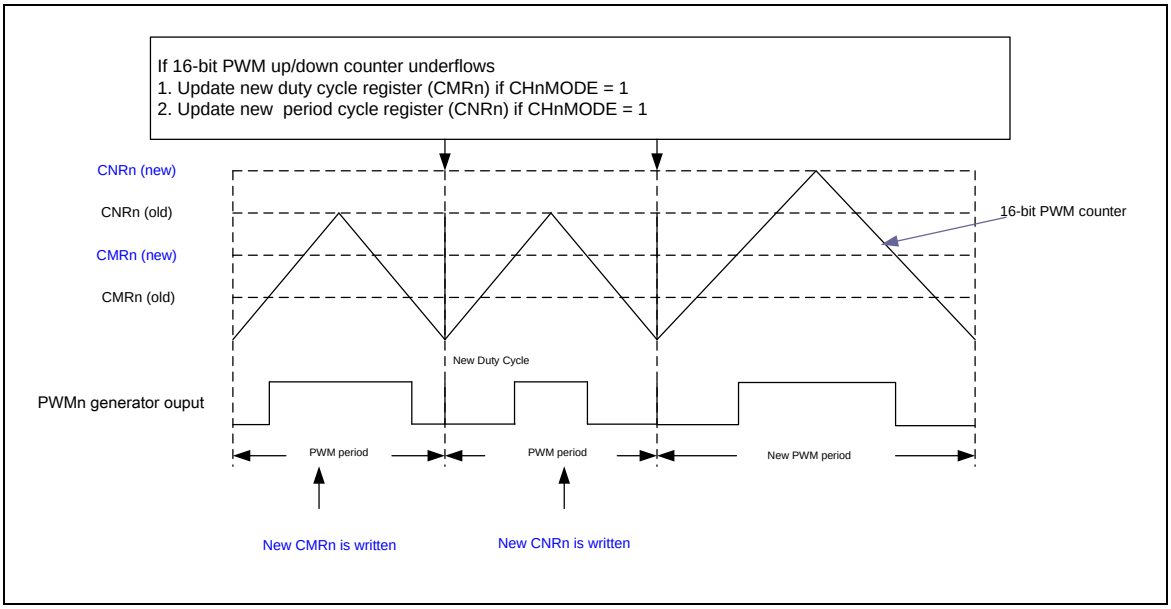


Figure 6.7-11 Center-aligned Mode

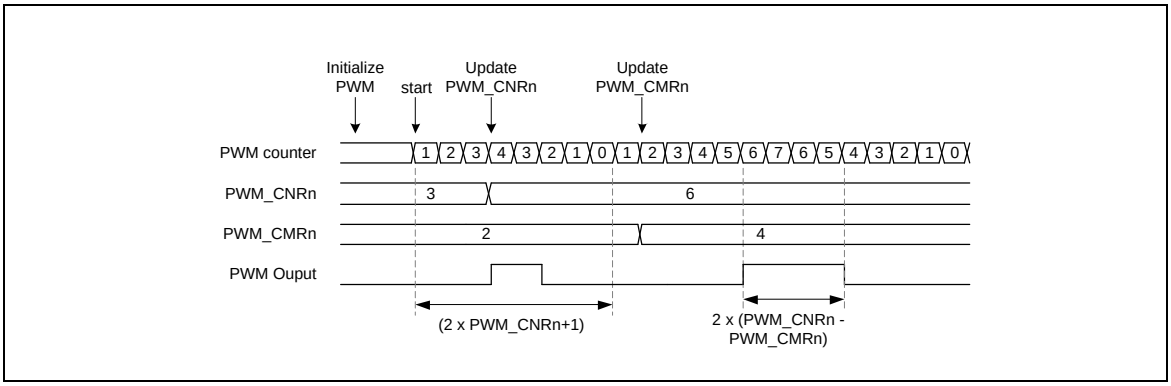


Figure 6.7-12 Center-aligned Mode Operation Timing

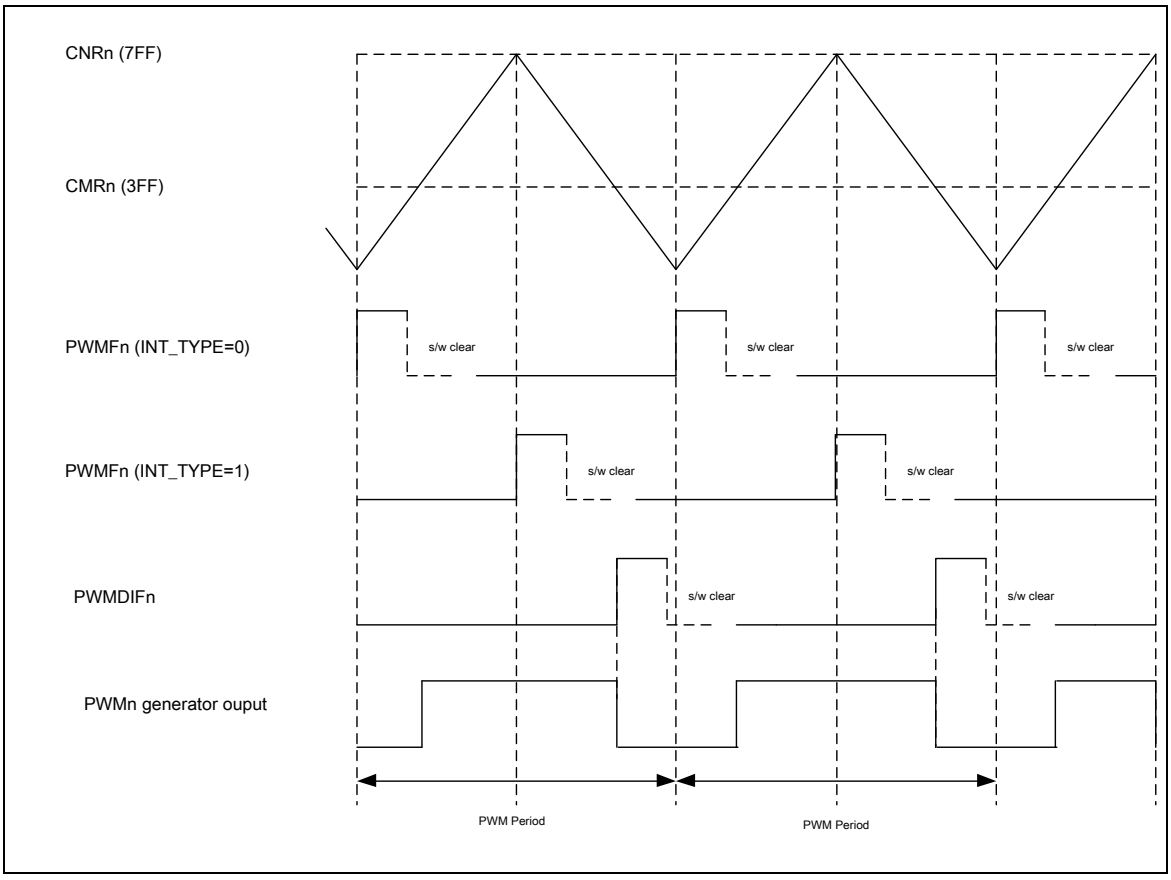


Figure 6.7-13 PWM Center-aligned Waveform Output

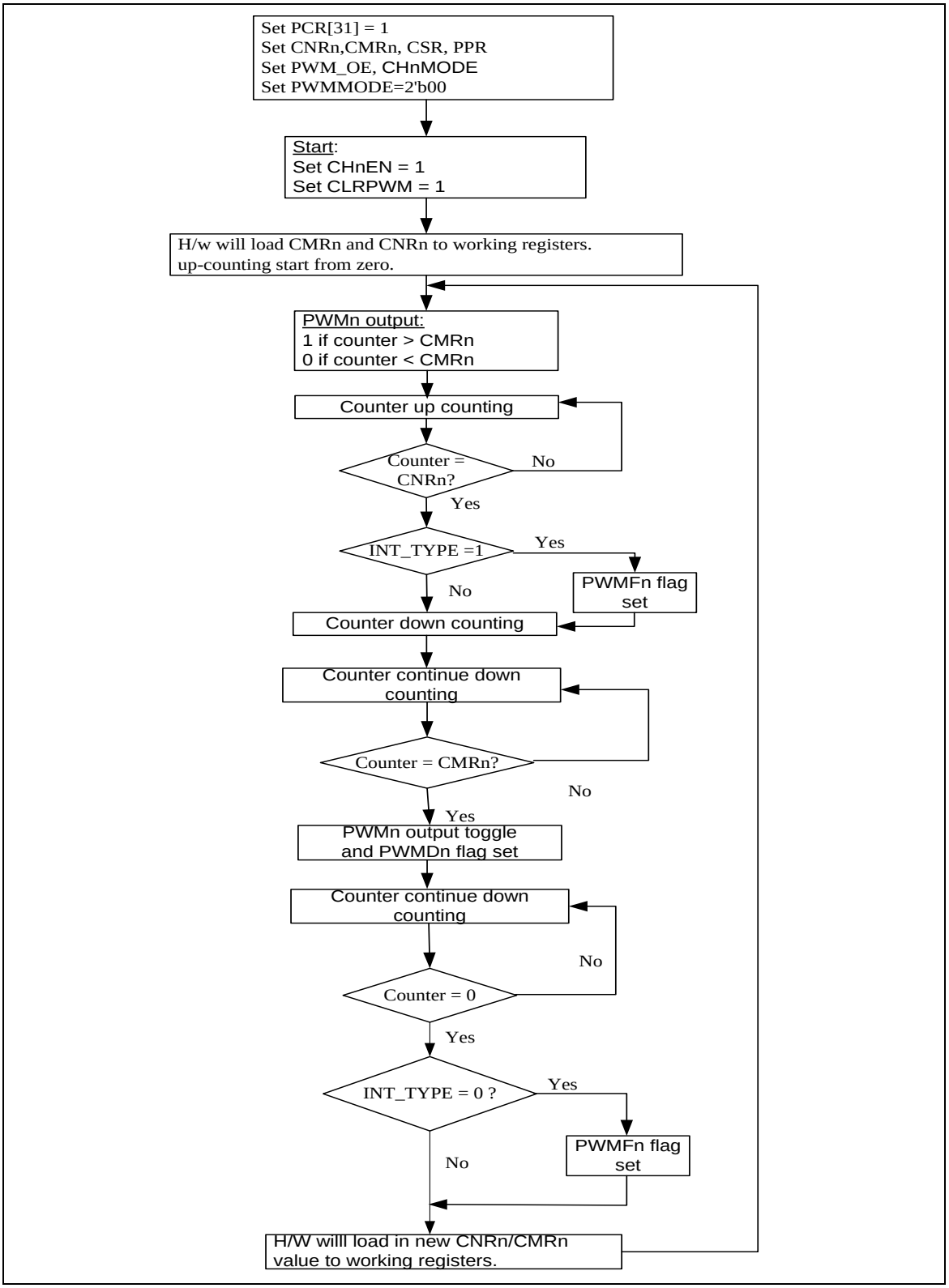


Figure 6.7-14 Center-aligned Flow Diagram (INT_TYPE = 0)

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

The NuMicro™ Mini51 series PWM Timers have double buffering function the reload value is updated at the start of next period without affecting current timer operation. The PWM counter value can be written into CNRn.

PWM0 will operate in One-shot mode if CH0MOD bit is set to 0, and operate in Auto-reload mode if CH0MOD bit is set to 1. It is recommend that switch PWM0 operating mode before set CH0EN bit to 1 to enable PWM0 counter start running because the content of CNR0 and CMR0 will be cleared to 0 to reset the PWM0 period and duty setting when PWM0 operating mode is changed. As PWM0 operate in One-shot mode, CMR0 and CNR0 should be written first and then set CH0EN bit to 1 to enable PWM0 counter start running. After PWM0 counter down count from CNR0 value to 0, CNR0 and CMR0 will be cleared to 0 by hardware and PWM counter will be held. Software need to write new CMR0 and CNR0 value to set next one-shot period and duty. When re-start next one-shot operation, the CMR0 should be written first because PWM0 counter will auto re-start counting when CNR0 is written a non-zero value. As PWM0 operates at auto-reload mode, CMR0 and CNR0 should be written first and then set CH0EN bit to 1 to enable PWM0 counter start running. The value of CNR0 will reload to PWM0 counter when it down count reaches 0. If CNR0 is set to 0, PWM0 counter will be held. PWM1~PWM5 performs the same function as PWM0

Note: One-shot operation only support edge alignment mode.

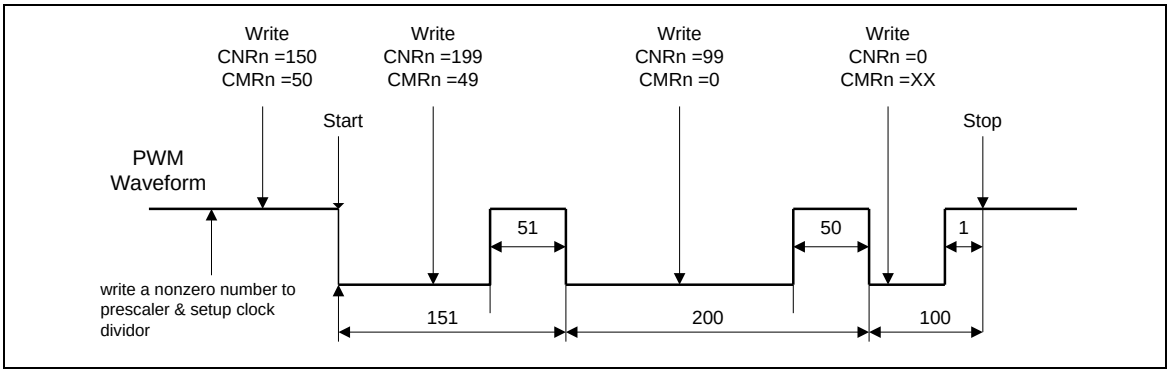


Figure 6.7-15 PWM Double Buffering Illustration

6.7.5.3 Modulate Duty Ratio

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

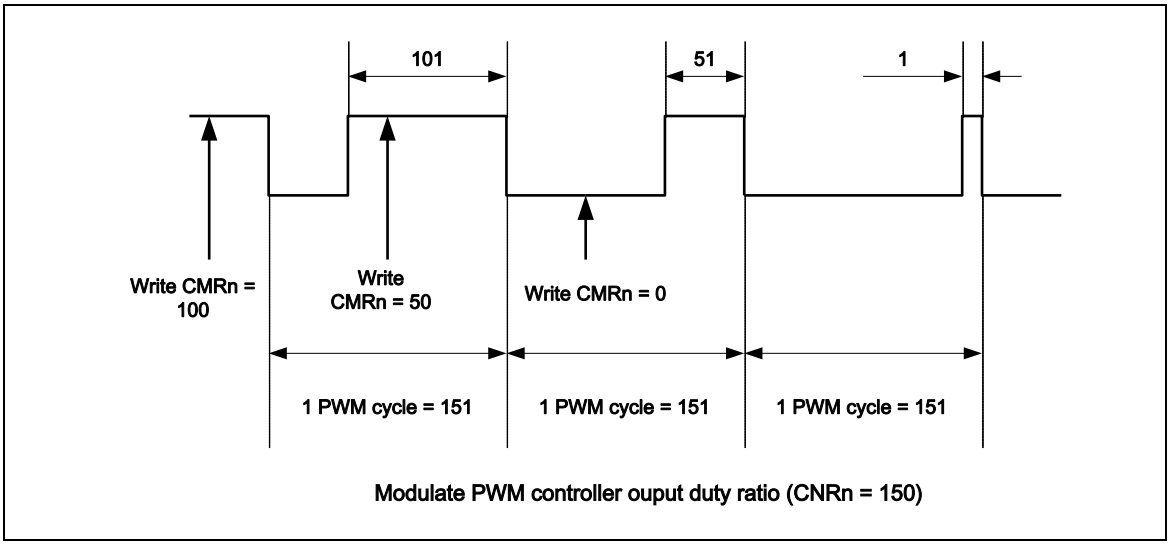


Figure 6.7-16 PWM Controller Output Duty Ratio

6.7.5.4 PWM Operation Modes

This powerful PWM unit supports independent mode which may be applied to DC or BLDC motor system, Complementary mode with dead-zone 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 PWM2 and PWM4 synchronous with PWM0 generator, may simplify updating duty control in DC and BLDC motor applications.

6.7.5.5 Independent Mode

Independent mode is enabled when PWMMOD[1:0] = 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.

6.7.5.6 Complementary Mode

Complementary mode is enabled when PWMMOD[1:0] = 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 PGn, always be the complement of the corresponding even PWM signal. For example, PG1 will be the complement of PG0. PG3 will be the complement of PG2 and PG5 will be the complement of PG4. The time base for the PWM module is provided by its own 16-bit timer, which also incorporates selectable pre-scalar options.

6.7.5.7 Dead-zone Insertion

The dead-zone generator inserts an “off” period called “dead-zone” 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-zone insertion. The complementary outputs are delayed until the timer counts down to 0.

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

dead-zone = PWM_CLK * (DZIxy[7:0]+1). where xy, could be 01, 23, 45

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

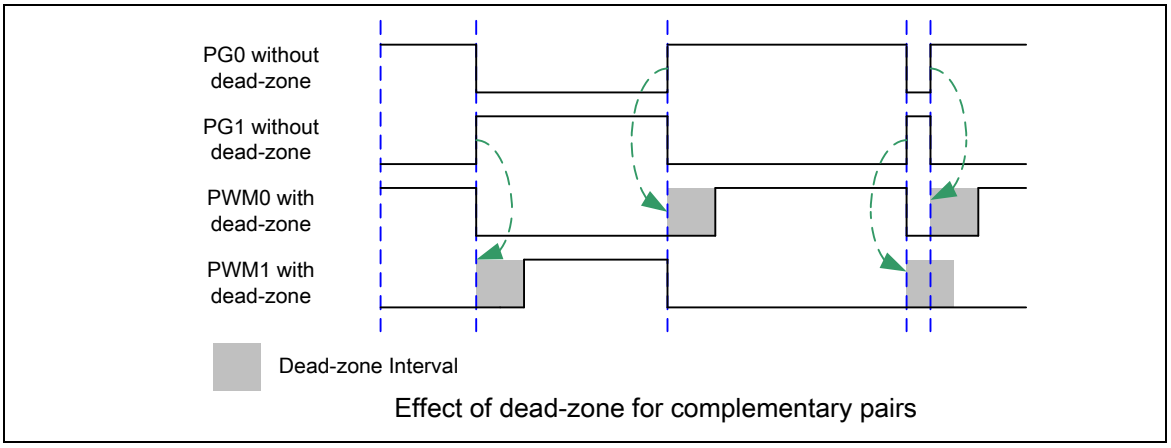


Figure 6.7-17 Dead-zone Insertion

In Power inverter applications, a dead-zone insertion avoids the upper and lower switches of the half bridge from being active at the same time. Hence the dead-zone 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.

6.7.5.8 Synchronous Mode

Synchronous mode is enabled when PWMMOD[1:0] = 10.

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

PG1=PG0, PG3=PG2 and PG5=PG4.

6.7.5.9 Group Mode

Group mode is enabled when GRP (PCR[30]) = 1.

This device supports Group mode control. This control allows all even PWM channels output to be duty controllable by PWM0 duty register.

If GRP = 1, both (PG2, PG3) and (PG4, PG5) pairs will follow (PG0, PG1), which imply;

PG4 = PG2 = PG0;

PG5 = PG3 = PG1 = invert (PG0) if Complementary mode is enabled (PWMMOD[1:0] = 01)

For Application, please do not use Group and Synchronous mode simultaneously because the Synchronous mode will be inactive.

6.7.5.10 Polarity Control

Each PWM port of PWM0 ~ PWM5 has independent polarity control to configure the polarity of active state of PWM output. By default, the PWM output is active high.

The following diagram shows the initial state before PWM starts with different polarity settings.

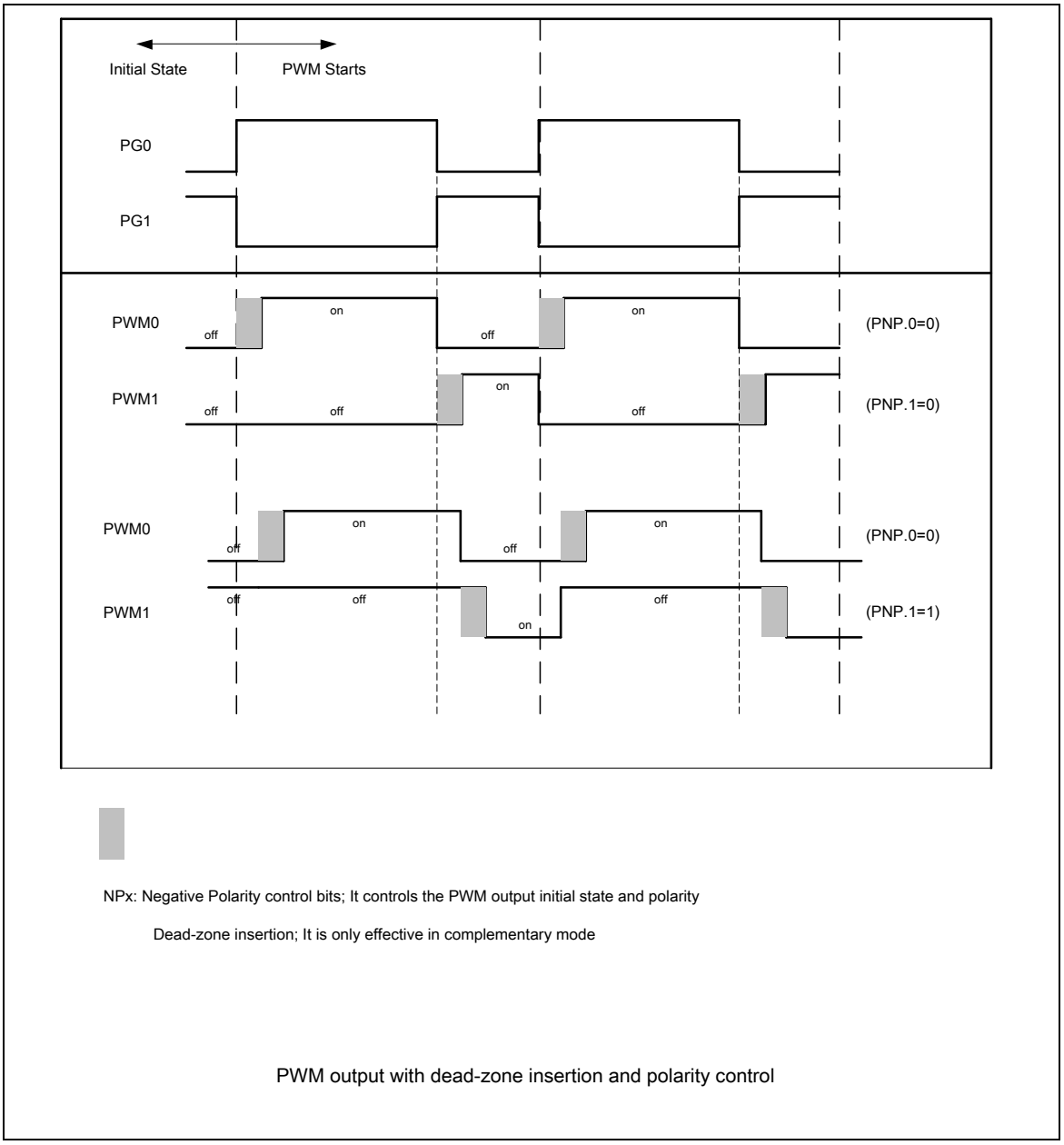


Figure 6.7-18 Initial State and Polarity Control with Rising Edge Dead-zone Insertion

6.7.6 PWM for Motor Control Interrupt Architecture

There are four interrupt sources for a PWM unit, which are PWM period flag (PWMPIF), PWM duty interrupt (PWMDIF), Brake0 flag (BKF0) and Brake1 flag (BKF1). The bit BRKIE (PIER[16]) controls the brake interrupt enable; the bit PWMPIEn (PIER[0] ~ PIER[5]) controls the PWM periodic interrupt enable; and the bit PWMDIEn (PIER[8] ~ PIER[13]) controls the PWM duty interrupt enable. Note that all the interrupt flags are set by hardware and must be cleared by software.

demonstrates the architecture of Motor Control PWM interrupts.

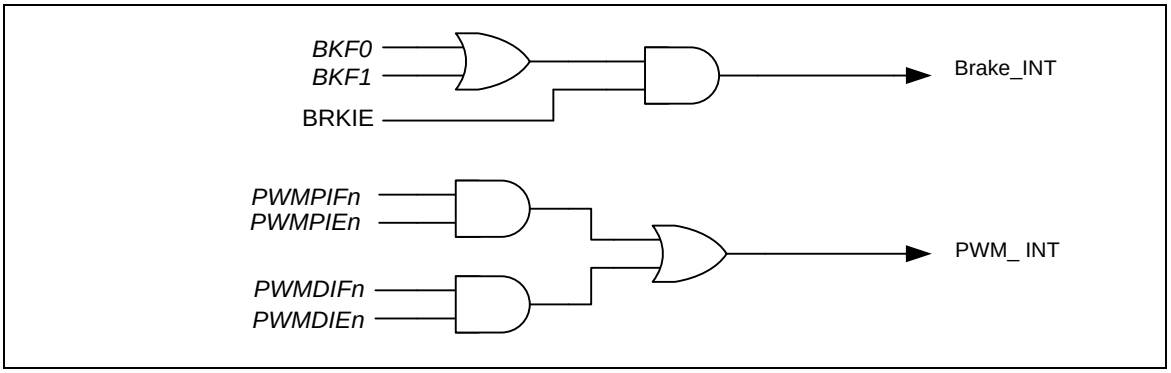


Figure 6.7-19 Motor Control PWM Architecture

6.7.6.1 PWM Brake Function

This device supported two external brake pins: BKP0 and BKP1 pins. The Brake function is controlled by the contents of the PFBCON registers.

Since the both brake conditions being asserted will automatically raise the BKF 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 ~ PWM5 enable bits will be disabled by hardware. User program must clear fault brake event flag (BKF) first, then write the PWM enable bits again to release PWM Brake state.

6.7.6.2 PWM Phase Change Function

PWM supports phase change function, by configuring both PHCHG and PHCHG_NXT register, each time when time-out event coming, PHCHG's value will be updated by PHCHG_NXT's value automatically, PHCHG's bit field is identical with PHCHG_NXT's, each time when PHCHG updated, the related function will also change,

6.7.7 PWM Timer Start Procedure

The following procedure is recommended for starting PWM-Timer.

1. Configure prescaler register (PPR) for setting clock prescaler (CPxx).
2. Configure clock select register (CSR) for setting clock source select (CSRx).
3. Configure PWM control register (PCR) for setting auto-reload mode (CHxMOD = 1), PWM aligned type (PWMTYPE) and DISABLE PWM-Timer (CHxEN = 0).
4. Configure PWM control register (PCR) for setting inverter on/off (CHxINV), and Dead-zone generator on/off (DZENn). (Optional)
5. Configure PDZIR register to set dead-zone interval. (Optional)
6. Configure comparator register (CMRx) for setting PWM duty (CMRx).
7. Configure PWM counter register (CNRx) for setting PWM-Timer loaded value (CNRx).
8. Configure PWM interrupt enable register (PIER) for setting PWM period interrupt type (INTTYPE), PWM period interrupt enable bit (PWMPIEx) and PWM duty interrupt enable bit (PWMDIEx). (Optional)
9. Configure PWM output enable register (POE) to enable PWM output channel (PWMx)

10. Configure PWM control register (PCR) to enable PWM-Timer (CHxEN = 1)

6.7.8 PWM Timer Stop Procedure

Method 1:

Set 16-bit counter register (CNRx) as 0. When interrupt request happened, disable PWM-Timer (CHxEN in PCR). (Recommended)

Method 2:

Disable PWM-Timer directly (CHxEN in PCR) (Not recommended)

The reason why this method is not recommended is that disabling CHxEN 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.

6.7.9 Register Map

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

Register	Offset	R/W	Description	Reset Value
PWM Base Address: **PWM_BA = 0x4004_0000**				
PPR	PWM_BA+0x00	R/W	PWM Pre-scale Register	0x0000_0000
CSR	PWM_BA+0x04	R/W	PWM Clock Select Register	0x0000_0000
PCR	PWM_BA+0x08	R/W	PWM Control Register	0x0000_0000
CNR0	PWM_BA+0x0C	R/W	PWM Counter Register 0	0x0000_0000
CNR1	PWM_BA+0x10	R/W	PWM Counter Register 1	0x0000_0000
CNR2	PWM_BA+0x14	R/W	PWM Counter Register 2	0x0000_0000
CNR3	PWM_BA+0x18	R/W	PWM Counter Register 3	0x0000_0000
CNR4	PWM_BA+0x1C	R/W	PWM Counter Register 4	0x0000_0000
CNR5	PWM_BA+0x20	R/W	PWM Counter Register 5	0x0000_0000
CMR0	PWM_BA+0x24	R/W	PWM Comparator Register 0	0x0000_0000
CMR1	PWM_BA+0x28	R/W	PWM Comparator Register 1	0x0000_0000
CMR2	PWM_BA+0x2C	R/W	PWM Comparator Register 2	0x0000_0000
CMR3	PWM_BA+0x30	R/W	PWM Comparator Register 3	0x0000_0000
CMR4	PWM_BA+0x34	R/W	PWM Comparator Register 4	0x0000_0000
CMR5	PWM_BA+0x38	R/W	PWM Comparator Register 5	0x0000_0000
PIER	PWM_BA+0x54	R/W	PWM Interrupt Enable Control Register	0x0000_0000
PIIR	PWM_BA+0x58	R/W	PWM Interrupt Indication Register	0x0000_0000
PWMPOE	PWM_BA+0x5C	R/W	PWM Output Enable for Channel 0~5	0x0000_0000
PFBCON	PWM_BA+0x60	R/W	PWM Fault Brake Control Register	0x0000_0000
PDZIR	PWM_BA+0x64	R/W	PWM Dead-zone Interval Register	0x0000_0000
TRGCON0	PWM_BA+0x68	R/W	PWM Trigger Control Register 0	0x0000_0000
TRGCON1	PWM_BA+0x6C	R/W	PWM Trigger Control Register 1	0x0000_0000
TRGSTS0	PWM_BA+0x70	R/W	PWM Trigger Status Register 0	0x0000_0000
TRGSTS1	PWM_BA+0x74	R/W	PWM Trigger Status Register 1	0x0000_0000
PHCHG	PWM_BA+0x78	R/W	Phase Change Register	0x0000_3F00
PHCHGNXT	PWM_BA+0x7C	R/W	Next Phase Change Register	0x0000_3F00

PHCHGMASK	PWM_BA+0x80	R/W	Phase Change MASK Register	0x0000_0000
INTACCUCTL	PWM_BA+0x84	R/W	Period Interrupt Accumulation Control Register	0x0000_00F0

6.7.10 Register Description

PWM Pre-Scale Register (PPR)

Register	Offset	R/W	Description	Reset Value
PPR	PWM_BA+0x00	R/W	PWM Pre-scale Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CP45							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CP23							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CP01							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**CP45[7:0]**	**Clock Prescaler 4 For PWM Counter 4 And 5** Clock input is divided by (CP45 + 1) before it is fed to the corresponding PWM counter. If CP45 = 0, the clock prescaler 4 output clock will be stopped. So the corresponding PWM counter will also be stopped.
[15:8]	**CP23[7:0]**	**Clock Prescaler 2 For PWM Counter 2 And 3** Clock input is divided by (CP23 + 1) before it is fed to the corresponding PWM counter. If CP23 = 0, the clock prescaler 2 output clock will be stopped. So the corresponding PWM counter will also be stopped.
[7:0]	**CP01[7:0]**	**Clock Prescaler 0 For PWM Counter 0 And 1** Clock input is divided by (CP01 + 1) before it is fed to the corresponding PWM counter. If CP01 = 0, the clock prescaler 0 output clock will be stopped. So the corresponding PWM counter will also be stopped.

PWM Clock Selector Register (CSR)

Register	Offset	R/W	Description	Reset Value
CSR	PWM_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	CSR5			Reserved	CSR4		
15	14	13	12	11	10	9	8
Reserved	CSR3			Reserved	CSR2		
7	6	5	4	3	2	1	0
Reserved	CSR1			Reserved	CSR0		

Bits	Description	
[31:23]	**Reserved**	Reserved.
[22:20]	**CSR5[2:0]**	**Timer 5 Clock Source Selection** Select clock input for PWM timer. 000 = Input Clock Divided by 2. 001 = Input Clock Divided by 4. 010 = Input Clock Divided by 8. 011 = Input Clock Divided by 16. 100 = Input Clock Divided by 1.
[19]	**Reserved**	Reserved.
[18:16]	**CSR4[2:0]**	**Timer 4 Clock Source Selection** Select clock input for PWM timer. (Table is the same as CSR5.)
[15]	**Reserved**	Reserved.
[14:12]	**CSR3[2:0]**	**Timer 3 Clock Source Selection** Select clock input for PWM timer. (Table is the same as CSR5.)
[11]	**Reserved**	Reserved.
[10:8]	**CSR2[2:0]**	**Timer 2 Clock Source Selection** Select clock input for PWM timer. (Table is the same as CSR5.)
[7]	**Reserved**	Reserved.
[6:4]	**CSR1[2:0]**	**Timer 1 Clock Source Selection** Select clock input for PWM timer. (Table is the same as CSR5.)

Bits	Description	
[3]	**Reserved**	Reserved.
[2:0]	**CSR0[2:0]**	**Timer 0 Clock Source Selection** Select clock input for PWM timer. (Table is the same as CSR5.)

PWM Control Register (PCR)

Register	Offset	R/W	Description	Reset Value
PCR	PWM_BA+0x08	R/W	PWM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PWMTYPE	**GRP**	**PWMMOD**		**CLRPWM**	**DZEN45**	**DZEN23**	**DZEN01**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CH5MOD	**CH5INV**	**Reserved**	**CH5EN**	**CH4MOD**	**CH4INV**	**Reserved**	**CH4EN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CH3MOD	**CH3INV**	**Reserved**	**CH3EN**	**CH2MOD**	**CH2INV**	**Reserved**	**CH2EN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CH1MOD	**CH1INV**	**Reserved**	**CH1EN**	**CH0MOD**	**CH0INV**	**DB_MODE**	**CH0EN**

Bits	Description	
[31]	**PWMTYPE**	**PWM Aligned Type Selection Bit** 0 = Edge-aligned type. 1 = Center-aligned type.
[30]	**GRP**	**Group Bit** 0 = The signals timing of all PWM channels are independent. 1 = Unify the signals timing of PWM0, PWM2 and PWM4 in the same phase which is controlled by PWM0 and also unify the signals timing of PWM1, PWM3 and PWM5 in the same phase which is controlled by PWM1.
[29:28]	**PWMMOD[1:0]**	**PWM Operating Mode Selection** 00 = Independent mode. 01 = Complementary mode. 10 = Synchronized mode. 11 = Reserved.
[27]	**CLRPWM**	**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]	**DZEN45**	**Dead-zone 4 Generator Enable Control (PWM4 And PWM5 Pair For PWM Group)** 0 = Disabled. 1 = Enabled. **Note:** When the dead-zone generator is enabled, the pair of PWM4 and PWM5 becomes a complementary pair for PWM group.
[25]	**DZEN23**	**Dead-zone 2 Generator Enable Control (PWM2 And PWM3 Pair For PWM Group)** 0 = Disabled. 1 = Enabled. **Note:** When the dead-zone generator is enabled, the pair of PWM2 and PWM3 becomes a complementary pair for PWM group.

Bits	Description	
[24]	**DZEN01**	**Dead-zone 0 Generator Enable Control (PWM0 And PWM1 Pair For PWM Group)** 0 = Disabled. 1 = Enabled. **Note:** When the dead-zone generator is enabled, the pair of PWM0 and PWM1 becomes a complementary pair for PWM group.
[23]	**CH5MOD**	**PWM-timer 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 CNR5 and CMR5 cleared.
[22]	**CH5INV**	**PWM-timer 5 Output Inverter Enable Control** 0 = Inverter Disabled. 1 = Inverter Enabled.
[21]	**Reserved**	Reserved.
[20]	**CH5EN**	**PWM-timer 5 Enable/Disable Start Run** 0 = Corresponding PWM-timer running Stopped. 1 = Corresponding PWM-timer start run Enabled.
[19]	**CH4MOD**	**PWM-timer 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 CNR4 and CMR4 cleared.
[18]	**CH4INV**	**PWM-timer 4 Output Inverter Enable Control** 0 = Inverter Disabled. 1 = Inverter Enabled.
[17]	**Reserved**	Reserved.
[16]	**CH4EN**	**PWM-timer 4 Enable/Disable Start Run** 0 = Corresponding PWM-timer running Stopped. 1 = Corresponding PWM-timer start run Enabled.
[15]	**CH3MOD**	**PWM-timer 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 CNR3 and CMR3 cleared.
[14]	**CH3INV**	**PWM-timer 3 Output Inverter Enable Control** 0 = Inverter Disabled. 1 = Inverter Enabled.
[13]	**Reserved**	Reserved.
[12]	**CH3EN**	**PWM-timer 3 Enable/Disable Start Run** 0 = Corresponding PWM-timer running Stopped. 1 = Corresponding PWM-timer start run Enabled.
[11]	**CH2MOD**	**PWM-timer 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 CNR2 and CMR2 cleared.

Bits	Description	
[10]	**CH2INV**	**PWM-timer 2 Output Inverter Enable Control** 0 = Inverter Disabled. 1 = Inverter Enabled.
[9]	**Reserved**	Reserved.
[8]	**CH2EN**	**PWM-timer 2 Enable/Disable Start Run** 0 = Corresponding PWM-timer running Stopped. 1 = Corresponding PWM-timer start run Enabled.
[7]	**CH1MOD**	**PWM-timer 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 CNR1 and CMR1 cleared.
[6]	**CH1INV**	**PWM-timer 1 Output Inverter ON/OFF** 0 = Inverter OFF. 1 = Inverter ON.
[5]	**Reserved**	Reserved.
[4]	**CH1EN**	**PWM-timer 1 Enable/Disable Start Run** 0 = Corresponding PWM-timer running Stopped. 1 = Corresponding PWM-timer start run Enabled.
[3]	**CH0MOD**	**PWM-timer 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 CNR0 and CMR0 cleared.
[2]	**CH0INV**	**PWM-timer 0 Output Inverter Enable Control** 0 = Inverter Disabled. 1 = Inverter Enabled.
[1]	**DB_MODE**	**PWM Debug Mode Configuration Bit (Available In DEBUG Mode Only)** 0 = Safe mode: The timer is frozen and PWM outputs are shut down Safe state for the inverter. The timer can still be re-started from where it stops. 1 = Normal mode: The timer 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]	**CH0EN**	**PWM-timer 0 Enable/Disable Start Run** 0 = Corresponding PWM-timer running Stopped. 1 = Corresponding PWM-timer start run Enabled.

PWM Counter Register 0-5 (CNR0-5)

Register	Offset	R/W	Description	Reset Value
CNR0	PWM_BA+0x0C	R/W	PWM Counter Register 0	0x0000_0000
CNR1	PWM_BA+0x10	R/W	PWM Counter Register 1	0x0000_0000
CNR2	PWM_BA+0x14	R/W	PWM Counter Register 2	0x0000_0000
CNR3	PWM_BA+0x18	R/W	PWM Counter Register 3	0x0000_0000
CNR4	PWM_BA+0x1C	R/W	PWM Counter Register 4	0x0000_0000
CNR5	PWM_BA+0x20	R/W	PWM Counter 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**
CNRn							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNRn							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CNRn**	**PWM Counter/Timer Loaded Value** CNRn determines the PWM period. n = 0, 1..5. Edge-aligned mode: PWM frequency = HCLK/((prescale+1)*(clock divider))/(CNRn+1); where xy, could be 01, 23, 45 depending on the selected PWM channel. Duty ratio = (CMRn+1)/(CNRn+1). CMRn >= CNRn: PWM output is always high. CMRn < CNRn: PWM low width = (CNRn-CMRn) unit; PWM high width = (CMR+1) unit. CMRn = 0: PWM low width = (CNRn) unit; PWM high width = 1 unit. Center-aligned mode: PWM frequency = HCLK/((prescale+1)*(clock divider))/ (2*CNRn+1); where xy, could be 01, 23, 45 depending on the selected PWM channel. Duty ratio = (CNRn - CMRn)/(CNRn+1). CMRn >= CNRn: PWM output is always low. CMRn < CNRn: PWM low width = (CMRn + 1) * 2 unit; PWM high width = (CNRn - CMRn) * 2 unit. CMRn = 0: PWM low width = 2 unit; PWM high width = (CNRn) * 2 unit. (Unit = One PWM clock cycle). **Note:** Any write to CNRn will take effect in next PWM cycle.

PWM Comparator Register 0-5 (CMR0-5)

Register	Offset	R/W	Description	Reset Value
CMR0	PWM_BA+0x24	R/W	PWM Comparator Register 0	0x0000_0000
CMR1	PWM_BA+0x28	R/W	PWM Comparator Register 1	0x0000_0000
CMR2	PWM_BA+0x2C	R/W	PWM Comparator Register 2	0x0000_0000
CMR3	PWM_BA+0x30	R/W	PWM Comparator Register 3	0x0000_0000
CMR4	PWM_BA+0x34	R/W	PWM Comparator Register 4	0x0000_0000
CMR5	PWM_BA+0x38	R/W	PWM Comparator Register 5	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CMRn**	**PWM Comparator Bits** CMR determines the PWM duty. n = 0, 1..5. Edge-aligned mode: PWM frequency = HCLK/((prescale+1)*(clock divider))/(CNRn+1); where xy, could be 01, 23, 45 depending on the selected PWM channel. Duty ratio = (CMRn+1)/(CNRn+1). CMRn >= CNRn: PWM output is always high. CMRn < CNRn: PWM low width = (CNRn-CMRn) unit; PWM high width = (CMR+1) unit. CMRn = 0: PWM low width = (CNRn) unit; PWM high width = 1 unit. Center-aligned mode: PWM frequency = HCLK/((prescale+1)*(clock divider)) /(2*CNRn+1); where xy, could be 01, 23, 45 depending on the selected PWM channel. Duty ratio = (CNRn - CMRn)/(CNRn+1). CMRn >= CNRn: PWM output is always low. CMRn < CNRn: PWM low width = (CMRn + 1) * 2 unit; PWM high width = (CNRn - CMRn) * 2 unit. CMRn = 0: PWM low width = 2 unit; PWM high width = (CNRn) * 2 unit. (Unit = One PWM clock cycle).

Bits	Description	
		Note: Any write to CMRn will take effect in next PWM cycle.

PWM Interrupt Enable Control Register (PIER)

Register	Offset	R/W	Description	Reset Value
PIER	PWM_BA+0x54	R/W	PWM Interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**INT_TYPE**	**BRKIE**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**PWMDIE5**	**PWMDIE4**	**PWMDIE3**	**PWMDIE2**	**PWMDIE1**	**PWMDIE0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**PWMPIE5**	**PWMPIE4**	**PWMPIE3**	**PWMPIE2**	**PWMPIE1**	**PWMPIE0**

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**INT_TYPE**	**PWM Interrupt Type Selection Bit** 0 = PWMPIFn will be set if PWM counter underflows. 1 = PWMPIFn will be set if PWM counter matches CNRn register. **Note:** This bit is effective when PWM in central align mode only.
[16]	**BRKIE**	**Fault Brake0 And 1 Interrupt Enable Control** 0 = Disabling flags BKF0 and BKF1 to trigger PWM interrupt. 1 = Enabling flags BKF0 and BKF1 can trigger PWM interrupt.
[15:14]	**Reserved**	Reserved.
[13]	**PWMDIE5**	**PWM Channel 5 Duty Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[12]	**PWMDIE4**	**PWM Channel 4 Duty Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[11]	**PWMDIE3**	**PWM Channel 3 Duty Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[10]	**PWMDIE2**	**PWM Channel 2 Duty Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[9]	**PWMDIE1**	**PWM Channel 1 Duty Interrupt Enable Control** 0 = Disabled. 1 = Enabled.

Bits	Description	
[8]	**PWMDIE0**	**PWM Channel 0 Duty Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[7:6]	**Reserved**	Reserved.
[5]	**PWMPIE5**	**PWM Channel 5 Period Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[4]	**PWMPIE4**	**PWM Channel 4 Period Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[3]	**PWMPIE3**	**PWM Channel 3 Period Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[2]	**PWMPIE2**	**PWM Channel 2 Period Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[1]	**PWMPIE1**	**PWM Channel 1 Period Interrupt Enable Control** 0 = Disabled. 1 = Enabled.
[0]	**PWMPIE0**	**PWM Channel 0 Period Interrupt Enable Control** 0 = Disabled. 1 = Enabled.

PWM Interrupt Indication Register (PIIR)

Register	Offset	R/W	Description	Reset Value
PIIR	PWM_BA+0x58	R/W	PWM Interrupt Indication Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**BKF1**	**BKF0**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**PWMDIF5**	**PWMDIF4**	**PWMDIF3**	**PWMDIF2**	**PWMDIF1**	**PWMDIF0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**PWMPIF5**	**PWMPIF4**	**PWMPIF3**	**PWMPIF2**	**PWMPIF1**	**PWMPIF0**

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**BKF1**	**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:** Software can write 1 to clear this bit.
[16]	**BKF0**	**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:** Software can write 1 to clear this bit.
[15:14]	**Reserved**	Reserved.
[13]	**PWMDIF5**	**PWM Channel 5 Duty Interrupt Flag** Flag is set by hardware when a channel 5 PWM counter reaches CMR5 in down-count direction. **Note:** Software can write 1 to clear this bit.
[12]	**PWMDIF4**	**PWM Channel 4 Duty Interrupt Flag** Flag is set by hardware when a channel 4 PWM counter reaches CMR4 in down-count direction. **Note:** Software can write 1 to clear this bit.
[11]	**PWMDIF3**	**PWM Channel 3 Duty Interrupt Flag** Flag is set by hardware when a channel 3 PWM counter reaches CMR3 in down-count direction. **Note:** Software can write 1 to clear this bit.
[10]	**PWMDIF2**	**PWM Channel 2 Duty Interrupt Flag** Flag is set by hardware when a channel 2 PWM counter reaches CMR2 in down-count direction. **Note:** Software can write 1 to clear this bit.

Bits	Description	
[9]	**PWMDIF1**	**PWM Channel 1 Duty Interrupt Flag** Flag is set by hardware when a channel 1 PWM counter reaches CMR1 in down-count direction. **Note:** Software can write 1 to clear this bit.
[8]	**PWMDIF0**	**PWM Channel 0 Duty Interrupt Flag** Flag is set by hardware when a channel 0 PWM counter reaches CMR0 in down-count direction. **Note:** Software can write 1 to clear this bit.
[7:6]	**Reserved**	Reserved.
[5]	**PWMPIF5**	**PWM Channel 5 Period Interrupt Flag** Flag is set by hardware when PWM5 down counter reaches zero. **Note:** Software can write 1 to clear this bit.
[4]	**PWMPIF4**	**PWM Channel 4 Period Interrupt Flag** Flag is set by hardware when PWM4 down counter reaches zero. **Note:** Software can write 1 to clear this bit.
[3]	**PWMPIF3**	**PWM Channel 3 Period Interrupt Flag** Flag is set by hardware when PWM3 down counter reaches zero. **Note:** Software can write 1 to clear this bit.
[2]	**PWMPIF2**	**PWM Channel 2 Period Interrupt Flag** Flag is set by hardware when PWM2 down counter reaches zero. **Note:** Software can write 1 to clear this bit.
[1]	**PWMPIF1**	**PWM Channel 1 Period Interrupt Flag** Flag is set by hardware when PWM1 down counter reaches zero. **Note:** Software can write 1 to clear this bit.
[0]	**PWMPIF0**	**PWM Channel 0 Period Interrupt Flag** Flag is set by hardware when PWM0 down counter reaches zero. **Note:** Software can write 1 to clear this bit.

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

PWM Output Control Register (PWMPOE)

Register	Offset	R/W	Description	Reset Value
PWMPOE	PWM_BA+0x5C	R/W	PWM Output Enable for Channel 0~5	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**PWM5**	**PWM4**	**PWM3**	**PWM2**	**PWM1**	**PWM0**

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**PWM5**	**PWM Channel 5 Output Enable Control** 0 = PWM channel 5 output to pin Disabled. 1 = PWM channel 5 output to pin Enabled. **Note:** The corresponding GPIO pin must be switched to PWM function.
[4]	**PWM4**	**PWM Channel 4 Output Enable Control** 0 = PWM channel 4 output to pin Disabled. 1 = PWM channel 4 output to pin Enabled. **Note:** The corresponding GPIO pin must be switched to PWM function.
[3]	**PWM3**	**PWM Channel 3 Output Enable Control** 0 = PWM channel 3 output to pin Disabled. 1 = PWM channel 3 output to pin Enabled. **Note:** The corresponding GPIO pin must be switched to PWM function.
[2]	**PWM2**	**PWM Channel 2 Output Enable Control** 0 = PWM channel 2 output to pin Disabled. 1 = PWM channel 2 output to pin Enabled. **Note:** The corresponding GPIO pin must be switched to PWM function.
[1]	**PWM1**	**PWM Channel 1 Output Enable Control** 0 = PWM channel 1 output to pin Disabled. 1 = PWM channel 1 output to pin Enabled. **Note:** The corresponding GPIO pin must be switched to PWM function.
[0]	**PWM0**	**PWM Channel 0 Output Enable Control** 0 = PWM channel 0 output to pin Disabled. 1 = PWM channel 0 output to pin Enabled. **Note:** The corresponding GPIO pin must be switched to PWM function.

PWM Fault Brake Control Register (PFBCON)

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

31	30	29	28	27	26	25	24
D7BKO7	**D6BKO6**	**PWMBKO5**	**PWMBKO4**	**PWMBKO3**	**PWMBKO2**	**PWMBKO1**	**PWMBKO0**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BKF	**Reserved**			**CPO1BKEN**	**CPO0BKEN**	**BKEN1**	**BKEN0**

Bits	Description	
[31]	**D7BKO7**	**D7 Brake Output Select Bit** 0 = D7 output low when fault brake conditions asserted. 1 = D7 output high when fault brake conditions asserted.
[30]	**D6BKO6**	**D6 Brake Output Select Bit** 0 = D6 output low when fault brake conditions asserted. 1 = D6 output high when fault brake conditions asserted.
[29]	**PWMBKO5**	**PWM Channel 5 Brake Output Select Bit** 0 = PWM output low when fault brake conditions asserted. 1 = PWM output high when fault brake conditions asserted.
[28]	**PWMBKO4**	**PWM Channel 4 Brake Output Select Bit** 0 = PWM output low when fault brake conditions asserted. 1 = PWM output high when fault brake conditions asserted.
[27]	**PWMBKO3**	**PWM Channel 3 Brake Output Select Bit** 0 = PWM output low when fault brake conditions asserted. 1 = PWM output high when fault brake conditions asserted.
[26]	**PWMBKO2**	**PWM Channel 2 Brake Output Select Bit** 0 = PWM output low when fault brake conditions asserted. 1 = PWM output high when fault brake conditions asserted.
[25]	**PWMBKO1**	**PWM Channel 1 Brake Output Select Bit** 0 = PWM output low when fault brake conditions asserted. 1 = PWM output high when fault brake conditions asserted.
[24]	**PWMBKO0**	**PWM Channel 0 Brake Output Select Bit** 0 = PWM output low when fault brake conditions asserted. 1 = PWM output high when fault brake conditions asserted.
[23:8]	**Reserved**	Reserved.

Bits	Description	
[7]	**BKF**	**PWM Fault Brake Event Flag (Write 1 Clear)** 0 = PWM output initial state when fault brake conditions asserted. 1 = PWM output fault brake state when fault brake conditions asserted. Software can write 1 to clear this bit and must clear this bit before restart PWM counter.
[6:4]	**Reserved**	Reserved.
[3]	**CPO1BKEN**	**BKP0 Fault Brake Function Source Selection** 0 = EINT0 as one brake source in BKP0. 1 = CPO1 as one brake source in BKP0.
[2]	**CPO0BKEN**	**BKP1 Fault Brake Function Source Selection** 0 = EINT1 as one brake source in BKP1. 1 = CPO0 as one brake source in BKP1.
[1]	**BKEN1**	**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]	**BKEN0**	**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-Zone Interval Register (PDZIR)

Register	Offset	R/W	Description	Reset Value
PDZIR	PWM_BA+0x64	R/W	PWM Dead-zone Interval Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DZI45							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DZI23							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DZI01							

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**DZI45[7:0]**	**Dead-zone Interval Register For Pair Of Channel4 And Channel5 (PWM4 And PWM5 Pair)** These 8 bits determine dead-zone length. The unit time of dead-zone length is received from corresponding CSR bits.
[15:8]	**DZI23[7:0]**	**Dead-zone Interval Register For Pair Of Channel2 And Channel3 (PWM2 And PWM3 Pair)** These 8 bits determine dead-zone length. The unit time of dead-zone length is received from corresponding CSR bits.
[7:0]	**DZI01[7:0]**	**Dead-zone Interval Register For Pair Of Channel0 And Channel1 (PWM0 And PWM1 Pair)** These 8 bits determine dead-zone length. The unit time of dead-zone length is received from corresponding CSR bits.

PWM Trigger ADC Control Register (TRGCON0)

Register	Offset	R/W	Description	Reset Value
TRGCON0	PWM_BA+0x68	R/W	PWM Trigger Control Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				P3TRGEN	CM3TRGFEN	CNT3TRGEN	CM3TRGREN
23	22	21	20	19	18	17	16
Reserved				P2TRGEN	CM2TRGFEN	CNT2TRGEN	CM2TRGREN
15	14	13	12	11	10	9	8
Reserved				P1TRGEN	CM1TRGFEN	CNT1TRGEN	CM1TRGREN
7	6	5	4	3	2	1	0
Reserved				P0TRGEN	CM0TRGFEN	CNT0TRGEN	CM0TRGREN

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**P3TRGEN**	**Enable PWM Trigger ADC Function While Channel3's Counter Matching 0** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.
[26]	**CM3TRGFEN**	**Enable PWM Trigger ADC Function While Channel3's Counter Matching CMR3 In Down-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.
[25]	**CNT3TRGEN**	**Enable PWM Trigger ADC Function While Channel3's Counter Matching CNR3** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.
[24]	**CM3TRGREN**	**Enable PWM Trigger ADC Function While Channel3's Counter Matching CMR3 In Up-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.
[23:20]	**Reserved**	Reserved.
[19]	**P2TRGEN**	**Enable PWM Trigger ADC Function While Channel2's Counter Matching 0** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.

Bits	Description	
[18]	**CM2TRGFEN**	**Enable PWM Trigger ADC Function While Channel2's Counter Matching CMR2 In Down-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.
[17]	**CNT2TRGEN**	**Enable PWM Trigger ADC Function While Channel2's Counter Matching CNR2** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.
[16]	**CM2TRGREN**	**Enable PWM Trigger ADC Function While Channel2's Counter Matching CMR2 In Up-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.
[15:12]	**Reserved**	Reserved.
[11]	**P1TRGEN**	**Enable PWM Trigger ADC Function While Channel1's Counter Matching 0** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.
[10]	**CM1TRGFEN**	**Enable PWM Trigger ADC Function While Channel1's Counter Matching CMR1 In Down-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.
[9]	**CNT1TRGEN**	**Enable PWM Trigger ADC Function While Channel1's Counter Matching CNR1** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.
[8]	**CM1TRGREN**	**Enable PWM Trigger ADC Function While Channel1's Counter Matching CMR1 In Up-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.
[7:4]	**Reserved**	Reserved.
[3]	**P0TRGEN**	**Enable PWM Trigger ADC Function While Channel0's Counter Matching 0** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.

Bits	Description	
[2]	**CM0TRGFEN**	**Enable PWM Trigger ADC Function While Channel0's Counter Matching CMR0 In Down-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.
[1]	**CNT0TRGEN**	**Enable PWM Trigger ADC Function While Channel0's Counter Matching CNR0** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.
[0]	**CM0TRGREN**	**Enable PWM Trigger ADC Function While Channel0's Counter Matching CMR0 In Up-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.

PWM Trigger ADC Control Register (TRGCON1)

Register	Offset	R/W	Description	Reset Value
TRGCON1	PWM_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				**P5TRGEN**	**CM5TRGFEN**	**CNT5TRGEN**	**CM5TRGREN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**P4TRGEN**	**CM4TRGFEN**	**CNT4TRGEN**	**CM4TRGREN**

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11]	**P5TRGEN**	**Enable PWM Trigger ADC Function While Channel5's Counter Matching 0** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.
[10]	**CM5TRGFEN**	**Enable PWM Trigger ADC Function While Channel5's Counter Matching CMR5 In Down-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.
[9]	**CNT5TRGEN**	**Enable PWM Trigger ADC Function While Channel5's Counter Matching CNR5** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.
[8]	**CM5TRGREN**	**Enable PWM Trigger ADC Function While Channel5's Counter Matching CMR5 In Up-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.
[7:4]	**Reserved**	Reserved.
[3]	**P4TRGEN**	**Enable PWM Trigger ADC Function While Channel4's Counter Matching 0** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.

Bits	Description	
[2]	**CM4TRGFEN**	**Enable PWM Trigger ADC Function While Channel4's Counter Matching CMR4 In Down-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is valid for both center aligned mode and edged aligned mode.
[1]	**CNT4TRGEN**	**Enable PWM Trigger ADC Function While Channel4's Counter Matching CNR4** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.
[0]	**CM4TRGREN**	**Enable PWM Trigger ADC Function While Channel4's Counter Matching CMR4 In Up-count Direction** 0 = Disabled. 1 = Enabled. **Note:** This bit is only valid for PWM in center aligned mode. When PWM is in edged aligned mode, setting this bit is meaningless and will not take any effect.

PWM Trigger Status Register (TRGSTS0)

Register	Offset	R/W	Description	Reset Value
TRGSTS0	PWM_BA+0x70	R/W	PWM Trigger Status Register 0	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**PERID3FLAG**	**CMR3FLAG_F**	**CNT3FLAG**	**CMR3FLAG_R**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**PERID2FLAG**	**CMR2FLAG_F**	**CNT2FLAG**	**CMR2FLAG_R**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**PERID1FLAG**	**CMR1FLAG_F**	**CNT1FLAG**	**CMR1FLAG_R**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**PERID0FLAG**	**CMR0FLAG_F**	**CNT0FLAG**	**CMR0FLAG_R**

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**PERID3FLAG**	**When Counter Counting To Period, This Bit Will Be Set For Trigger ADC** **Note:** Software can write 1 to clear this bit.
[26]	**CMR3FLAG_F**	**When Counter Counting Down To CMR, This Bit Will Be Set For Trigger ADC** **Note:** Software can write 1 to clear this bit.
[25]	**CNT3FLAG**	**When Counter Counting To CNR, This Bit Will Be Set For Trigger ADC** **Note:** Software can write 1 to clear this bit.
[24]	**CMR3FLAG_R**	**When Counter Counting Up To CMR, This Bit Will Be Set For Trigger ADC** **Note:** Software can write 1 to clear this bit.
[23:20]	**Reserved**	Reserved.
[19]	**PERID2FLAG**	**ADC Trigger Flag By Period** **Note:** Software can write 1 to clear this bit.
[18]	**CMR2FLAG_F**	**ADC Trigger Flag By Counting Down To CMR** **Note:** Software can write 1 to clear this bit.
[17]	**CNT2FLAG**	**ADC Trigger Flag By Counting To CNR** **Note:** Software can write 1 to clear this bit.
[16]	**CMR2FLAG_R**	**ADC Trigger Flag By Counting Up To CMR** **Note:** Software can write 1 to clear this bit.
[15:12]	**Reserved**	Reserved.
[11]	**PERID1FLAG**	**ADC Trigger Flag By Period** **Note:** Software can write 1 to clear this bit.

Bits	Description	
[10]	**CMR1FLAG_F**	**ADC Trigger Flag By Counting Down To CMR** **Note:** Software can write 1 to clear this bit.
[9]	**CNT1FLAG**	**ADC Trigger Flag By Counting To CNR** **Note:** Software can write 1 to clear this bit.
[8]	**CMR1FLAG_R**	**ADC Trigger Flag By Counting Up To CMR** **Note:** Software can write 1 to clear this bit.
[7:4]	**Reserved**	Reserved.
[3]	**PERID0FLAG**	**ADC Trigger Flag By Period** **Note:** Software can write 1 to clear this bit.
[2]	**CMR0FLAG_F**	**ADC Trigger Flag By Counting Down To CMR** **Note:** Software can write 1 to clear this bit.
[1]	**CNT0FLAG**	**ADC Trigger Flag By Counting To CNR** **Note:** Software can write 1 to clear this bit.
[0]	**CMR0FLAG_R**	**ADC Trigger Flag By Counting Up To CMR** **Note:** Software can write 1 to clear this bit.

PWM Trigger Status Register (TRGSTS1)

Register	Offset	R/W	Description	Reset Value
TRGSTS1	PWM_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				**PERID5FLAG**	**CMR5FLAG_F**	**CNT5FLAG**	**CMR5FLAG_R**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**PERID4FLAG**	**CMR4FLAG_F**	**CNT4FLAG**	**CMR4FLAG_R**

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11]	**PERID5FLAG**	**ADC Trigger Flag By Period** **Note:** Software can write 1 to clear this bit.
[10]	**CMR5FLAG_F**	**ADC Trigger Flag By Counting Down To CMR** **Note:** Software can write 1 to clear this bit.
[9]	**CNT5FLAG**	**ADC Trigger Flag By Counting To CNR** **Note:** Software can write 1 to clear this bit.
[8]	**CMR5FLAG_R**	**ADC Trigger Flag By Counting Up To CMR** **Note:** Software can write 1 to clear this bit.
[7:4]	**Reserved**	Reserved.
[3]	**PERID4FLAG**	**ADC Trigger Flag By Period** **Note:** Software can write 1 to clear this bit.
[2]	**CMR4FLAG_F**	**ADC Trigger Flag By Counting Down To CMR** **Note:** Software can write 1 to clear this bit.
[1]	**CNT4FLAG**	**ADC Trigger Flag By Counting To CNR** **Note:** Software can write 1 to clear this bit.
[0]	**CMR4FLAG_R**	**ADC Trigger Flag By Counting Up To CMR** **Note:** Software can write 1 to clear this bit.

Phase Change Register (PHCHG)

Register	Offset	R/W	Description	Reset Value
PHCHG	PWM_BA+0x78	R/W	Phase Change Register	0x0000_3F00

31	30	29	28	27	26	25	24
CE0	T0	CMP0SEL		CH31TOFF0	CH21TOFF0	CH11TOFF0	CH01TOFF0
23	22	21	20	19	18	17	16
CE1	T1	CMP1SEL		CH31TOFF1	CH21TOFF1	CH11TOFF1	CH01TOFF1
15	14	13	12	11	10	9	8
ACCNT1	ACCNT0	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0
7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0

Bits	Description	
[31]	**CE0**	**ACMP0 Trigger Function Enable Control** 0 = Disabled. 1 = Enabled. **Note:** This bit will be auto cleared when ACMP0 trigger PWM if ACCNT0 is set.
[30]	**T0**	**Timer0 Trigger PWM Function Enable Control** 0 = Disabled. 1 = Enabled. When this bit is set, timer0 time-out event will update PHCHG with PHCHG_NXT register.
[29:28]	**CMP0SEL[1:0]**	**CMP0SEL** Select the positive input source of ACMP0. 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]	**CH31TOFF0**	**Setting This Bit Will Force PWM3 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.
[26]	**CH21TOFF0**	**Setting This Bit Will Force PWM2 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.

Bits	Description	
[25]	**CH11TOFF0**	**Setting This Bit Will Force PWM1 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.
[24]	**CH01TOFF0**	**Setting This Bit Will Force PWM0 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.
[23]	**CE1**	**ACMP1 Trigger Function Enable Control** 0 = Disabled. 1 = Enabled. **Note: T**his bit will be auto cleared when ACMP1 trigger PWM if ACCNT1 is set.
[22]	**T1**	**Timer1 Trigger PWM Function Enable Control** 0 = Disabled. 1 = Enabled. When this bit is set, timer1 time-out event will update PHCHG with PHCHG_NXT register.
[21:20]	**CMP1SEL[1:0]**	**CMP1SEL** Select the positive input source of ACMP1. 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]	**CH31TOFF1**	**Setting This Bit Will Force PWM3 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.
[18]	**CH21TOFF1**	**Setting This Bit Will Force PWM2 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.
[17]	**CH11TOFF1**	**Setting This Bit Will Force PWM1 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.

Bits	Description	
[16]	**CH01TOFF1**	**Setting This Bit Will Force PWM0 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 = Disabled. 1 = Enabled. **Note:** only for PWM0,PWM1,PWM2,PWM3.
[15]	**ACCNT1**	**Hardware Auto Clear CE1 When ACMP1 Trigger It** 0 = Enabled. 1 = Disabled.
[14]	**ACCNT0**	**Hardware Auto Clear CE0 When ACMP0 Trigger It** 0 = Enabled. 1 = Disabled.
[13]	**PWM5**	**PWM Channel 5 Output Enable Control** 0 = Output D5 specified in bit 5 of PHCHG register. 1 = Output the original channel 5 waveform.
[12]	**PWM4**	**PWM Channel 4 Output Enable Control** 0 = Output D4 specified in bit 4 of PHCHG register. 1 = Output the original channel 4 waveform.
[11]	**PWM3**	**PWM Channel 3 Output Enable Control** 0 = Output D3 specified in bit 3 of PHCHG register. 1 = Output the original channel 3 waveform.
[10]	**PWM2**	**PWM Channel 2 Output Enable Control** 0 = Output D2 specified in bit 2 of PHCHG register. 1 = Output the original channel 2 waveform.
[9]	**PWM1**	**PWM Channel 1 Output Enable Control** 0 = Output D1 specified in bit 1 of PHCHG register. 1 = Output the original channel 1 waveform.
[8]	**PWM0**	**PWM Channel 0 Output Enable Control** 0 = Output D0 specified in bit 0 of PHCHG register. 1 = Output the original channel 0 waveform.
[7]	**D7**	**D7: When MASK7 Is 1, Channel 7's Output Waveform Is D7** 0 = Output low. 1 = Output high.
[6]	**D6**	**D6: When MASK6 Is 1, Channel 6's Output Waveform Is D6** 0 = Output low. 1 = Output high.
[5]	**D5**	**D5: When PWM5 Is Zero, Channel 5's Output Waveform Is D5** 0 = Output low. 1 = Output high.
[4]	**D4**	**D4: When PWM4 Is Zero, Channel 4's Output Waveform Is D4** 0 = Output low. 1 = Output high.

Bits	Description	
[3]	**D3**	**D3: When PWM3 Is Zero, Channel 3's Output Waveform Is D3** 0 = Output low. 1 = Output high.
[2]	**D2**	**D2: When PWM2 Is Zero, Channel 2's Output Waveform Is D2** 0 = Output low. 1 = Output high.
[1]	**D1**	**D1: When PWM1 Is Zero, Channel 1's Output Waveform Is D1** 0 = Output low. 1 = Output high.
[0]	**D0**	**D0: When PWM0 Is Zero, Channel 0's Output Waveform Is D0** 0 = Output low. 1 = Output high.

Next Phase Change Register (PHCHGNXT)

Register	Offset	R/W	Description	Reset Value
PHCHGNXT	PWM_BA+0x7C	R/W	Next Phase Change Register	0x0000_3F00

31	30	29	28	27	26	25	24
CE0	**T0**	**CMP0SEL**		**OFF03**	**OFF02**	**OFF01**	**OFF00**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CE1	**T1**	**CMP1SEL**		**OFF13**	**OFF12**	**OFF11**	**OFF10**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ACCNT1	**ACCNT0**	**PWM5**	**PWM4**	**PWM3**	**PWM2**	**PWM1**	**PWM0**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**

Bits	Description	
[31]	**CE0**	**ACMP0 Trigger Function Enable Control** 0 = Disabled. 1 = Enabled. **Note:** This bit will be auto cleared when ACMP0 trigger PWM if ACCNT0 is set.
[30]	**T0**	**Timer0 Trigger PWM Function Enable Control** 0 = Disabled. 1 = Enabled. When this bit is set, timer0 time-out event will update PHCHG with PHCHG_NXT register.
[29:28]	**CMP0SEL[1:0]**	**CMP0SEL** Select the positive input source of ACMP0. 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]	**CH31TOFF0**	**Setting This Bit Will Force PWM3 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.
[26]	**CH21TOFF0**	**Setting This Bit Will Force PWM2 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.

Bits	Description	
[25]	**CH11TOFF0**	**Setting This Bit Will Force PWM1 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.
[24]	**CH01TOFF0**	**Setting This Bit Will Force PWM0 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.
[23]	**CE1**	**ACMP1 Trigger Function Enable Control** 0 = Disabled. 1 = Enabled. **Note: T**his bit will be auto cleared when ACMP1 trigger PWM if ACCNT1 is set.
[22]	**T1**	**Timer1 Trigger PWM Function Enable Control** 0 = Disabled. 1 = Enabled. When this bit is set, timer1 time-out event will update PHCHG with PHCHG_NXT register.
[21:20]	**CMP1SEL[1:0]**	**CMP1SEL** Select the positive input source of ACMP1. 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]	**CH31TOFF1**	**Setting This Bit Will Force PWM3 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.
[18]	**CH21TOFF1**	**Setting This Bit Will Force PWM2 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.
[17]	**CH11TOFF1**	**Setting This Bit Will Force PWM1 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 = Disabled. 1 = Enabled. **Note:** Only for PWM0, PWM1, PWM2, PWM3.

Bits	Description	
[16]	**CH01TOFF1**	**Setting This Bit Will Force PWM0 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 = Disabled. 1 = Enabled. **Note:** only for PWM0,PWM1,PWM2,PWM3.
[15]	**ACCNT1**	**Hardware Auto Clear CE1 When ACMP1 Trigger It** 0 = Enabled. 1 = Disabled.
[14]	**ACCNT0**	**Hardware Auto Clear CE0 When ACMP0 Trigger It** 0 = Enabled. 1 = Disabled.
[13]	**PWM5**	**PWM Channel 5 Output Enable Control** 0 = Output D5 specified in bit 5 of PHCHG register. 1 = Output the original channel 5 waveform.
[12]	**PWM4**	**PWM Channel 4 Output Enable Control** 0 = Output D4 specified in bit 4 of PHCHG register. 1 = Output the original channel 4 waveform.
[11]	**PWM3**	**PWM Channel 3 Output Enable Control** 0 = Output D3 specified in bit 3 of PHCHG register. 1 = Output the original channel 3 waveform.
[10]	**PWM2**	**PWM Channel 2 Output Enable Control** 0 = Output D2 specified in bit 2 of PHCHG register. 1 = Output the original channel 2 waveform.
[9]	**PWM1**	**PWM Channel 1 Output Enable Control** 0 = Output D1 specified in bit 1 of PHCHG register. 1 = Output the original channel 1 waveform.
[8]	**PWM0**	**PWM Channel 0 Output Enable Control** 0 = Output D0 specified in bit 0 of PHCHG register. 1 = Output the original channel 0 waveform.
[7]	**D7**	**D7: When MASK7 Is 1, Channel 7's Output Waveform Is D7** 0 = Output low. 1 = Output high.
[6]	**D6**	**D6: When MASK6 Is 1, Channel 6's Output Waveform Is D6** 0 = Output low. 1 = Output high.
[5]	**D5**	**D5: When PWM5 Is Zero, Channel 5's Output Waveform Is D5** 0 = Output low. 1 = Output high.
[4]	**D4**	**D4: When PWM4 Is Zero, Channel 4's Output Waveform Is D4** 0 = Output low. 1 = Output high.

Bits	Description	
[3]	**D3**	**D3: When PWM3 Is Zero, Channel 3's Output Waveform Is D3** 0 = Output low. 1 = Output high.
[2]	**D2**	**D2: When PWM2 Is Zero, Channel 2's Output Waveform Is D2** 0 = Output low. 1 = Output high.
[1]	**D1**	**D1: When PWM1 Is Zero, Channel 1's Output Waveform Is D1** 0 = Output low. 1 = Output high.
[0]	**D0**	**D0: When PWM0 Is Zero, Channel 0's Output Waveform Is D0** 0 = Output low. 1 = Output high.

Phase Change Mask Register (PHCHGMASK)

Register	Offset	R/W	Description	Reset Value
PHCHGMASK	PWM_BA+0x80	R/W	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						**CMPMASK**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MASK7	**MASK6**	**Reserved**					

Bits	Description	
[31:10]	**Reserved**	Reserved.
[9]	**CMPMASK[1]**	**MASK For ACMP1** 0 = The input of ACMP is controlled by CMP1CR. 1 = The input of ACMP is controlled by CMP1SEL of PHCHG register. **Note:** Register CMP1CR is describe in Comparator Controller chapter
[8]	**CMPMASK[0]**	**MASK For ACMP0** 0 = The input of ACMP is controlled by CMP0CR. 1 = The input of ACMP is controlled by CMP0SEL of PHCHG register. **Note:** Register CMP0CR is describe in Comparator Controller chapter
[7]	**MASK7**	**MASK For D7** 0 = Original GPIO P0.0. 1 = D7.
[6]	**MASK6**	**MASK For D6** 0 = Original GPIO P0.1. 1 = D6.
[5:0]	**Reserved**	Reserved.

Interrupt Accumulation Control Register (INTACCUCTL)

Register	Offset	R/W	Description	Reset Value
INTACCUCTL	PWM_BA+0x84	R/W	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**
PERIODCNT				**Reserved**			**INTACCUEN0**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:4]	**PERIODCNT[3:0]**	**Interrupt Accumulation Bits** When **INTACCUEN0** is set, PERIODCNT will decrease when every PWMPIF0 flag is set and when PERIODCNT reach to zero, the PWM0 interrupt will occurred and PERIODCNT will reload itself.
[3:1]	**Reserved**	Reserved.
[0]	**INTACCUEN0**	**Interrupt Accumulation Function Enable Control** 0 = Disabled. 1 = Enabled.

6.8 Watchdog Timer (WDT)

6.8.1 Overview

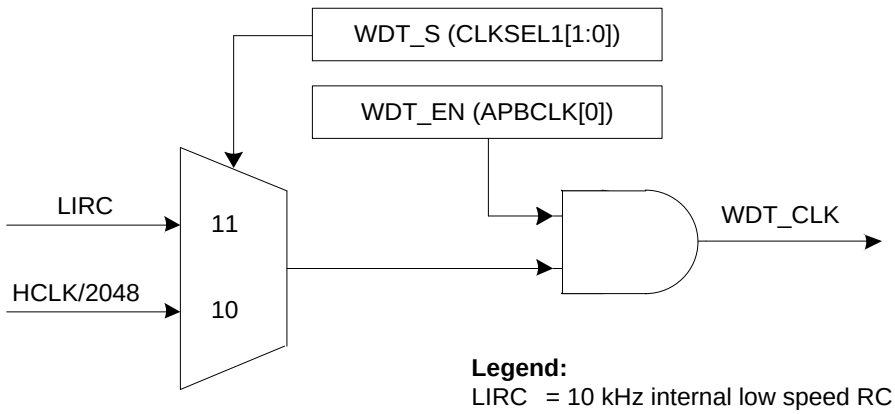
The purpose of Watchdog Timer is to perform a system reset when system runs into an unknown state. This prevents system from hanging for an infinite period of time. Besides, this Watchdog Timer supports the function to wake-up system from Idle/Power-down mode.

6.8.2 Features

- 18-bit free running up counter for Watchdog Timer time-out interval
- Selectable time-out interval (2^4 ~ 2^{18}) WDT_CLK cycle and the time-out interval period is 104 ms ~ 26.3168 s if WDT_CLK = 10 kHz
- System kept in reset state for a period of (1 / WDT_CLK) * 63
- Supports Watchdog Timer time-out wake-up function only if WDT clock source is selected as 10 kHz

6.8.3 Block Diagram

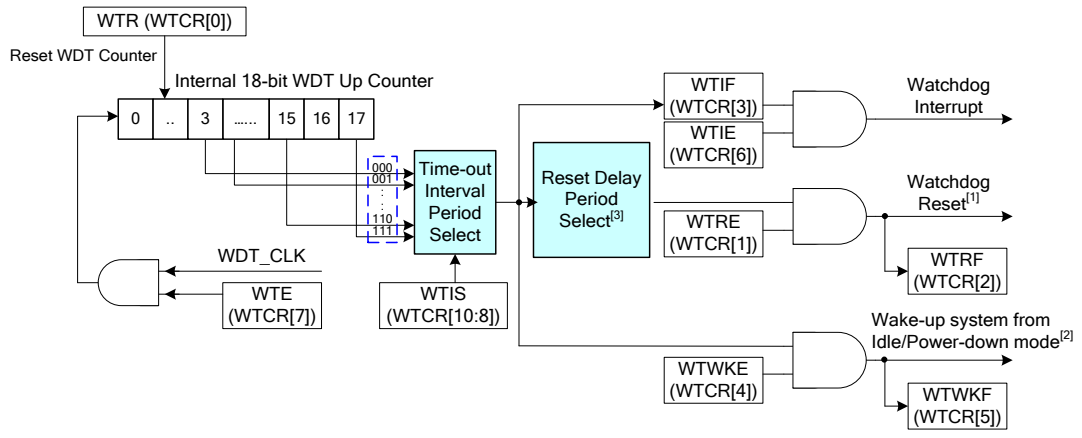
The Watchdog Timer clock control and block diagram are shown as follows.



Legend:
LIRC = 10 kHz internal low speed RC oscillator

Note: Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6.8-1 Watchdog Timer Clock Control



Note1: WDT resets CPU and lasts 63 WDT_CLK.

Note2: Chip can be woken-up by WDT time-out interrupt signal generated only if WDT clock source is selected to 10kHz oscillator.

Note3: The WDT reset delay period is fixed to 1024 WDT_CLK.

Figure 6.8-2 Watchdog Timer Block Diagram

6.8.4 Basic Configuration

The WDT peripheral clock is enabled in WDT_EN(APBCLK[0]) and clock source can be selected in WDT_S(CLKSEL1[1:0]).

6.8.5 Functional Description

The Watchdog Timer (WDT) includes an 18-bit free running up counter with programmable time-out intervals. shows the WDT time-out interval and reset period timing.

6.8.5.1 WDT Time-out Interrupt

Writing WTE(WTCR[7]) bit to 1 will enable the WDT function and the WDT counter to start counting up. There are eight time-out interval periods can be selected by writing WTIS(WTCR[10:8]). When the WDT up counter reaches the WTIS settings, WDT time-out interrupt will occur then WTIF(WTCR[3]) flag will be set to 1 immediately.

6.8.5.2 WDT Reset Delay Period and Reset System

There is a specified T_{RSTD} delay period follows the WTIF flag is setting to 1. User must enabled WTR(WTCR[0]) bit to reset the 18-bit WDT up counter value to avoid generate WDT time-out reset signal before the T_{RSTD} delay period expires. If the WDT up counter value has not been cleared after the specific T_{RSTD} delay period expires, the WDT control will set WTRF(WTCR[2]) flag to 1 if WTRE(WTCR[1]) bit is enabled, then chip enters to reset state immediately. Refer to , the T_{RST} reset period will keep last 63 WDT clocks then chip restart executing program from reset vector (0x0000_0000). The WTRF flag will keep 1 after WDT time-out reset the chip, user can check WTRF flag by software to recognize the system has been reset by WDT time-out reset or not.

6.8.5.3 WDT Wake-up

If WDT clock source is selected to 10 kHz, system can be woken up from Power-down mode while WDT time-out interrupt signal is generated and WTWKE bit enabled. In the meanwhile, the WTWKF flag will set to 1 automatically, user can check WTWKF flag by software to recognize the system has been woken up by WDT time-out interrupt or not.

WTIS	Time-Out Interval Period T_{TIS}	Reset Delay Period T_{RSTD}
000	$2^4 * T_{WDT}$	$1024 * T_{WDT}$
001	$2^6 * T_{WDT}$	$1024 * T_{WDT}$
010	$2^8 * T_{WDT}$	$1024 * T_{WDT}$
011	$2^{10} * T_{WDT}$	$1024 * T_{WDT}$
100	$2^{12} * T_{WDT}$	$1024 * T_{WDT}$
101	$2^{14} * T_{WDT}$	$1024 * T_{WDT}$
110	$2^{16} * T_{WDT}$	$1024 * T_{WDT}$
111	$2^{18} * T_{WDT}$	$1024 * T_{WDT}$

Table 6.8-1 Watchdog Timer Time-out Interval Period Selection

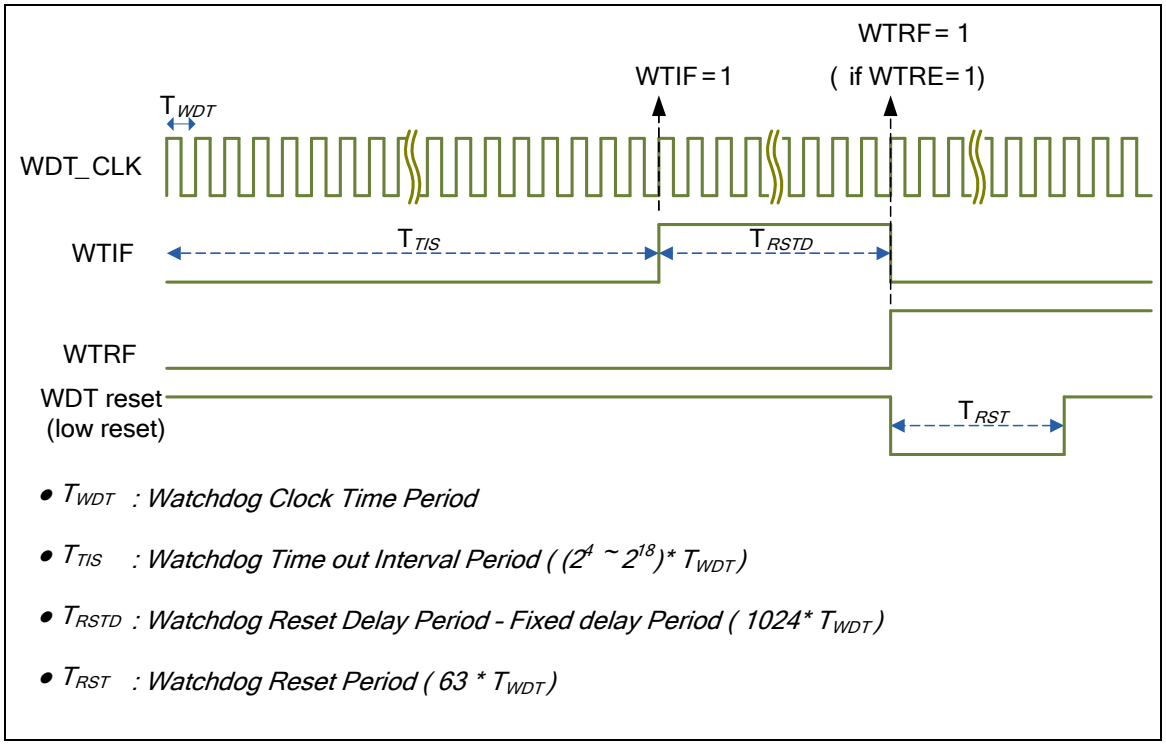


Figure 6.8-3 Watchdog Timer Time-out Interval and Reset Period Timing

6.8.6 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**				
WTCR	WDT_BA+0x00	R/W	Watchdog Timer Control Register	0x0000_0700

6.8.7 Register Description

Watchdog Timer Control Register (WTCR)

Register	Offset	R/W	Description	Reset Value
WTCR	WDT_BA+0x00	R/W	Watchdog Timer Control Register	0x0000_0700

Note: All bits in this register are write-protected. To program it, an open lock sequence is needed, by sequentially writing 0x59, 0x16, and 0x88 to register REGWRPROT at address GCR_BA + 0x100.

31	30	29	28	27	26	25	24
DBGACK_WDT	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**WTIS**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WTE	**WTIE**	**WTWKF**	**WTWKE**	**WTIF**	**WTRF**	**WTRE**	**WTR**

Bits	Description	
[31]	**DBGACK_WDT**	**ICE Debug Mode Acknowledge Disable Control (Write Protect)** 0 = ICE debug mode acknowledgement effects WDT counting. WDT up counter will be kept while CPU is hanging by ICE. 1 = ICE debug mode acknowledgement Disabled. WDT up counter will keep going no matter CPU is hanging by ICE or not.
[30:11]	**Reserved**	Reserved.
[10:8]	**WTIS[2:0]**	**Watchdog Timer Interval Selection** These three bits select the time-out interval for the Watchdog Timer. 000 = $2^4 * T_{WDT}$. 001 = $2^6 * T_{WDT}$. 010 = $2^8 * T_{WDT}$. 011 = $2^{10} * T_{WDT}$. 100 = $2^{12} * T_{WDT}$. 101 = $2^{14} * T_{WDT}$. 110 = $2^{16} * T_{WDT}$. 111 = $2^{18} * T_{WDT}$.
[7]	**WTE**	**Watchdog Timer Enable Control (Write Protect)** 0 = WDT Disabled. (This action will reset the internal up counter value.) 1 = WDT Enabled.
[6]	**WTIE**	**Watchdog Timer Time-out Interrupt Enable Control (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.

Bits	Description	
[5]	**WTWKF**	**Watchdog Timer Time-out Wake-up Flag** 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. **Note:** This bit is cleared by writing 1 to it.
[4]	**WTWKE**	**Watchdog Timer Time-out Wake-up Function Control (Write Protect)** If this bit is set to 1, while WTIF is generated to 1 and WTIE 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. **Note:** Chip can be woken-up by WDT time-out interrupt signal generated only if WDT clock source is selected to 10 kHz oscillator.
[3]	**WTIF**	**Watchdog Timer 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]	**WTRF**	**Watchdog Timer 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]	**WTRE**	**Watchdog Timer Time-out Reset Enable Control (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 (1024 * T_{WDT}) expires. 0 = WDT time-out reset function Disabled. 1 = WDT time-out reset function Enabled.
[0]	**WTR**	**Reset Watchdog Timer Up Counter (Write Protect)** 0 = No effect. 1 = Reset the internal 18-bit WDT up counter value. **Note:** This bit will be automatically cleared by hardware.

6.9 UART Controller (UART)

6.9.1 Overview

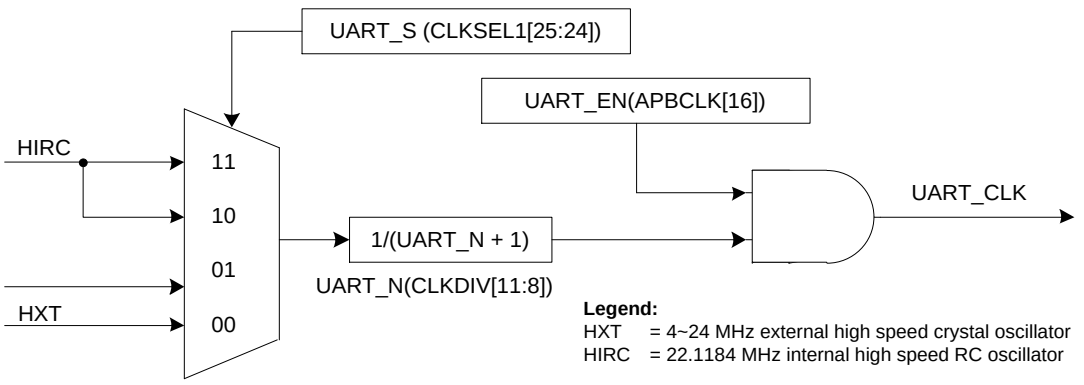
The NuMicro™ Mini51 series provides one channel of Universal Asynchronous Receiver/Transmitters (UART). UART Controller performs Normal Speed UART, and supports flow control function. The UART Controller performs a serial-to-parallel conversion on data received from the peripheral, and a parallel-to-serial conversion on data transmitted from the CPU. The UART controller also supports IrDA SIR Function, and RS-485 function mode.

6.9.2 Features

- Full duplex, asynchronous communications
- Separates 16-byte receive and transmitted FIFO for data payloads
- Supports hardware auto flow control, flow control function (CTS, RTS) and programmable RTS flow control trigger level
- Programmable receiver buffer trigger level
- Supports programmable baud-rate generator for each channel individually
- Supports CTS wake-up function
- Supports 8-bit receiver buffer time-out detection function
- Programmable transmitting data delay time between the last stop and the next start bit by setting DLY(UA_TOR[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- bit character
  - Programmable parity bit, even, odd, no parity or stick parity bit
  - Programmable stop bit, 1, 1.5, or 2 stop bit
- Supports IrDA SIR function mode
  - Supports 3/16-bit duration for normal mode
- Supports RS-485 function mode
  - Supports RS-485 9-bit mode
  - Supports hardware or software enable to program RTS pin to control RS-485 transmission direction directly

6.9.3 Block Diagram

The UART clock control and block diagram are shown as follows.



Note: Before clock switching, both the pre-selected and newly selected clock sources must be turned on and stable.

Figure 6.9-1 UART Controller Clock Control

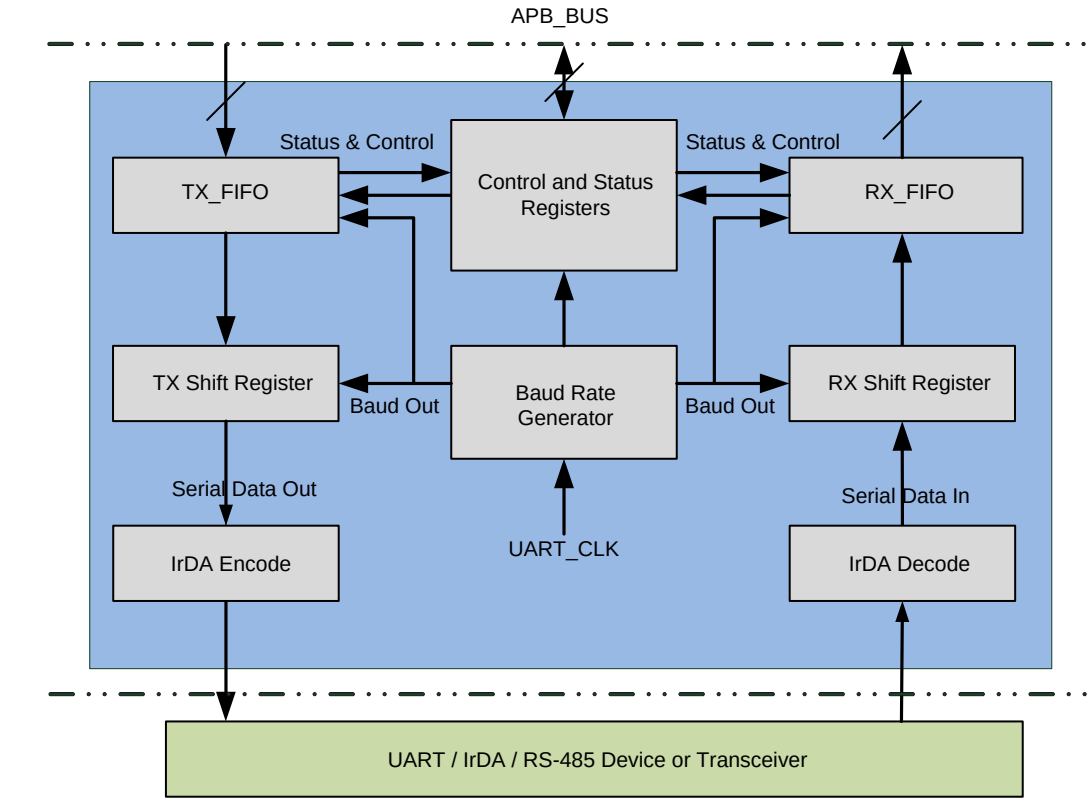


Figure 6.9-2 UART Controller Block Diagram

Each block is described in detail as follows:

TX_FIFO

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

RX_FIFO

The receiver is buffered with a 16-byte FIFO (plus parity error flag, framing error flag and break interrupt flag per byte) to reduce the number of interrupts presented to the CPU.

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

Dividing the external clock by the divisor to get the desired baud rate clock. Refer to baud rate equation.

IrDA Encode

This block is IrDA encode control block.

IrDA Decode

This block is IrDA decode control block.

Control and Status Register

This field is register set that includes the FIFO control registers (UA_FCR), FIFO status registers (UA_FSR), and line control register (UA_LCR) for transmitter and receiver. The time-out control register (UA_TOR) identifies the condition of time-out interrupt. This register set also includes the interrupt enable register (UA_IER) and interrupt status register (UA_ISR) to enable or disable the responding interrupt and to identify the occurrence of the responding interrupt. There are six types of interrupts, including transmitter FIFO empty interrupt (THRE_INT), receiver threshold level reaching interrupt (RDA_INT), line status interrupt (parity error or framing error or break interrupt) (RLS_INT), time-out interrupt (TOUT_INT), MODEM/Wake-up status interrupt (MODEM_INT), and Buffer error interrupt (BUF_ERR_INT).

In addition, the block diagram of auto-flow control is demonstrated in the following diagram.

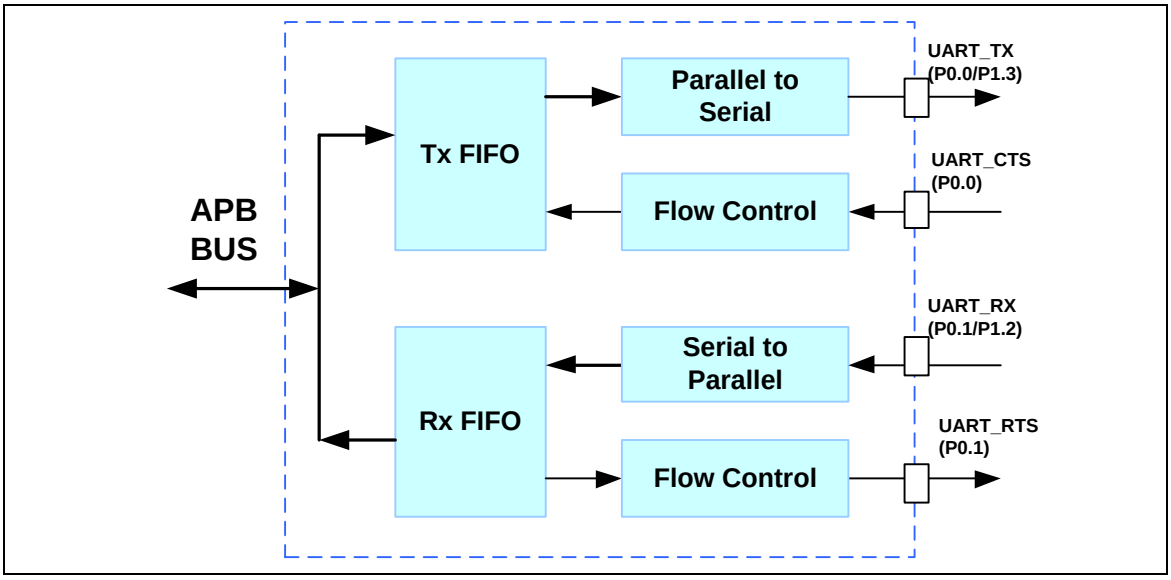


Figure 6.9-3 UART Auto Flow Control Block Diagram

6.9.4 Basic Configuration

The UART Controller function pins are configured in P0_MFP, P1_MFP registers for UART.

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

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

The UART Controller clock prescaler is determined by UART_N(CLKDIV[11:8]).

6.9.5 Functional Description

The UART Controller supports three function modes including UART, IrDA, and RS-485 mode. User can select a function by setting the UA_FUN_SEL register.

6.9.5.1 UART Controller Baud Rate Generator

The UART Controller includes a programmable baud rate generator capable of dividing clock input by dividers 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 (UA_BAUD). The following tables list 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.

Mode	DIV_X_EN	DIV_X_ONE	Divider X	BRD	M	Baud Rate Equation
Mode 0	0	0	Don't care	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 6.9-4 UART 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	0x2B00_0000	A=22	0x3000_0016
460800	A=1	0x0000_0001	A=1, B=15 A=2, B=11	0x2F00_0001 0x2B00_0002	A=46	0x3000_002E
230400	A=4	0x0000_0004	A=4, B=15 A=6, B=11	0x2F00_0004 0x2B00_0006	A=94	0x3000_005E
115200	A=10	0x0000_000A	A=10, B=15 A=14, B=11	0x2F00_000A 0x2B00_000E	A=190	0x3000_00BE
57600	A=22	0x0000_0016	A=22, B=15 A=30, B=11	0x2F00_0016 0x2B00_001E	A=382	0x3000_017E
38400	A=34	0x0000_0022	A=62, B=8 A=46, B=11 A=34, B=15	0x2800_003E 0x2B00_002E 0x2F00_0022	A=574	0x3000_023E
19200	A=70	0x0000_0046	A=126, B=8 A=94, B=11 A=70, B=15	0x2800_007E 0x2B00_005E 0x2F00_0046	A=1150	0x3000_047E
9600	A=142	0x0000_008E	A=254, B=8 A=190, B=11 A=142, B=15	0x2800_00FE 0x2B00_00BE 0x2F00_008E	A=2302	0x3000_08FE
4800	A=286	0x0000_011E	A=510, B=8 A=382, B=11 A=286, B=15	0x2800_01FE 0x2B00_017E 0x2F00_011E	A=4606	0x3000_11FE

Figure 6.9-5 UART Controller Baud Rate Parameter And Register Setting Table

6.9.5.2 UART Controller FIFO Control and Status

The UART Controller is built-in with a 16-bytes transmitter FIFO (TX_FIFO) and a 16-bytes receiver FIFO (RX_FIFO) that reduces the number of interrupts presented to the CPU. The CPU can read the status of the UART at any time during operation. The reported status information includes the 6 types of interrupts and condition of the transfer operations being performed by the UART, 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, IrDA, and RS-485 function mode.

6.9.5.3 UART Controller Wake-up 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 MODEN_INT interrupt to use the wake-up function.

6.9.5.4 UART Controller Interrupt and Status

Each UART Controller supports 6 types of interrupts including:

- Receiver threshold level reached interrupt (RDA_INT)
- Transmitter FIFO empty interrupt (THRE_INT)
- Line status interrupt (parity error, frame error or break interrupt) (RLS_INT)
- MODEM/Wake-up status interrupt (MODEM_INT)
- Receiver buffer time-out interrupt (TOUT_INT)
- Buffer error interrupt (BUF_ERR_INT)

The following tables describe 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	RDA_INT	RDA_IEN	**RDA_IF**	Read UA_RBR
Transmit Holding Register Empty Interrupt	THRE_INT	THRE_IEN	**THRE_IF**	Write UA_THR
Receive Line Status Interrupt	RLS_INT	RLS_IEN	**RLS_IF** = (BIF or FEF or PEF)	**Writing '1' to** BIF/FEF/ PEF
			RLS_IF = (BIF or FEF or PEF or RS485_ADD_DETF)	**Writing '1' to** BIF/FEF/PEF/RS485_ADD_DETF
Modem Status Interrupt	MODEM_INT	MODEM_IEN	**MODEM_IF** = DCTSF	**Write '1' to** DCTSF
RX Time-out Interrupt	TOUT_INT	RTO_IEN	**TOUT_IF**	Read UA_RBR

Buffer Error Interrupt	BUF_ERR_INT	BUF_ERR_IEN	**BUF_ERR_IF** = (TX_OVER_IF or RX_OVER_IF)	**Writing '1' to** TX_OVER_IF / RX_OVER_IF

Figure 6.9-6 UART Controller Interrupt Source and Flag

6.9.5.5 UART Function Mode

The UART Controller provides UART function (user must set UA_FUN_SEL [1:0] to '00' to enable UART function mode). The UART baud rate is up to 1 Mbps.

The UART provides full-duplex and asynchronous communications. The transmitter and receiver contain 16-bytes FIFO for payloads. User can program receiver buffer trigger level and receiver buffer time-out detection for receiver. The transmitting data delay time between the last stop and the next start bit can be programed by setting DLY (UA_TOR [15:8]) register. The UART supports hardware auto-flow control and flow control function (CTS, RTS), programmable RTS flow control trigger level and fully programmable serial-interface characteristics.

UART Line Control Function

The UART Controller supports fully programmable serial-interface characteristics by setting the UA_LCR register. Software can use the UA_LCR register to program the word length, stop bit and parity bit. The following tables list the UART word and stop bit length settings and the UART parity bit settings.

NSB (UA_LCR[2])	WLS (UA_LCR[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 6.9-7 UART Line Control of Word and Stop Length Setting

Parity Type	SPE (UA_LCR[5])	EPE (UA_LCR[4])	PBE (UA_LCR[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 6.9-8 UART Line Control of Parity Bit Setting

Note: User cannot change line controller setting when TE_FLAG(UA_FSR[28]) is not empty

UART Auto-Flow Control Function

The UART 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 the number of bytes in the RX FIFO equals the value of RTS_TRI_LEV (UA_FCR [19:16]), the RTS is de-asserted. The UART sends data out when UART detects CTS is asserted from external device. If the valid asserted CTS is not detected, the UART will not send data out.

The following diagram demonstrates the CTS auto flow control of UART function mode. User must set AUTO_CTS_EN (UA_IER [13]) to enable CTS auto flow control function. The LEV_CTS (UA_MCR [8]) can set CTS pin input active state. The DCTSF (UA_MSR [0]) is set when any state change of CTS pin input has occurred, and then TX data will be automatically transmitted from TX FIFO.

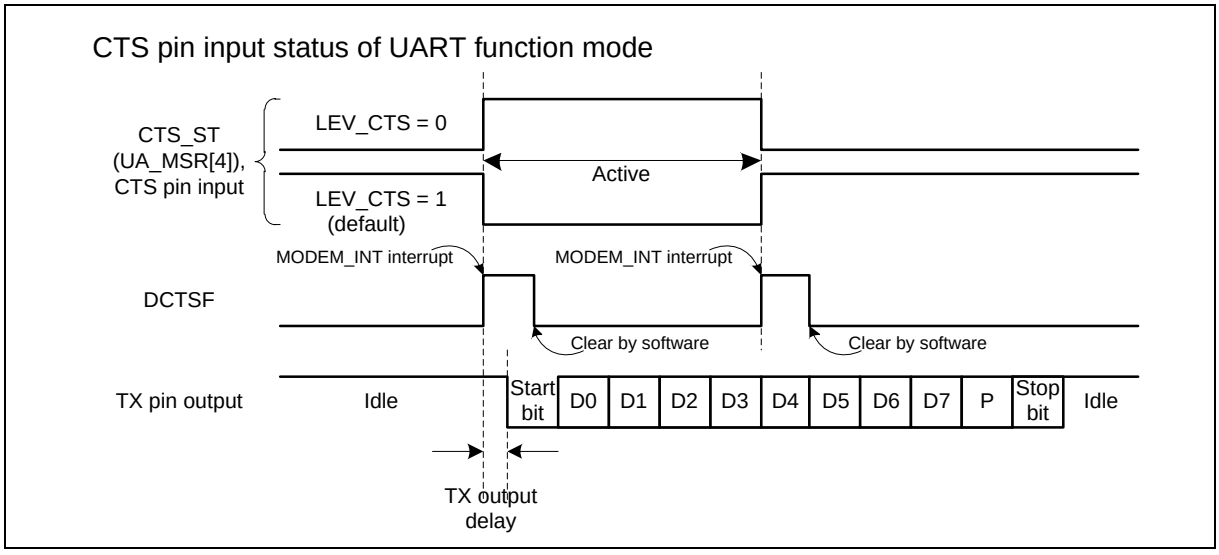


Figure 6.9-9 CTS Auto Flow Control Enabled

As shown in the following figure, in UART RTS Auto Flow control mode (AUTO_RTS_EN(UA_IER[12])=1), the nRTS internal signal is controlled by UART FIFO controller with RTS_RTI_LEV(UA_FCR[19:16]) trigger level.

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

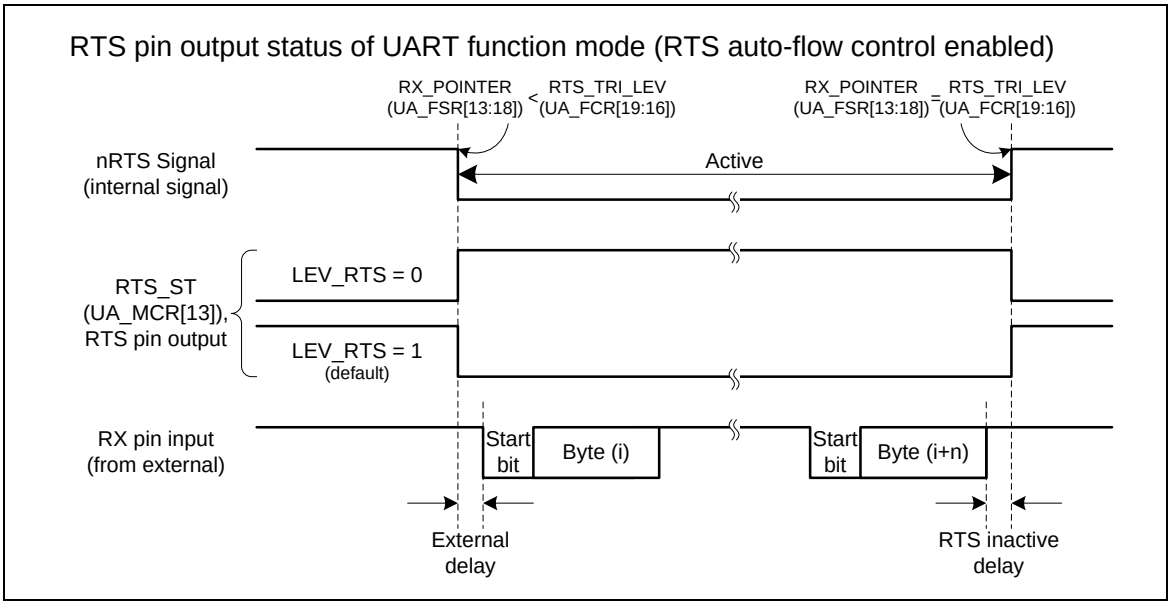


Figure 6.9-10 RTS Auto Flow Control Enabled

As shown in the following figure, in software mode (AUTO_RTS_EN(UA_IER[12])=0) the RTS flow is directly controlled by software programming of RTS(UA_MCR[1]) control bit.

Setting LEV_RTS(UA_MCR[9]) can control the RTS pin output is inverse or non-inverse from RTS(UA_MCR[1]) control bit. User can read the RTS_ST(UA_MCR[13]) bit to get real RTS pin output voltage logic status.

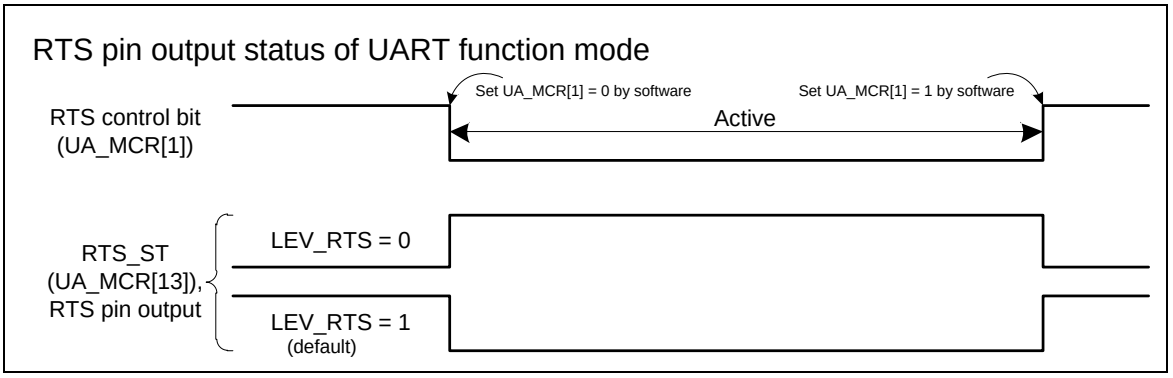


Figure 6.9-11 RTS Flow with Software Control

6.9.5.6 IrDA Function

The UART controllers also provides Serial IrDA (SIR, Serial Infrared) function (user must set IrDA_EN (UA_FUN_SEL[1:0]) 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 DIV_X_EN (UA_BAUD [29]) register must be disabled.

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

The following diagram demonstrates the IrDA control block diagram.

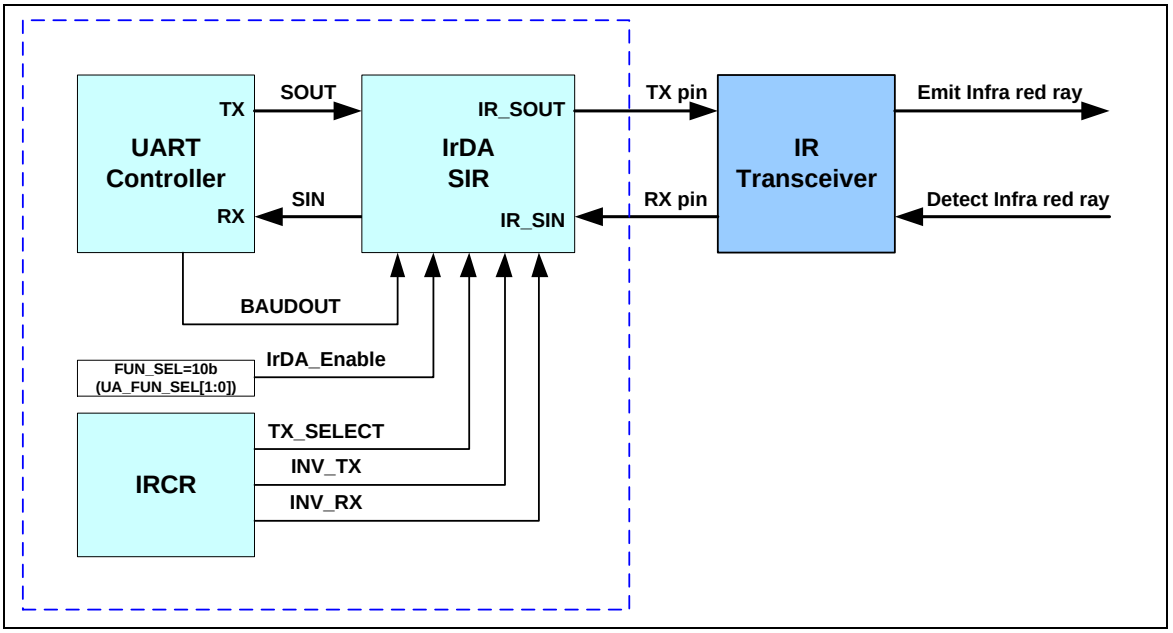


Figure 6.9-12 IrDA Control Block Diagram

IrDA SIR Transmit Encoder

The IrDA SIR Transmit Encoder modulate Non-Return-to Zero (NRZ) transmit bit stream output from UART. 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 UART received data input. The decoder input is normally high in the idle state. (Because of this, IRCR (INV_RX [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. The following diagram is IrDA encoder/decoder waveform:

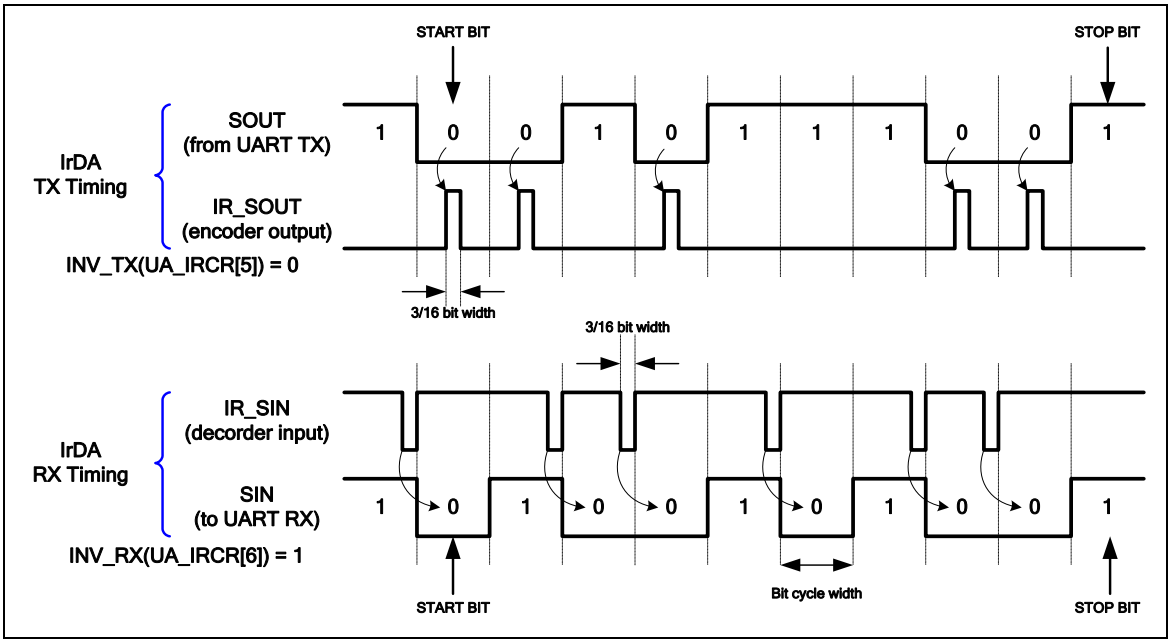


Figure 6.9-13 IrDA TX/RX Timing Diagram

6.9.5.7 RS-485 Function Mode

Another alternate function of UART Controller is RS-485 function (user must set UA_FUN_SEL [1:0] to '11b' to enable RS-485 function), and direction control provided by RTS pin or can program GPIO (P0.1 for RTSn) 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 UA_LCR 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 UA_ALT_CSR register, and drive the transfer delay time between the last stop bit leaving the TX FIFO and the de-assertion of by setting DLY (UA_TOR [15:8]) register.

RS-485 Normal Multi-drop Operation Mode (NMM)

In RS-485 Normal Multi-drop Operation Mode, in first, software must decide the data which before the address byte be detected will be stored in RX FIFO or not. If software wants to ignore any data before address byte detected, the flow is set RX_DIS (UA_FCR [8]) then enable RS485_NMM (UA_ALT_CSR [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 RX_DIS (UA_FCR [8]) then enable RS485_NMM (UA_ALT_CSR [8]) and the receiver will received any data.

If an address byte is detected (bit 9 = 1), it will generate an interrupt(UA_IER[0]: RDA_IEN) to CPU and RX_DIS (UA_FCR [8]) can decide whether accepting the following data bytes are stored

in the RX FIFO. If software disables receiver by setting RX_DIS (UA_FCR [8]) register, when a next address byte is detected, the controller will clear the RX_DIS (UA_FCR [8]) bit and the address byte data will be stored in the RX FIFO.

RS-485 Auto Address Detection Operation Mode (AAD)

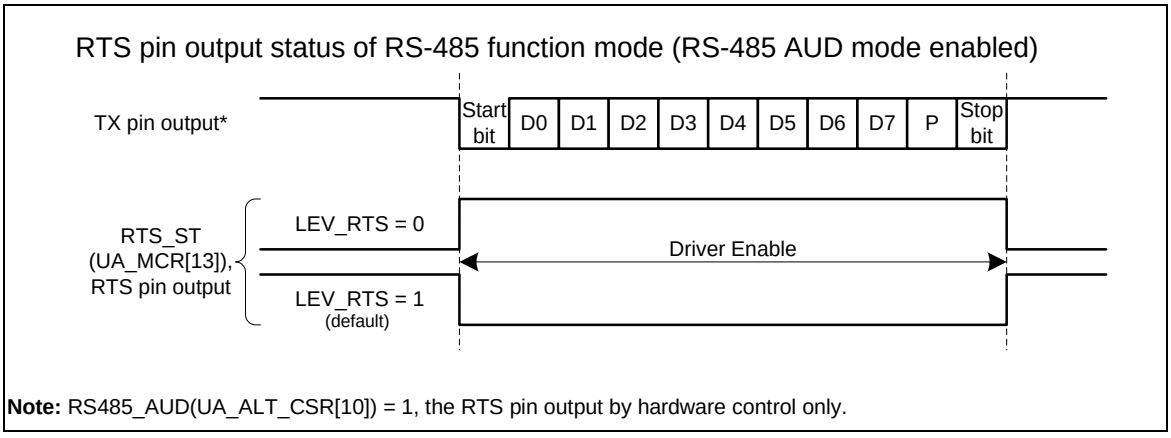
In RS-485 Auto Address Detection Operation Mode, the receiver will ignore any data until an address byte is detected (bit 9 = 1) and the address byte data matches the ADDR_MATCH (UA_ALT_CSR[31:24]) value. The address byte data will be stored in the RX FIFO. The all received byte data will be accepted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

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

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



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

Figure 6.9-14 RS-485 RTS Driving Level in Auto Direction Mode

The following diagram demonstrates the RS-485 RTS driving level in software control (RS485_AUD(UA_ALT_CSR[10])=0). The RTS driving level is controlled by programing the RTS(UA_MCR[1]) control bit.

Setting LEV_RTS(UA_MCR[9]) can control the RTS pin output is inverse or non-inverse from RTS(UA_MCR[1]) control bit. User can read the RTS_ST(UA_MCR[13]) bit to get real RTS pin output voltage logic status.

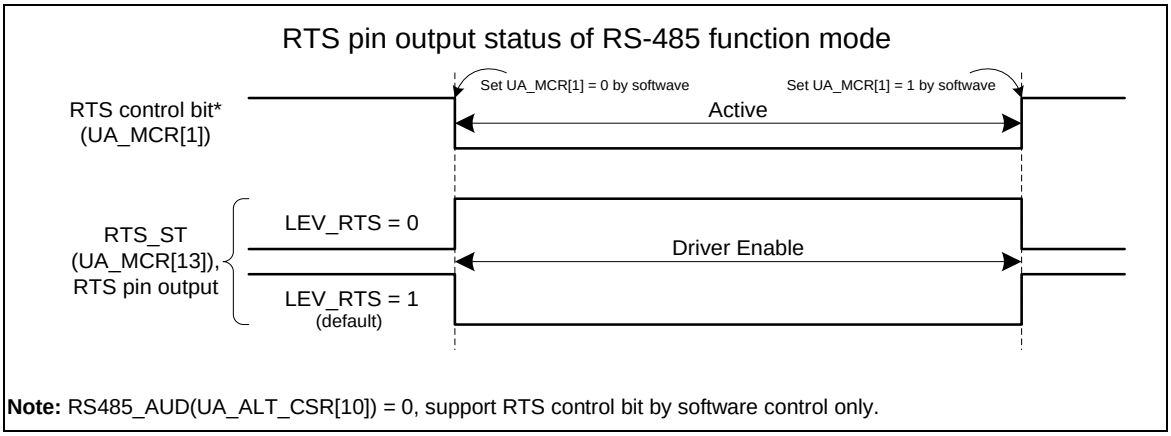


Figure 6.9-15 RS-485 RTS Driving Level with Software Control

Program Sequence Example:

1. Program FUN_SEL in UA_FUN_SEL to select RS-485 function.
2. Program the RX_DIS bit in UA_FCR register to determine enable or disable RS-485 receiver.
3. Program the RS-485_NMM or RS-485_AAD mode.
4. If the RS-485_AAD mode is selected, the ADDR_MATCH is programmed for auto address match value.
5. Determine auto direction control by programming RS-485_AUD.

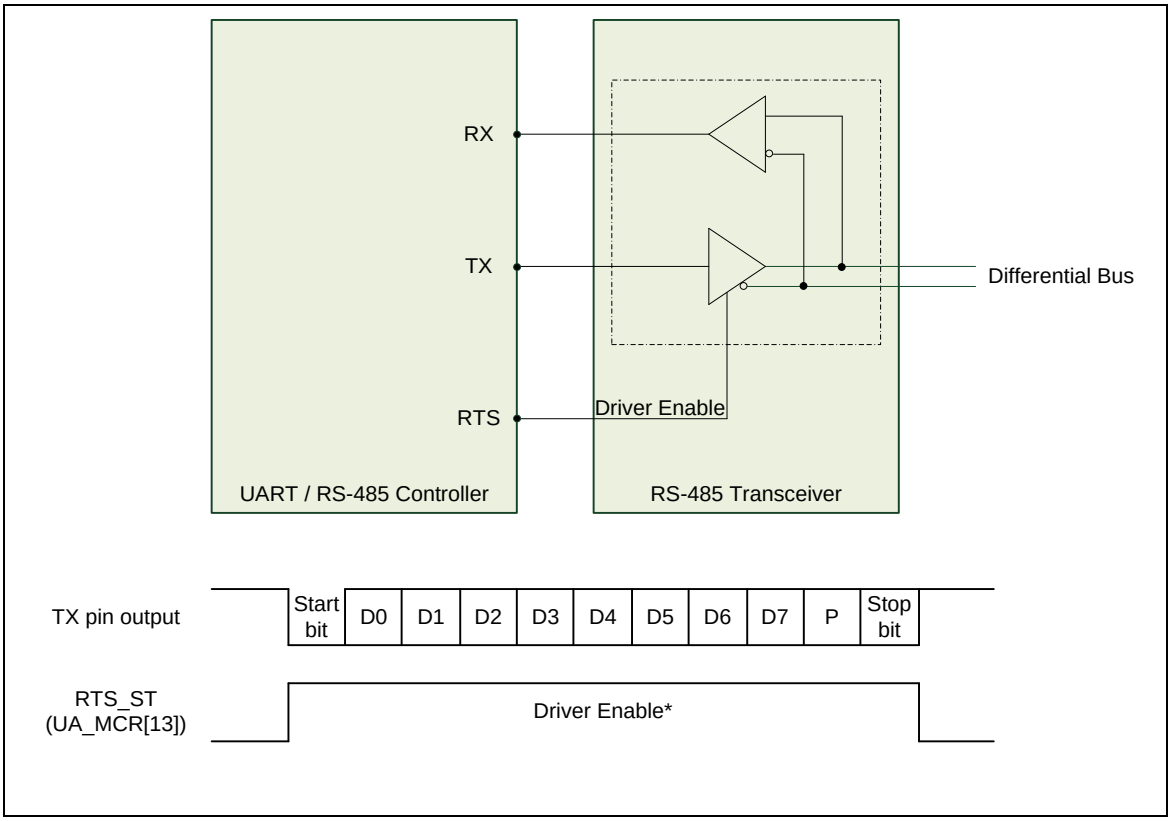


Figure 6.9-16 Structure of RS-485 Frame

6.9.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
UART Base Address: **UART_BA = 0x4005_0000**				
UA_RBR	UART_BA+0x00	R	UART Receive Buffer Register	Undefined
UA_THR	UART_BA+0x00	W	UART Transmit Holding Register	Undefined
UA_IER	UART_BA+0x04	R/W	UART Interrupt Enable Control Register	0x0000_0000
UA_FCR	UART_BA+0x08	R/W	UART FIFO Control Register	0x0000_0101
UA_LCR	UART_BA+0x0C	R/W	UART Line Control Register	0x0000_0000
UA_MCR	UART_BA+0x10	R/W	UART Modem Control Register	0x0000_0200
UA_MSR	UART_BA+0x14	R/W	UART Modem Status Register	0x0000_0110
UA_FSR	UART_BA+0x18	R/W	UART FIFO Status Register	0x1040_4000
UA_ISR	UART_BA+0x1C	R/W	UART Interrupt Status Register	0x0000_0002
UA_TOR	UART_BA+0x20	R/W	UART Time-out Register	0x0000_0000
UA_BAUD	UART_BA+0x24	R/W	UART Baud Rate Divisor Register	0x0F00_0000
UA_IRCR	UART_BA+0x28	R/W	UART IrDA Control Register	0x0000_0040
UA_ALT_CSR	UART_BA+0x2C	R/W	UART Alternate Control/Status Register	0x0000_0000
UA_FUN_SEL	UART_BA+0x30	R/W	UART Function Select Register	0x0000_0000

6.9.7 Register Description

Receive Buffer Register (UA_RBR)

Register	Offset	R/W	Description	Reset Value
UA_RBR	UART_BA+0x00	R	UART Receive 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**
RBR							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**RBR[7:0]**	**Receive Buffer Bits (Read Only)** By reading this register, the UART Controller will return an 8-bit data received from RX pin (LSB first).

Transmit Holding Register (UA_THR)

Register	Offset	R/W	Description	Reset Value
UA_THR	UART_BA+0x00	W	UART Transmit Holding 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**
THR							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**THR**	**Transmit Holding Bits** By writing to this register, the UART sends out an 8-bit data through the TX pin (LSB first).

Interrupt Enable Control Register (UA_IER)

Register	Offset	R/W	Description	Reset Value
UA_IER	UART_BA+0x04	R/W	UART Interrupt Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		AUTO_CTS_EN	AUTO_RTS_EN	TIME_OUT_EN	Reserved		
7	6	5	4	3	2	1	0
Reserved	WAKE_EN	BUF_ERR_IEN	RTO_IEN	MODEM_IEN	RLS_IEN	THRE_IEN	RDA_IEN

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**AUTO_CTS_EN**	**CTS Auto Flow Control Enable Control** 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]	**AUTO_RTS_EN**	**RTS Auto Flow Control Enable Control** 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 RTS_TRI_LEV (UA_FCR [19:16]), the UART will de-assert RTS signal.
[11]	**TIME_OUT_EN**	**Time-out Counter Enable Control** 0 = Time-out counter Disabled. 1 = Time-out counter Enabled.
[10:7]	**Reserved**	Reserved.
[6]	**WAKE_EN**	**Wake-up CPU Function Enable Control** 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.
[5]	**BUF_ERR_IEN**	**Buffer Error Interrupt Enable Control** 0 = INT_BUF_ERR Masked Disabled. 1 = INT_BUF_ERR Enabled.
[4]	**RTO_IEN**	**RX Time-out Interrupt Enable Control** 0 = TOUT_INT Masked off.

Bits	Description	
		1 = TOUT_INT Enabled.
[3]	**MODEM_IEN**	**Modem Status Interrupt Enable Control** 0 = MODEM_INT Masked off. 1 = MODEM_INT Enabled.
[2]	**RLS_IEN**	**Receive Line Status Interrupt Enable Control** 0 = RLS_INT Masked off. 1 = RLS_INT Enabled.
[1]	**THRE_IEN**	**Transmit Holding Register Empty Interrupt Enable Control** 0 = THRE_INT Masked off. 1 = THRE_INT Enabled.
[0]	**RDA_IEN**	**Receive Data Available Interrupt Enable Control** 0 = RDA_INT Masked off. 1 = RDA_INT Enabled.

FIFO Control Register (UA_FCR)

Register	Offset	R/W	Description	Reset Value
UA_FCR	UART_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				**RTS_TRI_LEV**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**RX_DIS**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RFITL				**Reserved**	**TFR**	**RFR**	**Reserved**

Bits	Description	
[31:20]	**Reserved**	Reserved.
[19:16]	**RTS_TRI_LEV[3:0]**	**RTS Trigger Level (For Auto-flow Control Use)** 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]	**RX_DIS**	**Receiver Disable Control** The receiver is disabled or not (setting 1 to disable the receiver). 0 = Receiver Enabled. 1 = Receiver Disabled. **Note1:** This field is only used for RS-485 Normal Multi-drop mode. It should be programmed firstly to avoid receiving unknown data before RS-485_NMM (UA_ALT_CSR [8]) is programmed. **Note2:** After RS-485 receives an address byte in RS-485 Normal Multi-drop mode, this bit (RX_DIS) will be cleared to “0” by hardware.
[7:4]	**RFITL[3:0]**	**RX FIFO Interrupt (RDA_INT) Trigger Level** When the number of bytes in the receive FIFO equals the RFITL then the RDA_IF will be set (if RDA_IEN in UA_IER 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.

Bits	Description	
[2]	**TFR**	**TX Field Software Reset** When TX_RST is set, all the byte in the transmit FIFO and TX internal state machine are cleared. 0 = No effect. 1 = The TX internal state machine and pointers reset. **Note:** This bit will auto clear needs at least 3 UART Controller peripheral clock cycles.
[1]	**RFR**	**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 = The 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 (UA_LCR)

Register	Offset	R/W	Description	Reset Value
UA_LCR	UART_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 Control** 0 = Stick parity Disabled. 1 = If PBE (UA_LCR[3]) and EBE (UA_LCR[4]) are logic 1, the parity bit is transmitted and checked as logic 0. If PBE (UA_LCR[3]) is 1 and EBE (UA_LCR[4]) is 0 then the parity bit is transmitted and checked as 1.
[4]	**EPE**	**Even Parity Enable Control** 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 (UA_LCR[3]) is set.
[3]	**PBE**	**Parity Bit Enable Control** 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[1:0]**	**Word Length Selection** 00 = Word length is 5-bit. 01 = Word length is 6-bit. 10 = Word length is 7-bit.

Bits	Description	
		11 = Word length is 8-bit.

MODEM Control Register (UA_MCR)

Register	Offset	R/W	Description	Reset Value
UA_MCR	UART_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		RTS_ST	Reserved			LEV_RTS	Reserved
7	6	5	4	3	2	1	0
Reserved						RTS	Reserved

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**RTS_ST**	**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]	**LEV_RTS**	**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. **Note1:** Refer to and UART function mode. **Note2:** Refer to and for RS-485 function mode.
[8:2]	**Reserved**	Reserved.
[1]	**RTSn**	**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 LEV_RTS 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 (AUTO_RTS_EN) is enabled in UART function mode. **Note2:** This RTS signal control bit is not effective when RS-485 auto direction mode (RS485_AUD) is enabled in RS-485 function mode.
[0]	**Reserved**	Reserved.

Modem Status Register (UA_MSR)

Register	Offset	R/W	Description	Reset Value
UA_MSR	UART_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							**LEV_CTS**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**CTS_ST**	**Reserved**			**DCTSF**

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**LEV_CTS**	**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. **Note:** Refer to
[7:5]	**Reserved**	Reserved.
[4]	**CTS_ST**	**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]	**DCTSF**	**Detect CTS State Change Flag** This bit is set whenever CTS input has change state, and it will generate Modem interrupt to CPU when MODEM_IEN (UA_IER [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 (UA_FSR)

Register	Offset	R/W	Description	Reset Value
UA_FSR	UART_BA+0x18	R/W	UART FIFO Status Register	0x1040_4000

31	30	29	28	27	26	25	24
Reserved			TE_FLAG	Reserved			TX_OVER_IF
23	22	21	20	19	18	17	16
TX_FULL	TX_EMPTY	TX_POINTER					
15	14	13	12	11	10	9	8
RX_FULL	RX_EMPTY	RX_POINTER					
7	6	5	4	3	2	1	0
Reserved	BIF	FEF	PEF	RS-485_ ADD_DETF	Reserved		RX_OVER_IF

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**TE_FLAG**	**Transmitter Empty Flag (Read Only)** This bit is set by hardware when TX FIFO (UA_THR) is empty and the STOP bit of the last byte has been transmitted. 0 = TX FIFO is not empty. 1 = TX FIFO is empty and the STOP bit of the last byte has been transmitted. **Note:** This bit is cleared automatically when TX FIFO is not empty or the last byte transmission has not completed.
[27:25]	**Reserved**	Reserved.
[24]	**TX_OVER_IF**	**TX Overflow Error Interrupt Flag** If TX FIFO (UA_THR) is full, an additional write to UA_THR will cause this bit to logic 1. 0 = TX FIFO is not overflow. 1 = TX FIFO is overflow. **Note:** This bit is cleared by writing 1 to it.
[23]	**TX_FULL**	**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]	**TX_EMPTY**	**Transmitter FIFO Empty (Read Only)** This bit indicates TX FIFO is 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]	**TX_POINTER [5:0]**	**TX FIFO Pointer (Read Only)** This field indicates the TX FIFO Buffer Pointer. When CPU writes one byte into UA_THR, TX_POINTER increases one. When one byte of TX FIFO is transferred to Transmitter Shift Register, TX_POINTER decreases one. The Maximum value shown in TX_POINTER is 15. When the using level of TX FIFO Buffer equal to 16, the TX_FULL bit is set to 1 and TX_POINTER will show 0. As one byte of TX FIFO is transferred to Transmitter Shift Register, the TX_FULL bit is cleared to 0 and TX_POINTER will show 15.
[15]	**RX_FULL**	**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]	**RX_EMPTY**	**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]	**RX_POINTER [5:0]**	**RX FIFO Pointer (Read Only)** This field indicates the RX FIFO Buffer Pointer. When UART receives one byte from external device, RX_POINTER increases one. When one byte of RX FIFO is read by CPU, RX_POINTER decreases one. The Maximum value shown in RX_POINTER is 15. When the using level of RX FIFO Buffer equal to 16, the RX_FULL bit is set to 1 and RX_POINTER will show 0. As one byte of RX FIFO is read by CPU, the RX_FULL bit is cleared to 0 and RX_POINTER 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 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]	**RS-485_**	**RS-485 Address Byte Detection Flag**

Bits	Description	
	ADD_DETF	This bit is set to 1 while RS485_ADD_EN (UA_ALT_CSR[15]) is set to 1 to enable Address 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]	**RX_OVER_IF**	**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 (UA_RBR) size 16 bytes, this bit will be set. 0 = RX FIFO is not overflow. 1 = RX FIFO is overflow. **Note:** This bit is cleared by writing 1 to it.

Interrupt Status Control Register (UA_ISR)

Register	Offset	R/W	Description	Reset Value
UA_ISR	UART_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							
15	14	13	12	11	10	9	8
Reserved		BUF_ERR_INT	TOUT_INT	MODEM_INT	RLS_INT	THRE_INT	RDA_INT
7	6	5	4	3	2	1	0
Reserved		BUF_ERR_IF	TOUT_IF	MODEM_IF	RLS_IF	THRE_IF	RDA_IF

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**BUF_ERR_INT**	**Buffer Error Interrupt Indicator (Read Only)** This bit is set if BUF_ERR_IEN and BUF_ERR_IF are both set to 1. 0 = No buffer error interrupt is generated. 1 = buffer error interrupt is generated.
[12]	**TOUT_INT**	**Time-out Interrupt Indicator (Read Only)** This bit is set if RTO_IEN and TOUT_IF are both set to 1. 0 = No Time-out interrupt is generated. 1 = Time-out interrupt is generated.
[11]	**MODEM_INT**	**MODEM Status Interrupt Indicator (Read Only)** This bit is set if MODEM_IEN and MODEM_IF are both set to 1. 0 = No Modem interrupt is generated. 1 = Modem interrupt is generated.
[10]	**RLS_INT**	**Receive Line Status Interrupt (Read Only)** This bit is set if RLS_IEN and RLS_IF are both set to 1. 0 = No RLS interrupt is generated. 1 = RLS interrupt is generated.
[9]	**THRE_INT**	**Transmit Holding Register Empty Interrupt Indicator (Read Only)** This bit is set if THRE_IEN and THRE_IF are both set to 1. 0 = No THRE interrupt is generated. 1 = THRE interrupt is generated.
[8]	**RDA_INT**	**Receive Data Available Interrupt Indicator (Read Only)** This bit is set if RDA_IEN and RDA_IF are both set to 1. This bit is set if RDA_IEN and RDA_IF are both set to 1. 0 = No RDA interrupt is generated.

Bits	Description	
		1 = RDA interrupt is generated.
[7:6]	**Reserved**	Reserved.
[5]	**BUF_ERR_IF**	**Buffer Error Interrupt Flag (Read Only)** This bit is set when the TX/RX FIFO overflow flag (TX_OVER_IF or RX_OVER_IF) is set. When BUF_ERR_IF is set, the transfer is not correct. If BUF_ERR_IEN (UA_IER [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 TX_OVER_IF and RX_OVER_IF are cleared.
[4]	**TOUT_IF**	**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. If RTO_IEN (UA_IER [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 UA_RBR (RX is in active) to clear it.
[3]	**MODEM_IF**	**MODEM Interrupt Flag (Read Only)** This bit is set when the CTS pin has state change (DCTSF = 1). If UA_IER [MODEM_IEN] 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 DCTSF is cleared by a write 1 on DCTSF.
[2]	**RLS_IF**	**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 RLS_IEN (UA_IER [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 detect and received address byte character (bit 9 = 1) bit". At the same time, the bit of RS485_ADD_DETF (UA_FSR[3]) is also set. **Note2:** This bit is read only and reset to 0 when all bits of BIF, FEF, PEF and RS485_ADD_DETF are cleared.
[1]	**THRE_IF**	**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 THRE_IEN (UA_IER [1]) is enabled, the THRE interrupt will be generated. 0 = No THRE interrupt flag is generated. 1 = THRE interrupt flag is generated. **Note:** This bit is read only and it will be cleared when writing data into THR (TX FIFO not empty).
[0]	**RDA_IF**	**Receive Data Available Interrupt Flag (Read Only)** When the number of bytes in the RX FIFO equals the RFITL then the RDA_IF will be set. If RDA_IEN (UA_IER [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).

Time-out Register (UA_TOR)

Register	Offset	R/W	Description	Reset Value
UA_TOR	UART_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[7:0]**	**TX Delay Time Value** This field is used to program the transfer delay time between the last stop bit and next start bit.
[7:0]	**TOIC[7:0]**	**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 (TOUT_INT) is generated if RTO_IEN (UA_IER [4]). A new incoming data word or RX FIFO empty clears TOUT_INT. 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 (UA_BAUD)

Register	Offset	R/W	Description	Reset Value
UA_BAUD	UART_BA+0x24	R/W	UART Baud Rate Divisor Register	0x0F00_0000

31	30	29	28	27	26	25	24
Reserved		**DIV_X_EN**	**DIV_X_ONE**	**DIVIDER_X**			
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]	**DIV_X_EN**	**Divider X Enable Control** 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 DIVIDER_X [27:24] must >= 8). **Note:** When in IrDA mode, this bit must be disabled.
[28]	**DIV_X_ONE**	**Divider X Equal 1** 0 = Divider M = X (the equation of M = X+1, but DIVIDER_X[27:24] must >= 8). 1 = Divider M = 1 (the equation of M = 1, but BRD [15:0] must >= 8). Refer to section “UART Controller Baud Rate Generator” for more information.
[27:24]	**DIVIDER_X[3:0]**	**Divider X** The baud rate divider M = X+1.
[23:16]	**Reserved**	Reserved.
[15:0]	**BRD[15:0]**	**Baud Rate Divider** The field indicates the baud rate divider.

IrDA Control Register (IRCR)

Register	Offset	R/W	Description	Reset Value
UA_IRCR	UART_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	**INV_RX**	**INV_TX**	**Reserved**			**TX_SELECT**	**Reserved**

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**INV_RX**	**INV_RX** 0 = No inversion. 1 = Inverse RX input signal.
[5]	**INV_TX**	**INV_TX** 0 = No inversion. 1 = Inverse TX output signal.
[4:2]	**Reserved**	Reserved.
[1]	**TX_SELECT**	**TX_SELECT** 0 = IrDA receiver Enabled. 1 = IrDA transmitter Enabled.
[0]	**Reserved**	Reserved.

UART Alternate Control/Status Register (UA_ALT_CSR)

Register	Offset	R/W	Description	Reset Value
UA_ALT_CSR	UART_BA+0x2C	R/W	UART Alternate Control/Status Register	0x0000_0000

31	30	29	28	27	26	25	24
ADDR_MATCH							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
RS485_ADD_EN	Reserved				RS485_AUD	RS485_AAD	RS485_NMM
7	6	5	4	3	2	1	0
Reserved							

Bits	Description	
[31:24]	**ADDR_MATCH [7:0]**	**Address Match Value** This field contains the RS-485 address match values. **Note:** This field is used for RS-485 auto address detection mode.
[23:16]	**Reserved**	Reserved.
[15]	**RS485_ADD_EN**	**RS-485 Address Detection Enable Control** 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 Address Detection Operation Mode (AAD) Disabled. 1 = RS-485 Auto Address Detection Operation Mode (AAD) Enabled. **Note:** This bit 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:** This bit 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:** This bit cannot be active with RS485_AAD operation mode.
[7:0]	**Reserved**	Reserved.

UART Function Select Register (UA_FUN_SEL)

Register	Offset	R/W	Description	Reset Value
UA_FUN_SEL	UART_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. 11 = RS-485 function mode.

6.10 I^{2}C Serial Interface Controller (I^{2}C)

6.10.1 Overview

I^{2}C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. The I^{2}C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously. The I^{2}C also supports Power-down wake-up function.

6.10.2 Features

The I^{2}C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the bus include:

- 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 I^{2}C interrupt if the I^{2}C bus hangs up and timer-out counter overflows
- External pull-up needed for higher output pull-up speed
- 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
- Support FIFO function

6.10.3 Basic Configuration

The basic configurations of I^{2}C are as follows:

- I^{2}C pins are configured on P4_MFP[7:6] register.
- Enable I^{2}C clock (I2C_EN) on APBCLK [8] register.
- Reset I^{2}C controller (I2C_RST) on IPRSTC2 [8] register.

6.10.4 Functional Description

On I^{2}C bus, data is transferred between a Master and a Slave. Data bits transfer on the SCL and SDA lines are synchronously on a byte-by-byte basis. Each data byte is 8-bit long. There is one SCL clock pulse for each data bit with the MSB being transmitted first, and an acknowledge bit

follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (START or STOP). Please refer to the following figure for more detailed I^{2}C BUS Timing.

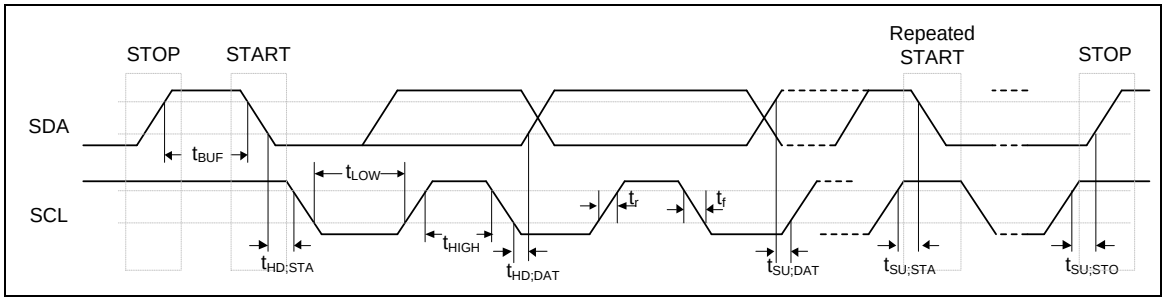


Figure 6.10-1 I^{2}C Bus Timing

The device on-chip I^{2}C provides the serial interface that meets the I^{2}C bus standard mode specification. The I^{2}C port handles byte transfers autonomously. To enable this port, the bit ENS1 in I2CON should be set to '1'. The I^{2}C hardware interfaces to the I^{2}C bus via two pins: SDA and SCL. When I/O pins are used as I^{2}C ports, user must set the pins function to I^{2}C in advance.

There is two-level FIFO to improve the performance of I^{2}C bus. In two-level FIFO 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 I^{2}C SCL bus is stretched low when there is SI event. The NOSTRETCH control bit is used to force the I^{2}C SCL bus is no stretched under the SI event.

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

Note: Pull-up resistor is needed for I^{2}C operation as the SDA and SCL are open-drain pins.

6.10.5 I^{2}C Protocol

The following figure shows the typical I^{2}C 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

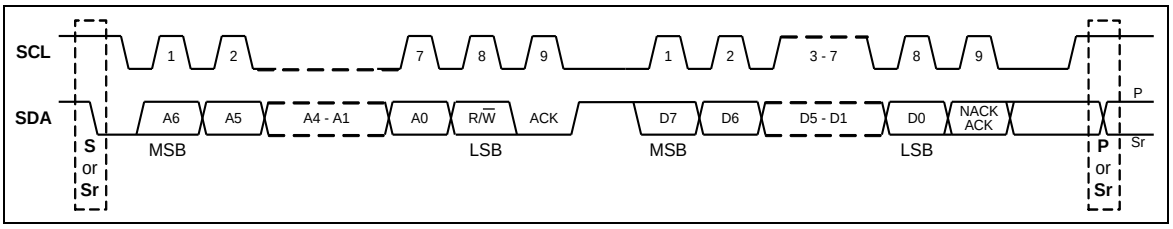


Figure 6.10-2 I^{2}C Protocol

When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred 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.

6.10.5.1 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.

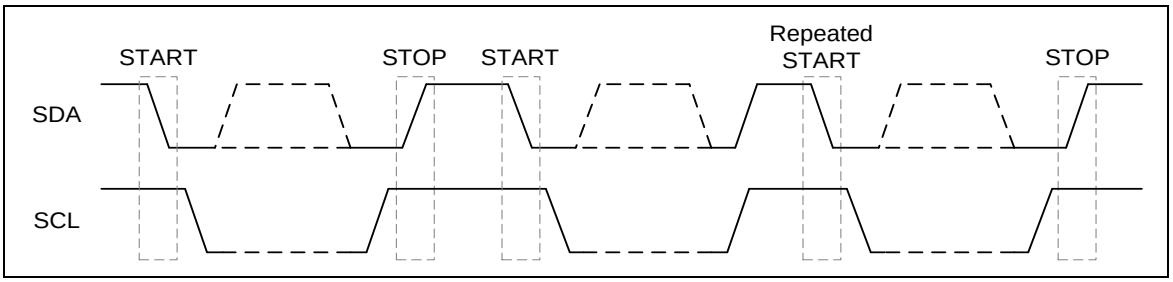


Figure 6.10-3 START and STOP Condition

6.10.5.2 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.

6.10.5.3 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.

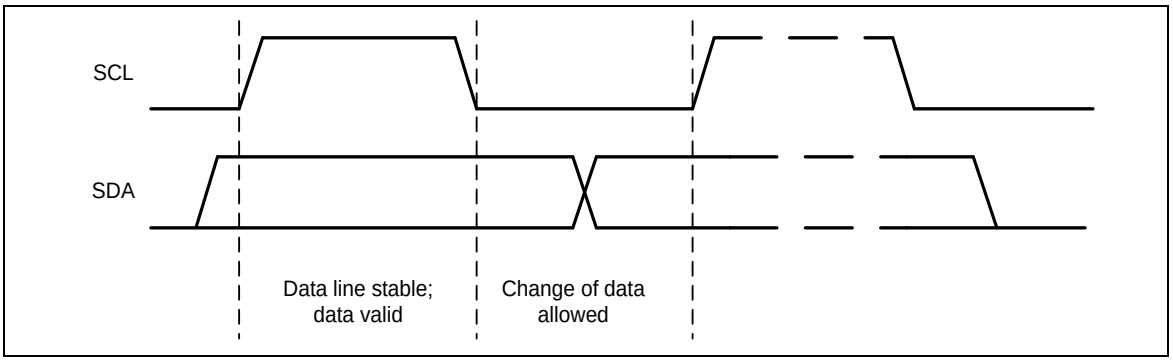


Figure 6.10-4 Bit Transfer on the I^{2}C Bus

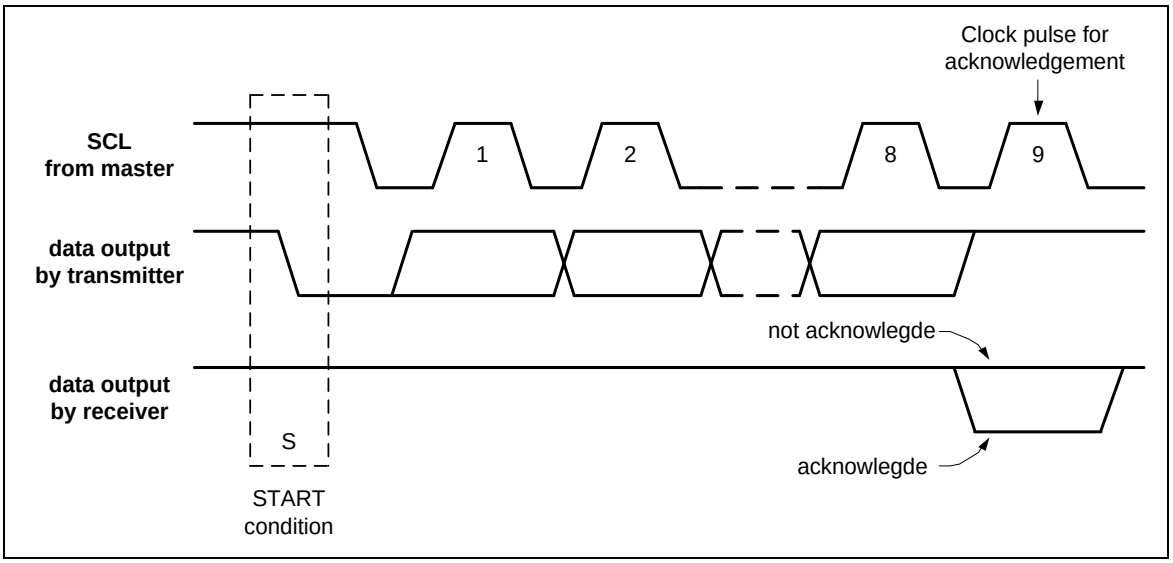


Figure 6.10-5 Acknowledge on the I^{2}C Bus

6.10.5.4 Data transfer on I^{2}C bus

The following figure 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.

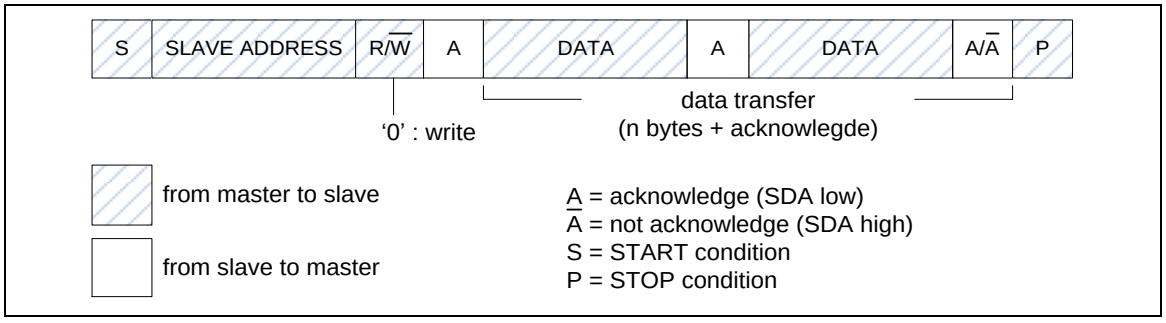


Figure 6.10-6 Acknowledge on the I^{2}C Bus

The following figure 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.

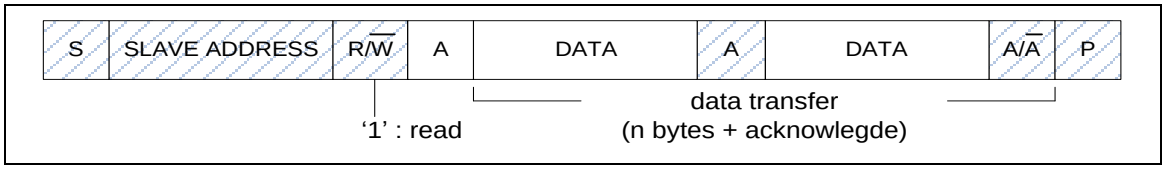


Figure 6.10-7 Master Reads Data from Slave

6.10.5.5 Two Level FIFO Mode on I²C bus

Set to enable the two-level FIFO 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 being clear. In this time, the 4th data can be transmitted and the I²C stop after the 4th data transmission done.

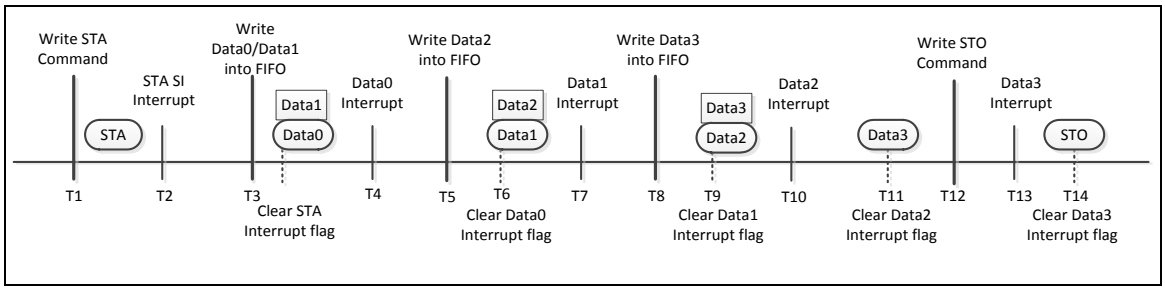


Figure 6.10-8 Timing of Two Level FIFO Transmit in Master Write

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 be cleared.

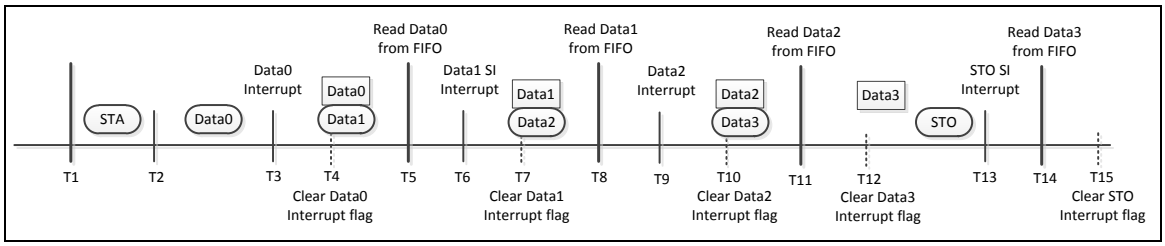


Figure 6.10-9 Timing of Two Level FIFO Transmit in Slave Read

6.10.6 I^{2}C Protocol Registers

To control I^{2}C port through the following fifteen special function registers: I2CON (control register), I2CSTATUS (status register), I2CDAT (data register), I2CADDRn (address registers, n=0~3), I2CADMn (address mask registers, n=0~3), I2CLK (clock rate register) and I2CTOC (time-out counter register), I2CWKUPCON (wake-up control register) and I2CWKUPSTS(wake-up status register).

6.10.6.1 Address Registers (I2CADDR)

The I^{2}C port is equipped with four slave address registers I2CADDRn (n=0~3). The contents of the register are irrelevant when I^{2}C is in Master mode. In Slave mode, the bit field I2CADDRn[7:1] must be loaded with the chip's own slave address. The I^{2}C hardware will react if the contents of I2CADDRn are matched with the received slave address.

The I^{2}C ports support the "General Call" function. If the GC bit (I2CADDRn[0]) is set, the I^{2}C 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^{2}C is in Slave mode, it can receive the general call address by 00H after Master send general call address to I^{2}C bus, then it will follow status of GC mode.

6.10.6.2 Slave Address Mask Registers (I2CADM)

The I^{2}C bus controller supports multiple address recognition with four address mask registers I2CADMn (n=0~3). When the bit in the address mask register is set to 1, it means the received corresponding address bit is "Don't care". If the bit is set to 0, it means the received corresponding register bit should be exactly the same as address register.

6.10.6.3 Data Register (I2CDAT)

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 (I2CDAT[7:0]) directly while it is not in the process of shifting a byte. When I^{2}C is in a defined state and the serial interrupt flag (SI) is set, data in I2CDAT[7:0] remains stable. While data is being shifted out, data on the bus is simultaneously being shifted in; I2CDAT[7:0] always contains the last data byte present on the bus.

The acknowledge bit is controlled by the I^{2}C hardware and cannot be accessed by the CPU. Serial data is shifted through into I2CDAT[7:0] on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into I2CDAT[7:0], the serial data is available in I2CDAT[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 I2CDATA[7:0] when sending I2CDATA[7:0] to bus. In the case of sending data, serial data bits are shifted out from I2CDAT [7:0] on the falling edge of SCL clocks, and is shifted to I2CDAT [7:0] on the rising edge of SCL clocks.

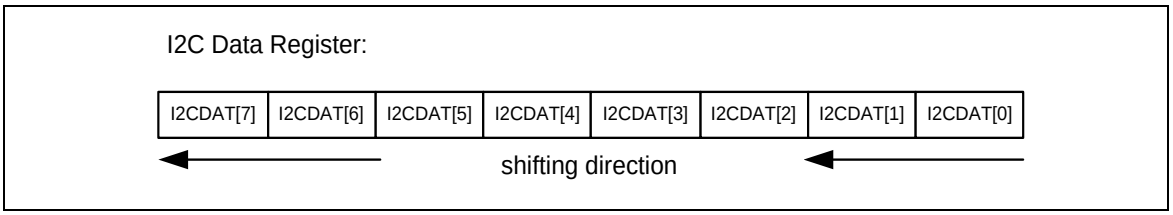


Figure 6.10-10 I^{2}C Data Shifting Direction

6.10.6.4 Control Register (I2CON)

The CPU can read from and write to I2CON[7:0] directly. When the I^{2}C port is enabled by setting ENS1 (I2CON [6]) to high, the internal states will be controlled by I2CON and I^{2}C logic hardware.

There are two bits are affected by hardware: the SI bit is set when the I^{2}C hardware requests a serial interrupt, and the STO bit is cleared when a STOP condition is present on the bus. The STO bit is also cleared when ENS1 = 0.

Once a new status code is generated and stored in I2CSTATUS, the I^{2}C Interrupt Flag bit SI (I2CON [3]) will be set automatically. If the Enable Interrupt bit EI (I2CON [7]) is set at this time, the I^{2}C interrupt will be generated. The bit field I2CSTATUS[7:0] stores the internal state code, the content keeps stable until SI is cleared by software.

6.10.6.5 Status Register (I2CSTATUS)

I2CSTATUS[7:0] is an 8-bit read-only register. The bit field I2CSTATUS[7:0] contain the status code. There are 26 possible status codes. All states are listed in . When I2CSTATUS[7:0] is F8H, no serial interrupt is requested. All other I2CSTATUS[7:0] values correspond to defined I^{2}C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2CSTATUS[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 I^{2}C format frame. A Bus Error may occur during the serial transfer of an address byte, a data byte or an acknowledge bit. To recover I^{2}C from bus error, STO should be set and SI should be clear to enter Not Addressed Slave mode. Then STO is cleared to release bus and to wait for a new communication. I^{2}C 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:** Status "0xF8" exists in both master/slave modes, and it won't raise interrupt.		

Table 6.10-1 I^2C Status Code Description

6.10.6.6 I^2C Clock Baud Rate Bits (I2CLK)

The data baud rate of I^2C is determined by I2CLK[7:0] register when I^2C is in Master mode. It is not necessary in a Slave mode. In Slave mode, I^2C will automatically synchronize with any clock frequency from master I^2C device.

The data baud rate of I^2C setting is Data Baud Rate of I^2C = (system clock) / (4x (I2CLK[7:0] +1)). If system clock = 16 MHz, the I2CLK[7:0] = 40 (28H), the data baud rate of I^2C = 16 MHz / (4x (40 +1)) = 97.5K bits/sec.

6.10.6.7 I^2C Time-out Counter Register (I2CTOC)

There is a 14-bit time-out counter which can be used to deal with the I^2C bus hang-up. If the time-out counter is enabled, the counter starts up counting until it overflows (TIF=1) and generates I^2C interrupt to CPU or stops counting by clearing ENTI 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^2C bus hangs up, it causes the I2CSTATUS and flag SI are not updated for a period, the 14-bit time-out counter may overflow and acknowledge CPU the I^2C interrupt. Refer to the following figure for the 14-bit time-out counter. User may write 1 to clear TIF to 0.

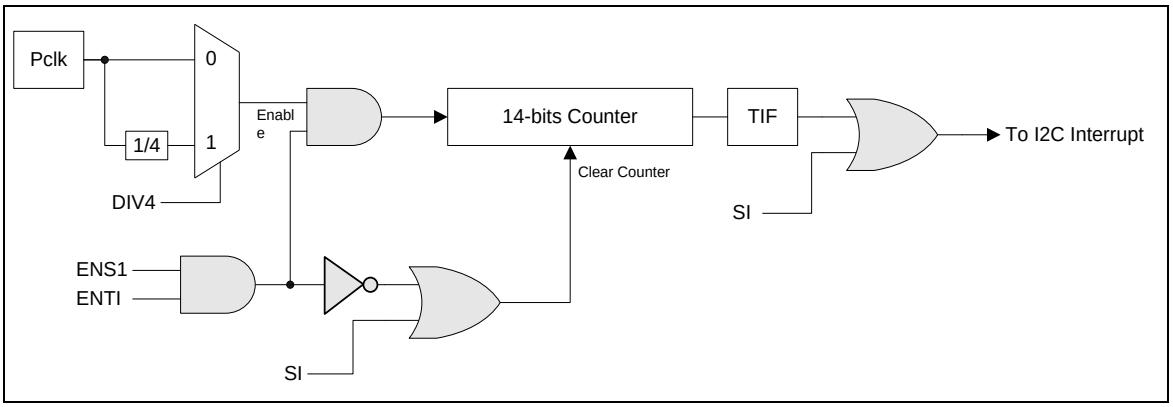


Figure 6.10-11 I^{2}C Time-out Count Block Diagram

6.10.6.8 I^{2}C Control Register 2 (I2CON2)

When entering Power-down mode, other I^{2}C master can wake-up the chip by addressing our I^{2}C device, 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.

6.10.6.9 I^{2}C Status Register 2 (I2CSTATUS2)

When system is woken up by other I^{2}C master device, WAKEIF is set to indicate this event. User needs write “1” to clear this bit. The other status bits are used to indicate the current FIFO status when the TWOFF_EN is set.

6.10.7 Operation Modes

The on-chip I^{2}C ports support three operation modes, Master, Slave, and General Call Mode.

In a given application, I^{2}C port may operate as a master or as a slave. In Slave mode, the I^{2}C port hardware looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to master(by setting the AA bit), acknowledge pulse will be transmitted out on the 9th clock, hence an interrupt is requested on both master and slave devices if interrupt is enabled. When the microcontroller wishes to become the bus master, hardware waits until the bus is free before entering Master mode so that a possible slave action is not be interrupted. If bus arbitration is lost in Master mode, I^{2}C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

To control the I^{2}C bus transfer in each mode, user needs to set I2CON, I2CDAT registers according to current status code of I2CSTATUS register. In other words, for each I^{2}C bus action, user needs to check current status by I2CSTATUS register, and then set I2CON, I2CDAT registers to take bus action. Finally, check the response status by I2CSTATUS.

The bits, STA, STO and AA in I2CON register are used to control the next state of the I^{2}C hardware after SI flag of I2CON [3] register is cleared. Upon completion of the new action, a new status code will be updated in I2CSTATUS register and the SI flag of I2CON register will be set. If the I^{2}C interrupt control bit EI (I2CON [7]) is set, appropriate action or software branch of the new status code can be performed in the Interrupt service routine.

The following figure shows the current I^{2}C status code is 0x08, and then set I2CDATA=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 I2CSTATUS will be updated by status code 0x18.

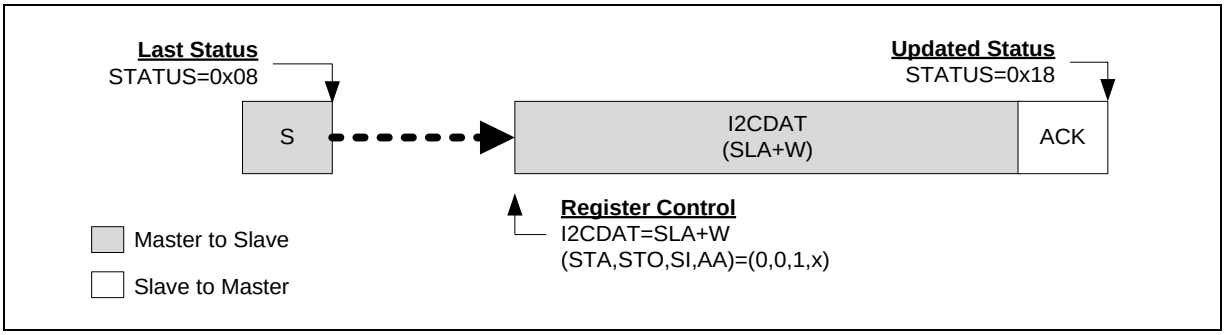


Figure 6.10-12 Control I^{2}C Bus according to Current I^{2}C Status

6.10.7.1 Master Mode

In the following figures, all possible protocols for I^{2}C master are shown. 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 () or Master receiver mode () 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.

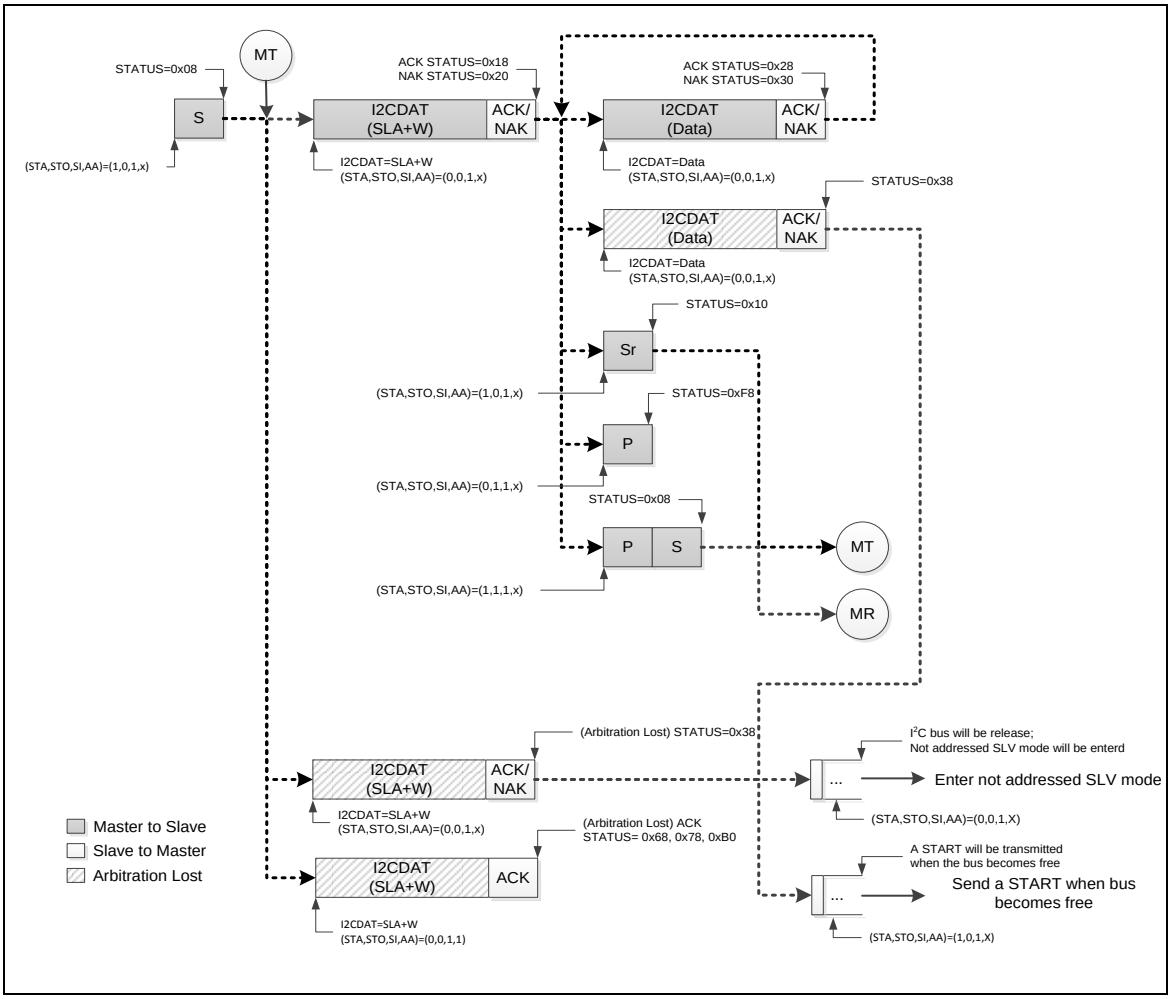


Figure 6.10-13 Master Transmitter Mode Control Flow

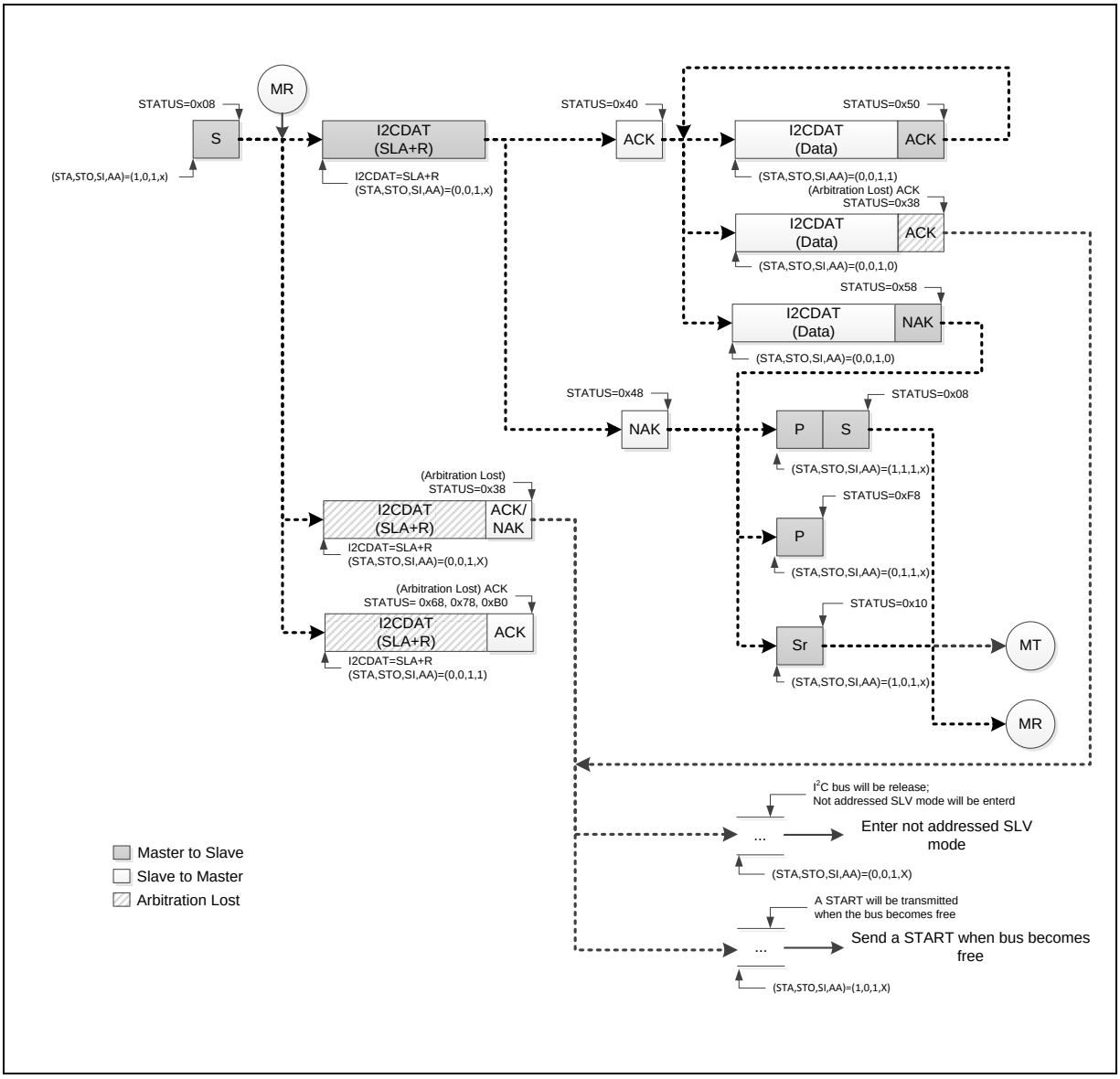


Figure 6.10-14 Master Receiver Mode Control Flow

If the I^{2}C is in Master mode and gets arbitration lost, the status code will be 0x38. In status 0x38, user may set (STA, STO, SI, AA) = (1, 0, 1, X) to send START to re-start Master operation when bus become free. Otherwise, user may set (STA, STO, SI, AA) = (0, 0, 1, X) to release I^{2}C bus and enter not addressed Slave mode.

6.10.7.2 Slave Mode

When reset default, I^{2}C is not addressed and will not recognize the address on I^{2}C bus. User can set slave address by I2CADDRx and set (STA, STO, SI, AA) = (0, 0, 1, 1) to let I^{2}C recognize the address sent by master. shows all the possible flow for I^{2}C in Slave mode. Users need to follow a proper flow (as shown in to implement their own I^{2}C protocol.

If bus arbitration is lost in Master mode, I^{2}C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer. If the detected address is SLA+W (Master want to write data to Slave) after arbitration lost, the status code is 0x68. If the detected address is SLA+R (Master want to read data from Slave) after arbitration lost, the status code is 0xB0.

Note: During I^{2}C communication, the SCL clock will be released when writing '1' to clear SI flag in Slave mode.

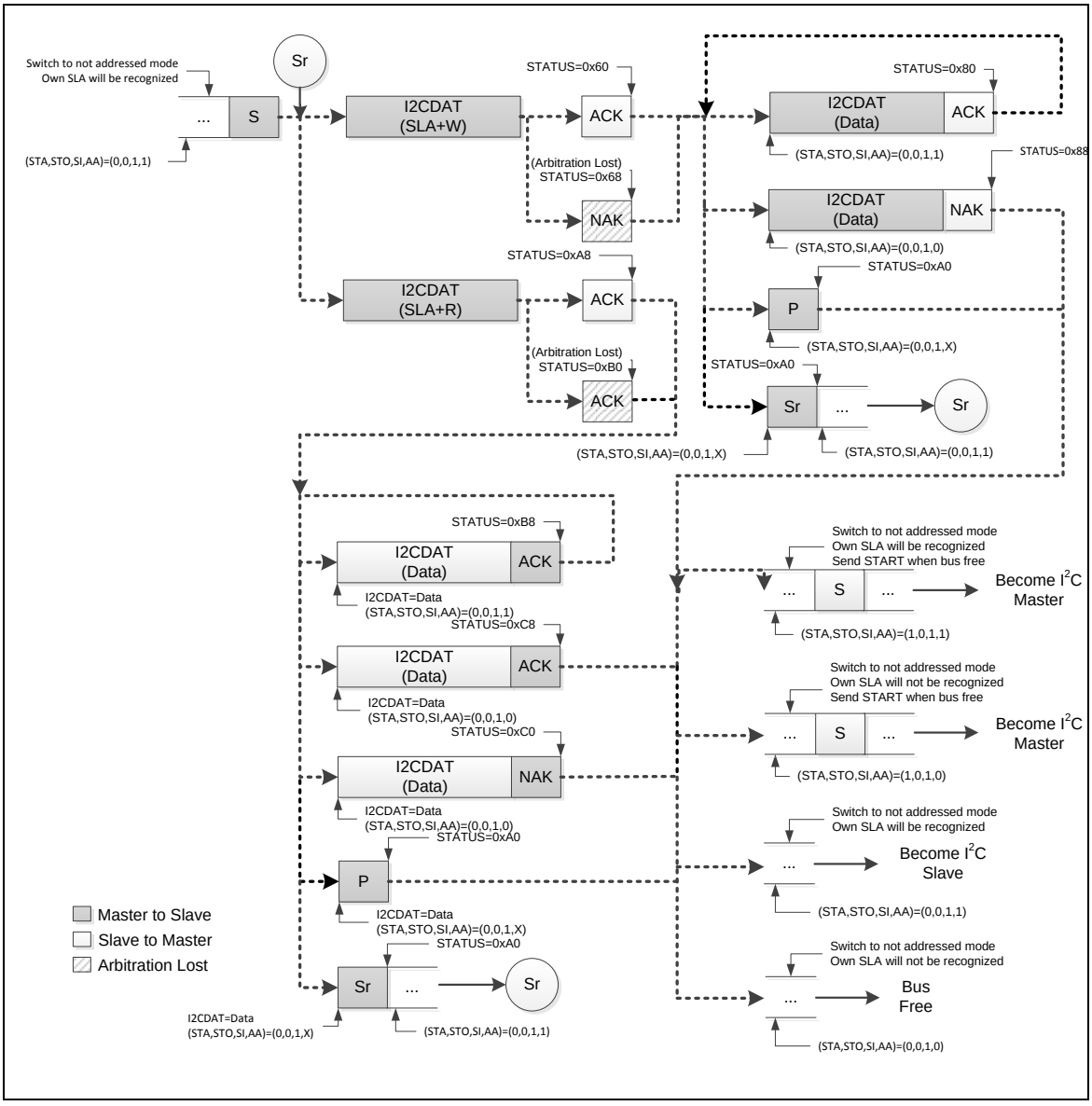


Figure 6.10-15 Slave Mode Control Flow

If I^2C is still receiving data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x88 as shown in the above figure when getting 0xA0 status.

If I^2C is still transmitting data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0xC8 as shown in the above figure when getting 0xA0 status.

Note: After slave gets status of 0x88, 0xC8, 0xC0 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I^2C signal or address from master. At this status, I^2C should be reset to leave this status.

6.10.7.3 General Call (GC) Mode

If the GC bit (I2CADDRn [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.

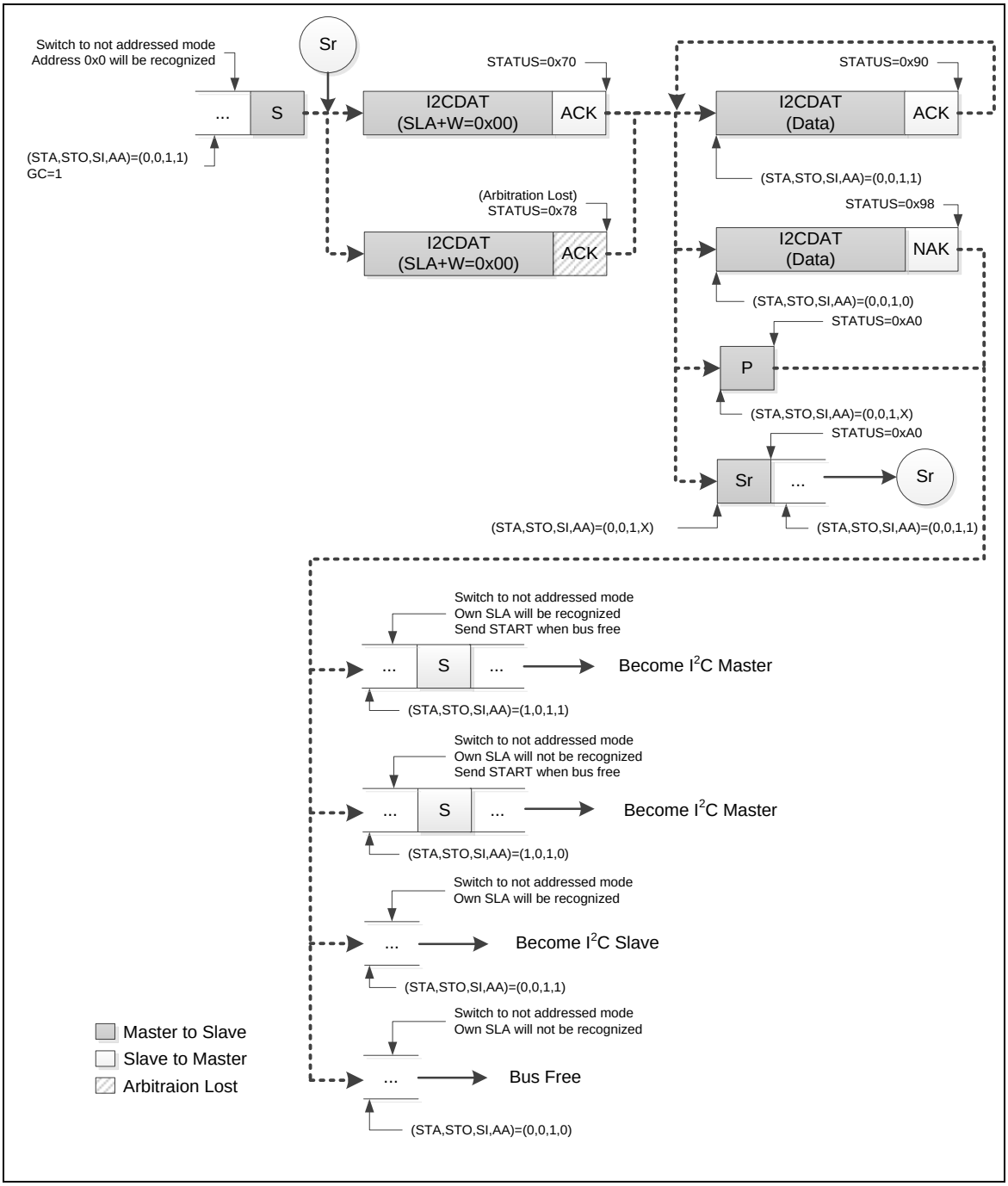


Figure 6.10-16 GC Mode

If I^{2}C is still receiving data in GC mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x98 in above figure when getting 0xA0 status.

Note: After slave gets status of 0x98 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I^{2}C signal or address from master. At this time, I^{2}C controller should be reset to leave this status.

6.10.7.4 Multi-Master

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

- When I2CSTATUS = 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 I2CSTATUS = 0x00, a “Bus Error” is received. To recover I^{2}C bus from a bus error, STO should be set and SI should be cleared, and then STO is cleared to release bus.
  - Set (STA, STO, SI, AA) = (0, 1, 1, X) to stop current transfer
  - Set (STA, STO, SI, AA) = (0, 0, 1, X) to release bus

6.10.7.5 Example for Random Read on EEPROM

The following steps are used to configure the I^{2}C related registers when using I^{2}C to read data from EEPROM.

1. Set the multi-function pin in the “P4_MFP” and “P4_ALT” registers as SCL and SDA pins.
2. Enable I^{2}C APB clock, I2C_EN=1 in the “APBCLK” register.
3. Set I2C_RST=1 to reset I^{2}C controller then set I^{2}C controller to normal operation, I2C_RST=0 in the “IPRSTC2” register.
4. Set ENS1=1 to enable I^{2}C controller in the “I2CON” register.
5. Give I^{2}C clock a divided register value for I^{2}C clock rate in the “I2CLK”.
6. Set SETENA=0x00040000 in the “NVIC_ISER” register to set I^{2}C IRQ.
7. Set EI=1 to enable I^{2}C Interrupt in the “I2CON” register.
8. Set I^{2}C address registers which are “I2CADDR0~I2CADDR3”.

Random read operation is one of the methods of access EEPROM. The method allows the master to access any address of EEPROM space. The following figure shows the EEPROM random read operation.

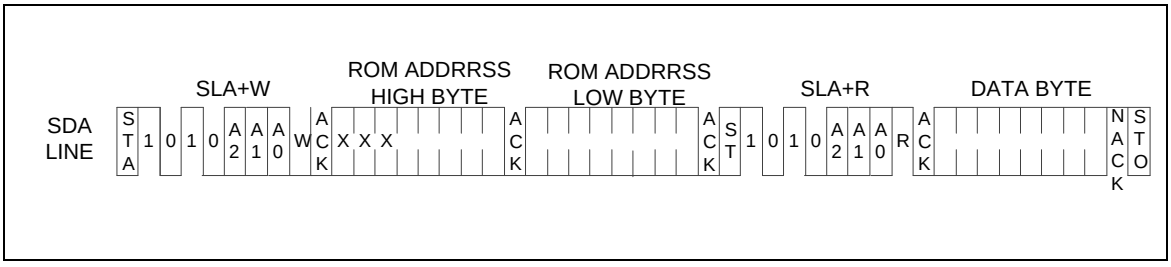


Figure 6.10-17 EEPROM Random Read

The following figure shows how to use I^{2}C controller to implement the protocol of EEPROM random read.

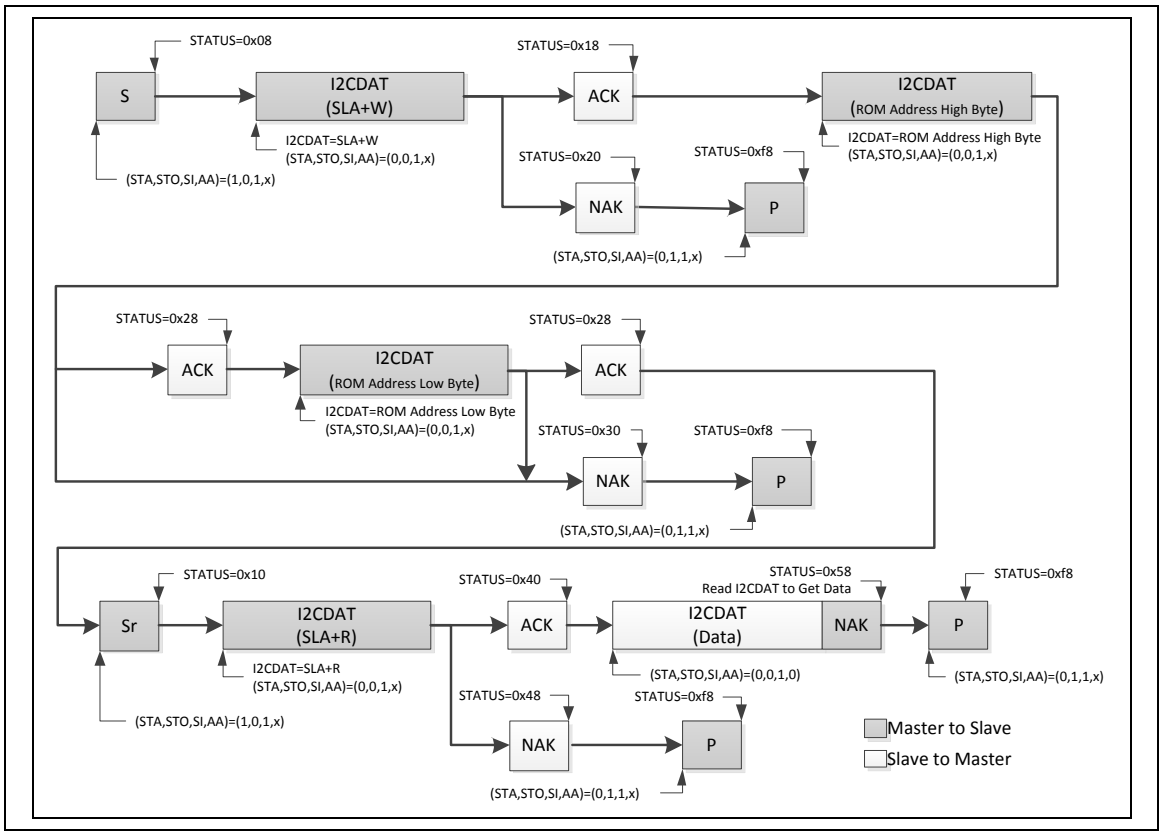


Figure 6.10-18 Protocol of EEPROM Random Read

The I^{2}C 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.

6.10.8 Register Map

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

Register	Offset	R/W	Description	Reset Value
I²C Base Address: **I²C_BA = 0x4002_0000**				
I2CON	I2C_BA+0x00	R/W	I²C Control Register	0x0000_0000
I2CADRR0	I2C_BA+0x04	R/W	I²C Slave Address Register 0	0x0000_0000
I2CDAT	I2C_BA+0x08	R/W	I²C DATA Register	0x0000_0000
I2CSTATUS	I2C_BA+0x0C	R	I²C Status Register	0x0000_00F8
I2CLK	I2C_BA+0x10	R/W	I²C Clock Divided Register	0x0000_0000
I2CTOC	I2C_BA+0x14	R/W	I²C Time-Out Counter Register	0x0000_0000
I2CADDR0	I2C_BA+0x04	R/W	I²C Slave Address Register 0	0x0000_0000
I2CADDR1	I2C_BA+0x18	R/W	I²C Slave Address Register 1	0x0000_0000
I2CADDR2	I2C_BA+0x1C	R/W	I²C Slave Address Register 2	0x0000_0000
I2CADDR3	I2C_BA+0x20	R/W	I²C Slave Address Register 3	0x0000_0000
I2CADM0	I2C_BA+0x24	R/W	I²C Slave Address Mask Register 0	0x0000_0000
I2CADM1	I2C_BA+0x28	R/W	I²C Slave Address Mask Register 1	0x0000_0000
I2CADM2	I2C_BA+0x2C	R/W	I²C Slave Address Mask Register 2	0x0000_0000
I2CADM3	I2C_BA+0x30	R/W	I²C Slave Address Mask Register 3	0x0000_0000
I2CCON2	I2C_BA+0x3C	R/W	I²C Control Register 2	0x0000_0000
I2CSTATUS2	I2C_BA+0x40	R/W	I²C Status Register 2	0x0000_0000

6.10.9 Register Description

I²C Control Register (I2CON)

Register	Offset	R/W	Description	Reset Value
I2CON	I2C_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**
EI	**ENS1**	**STA**	**STO**	**SI**	**AA**	**Reserved**	

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**EI**	**Interrupt Enable Control** 0 = I²C interrupt Disabled. 1 = I²C interrupt Enabled.
[6]	**ENS1**	**I²C Controller Enable Control** 0 = I²C Controller Disabled. 1 = I²C Controller Enabled. Set to enable I²C serial function controller. When ENS1=1 the I²C serial function enables. The function of multi-function pin must be set to I²C first.
[5]	**STA**	**I²C START Control Bit** Setting 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 I2CSTATUS register, the SI flag is set by hardware, and if bit EI (I2CON[7]) is set, the I²C interrupt is requested. SI must be cleared by software. Software can write 1 to clear this bit.

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 (I2CDAT)

Register	Offset	R/W	Description	Reset Value
I2CDAT	I2C_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**
I2CDAT							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**I2CDAT[7:0]**	**I²C Data Bits** Bit [7:0] is located with the 8-bit transferred data of the I²C serial port.

I²C STATUS REGISTER (I2CSTATUS)

Register	Offset	R/W	Description	Reset Value
I2CSTATUS	I2C_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**
I2CSTATUS							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**I2CSTATUS[7:0]**	**I²C Status Bits** The three least significant bits are always 0. The five most significant bits contain the status code. There are 26 possible status codes. When I2CSTATUS contains F8H, no serial interrupt is requested. All the other I2CSTATUS 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 I2CSTATUS 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 BAUD RATE CONTROL REGISTER (I2CLK)

Register	Offset	R/W	Description	Reset Value
I2CLK	I2C_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**
I2CLK							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**I2CLK[7:0]**	**I²C Clock Divided Bits** The I²C clock rate bits: Data Baud Rate of I²C = (system clock) / (4x (I2CLK+1)). **Note:** The minimum value of I2CLK is 4.

I²C TIME-OUT COUNTER REGISTER (I2CTOC)

Register	Offset	R/W	Description	Reset Value
I2CTOC	I2C_BA+0x14	R/W	I²C Time-Out Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**ENTI**	**DIV4**	**TIF**

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2]	**ENTI**	**Time-out Counter Enable Control** 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 clear. Setting 1to the SI flag will reset counter and re-start up counting after SI is cleared.
[1]	**DIV4**	**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]	**TIF**	**Time-out Flag** This bit is set by hardware when I²C time-out happened and it can interrupt CPU if I²C interrupt enable bit (EI) is set to 1. **Note:** Software can write 1 to clear this bit.

I²C SLAVE ADDRESS REGISTER (I2CADDRx)

Register	Offset	R/W	Description	Reset Value
I2CADDR0	I2C_BA+0x04	R/W	I²C Slave Address Register 0	0x0000_0000
I2CADDR1	I2C_BA+0x18	R/W	I²C Slave Address Register 1	0x0000_0000
I2CADDR2	I2C_BA+0x1C	R/W	I²C Slave Address Register 2	0x0000_0000
I2CADDR3	I2C_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**
I2CADDR[7:1]							**GC**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:1]	**I2CADDR**	**I²C Address Bits** 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** 0 = General Call Function Disabled. 1 = General Call Function Enabled.

I²C SLAVE ADDRESS MASK REGISTER (I2CADMx)

Register	Offset	R/W	Description	Reset Value
I2CADM0	I2C_BA+0x24	R/W	I²C Slave Address Mask Register 0	0x0000_0000
I2CADM1	I2C_BA+0x28	R/W	I²C Slave Address Mask Register 1	0x0000_0000
I2CADM2	I2C_BA+0x2C	R/W	I²C Slave Address Mask Register 2	0x0000_0000
I2CADM3	I2C_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**
I2CADM[7:1]							**Reserved**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:1]	**I2CADM**	**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 2 (I2CCON2)

Register	Offset	R/W	Description	Reset Value
I2CCON2	I2C_BA+0x3C	R/W	I²C Control Register 2	0x0000_0000

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

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**UNDER_INTEN**	**I²C UNDER RUN Interrupt Control Bit** Setting UNDER_INTEN to enable will send a interrupt to system when the TWOFF bit is enabled and there is under run event happened in transmitted FIFO. 0 = Disabled. 1 = Enabled.
[3]	**OVER_INTEN**	**I²C OVER RUN Interrupt Control Bit** Setting OVER_INTEN to enable will send a interrupt to system when the TWOFF bit is enabled and there is over run event in received FIFO. 0 = Disabled. 1 = Enabled.
[2]	**NOSTRETCH**	**NO STRETCH 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]	**TWOFF_EN**	**TWO LEVEL FIFO Enable Control** 0 = Disabled. 1 = Enabled. Set to enable the two-level FIFO 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.
[0]	**WAKEUPEN**	**Wake-up Enable Control** 0 = I²C wake-up function Disabled. 1 = I²C wake-up function Enabled. The system can be wake-up by I²C bus when the system is set into power mode and the received data matched one of the addresses in Address Register.

I²C STATUS REGISTER 2 (I2CSTATUS 2)

Register	Offset	R/W	Description	Reset Value
I2CSTATUS2	I2C_BA+0x40	R/W	I²C Status Register 2	0x0000_0000

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

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4]	**UNDERUN**	**I²C UNDER RUN Status Bit** This bit indicates the transmitted FIFO is under run when the TWOFF_EN = 1.
[3]	**OVERUN**	**I²C OVER RUN Status Bit** This bit indicates the received FIFO is over run when the TWOFF_EN = 1.
[2]	**EMPTY**	**I²C TWO LEVEL FIFO EMPTY** This bit indicates RX FIFO empty or not when the TWOFF_EN = 1.
[1]	**FULL**	**I²C TWO LEVEL FIFO FULL** This bit indicates TX FIFO full or not when the TWOFF_EN = 1.
[0]	**WAKEUPIF**	**I²C Wake-up Interrupt Flag** When chip is woken up from Power-Down mode by I²C, this bit is set to 1. Software can write 1 to clear this bit.

6.11 Serial Peripheral Interface (SPI)

6.11.1 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.

6.11.2 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

6.11.3 SPI Block Diagram

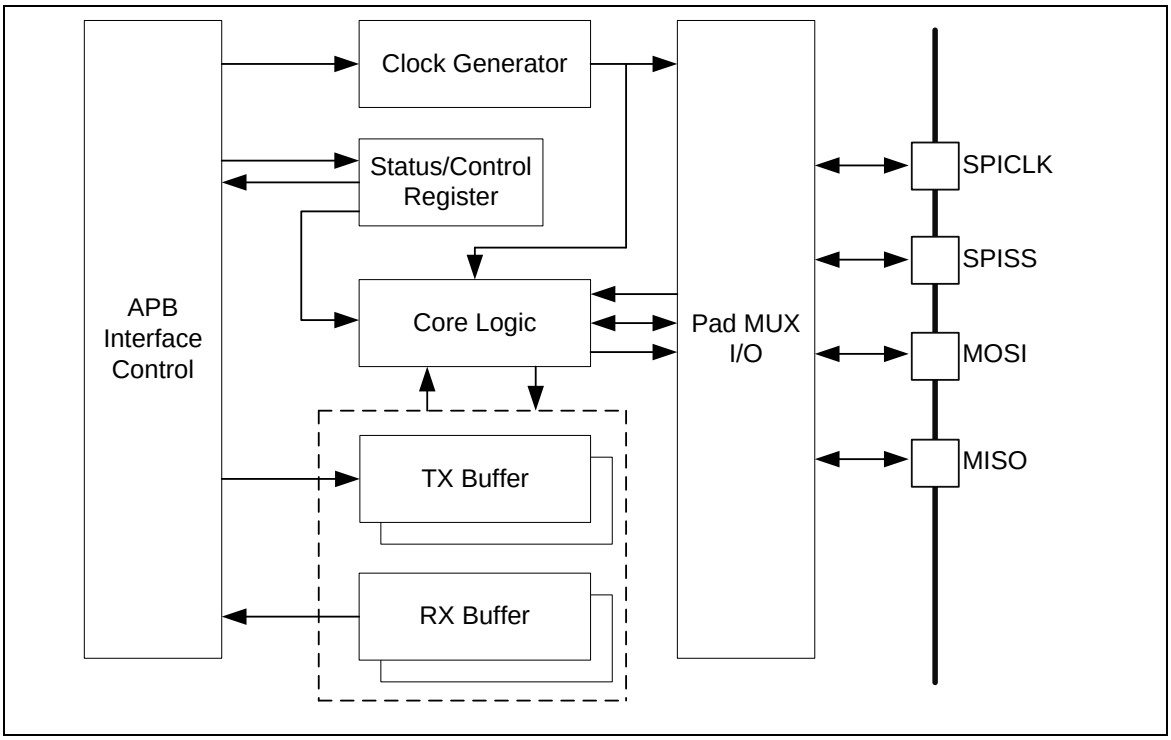


Figure 6.11-1 SPI Block Diagram

6.11.4 Basic Configuration

The SPI pin functions are configured in P0_MFP register. The SPI peripheral clock can be enabled in SPI_EN(APBCLK[12]) bit and its source can be selected in SPI_S(CLKSEL1[4]) bit.

6.11.5 Functional Description

6.11.5.1 Terminology

SPI Peripheral Clock and Serial Bus Clock

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 SPICLK pin. The SPI peripheral clock frequency is determined by the settings of clock source, BCn (SPI_CNTRL2[31]) option and clock divisor (DIVIDER(SPI_DIVIDER[7:0])). The SPI_S(CLKSEL1[14]) register determines the clock source of the SPI peripheral clock. Set the BCn bit to 0 for the compatible SPI clock frequency calculation of previous products. The DIVIDER setting of SPI_DIVIDER register determines the divisor of the clock rate calculation.

In SPI Master mode, the frequency of SPI bus clock is equal to SPI peripheral 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

This SPI controller can be set as Master or Slave mode by setting the SLAVE bit (SPI_CNTRL[18]) to communicate with the off-chip SPI slave or master device. The application block diagrams in Master and Slave mode are shown below.

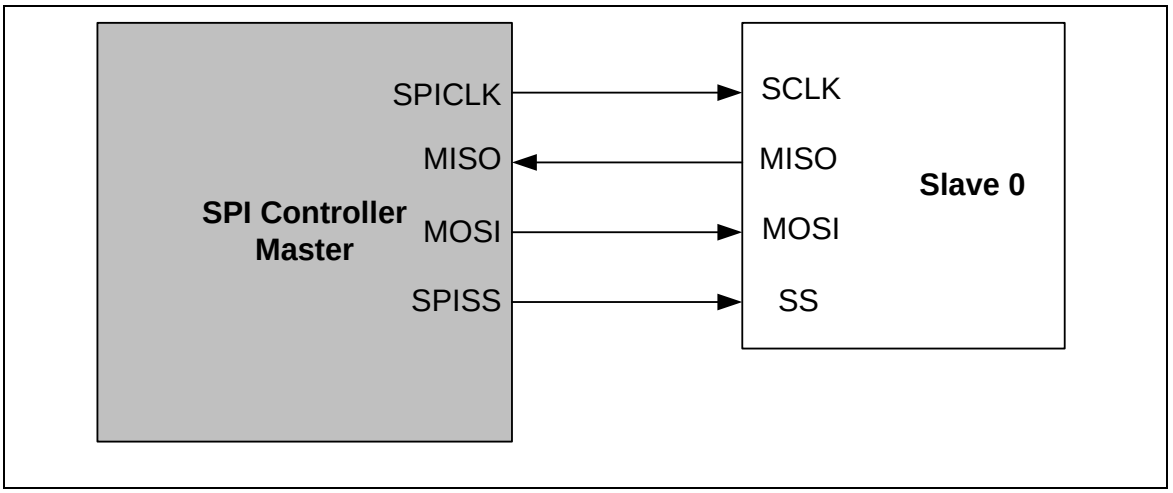


Figure 6.11-2 SPI Master Mode Application Block Diagram

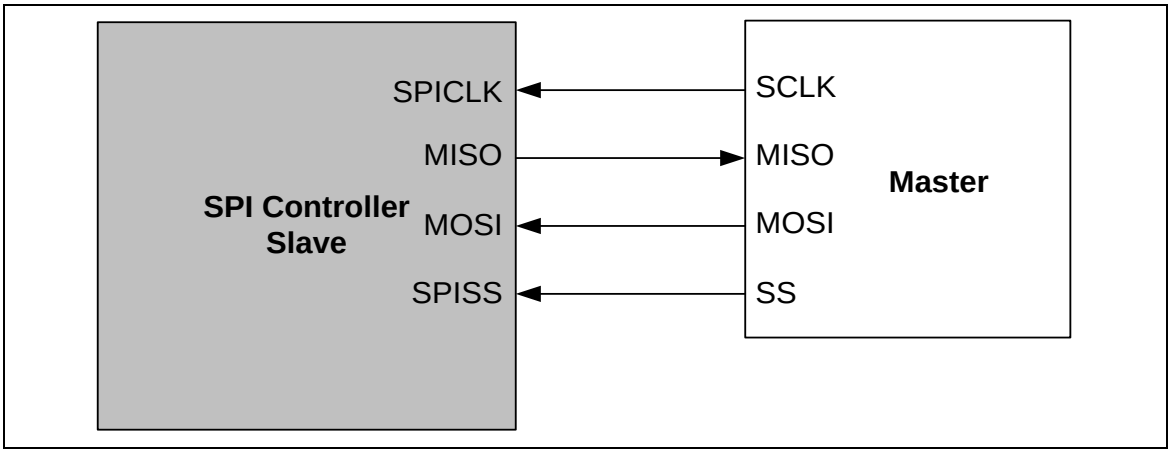


Figure 6.11-3 SPI Slave Mode Application Block Diagram

Clock Polarity

The CLKP bit (SPI_CNTRL [11]) defines the bus clock idle state. If CLKP = 1, the output SPICLK is idle at high state; otherwise it is at low state if CLKP = 0.

Transmit/Receive Bit Length

The bit length of a transaction word is defined in TX_BIT_LEN bit field (SPI_CNTRL[7:3]). It can be configured up to 32-bit length in a transaction word to transmitting and receiving.

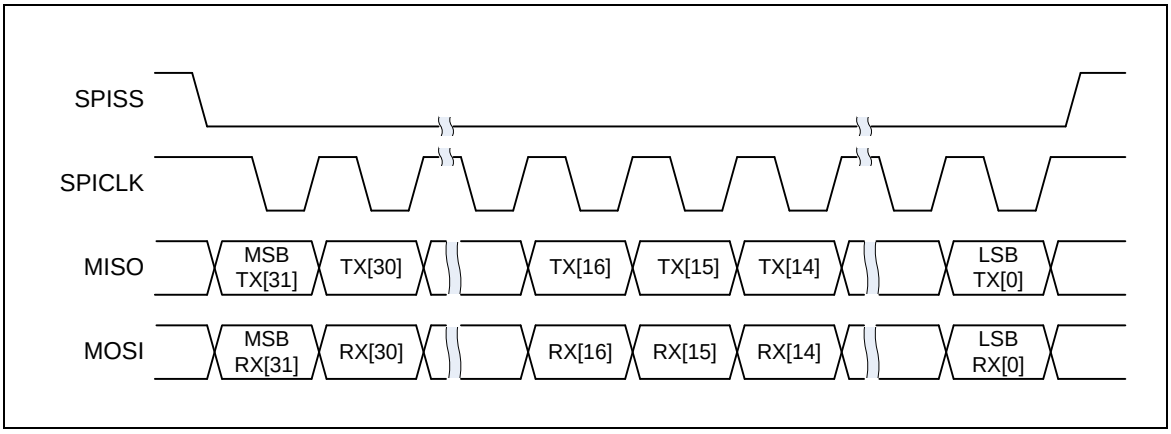


Figure 6.11-4 32-bit in One Transaction

LSB/MSB First

The LSB bit (SPI_CNTRL[10]) defines the data transmission either from LSB or MSB firstly to start to transmit/receive data.

Transmit Edge

The TX_NEG bit (SPI_CNTRL[2]) defines the data transmitted out either on negative edge or on positive edge of serial clock SPICLK.

Receive Edge

The RX_NEG bit (SPI_CNTRL[1]) defines the data received in either on negative edge or on positive edge of serial clock SPICLK.

Note: the settings of TX_NEG and RX_NEG are mutual exclusive. In other words, do not transmit and receive data on the same clock edge.

Word Suspend

The four bits field of SP_CYCLE (SPI_CNTRL[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 SP_CYCLE is 0x3 (3.5 serial clock cycles). This SP_CYCLE setting will not take effect to the word suspend interval if the software disables the FIFO mode.

Slave Section

In Master mode, the SPI controller can drive one off-chip slave device through the slave select output pin SPISS. In Slave mode, the off-chip master device drives the slave select signal from the SPISS input pin to this SPI controller. In Master/Slave mode, the active state of slave selected signal can be programmed to low active or high active in SS_LVL bit (SPI_SSR[2]), and the SS_LTRIG bit (SPI_SSR[4]) define the slave select signal SPISS 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 SS_LTRIG bit is configured as level trigger, the LTRIG_FLAG bit (SPI_SSR[5]) is used to indicate if both the number of received data and the number of received bits meet the requirement which defined in TX_BIT_LEN 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 (SPI_CNTRL[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 bits matches the settings of TX_BIT_LEN, 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 LTRIG_FLAG bit to check if the data has been completely transferred.

6.11.5.2 Automatic Slave Select

In Master mode, if the bit AUTOSS (SPI_SSR[3]) is set, the slave select signal will be generated automatically and output to SPISS pin according to SSR (SPI_SSR[0]) whether enabled or not. It means that the slave select signal, which is enabled is SSR register is asserted by the SPI controller when transmit/receive is started by setting the GO_BUSY bit (SPI_CNTRL[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 SSR[0] register. The active level of the slave select output signal is specified in SS_LVL bit (SPI_SSR[2]).

6.11.5.3 Byte Reorder Function

When the transfer is set as MSB first (LSB = 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 (TX_BIT_LEN = 0). The sequence of transmitted/received data will be BYTE0, BYTE1, BYTE2, and then BYTE3. If the TX_BIT_LEN 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 TX_BIT_LEN is configured as 16, 24, and 32 bits.

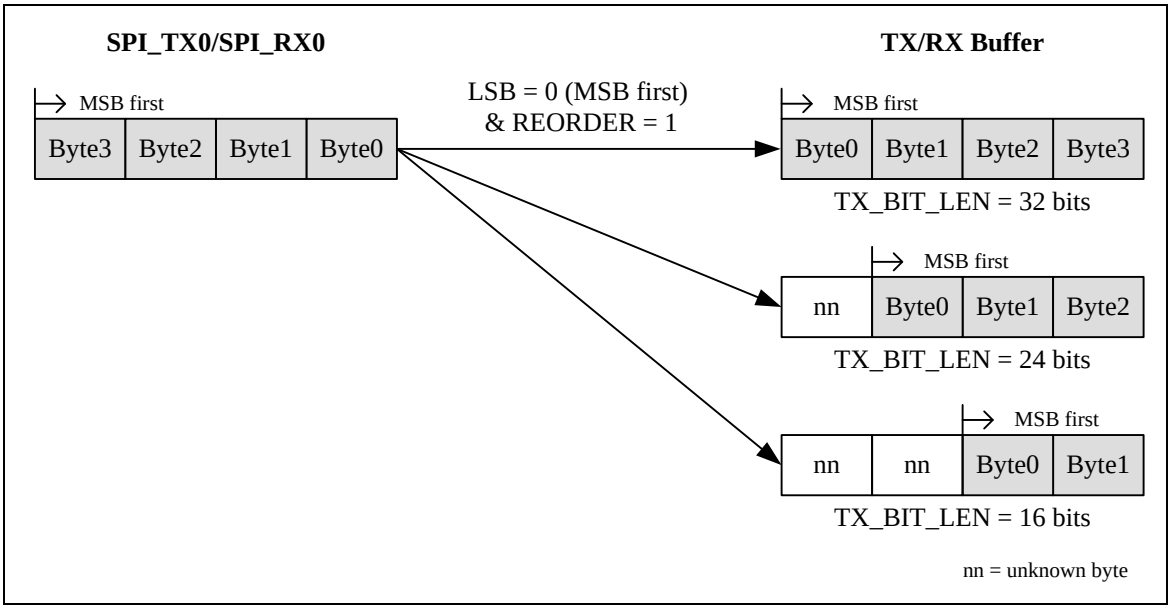


Figure 6.11-5 Byte Reorder

6.11.5.4 *Byte Suspend Function*

Both settings of byte suspend interval and word suspend interval are configured in SP_CYCLE. In Master mode, if byte reorder function is enabled by setting SPI_CNTRL[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 TX_BIT_LEN can be configured as 16, 24 or 32 bits.

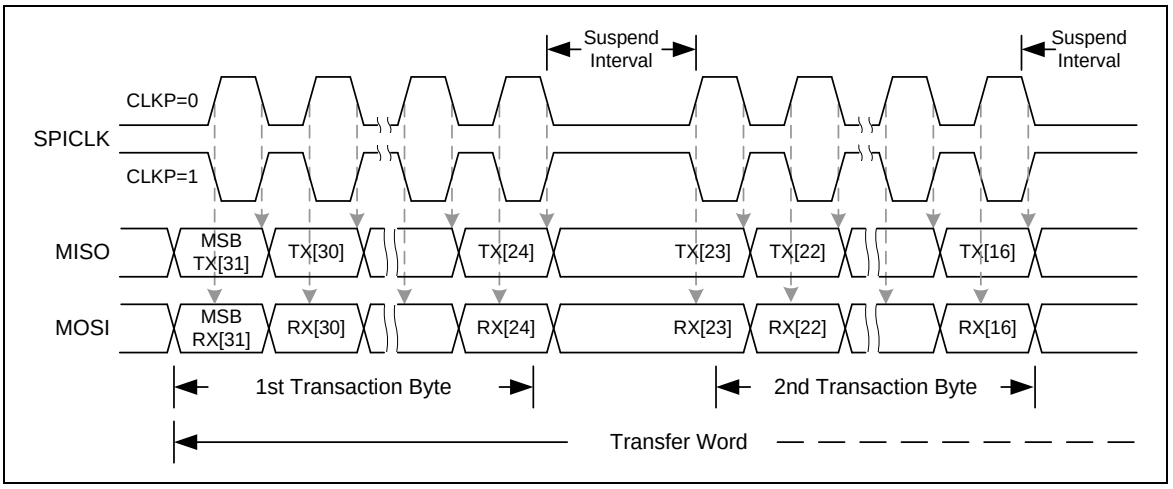


Figure 6.11-6 Timing Waveform for Byte Suspend

6.11.5.5 *Slave 3-Wire Mode*

When the NOSLVSEL bit (SPI_CNTRL2[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 NOSLVSEL bit only takes effect in Slave mode. Only three pins, SPICLK, MISO, and MOSI, are required to communicate with a SPI master. The SPISS pin can be configured as a GPIO. When the NOSLVSEL bit is set to 1, the SPI slave will be ready to transmit/receive data after the GO_BUSY bit is set to 1. As the number of received bits meets the requirement which defined in TX_BIT_LEN, the unit-transfer interrupt flag, IF (SPI_CNTRL[16]), will be set to 1.

Note: In Slave 3-wire mode, the SS_LTRIG bit (SPI_SSR[4]) should be set as 1.

6.11.5.6 FIFO Mode

The SPI controller supports FIFO mode when the FIFO(SPI_CNTRL[21]) bit is set as 1. The SPI controllers equip with four 32-bit wide transmit and receive FIFO buffers.

The transmit FIFO buffer is a 4-layer depth, 32-bit wide, first-in, 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 TX_FULL(SPI_STATUS[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 TX_EMPTY(SPI_STATUS[26]) bit will be set to 1. Notice that the TX_EMPTY flag 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, 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 RX_EMPTY(SPI_STATUS[24]) and RX_FULL(SPI_STATUS[25]), to indicate the current status of FIFO buffer.

In FIFO mode, the software can set the transmitting and receiving threshold by setting the TX_THRESHOLD(SPI_FIFO_CTL[29:28]) and RX_THRESHOLD(SPI_FIFO_CTL[25:24]) settings. When the count of valid data stored in transmit FIFO buffer is less than or equal to TX_THRESHOLD setting, the TX_INTSTS bit will be set to 1. When the count of valid data stored in receive FIFO buffer is larger than RX_THRESHOLD setting, the RX_INTSTS bit will be set to 1.

In FIFO mode, the software can write 4 transmit data to the SPI transmit FIFO buffer in advance. When the SPI controller operates with FIFO mode, the GO_BUSY bit of SPI_CNTRL register will be controlled by hardware, software should not modify the content of SPI_CNTRL register unless clearing the FIFO bit to disable the FIFO mode.

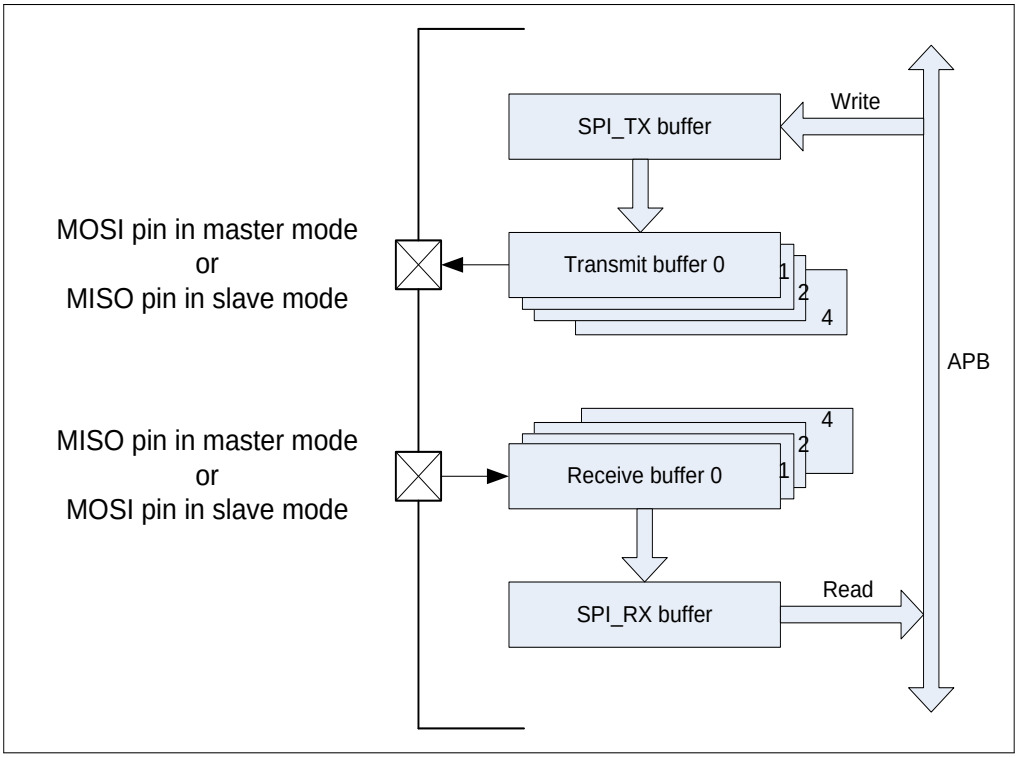


Figure 6.11-7 FIFO mode Block Diagram

SPI Master mode and FIFO mode Enabled

In Master mode transmission operation, the TX_EMPTY flag will be cleared to 0 when the FIFO bit is set to 1 and the software write the first datum 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 SP_CYCLE (SPI_CNTRL [15:12]). User can write data into SPI_TX register as long as the TX_FULL 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 MISO pin and stored to receive FIFO buffer. The RX_EMPTY 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 RX_EMPTY flag is 0. If the receive FIFO buffer contain 4 unread data, the RX_FULL 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 FIFO bit is set as 1, the GO_BUSY bit will be set as 1 by hardware automatically. If user wants to stop the slave mode SPI data transfer, both the FIFO bit and GO_BUSY 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 TX_EMPTY flag 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 TX_FULL flag 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 TX_EMPTY flag will be set to 1.

In Slave mode reception operation, the serial data is received from MOSI pin and stored to SPI_RX register. The reception mechanism is similar to master mode reception operation.

6.11.5.7 Interrupt

SPI unit-transfer interrupt

As the SPI controller finishes a unit transfer, the unit transfer interrupt flag IF (SPI_CNTRL[16]) will be set to 1. The unit transfer interrupt event will generate an interrupt to CPU if the unit transfer interrupt enable bit IE (SPI_CNTRL[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, SLV_START_INTSTS (SPI_CNTRL2[11]), will be set to 1 when the slave senses the SPI bus clock signal. The SPI controller will issue an interrupt if the SSTA_INTEN (SPI_CNTRL2[10]) is set to 1. If the count of the received bits is less than the setting of TX_BIT_LEN 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 SLV_ABORT bit (SPI_CNTRL2[9]) to abort the current transfer. The unit-transfer interrupt flag, IF, will be set to 1 if the SLV_ABORT 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, (SPI_FIFO_CTL[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 TX_THRESHOLD, 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, SPI_FIFO_CTL[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 RX_THRESHOLD, 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, SPI_FIFO_CTL[2], is set to 1.

6.11.6 Timing Diagram

The active state of slave select signal can be defined by the settings of SS_LVL bit (SPI_SSR[2]) and SS_LTRIG bit (SPI_SSR[4]). The serial clock (SPICLK) idle state can be configured as high state or low state by setting the CLKP bit (SPI_CNTRL[11]). It also provides the bit length of a transaction word in TX_BIT_LEN (SPI_CNTRL[7:3]), and transmit/receive data from MSB or LSB first in LSB bit (SPI_CNTRL[10]). User also can select which edge of bus clock to transmit/receive data in TX_NEG/RX_NEG (SPI_CNTRL[2:1]). Four SPI timing diagrams for master/slave operations and the related settings are shown below.

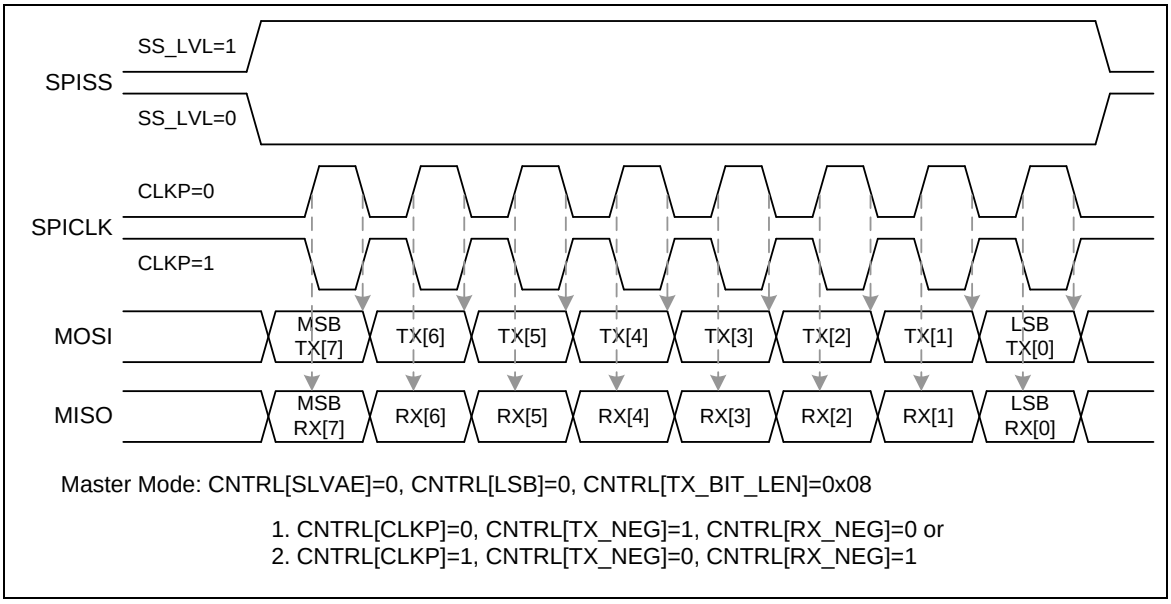


Figure 6.11-8 SPI Timing in Master Mode

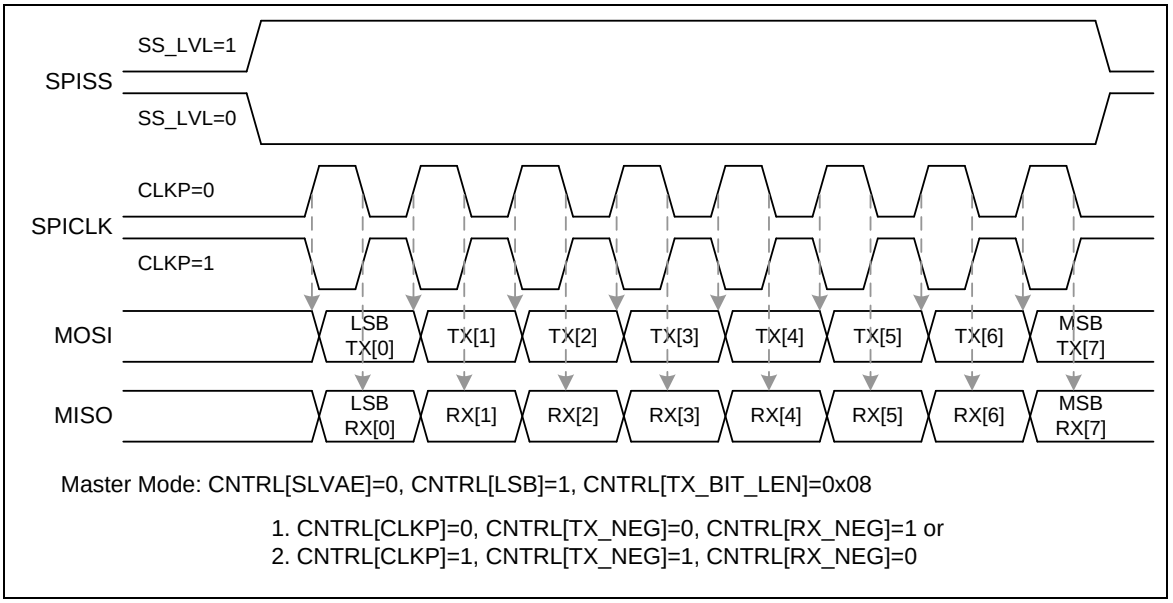


Figure 6.11-9 SPI Timing in Master Mode (Alternate Phase of SPICLK)

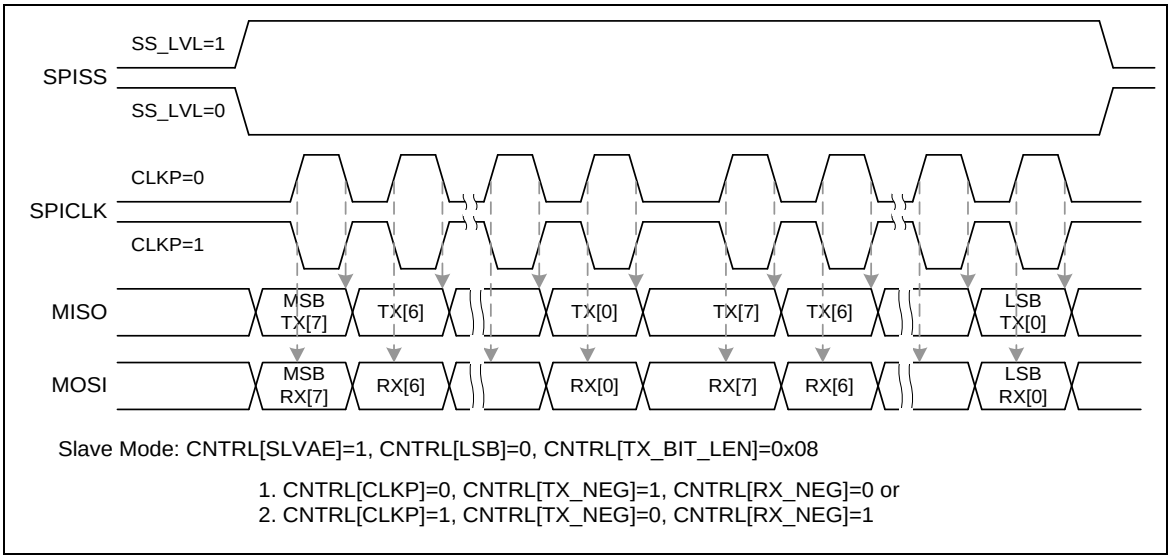


Figure 6.11-10 SPI Timing in Slave Mode

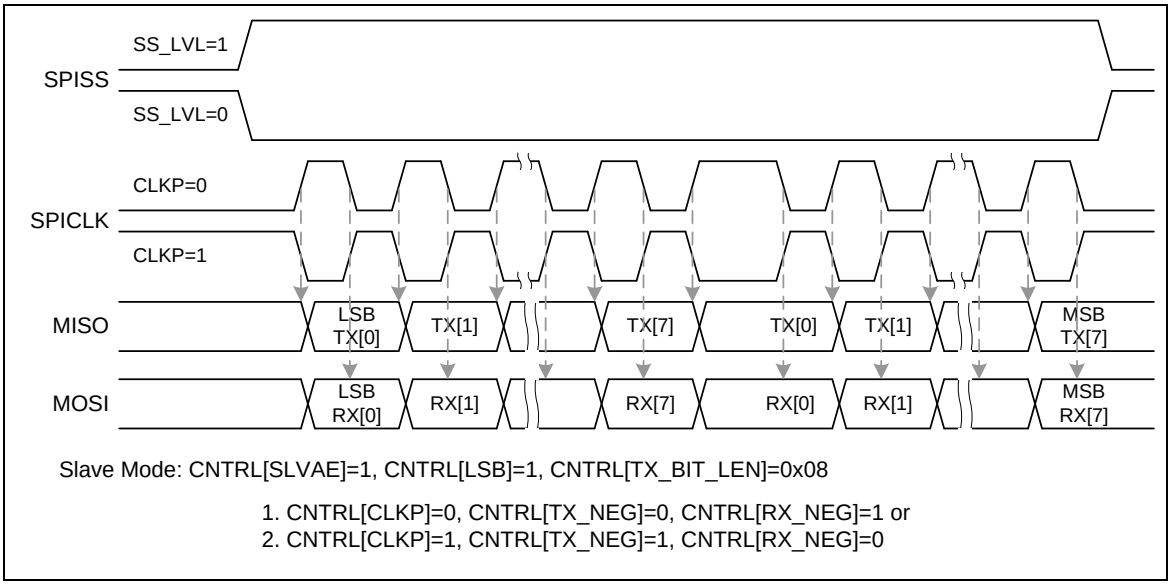


Figure 6.11-11 SPI Timing in Slave Mode (Alternate Phase of SPICLK)

6.11.7 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 on positive edge of bus clock
- Data bit is driven on negative edge of bus clock
- Data bit is transferred from MSB first
- Data width is 8 bits
- SPICLK idles at low 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_DIVIDER [7:0]) to determine the output frequency of bus clock.
2) Write the related settings into the SPI_CNTRL register to control this SPI master actions
    1. Set this SPI controller as master device in SLAVE bit (SPI_CNTRL[18] = 0)
    2. Force the serial clock idle state at low in CLKP bit (SPI_CNTRL[11] = 0)
    3. Select data transmitted at negative edge of bus clock in TX_NEG bit (SPI_CNTRL[2] = 1)
    4. Select data latched at positive edge of bus clock in RX_NEG bit (SPI_CNTRL[1] = 0)
    5. Set the bit length of word transfer as 8-bit in TX_BIT_LEN bit field (SPI_CNTRL[7:3] = 0x08)
    6. Set MSB transfer first in MSB bit (SPI_CNTRL[10] = 0), and need not to care the SP_CYCLE bit field (SPI_CNTRL[15:12]) due to it is not in FIFO mode in this case.

3) Write the SPI_SSR register a proper value for the related settings of Master mode.
   1. Clear the Automatic Slave Select bit AUTOSS(SPI_SSR[3] = 0).

      Select low level trigger output of slave select signal in the Slave Select Active Level control bit, SS_LVL (SPI_SSR[2] = 0).
   2. Set the slave select signal to be active by setting the Slave Select control bit SSR (SPI_SSR[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 GO_BUSY bit (SPI_CNTRL [0] = 1) to start the data transfer with the SPI interface.
7) Waiting for SPI interrupt (if the Interrupt Enable IE bit is set) or just polling the GO_BUSY bit 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 SSR bit (SPI_SSR[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 on positive edge of bus clock
- Data bit is driven on negative edge of bus clock
- Data bit is transferred from LSB first
- Data width is 8 bits
- SPICLK idles at high 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_DIVIDER[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_SSR 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 SS_LVL (SPI_SSR[2] = 1) and the Slave Select Level Trigger SS_LTRIG (SPI_SSR[4] = 1).
3) Write the related settings into the SPI_CNTRL register to control this SPI slave actions
   1. Set this SPI controller as slave device in SLAVE bit (SPI_CNTRL[18] = 1)
   2. Select the serial clock idle state at high in CLKP bit (SPI_CNTRL[11] = 1)
   3. Select data transmitted at negative edge of serial clock in TX_NEG bit (SPI_CNTRL[2] = 1)
   4. Select data latched at positive edge of serial clock in RX_NEG bit (SPI_CNTRL[1] = 0)

5. Set the bit length of word transfer as 8 bits in TX_BIT_LEN bit field (SPI_CNTRL[7:3] = 0x08)
6. Set LSB transfer first in LSB bit (SPI_CNTRL[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 update by software.
6) Set the GO_BUSY bit (SPI_CNTRL[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 at the SPI interface.
7) Waiting for SPI interrupt (if the Interrupt Enable IE bit is set) or just polling the GO_BUSY bit 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 GO_BUSY bit to stop data transfer.

6.11.8 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_CNTRL	SPI_BA+0x00	R/W	SPI Control and Status Register	0x0500_3004
SPI_DIVIDER	SPI_BA+0x04	R/W	SPI Clock Divider Register	0x0000_0000
SPI_SSR	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_CNTRL2	SPI_BA+0x3C	R/W	SPI Control and Status Register 2	0x0000_0000
SPI_FIFO_CTL	SPI_BA+0x40	R/W	SPI FIFO Control Register	0x2200_0000
SPI_STATUS	SPI_BA+0x44	R/W	SPI Status Register	0x0500_0000

6.11.9 Register Description

SPI Control and Status Register (SPI_CNTRL)

Register	Offset	R/W	Description	Reset Value
SPI_CNTRL	SPI_BA+0x00	R/W	SPI Control and Status Register	0x0500_3004

31	30	29	28	27	26	25	24
Reserved				TX_FULL	TX_EMPTY	RX_FULL	RX_EMPTY
23	22	21	20	19	18	17	16
Reserved		FIFO	Reserved	REORDER	SLAVE	IE	IF
15	14	13	12	11	10	9	8
SP_CYCLE				CLKP	LSB	Reserved	
7	6	5	4	3	2	1	0
TX_BIT_LEN					TX_NEG	RX_NEG	GO_BUSY

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27]	**TX_FULL**	**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]	**TX_EMPTY**	**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]	**RX_FULL**	**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]	**RX_EMPTY**	**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_CNTRL[24].
[23:22]	**Reserved**	Reserved.
[21]	**FIFO**	**FIFO Mode Enable Control** 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 GO_BUSY bit will be set to 1 automatically after writing data into the 4-depth transmit FIFO. When all data stored at

		transmit FIFO buffer are transferred, the GO_BUSY 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 TX_BIT_LEN is defined as 16, 24, and 32 bits.
[18]	**SLAVE**	**Slave Mode Control** 0 = Master mode. 1 = Slave mode.
[17]	**IE**	**Unit-transfer Interrupt Enable Control** 0 = SPI unit-transfer interrupt Disabled. 1 = SPI unit-transfer interrupt Enabled.
[16]	**IF**	**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]	**SP_CYCLE[3:0]**	**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: (SP_CYCLE[3:0] + 0.5) * period of SPICLK clock cycle Example: SP_CYCLE = 0x0 … 0.5 SPICLK clock cycle. SP_CYCLE = 0x1 … 1.5 SPICLK clock cycle. …… SP_CYCLE = 0xE … 14.5 SPICLK clock cycle. SP_CYCLE = 0xF … 15.5 SPICLK clock cycle.
[11]	**CLKP**	**Clock Polarity** 0 = SPICLK idle low. 1 = SPICLK 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]	**TX_BIT_LEN[4:0]**	**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. TX_BIT_LEN = 0x08 … 8 bits. TX_BIT_LEN = 0x09 … 9 bits. …… TX_BIT_LEN = 0x1F … 31 bits. TX_BIT_LEN = 0x00 … 32 bits.
[2]	**TX_NEG**	**Transmit On Negative Edge**

		0 = The transmitted data output signal is driven on the Rising edge of SPICLK. 1 = The transmitted data output signal is driven on the Falling edge of SPICLK.
[1]	**RX_NEG**	**Receive On Negative Edge** 0 = The received data input signal latched on the Rising edge of SPICLK. 1 = The received data input signal latched on the Falling edge of SPICLK.
[0]	**GO_BUSY**	**SPI Transfer Control Bit And Busy Status** If FIFO mode is enabled, this bit will be controlled by hardware and is 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 GO_BUSY 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 GO_BUSY bit will not be cleared to 0 when slave select signal goes to inactive state.

SPI Divider Register (SPI_DIVIDER)

Register	Offset	R/W	Description	Reset Value
SPI_DIVIDER	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[7:0]							

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**DIVIDER**	**Clock Divider Bits (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 SPICLK output pin. The frequency is obtained according to the following equation: If the bit of BCn, SPI_CNTRL2[31], is set to 0. $f_{spi} = \frac{f_{SPI_clock_src}}{(DIVIDER+1)*2}$ else if BCn 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 CLKSEL1 register.

SPI Slave Select Register (SPI_SSR)

Register	Offset	R/W	Description	Reset Value
SPI_SSR	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		**LTRIG_FLAG**	**SS_LTRIG**	**AUTOSS**	**SS_LVL**	**Reserved**	**SSR**

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**LTRIG_FLAG**	**Level Trigger Flag (Read Only, Slave Only)** When the SS_LTRIG 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 met the specified requirements which defined in TX_BIT_LEN.
[4]	**SS_LTRIG**	**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 = SPISS pin signal will be asserted/de-asserted by setting /clearing SSR bit. 1 = SPISS pin signal will be generated automatically, which means that slave select signal will be asserted by the SPI controller when transmit/receive is started by setting GO_BUSY, and will be de-asserted after each transmit/receive is finished.
[2]	**SS_LVL**	**Slave Select Active Level (Slave Only)** It defines the active status of slave select signal (SPISS). If SS_LTRIG bit is 1: 0 = The slave select signal SPISS is active at Low-level. 1 = The slave select signal SPISS is active at High-level. If SS_LTRIG bit is 0: 0 = The slave select signal SPISS is active at Falling-edge. 1 = The slave select signal SPISS is active at Rising-edge.
[1]	**Reserved**	Reserved.
[0]	**SSR**	**Slave Select Control Bit (Master Only)** If AUTOSS bit is 0,

		0 = Set the SPISS line to inactive state. 1 = Set the proper SPISS line to active state. If AUTOSS bit is 1, 0 = Keep the SPISS line at inactive state. 1 = Select the SPISS 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 SPISS is specified in SS_LVL 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[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RX[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RX[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX[7:0]							

Bits	Description	
[31:0]	**RX**	**Data Receive Bits (Read Only)** The Data Receive Registers hold the value of received data of the last executed transfer. Valid bits depend on the transmit bit length field in the SPI_CNTRL register. For example, if TX_BIT_LEN is set to 0x08, bit RX [7:0] holds the received data. The values of the other bits are unknown.

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[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TX[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TX[7:0]							

Bits	Description	
[31:0]	**TX**	**Data Transmit Bits (Write Only)** The Data Transmit Registers hold the data to be transmitted in the next transfer. Valid bits depend on the transmit bit length field in the CNTRL register. For example, if TX_BIT_LEN is set to 0x08, the bit TX [7:0] will be transmitted in next transfer.

SPI Control and Status Register 2 (SPI_CNTRL2)

Register	Offset	R/W	Description	Reset Value
SPI_CNTRL2	SPI_BA+0x3C	R/W	SPI Control and Status Register 2	0x0000_0000

31	30	29	28	27	26	25	24
BCn	**Reserved**						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**SS_INT_OPT**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**SLV_START _INTSTS**	**SSTA_ INTEN**	**SLV_ABORT**	**NOSLVSEL**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31]	**BCn**	**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_DIVIDER register for details.
[30:17]	**Reserved**	Reserved.
[16]	**SS_INT_OPT**	**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]	**SLV_START_INTSTS**	**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 SSTA_INTEN 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]	**SSTA_INTEN**	**Slave 3-wire Mode Start Interrupt Enable Control (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 SLV_ABORT 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 SLV_START_INTSTS bit is cleared to 0.

[9]	**SLV_ABORT**	**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 TX_BIT_LEN. 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 NOSLVSEL bit is set to 1. 1 = Force the transfer done when the NOSLVSEL 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]	**NOSLVSEL**	**Slave 3-wire Mode Enable Control (Slave Only)** The SPI controller work with 3-wire interface including SPICLK, SPI_MISO, and SPI_MOSI 0 = The controller is 4-wire bi-direction interface. 1 = The controller is 3-wire bi-direction interface in Slave mode. The controller will be ready to transmit/receive data after the GO_BUSY bit is set to 1. **Note:** In Slave 3-wire mode, the SS_LTRIG bit (SPI_SSR[4]) shall be set as 1.
[7:0]	**Reserved**	Reserved.

SPI FIFO Control Register (SPI_FIFO_CTL)

Register	Offset	R/W	Description	Reset Value
SPI_FIFO_CTL	SPI_BA+0x40	R/W	SPI FIFO Control Register	0x2200_0000

31	30	29	28	27	26	25	24
Reserved		**TX_THRESHOLD**		**Reserved**		**RX_THRESHOLD**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**TIMEOUT_ INTEN**	**Reserved**				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**RXOV_ INTEN**	**Reserved**		**TX_INTEN**	**RX_INTEN**	**TX_CLR**	**RX_CLR**

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:28]	**TX_THRESHOLD [1:0]**	**Transmit FIFO Threshold** If the valid data count of the transmit FIFO buffer is less than or equal to the TX_THRESHOLD setting, the TX_INTSTS bit will be set to 1, else the TX_INTSTS bit will be cleared to 0.
[27:26]	**Reserved**	Reserved.
[25:24]	**RX_THRESHOLD [1:0]**	**Received FIFO Threshold** If the valid data count of the receive FIFO buffer is larger than the RX_THRESHOLD setting, the RX_INTSTS bit will be set to 1, else the RX_INTSTS bit will be cleared to 0.
[23:22]	**Reserved**	Reserved.
[21]	**TIMEOUT_INTEN**	**Receive FIFO Time-out Interrupt Enable Control** 0 = Time-out interrupt Disabled. 1 = Time-out interrupt Enabled.
[20:7]	**Reserved**	Reserved.
[6]	**RXOV_INTEN**	**Receive FIFO Overrun Interrupt Enable Control** 0 = Receive FIFO overrun interrupt Disabled. 1 = Receive FIFO overrun interrupt Enabled.
[5:4]	**Reserved**	Reserved.
[3]	**TX_INTEN**	**Transmit Threshold Interrupt Enable Control** 0 = Transmit threshold interrupt Disabled. 1 = Transmit threshold interrupt Enabled.
[2]	**RX_INTEN**	**Receive Threshold Interrupt Enable Control** 0 = Receive threshold interrupt Disabled.

		1 = Receive threshold interrupt Enabled.
[1]	**TX_CLR**	**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]	**RX_CLR**	**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
TX_FIFO_COUNT				**TX_FULL**	**TX_EMPTY**	**RX_FULL**	**RX_EMPTY**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**TIMEOUT**	**Reserved**			**IF**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RX_FIFO_COUNT				**SLV_START _INTSTS**	**Reserved**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**TX_INTSTS**	**Reserved**	**RX_ OVERRUN**	**Reserved**	**RX_INTSTS**

Bits	Description	
[31:28]	**TX_FIFO_COUNT[3:0]**	**Transmit FIFO Data Count (Read Only)** Indicates the valid data count of transmit FIFO buffer.
[27]	**TX_FULL**	**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_CNTRL[27].
[26]	**TX_EMPTY**	**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_CNTRL[26].
[25]	**RX_FULL**	**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_CNTRL[25].
[24]	**RX_EMPTY**	**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_CNTRL[24].
[23:21]	**Reserved**	Reserved.
[20]	**TIMEOUT**	**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]	**IF**	**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_CNTRL[16].
[15:12]	**RX_FIFO_COUNT[3:0]**	**Receive FIFO Data Count (Read Only)** Indicates the valid data count of receive FIFO buffer.
[11]	**SLV_START_INTSTS**	**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 SSTA_INTEN bit was set to 1. 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_CNTRL2[11].
[10:5]	**Reserved**	Reserved.
[4]	**TX_INTSTS**	**Transmit FIFO Threshold Interrupt Status (Read Only)** 0 = The valid data count within the transmit FIFO buffer is larger than the setting value of TX_THRESHOLD. 1 = The valid data count within the transmit FIFO buffer is less than or equal to the setting value of TX_THRESHOLD. **Note:** If TX_INTEN = 1 and TX_INTSTS = 1, the SPI controller will generate a SPI interrupt request.
[3]	**Reserved**	Reserved.
[2]	**RX_OVERRUN**	**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]	**RX_INTSTS**	**Receive FIFO Threshold Interrupt Status (Read Only)** 0 = The valid data count within the Rx FIFO buffer is smaller than or equal to the setting value of RX_THRESHOLD. 1 = The valid data count within the receive FIFO buffer is larger than the setting value of RX_THRESHOLD. **Note:** If RX_INTEN = 1 and RX_INTSTS = 1, the SPI controller will generate a SPI interrupt request.

6.12 Analog-to-Digital Converter (ADC)

6.12.1 Overview

The NuMicro™ Mini51 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.

6.12.2 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
- 300 KSPS (AV_{DD} 4.5V - 5.5V) and 200 KSPS (AV_{DD} 2.5V - 5.5V) conversion rate
- An A/D conversion is performed one time on a specified channel
- An A/D conversion can be started by:
  - Software write 1 to ADST 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

6.12.3 Block Diagram

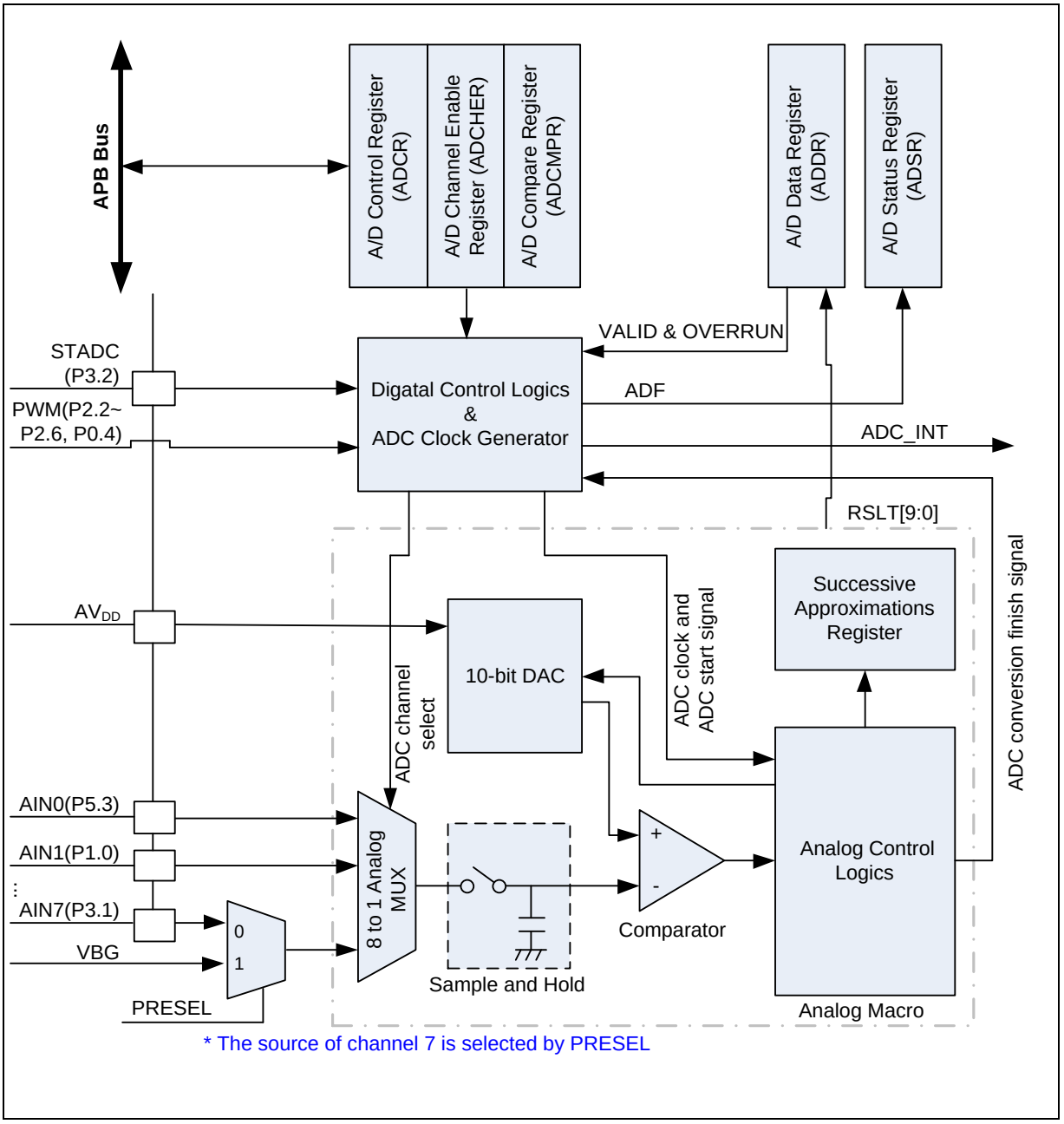


Figure 6.12-1 A/D Controller Block Diagram

6.12.4 Basic Configuration

The ADC pin functions are configured in P1_MFP and 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_OFFD and P3_OFFD register.

The ADC peripheral clock can be enabled in ADC_EN (APBCLK[28]). The ADC peripheral clock source is selected by ADC_S (CLKSEL1[3:2]). The clock prescalar is determined by ADC_N (CLKDIV[23:16]).

6.12.5 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 ADST (ADCR[11]) bit to 0 in the ADCR register. The A/D converter discards the current conversion immediately and enters idle state while ADST bit is cleared.

6.12.5.1 ADC Peripheral Clock Generator

The ADC engine has four clock sources selected by ADC_S (CLKSEL1[3:2]), and selected between HXT and LXT by XTLCLK_EN (PWRCON[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) / (ADC_N+1); where the 8-bit ADC_N is located in register CLKDIV[23:16].

In general, software can set ADC_S and ADC_N to get 6 MHz or slightly less.

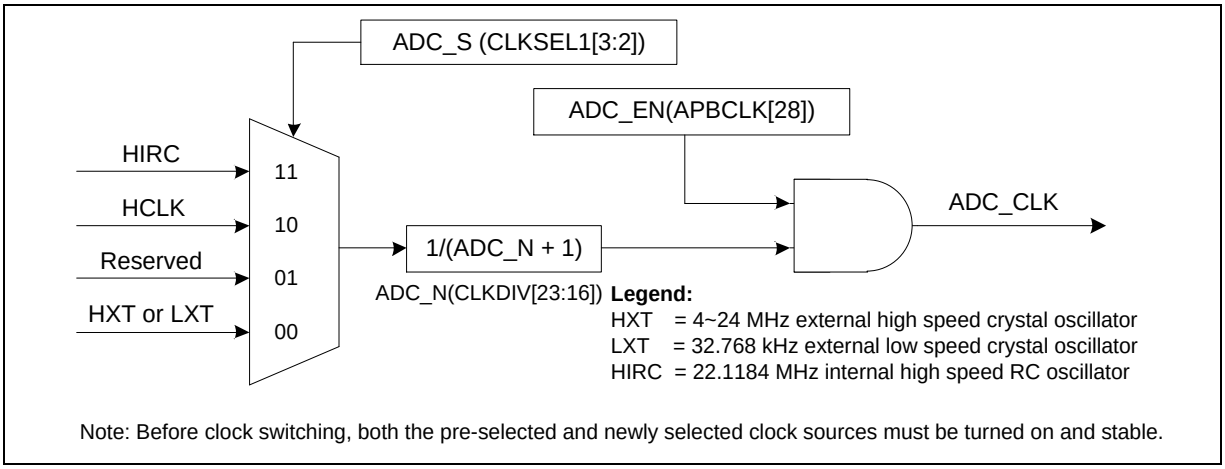


Figure 6.12-2 ADC Peripheral Clock Control

6.12.5.2 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 ADST bit of ADCR 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 ADF (ADSR[0]) bit will be set to 1. If the ADIE (ADCR[1]) bit is set to 1, the ADC interrupt will be asserted.
4. The ADST bit remains 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters idle state.

An example of timing diagram of ADC conversion is shown below.

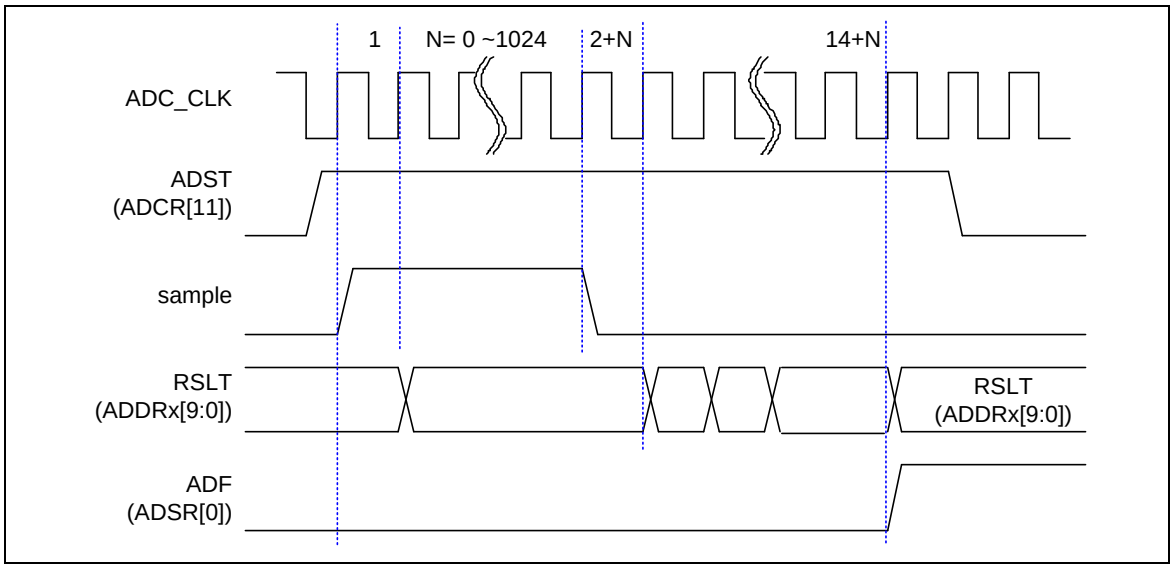


Figure 6.12-3 Single Mode Conversion Timing Diagram

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.

6.12.5.3 External Trigger Input Sampling and A/D Conversion Time

A/D conversion can be triggered by external pin request. When the TRGEN (ADCR[8]) bit is set to 1 to enable ADC external trigger function, setting the TRGS (ADCT[5:4] bits to 00b is to select external trigger input from the STADC pin. Software can set TRGCOND (ADCR[6]) 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 PLCKs. Pulse that is shorter than this specification will be ignored.

6.12.5.4 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.

$AV_{DD} = ((2 \wedge N) / R) * V_{BG}$

N: ADC resolution

R: A/D conversion result

V_{BG}: Band-gap voltage

The block diagram is shown as Figure 6.12-4.

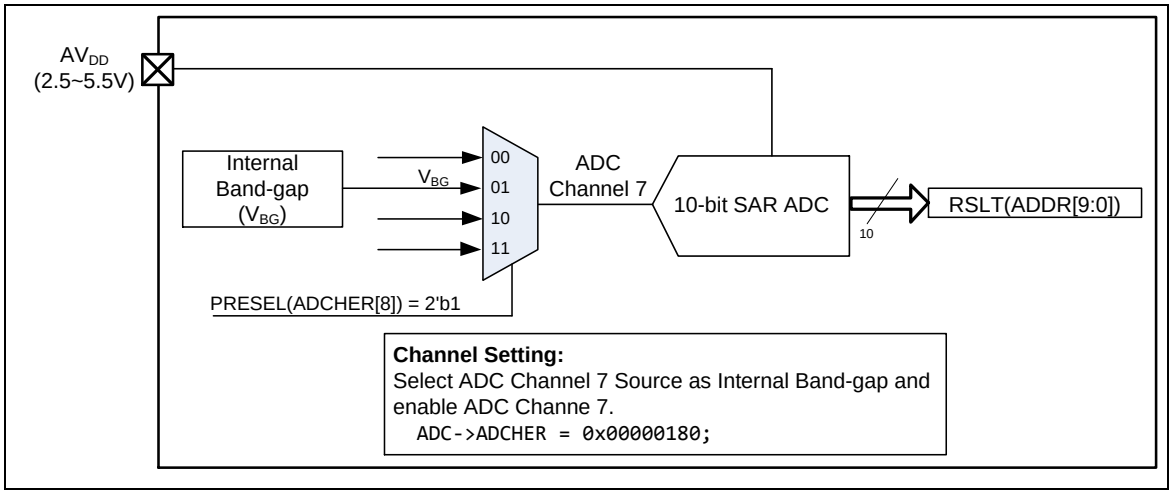


Figure 6.12-4 V_{BG} for Measuring AV_{DD} Application Block Diagram

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

6.12.5.5 PWM trigger

A/D conversion can also be triggered by PWM request. When the TRGEN is set to high to enable ADC external hardware trigger function, setting the TRGS bits to 11b is to select external hardware trigger input source from PWM trigger. When PWM trigger is enabled, setting PTDT (ADTDCR[7:0]) bits can insert a delay time between PWM trigger condition and ADC start conversion.

6.12.5.6 Conversion Result Monitor by Compare Mode Function

The NuMicro™ Mini51™ series ADC controller provides two compare registers, ADCMPR0 and ADCMPR1, to monitor maximum two specified channels. Software can select which channel to be monitored by setting CMPCH (ADCMPRx[5:3]). CMPCOND (ADCMPRx[2]) bit is used to determine the compare condition. If CMPCOND bit is cleared to 0, the internal match counter will increase one when the conversion result is less than the value specified in CMPD (ADCMPR[9:0]); if CMPCOND bit is set to 1, the internal match counter will increase one when the conversion result is greater than or equal to the value specified in CMPD(ADCMPR[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 (CMPMATCNT (ADCMPRx[11:8])+1) then CMPF bit will be set to 1, if CMPIE (ADCMPRx[1]) 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. The detailed logic diagram is shown below.

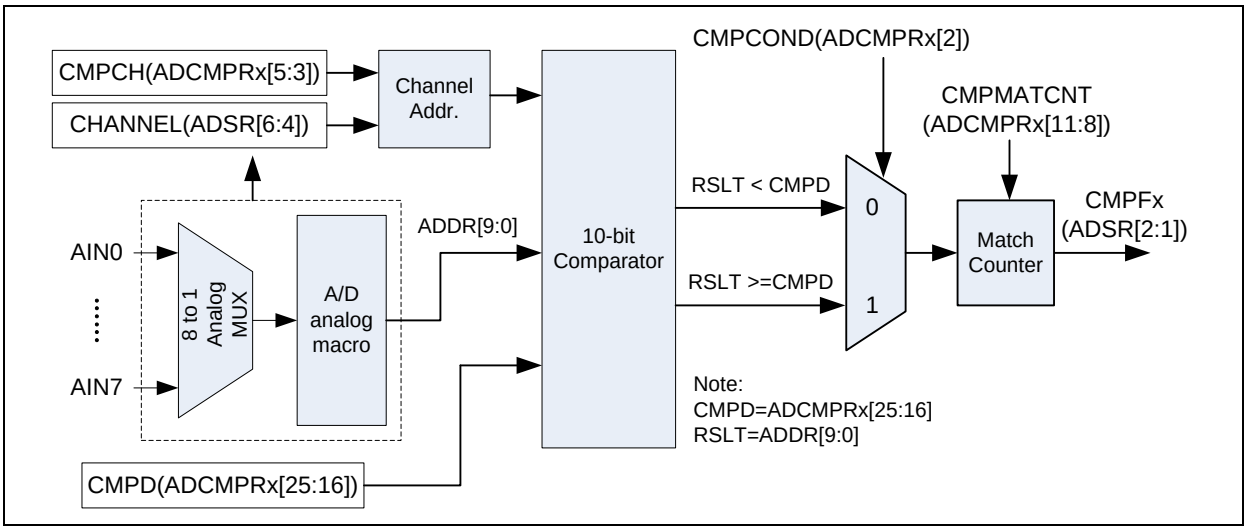


Figure 6.12-5 A/D Conversion Result Monitor Logics Diagram

6.12.5.7 Interrupt Sources

There are three interrupt sources of ADC interrupt. When an ADC operation mode finishes its conversion, the A/D conversion end flag, ADF, will be set to 1. The CMPF0 (ADSR[1]) and CMPF1 (ADCR[2]) are the compare flags of compare function. When the conversion result meets the settings of ADCMPR0/1, the corresponding flag will be set to 1. When one of the flags, ADF, CMPF0 and CMPF1, is set to 1 and the corresponding interrupt enable bit, ADIE of ADCR and CMPIE of ADCMPR0/1, is set to 1, the ADC interrupt will be asserted. Software can clear these flags to revoke the interrupt request.

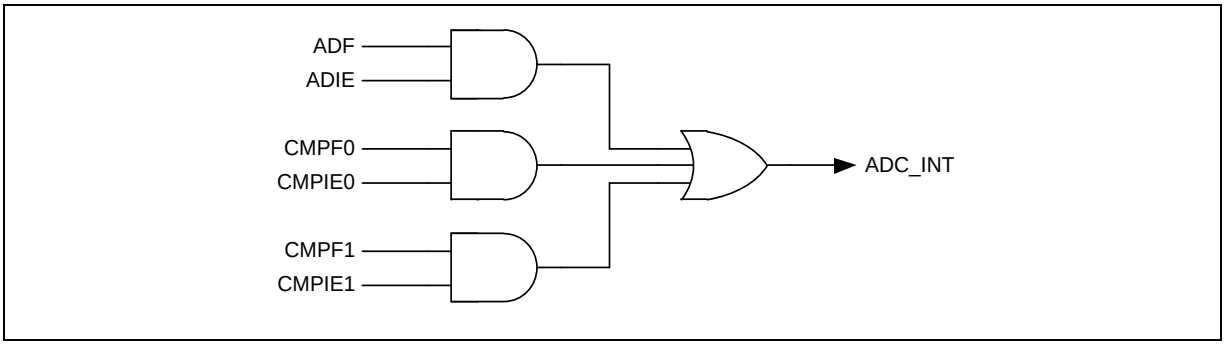


Figure 6.12-6 A/D Controller Interrupt

6.12.6 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**				
ADDR	ADC_BA+0x00	R	ADC Data Register	0x0000_0000
ADCR	ADC_BA+0x20	R/W	ADC Control Register	0x0000_0000
ADCHER	ADC_BA+0x24	R/W	ADC Channel Enable Control Register	0x0000_0000
ADCMPR0	ADC_BA+0x28	R/W	ADC Compare Register 0	0x0000_0000
ADCMPR1	ADC_BA+0x2C	R/W	ADC Compare Register 1	0x0000_0000
ADSR	ADC_BA+0x30	R/W	ADC Status Register	0x0000_0000
ADTDCR	ADC_BA+0x44	R/W	ADC Trigger Delay Control Register	0x0000_0000
ADSAMP	ADC_BA+0x48	R/W	ADC Sampling Time Counter Register	0x0000_0000

6.12.7 Register Description

ADC Data Registers (ADDR)

Register	Offset	R/W	Description	Reset Value
ADDR	ADC_BA+0x00	R	ADC Data Register	0x0000_0000

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

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**VALID**	**Valid Flag** 0 = Data in RSLT (ADDR[9:0]) bits not valid. 1 = Data in RSLT (ADDR[9:0]) bits valid. This bit is set to 1 when ADC conversion is completed and cleared by hardware after the ADDR register is read.
[16]	**OVERRUN**	**Over Run Flag** 0 = Data in RSLT (ADDR[9:0])is recent conversion result. 1 = Data in RSLT (ADDR[9:0])overwrote. If converted data in RSLT[9:0] has not been read before the new conversion result is loaded to this register, OVERRUN is set to 1. It is cleared by hardware after the ADDR register is read.
[15:10]	**Reserved**	Reserved.
[9:0]	**RSLT[9:0]**	**A/D Conversion Result** This field contains conversion result of ADC.

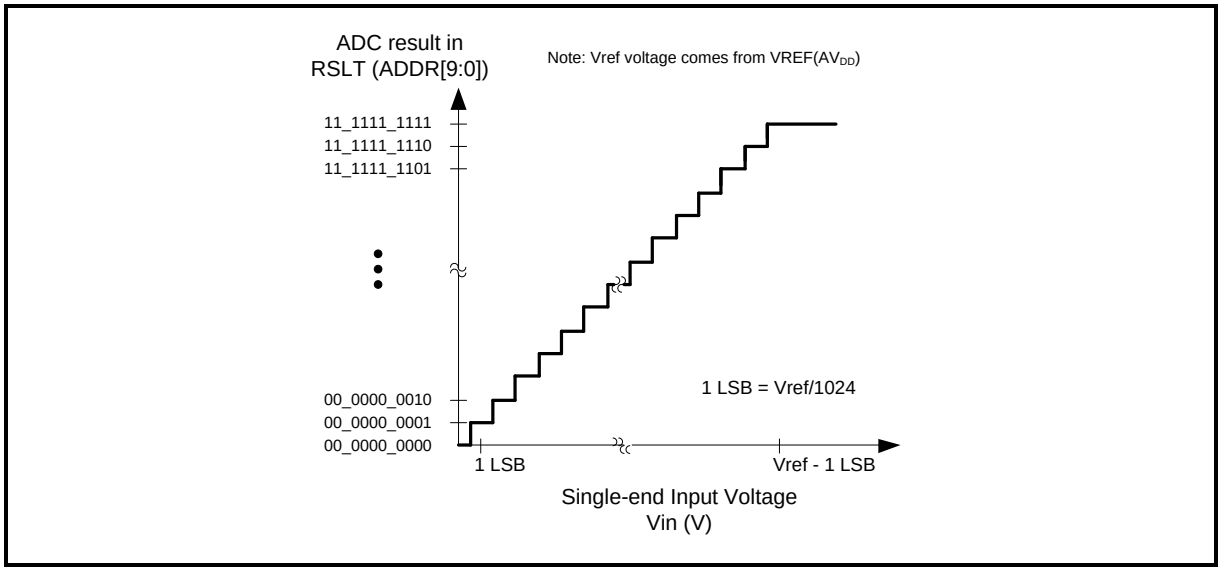


Figure 6.12-7 Conversion Result Mapping Diagram of ADC Single-end Input

ADC Control Register (ADCR)

Register	Offset	R/W	Description	Reset Value
ADCR	ADC_BA+0x20	R/W	ADC 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				ADST	Reserved		TRGEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	TRGCOND	TRGS		Reserved		ADIE	ADEN

Bits	Description	
[31:12]	**Reserved**	Reserved.
[11]	**ADST**	**A/D Conversion Start** ADST bit can be set to 1 from three sources: software or PWM trigger and external pin STADC. ADST 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]	**TRGEN**	**External Trigger Enable Control** Enable or disable triggering of A/D conversion by external STADC pin. If external trigger is enabled, the ADST bit can be set to 1 by the selected hardware trigger source. 0= External trigger Disabled. 1= External trigger Enabled.
[7]	**Reserved**	Reserved.
[6]	**TRGCOND**	**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]	**TRGS[1:0]**	**Hardware Trigger Source** 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 ADST before change TRGS.
[3:2]	**Reserved**	Reserved.
[1]	**ADIE**	**A/D Interrupt Enable Control** A/D conversion end interrupt request is generated if ADIE bit is set to 1.

Bits	Description	
		0 = A/D interrupt function Disabled. 1 = A/D interrupt function Enabled.
[0]	**ADEN**	**A/D Converter Enable Control** 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 Control Register (ADCHER)

Register	Offset	R/W	Description	Reset Value
ADCHER	ADC_BA+0x24	R/W	ADC Channel Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**PRESEL**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHEN7	**CHEN6**	**CHEN5**	**CHEN4**	**CHEN3**	**CHEN2**	**CHEN1**	**CHEN0**

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**PRESEL**	**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 Control** 0 = Channel 7 Disabled. 1 = Channel 7 Enabled.
[6]	**CHEN6**	**Analog Input Channel 6 Enable Control** 0 = Channel 6 Disabled. 1 = Channel 6 Enabled.
[5]	**CHEN5**	**Analog Input Channel 5 Enable Control** 0 = Channel 5 Disabled. 1 = Channel 5 Enabled.
[4]	**CHEN4**	**Analog Input Channel 4 Enable Control** 0 = Channel 4 Disabled. 1 = Channel 4 Enabled.
[3]	**CHEN3**	**Analog Input Channel 3 Enable Control** 0 = Channel 3 Disabled. 1 = Channel 3 Enabled.
[2]	**CHEN2**	**Analog Input Channel 2 Enable Control** 0 = Channel 2 Disabled. 1 = Channel 2 Enabled.
[1]	**CHEN1**	**Analog Input Channel 1 Enable Control**

Bits	Description	
		0 = Channel 1 Disabled. 1 = Channel 1 Enabled.
[0]	**CHEN0**	**Analog Input Channel 0 Enable Control** 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.

ADC Compare Register 0/1 (ADCMPR0/1)

Register	Offset	R/W	Description	Reset Value
ADCMPR0	ADC_BA+0x28	R/W	ADC Compare Register 0	0x0000_0000
ADCMPR1	ADC_BA+0x2C	R/W	ADC Compare Register 1	0x0000_0000

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

Bits	Description	
[31:26]	**Reserved**	Reserved.
[25:16]	**CMPD[9:0]**	**Comparison Data** The 10-bit data is used to compare with conversion result of specified channel.
[15:12]	**Reserved**	Reserved.
[11:8]	**CMPMATCNT [3:0]**	**Compare Match Count** When the specified A/D channel analog conversion result matches the compare condition defined by CMPCOND, the internal match counter will increase 1. When the internal counter reaches the value to (CMPMATCNT +1), the CMPFx bit will be set.
[7:6]	**Reserved**	Reserved.
[5:3]	**CMPCH[2:0]**	**Compare Channel Selection** 000 = Channel 0 conversion result is selected to be compared. 001 = Channel 1 conversion result is selected to be compared. 010 = Channel 2 conversion result is selected to be compared. 011 = Channel 3 conversion result is selected to be compared. 100 = Channel 4 conversion result is selected to be compared. 101 = Channel 5 conversion result is selected to be compared. 110 = Channel 6 conversion result is selected to be compared. 111 = Channel 7 conversion result is selected to be compared.
[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 CMPD (ADCMPRx[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 CMPD (ADCMPRx[25:16]), the internal match counter will increase one. **Note:** When the internal counter reaches the value to (CMPMATCNT +1), the CMPFx bit will be set.

Bits	Description	
[1]	**CMPIE**	**Compare Interrupt Enable Control** If the compare function is enabled and the compare condition matches the setting of CMPCOND and CMPMATCNT, CMPFx bit will be asserted, in the meanwhile, if CMPIE is set to 1, a compare interrupt request is generated. 0 = Compare function interrupt Disabled. 1 = Compare function interrupt Enabled.
[0]	**CMPEN**	**Compare Enable Control** Set 1 to this bit to enable comparing CMPD[9:0] with specified channel conversion results when converted data is loaded into the ADDR register. 0 = Compare function Disabled. 1 = Compare function Enabled.

ADC Status Register (ADSR)

Register	Offset	R/W	Description	Reset Value
ADSR	ADC_BA+0x30	R/W	ADC Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**OVERRUN**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**VALID**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**CHANNEL**			**BUSY**	**CMPF1**	**CMPF0**	**ADF**

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**OVERRUN**	**Overrun Flag (Read Only)** It is a mirror to OVERRUN (ADSR[16]) bit in ADDR register.
[15:9]	**Reserved**	Reserved.
[8]	**VALID**	**Data Valid Flag (Read Only)** It is a mirror of VALID (ADDR[17]) bit in ADDR register.
[7]	**Reserved**	Reserved.
[6:4]	**CHANNEL [2:0]**	**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 ADST bit in ADCR 0 = A/D converter is in idle state. 1 = A/D converter is busy at conversion.
[2]	**CMPF1**	**Compare Flag 1** When the selected channel A/D conversion result meets the setting condition in ADCMPR1, this bit is set to 1. Software can write 1 to clear this bit to 0. 0 = Conversion result in ADDR does not meet the ADCMPR1 setting. 1 = Conversion result in ADDR meets the ADCMPR1 setting.
[1]	**CMPF0**	**Compare Flag 0** When the selected channel A/D conversion result meets the setting condition in ADCMPR0, this bit is set to 1. Software can write 1 to clear this bit to 0. 0 = Conversion result in ADDR does not meet the ADCMPR0 setting. 1 = Conversion result in ADDR meets the ADCMPR0 setting.
[0]	**ADF**	**A/D Conversion End Flag** A status flag that indicates the end of A/D conversion. ADF is set to 1 When A/D conversion

Bits	Description	
		ends. Software can write 1 to clear this bit to 0.

ADC Trigger Delay Controller Register (ADTDCR)

Register	Offset	R/W	Description	Reset Value
ADTDCR	ADC_BA+0x44	R/W	ADC Trigger Delay Control Register	0x0000_0000

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

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**PTDT[7:0]**	**PWM Trigger Delay Timer** Set this field will delay ADC start conversion time after PWM trigger. PWM trigger delay time is (4 * PTDT) * system clock.

ADC Sampling Register (ADSAMP)

Register	Offset	R/W	Description	Reset Value
ADSAMP	ADC_BA+0x48	R/W	ADC 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				**ADSAMPCNT**			

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3:0]	**ADSAMPCNT [3:0]**	**ADC Sampling Counter** 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 clock. The additional clock number will be inserted to lengthen the sampling clock. 0000 = 0 additional ADC sample clock. 0001 = 1 additional ADC sample clock. 0010 = 2 additional ADC sample clock. 0011 = 4 additional ADC sample clock. 0100 = 8 additional ADC sample clock. 0101 = 16 additional ADC sample clock. 0110 = 32 additional ADC sample clock. 0111 = 64 additional ADC sample clock. 1000 = 128 additional ADC sample clock. 1001 = 256 additional ADC sample clock. 1010 = 512 additional ADC sample clock. 1011 = 1024 additional ADC sample clock. 1100 = 1024 additional ADC sample clock. 1101 = 1024 additional ADC sample clock. 1110 = 1024 additional ADC sample clock. 1111 = 1024 additional ADC sample clock.

6.13 Analog Comparator (ACMP)

6.13.1 Overview

The NuMicro™ Mini51 Series contains two comparators which can be used in a number of different configurations. The comparator output is logic 1 when positive input greater than negative input, otherwise the output is 0. Each comparator can be configured to generate interrupt when the comparator output value changes.

6.13.2 Features

- Analog input voltage range: 0 ~ AV_{DD}
- Supports Hysteresis function
- Optional internal reference voltage source for each comparator negative input

6.13.3 Block Diagram

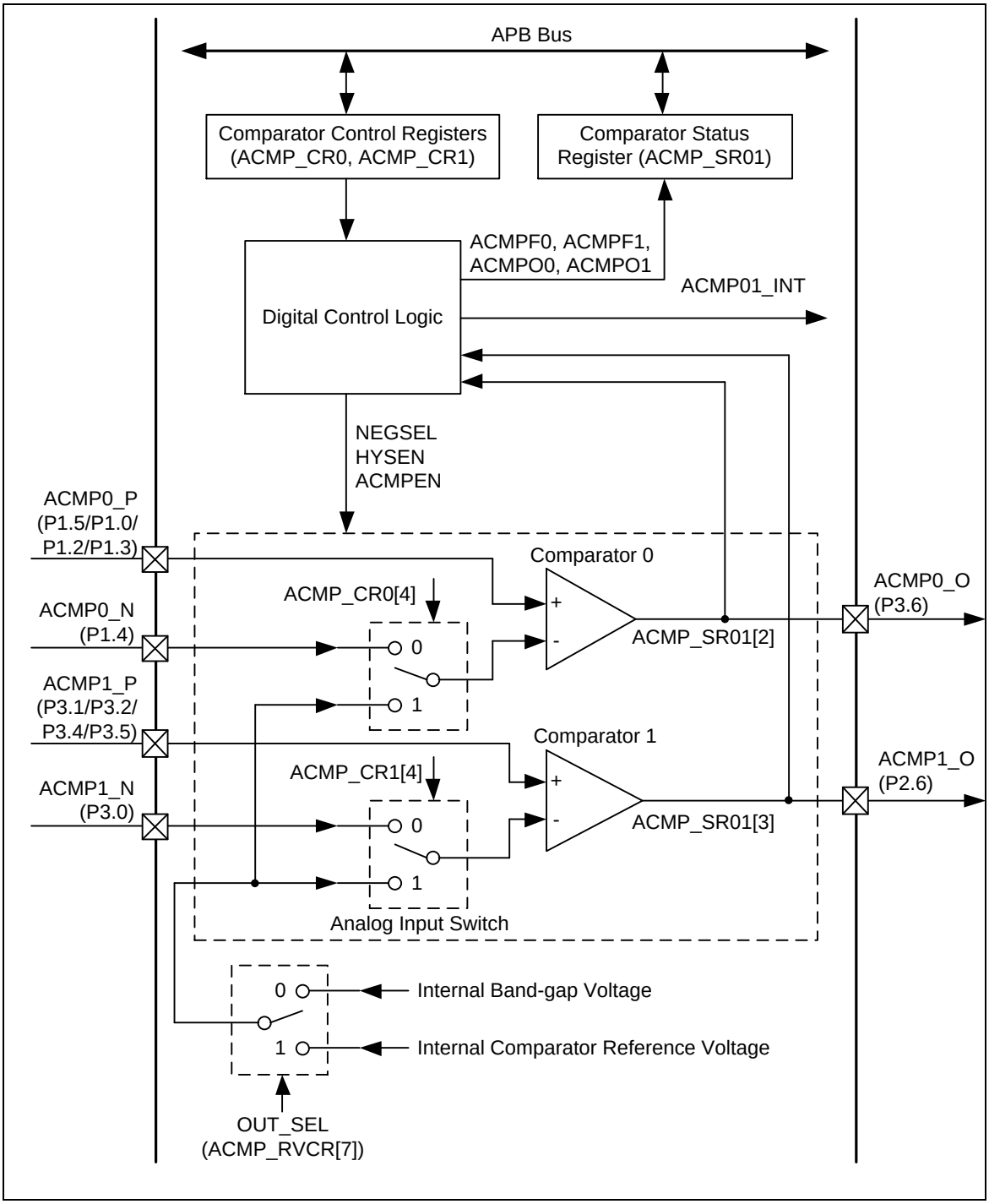


Figure 6.13-1 Analog Comparator Block Diagram

6.13.4 Basic Configuration

The ACMP pin functions are configured in P1_MFP, P2_MFP and 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 P1_OFFD and P3_OFFD registers.

The ACMP peripheral clocks can be enabled by set ACMP_EN(APBCLK[30]) to 1.

6.13.5 Functional Description

6.13.5.1 Interrupt Sources

The output of comparators are sampled by PCLK and reflected at ACMPOx(ACMP_SR01[3] and ACMP_SR01[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_SR01[1] and ACMP_SR01[0]), will be set. Software can clear the flag to 0 by writing 1 to it.

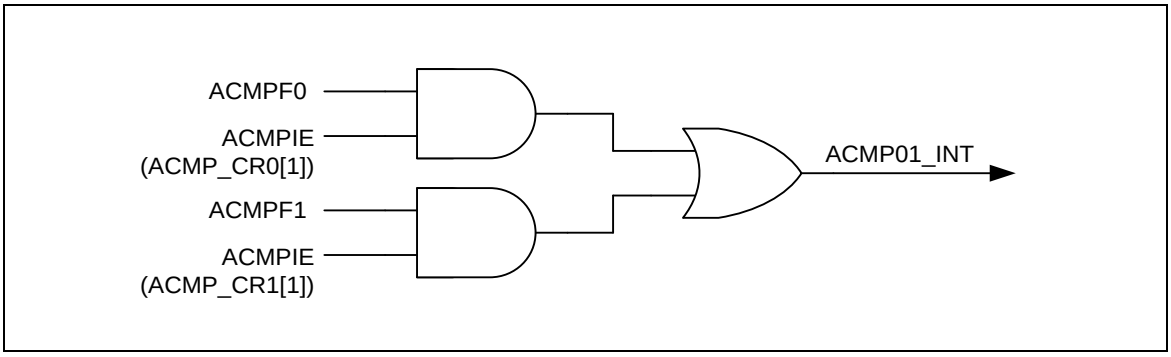


Figure 6.13-2 Analog Comparator Controller Interrupt Sources

6.13.5.2 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.

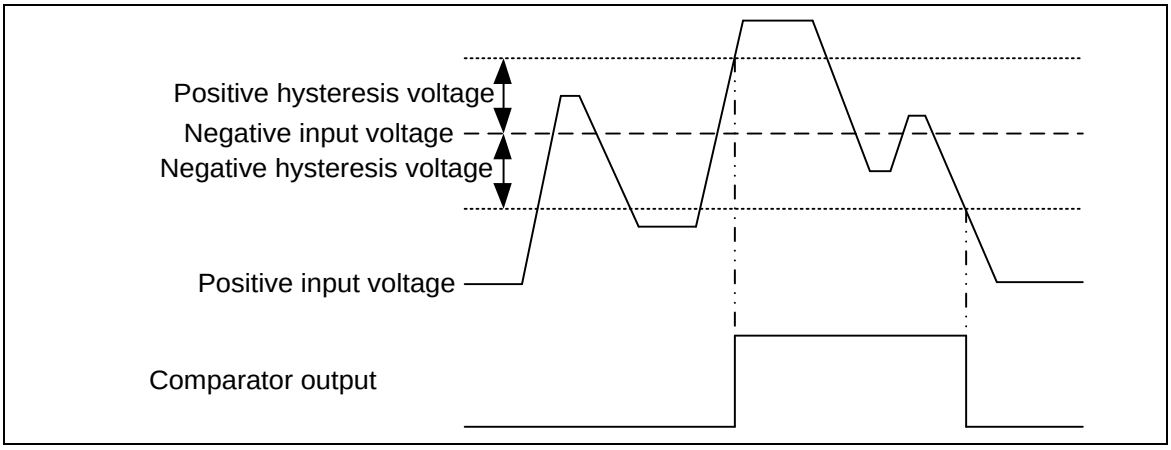


Figure 6.13-3 Comparator Hysteresis Function

6.13.6 Comparator Reference Voltage (CRV)

6.13.6.1 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 CRVS(ACMP_RVCR[3:0]) and select the reference voltage to ACMP by setting OUT_SEL(ACMP_RVCR[7]).

6.13.6.2 Features

- User selectable references voltage by setting CRVS(ACMP_RVCR[3:0])
- Automatic disable resisters ladder for reducing power consumption when setting OUT_SEL(ACMP_RVCR[7]) = 0 (selecting Band-gap source voltage)

The block diagram of the CRV module is shown below:

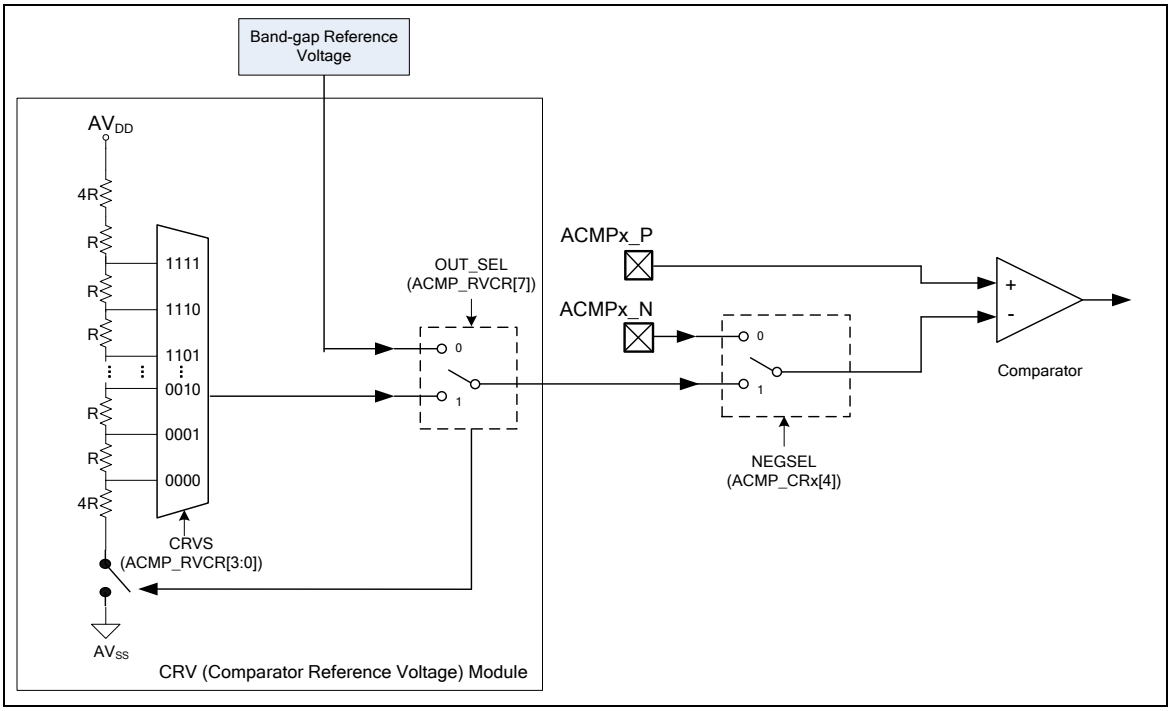


Figure 6.13-4 Comparator Reference Voltage Block Diagram

6.13.7 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_CR0	ACMP_BA+0x00	R/W	Analog Comparator 0 Control Register	0x0000_0000
ACMP_CR1	ACMP_BA+0x04	R/W	Analog Comparator 1 Control Register	0x0000_0000
ACMP_SR01	ACMP_BA+0x08	R/W	Analog Comparator 0/1 Status Register	0x0000_0000
ACMP_RVCR	ACMP_BA+0x0C	R/W	Analog Comparator Reference Voltage Control Register	0x0000_0000

6.13.8 Register Description

Analog Comparator 0 Control Register (ACMP_CR0)

Register	Offset	R/W	Description	Reset Value
ACMP_CR0	ACMP_BA+0x00	R/W	Analog Comparator 0 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	**CPP0SEL**		**Reserved**				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**FALLING**	**RISING**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**NEGSEL**	**Reserved**	**HYSEN**	**ACMPIE**	**ACMPEN**

Bits	Description	
[31]	**Reserved**	Reserved.
[30:29]	**CPP0SEL[1:0]**	**Analog Comparator 0 Positive Input Selection** 00 = CPP0 is from P1.5 pin. 01 = CPP0 is from P1.0 pin. 10 = CPP0 is from P1.2 pin. 11 = CPP0 is from P1.3 pin.
[28:10]	**Reserved**	Reserved.
[9]	**FALLING**	**Analog Comparator 0 Falling Edge Trigger Enable Control** 0 = Analog comparator 0 falling edge trigger Disabled. 1 = Analog comparator 0 falling edge trigger PWM or Timer Enabled. **Note:** The bit is only effective while analog comparator 0 triggers PWM or Timer.
[8]	**RISING**	**Analog Comparator 0 Rising Edge Trigger Enable Control** 0 = Analog comparator 0 rising edge trigger Disabled. 1 = Analog comparator 0 rising edge trigger PWM or Timer Enabled. **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 Selection** 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]	**HYSEN**	**Analog Comparator 0 Hysteresis Enable Control** 0 = Hysteresis function Disabled.

Bits	Description	
		1 = Hysteresis function Enabled.
[1]	**ACMPIE**	**Analog Comparator 0 Interrupt Enable Control** 0 = Interrupt function Disabled. 1 = Interrupt function Enabled.
[0]	**ACMPEN**	**Analog Comparator 0 Enable Control** 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_CR1)

Register	Offset	R/W	Description	Reset Value
ACMP_CR1	ACMP_BA+0x04	R/W	Analog Comparator 1 Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	**CPP1SEL**		**Reserved**				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**FALLING**	**RISING**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**NEGSEL**	**Reserved**	**HYSEN**	**ACMPIE**	**ACMPEN**

Bits	Description	
[31]	**Reserved**	Reserved.
[30:29]	**CPP1SEL[1:0]**	**Analog Comparator 1 Positive Input Selection** 00 = CPP1 is from P3.1 pin. 01 = CPP1 is from P3.2 pin. 10 = CPP1 is from P3.4 pin. 11 = CPP1 is from P3.5 pin.
[28:10]	**Reserved**	Reserved.
[9]	**FALLING**	**Analog Comparator 1 Falling Edge Trigger Enable Control** 0 = Analog comparator 1 falling edge trigger Disabled. 1 = Analog comparator 1 falling edge trigger PWM or Timer Enabled. **Note:** The bit is only effective while analog comparator 1 triggers PWM or Timer.
[8]	**RISING**	**Analog Comparator 1 Rising Edge Trigger Enable Control** 0 = Analog comparator 1 rising edge trigger Disabled. 1 = Analog comparator 1 rising edge trigger PWM or Timer Enabled. **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 Selection** 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]	**HYSEN**	**Analog Comparator 1 Hysteresis Enable Control** 0 = Hysteresis function Disabled. 1 = Hysteresis function Enabled.
[1]	**ACMPIE**	**Analog Comparator 1 Interrupt Enable Control**

Bits	Description	
		0 = Interrupt function Disabled. 1 = Interrupt function Enabled.
[0]	**ACMPEN**	**Analog Comparator 1 Enable Control** 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_SR01)

Register	Offset	R/W	Description	Reset Value
ACMP_SR01	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**	**ACMPF1**	**ACMPF0**

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_CR1[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_CR0[0]) = 0. 0 = Analog comparator 0 outputs 0. 1 = Analog comparator 0 outputs 1.
[1]	**ACMPF1**	**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_CR1[1]) = 1. 0 = Analog comparator 1 output does not change. 1 = Analog comparator 1 output changed. **Note:** Software can write 1 to clear this bit to 0.
[0]	**ACMPF0**	**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_CR0[1]) = 1. 0 = Analog comparator 0 output does not change. 1 = Analog comparator 0 output changed. **Note:** Software can write 1 to clear this bit to 0.

Analog Comparator Reference Voltage Control Register (ACMP_RVCR)

Register	Offset	R/W	Description	Reset Value
ACMP_RVCR	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**
OUT_SEL	**Reserved**			**CRVS**			

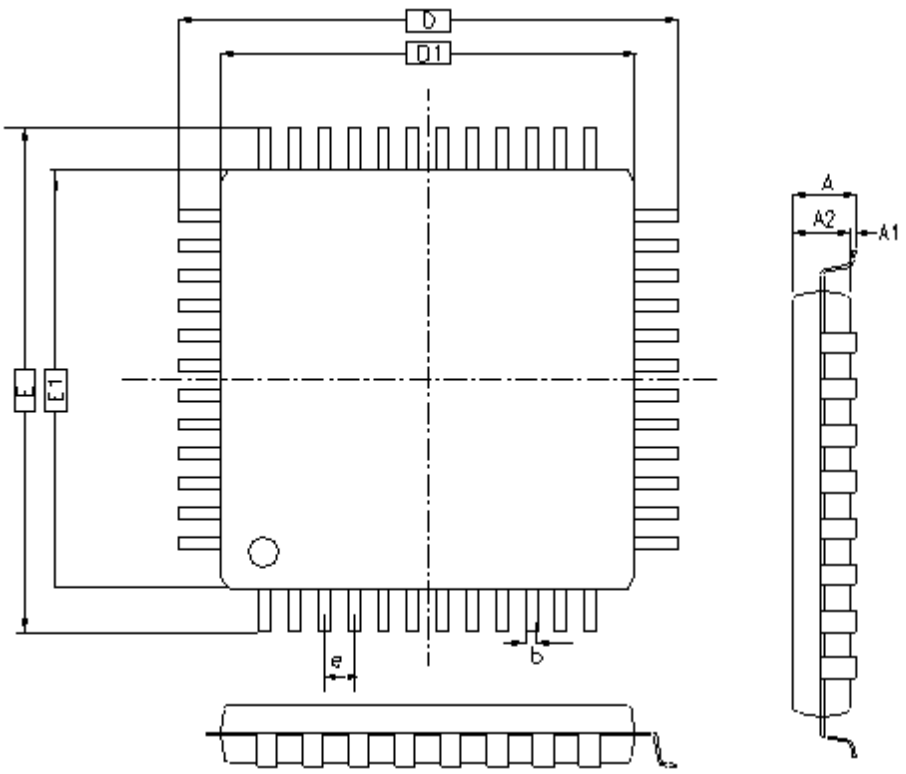
Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**OUT_SEL**	**CRV Module Output Selection** 0 = Band-gap voltage. 1 = Internal comparator reference voltage.
[6:4]	**Reserved**	Reserved.
[3:0]	**CRVS[3:0]**	**Comparator Reference Voltage Setting** Comparator reference voltage = AV_{DD} * (1 / 6 + CRVS[3:0] / 24).

7 ELECTRICAL CHARACTERISTICS

For information on NuMicro Mini51™ series electrical characteristics, please refer to NuMicro™ Mini51 DE Series Datasheet.

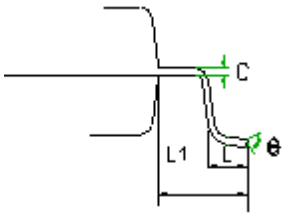
8 PACKAGE DIMENSIONS

8.1 48-pin LQFP

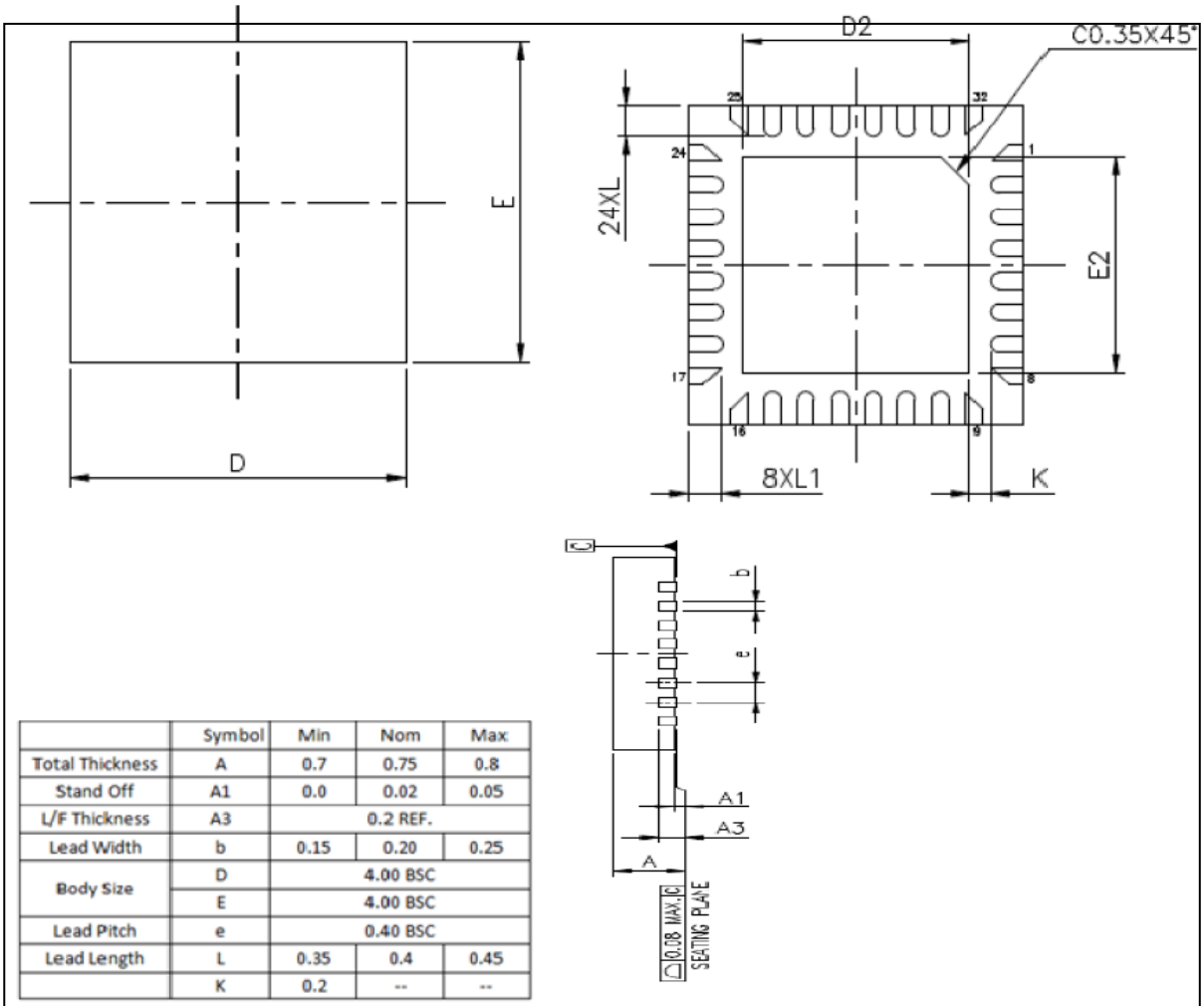


CONTROL DIMENSIONS ARE IN MILLIMETERS

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



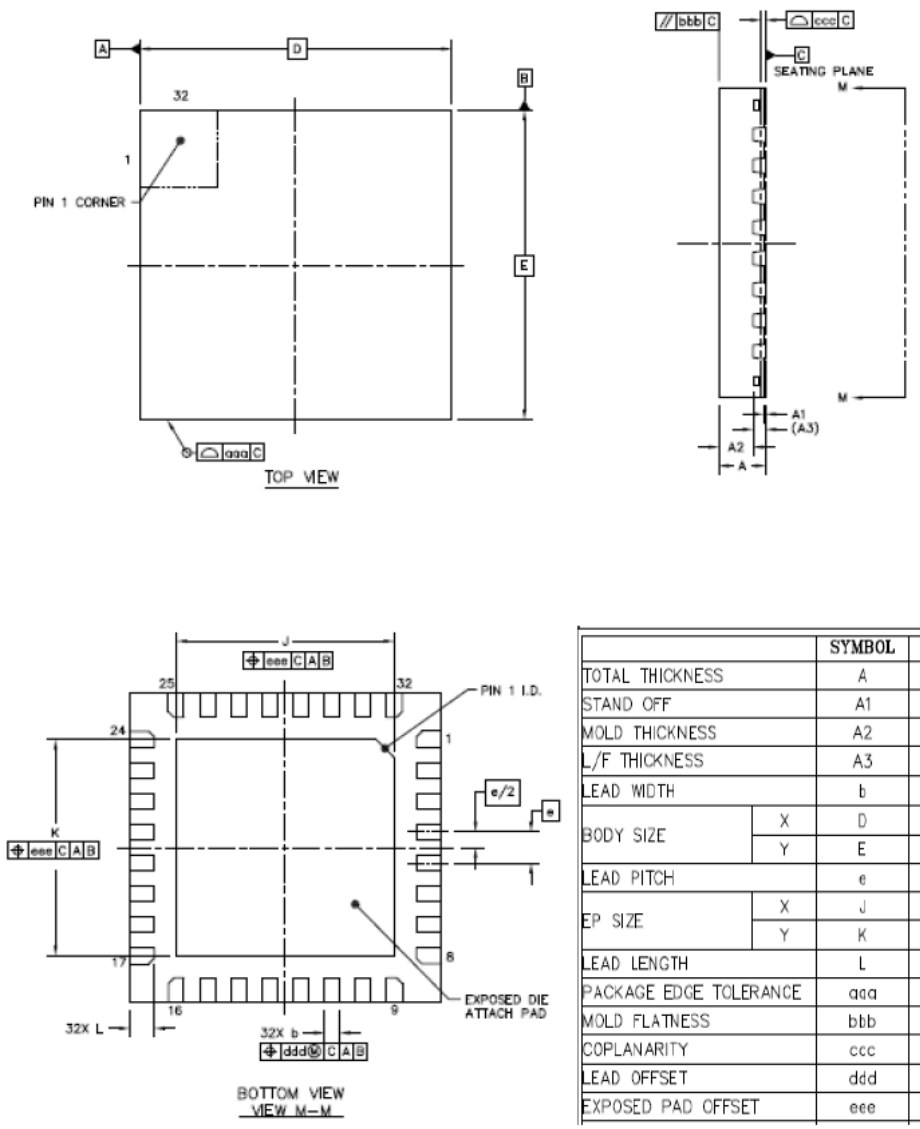
8.2 33-pin QFN (4 mm x 4 mm)



	Symbol	Min	Nom	Max
Total Thickness	A	0.7	0.75	0.8
Stand Off	A1	0.0	0.02	0.05
L/F Thickness	A3		0.2 REF.	
Lead Width	b	0.15	0.20	0.25
Body Size	D		4.00 BSC	
	E		4.00 BSC	
Lead Pitch	e		0.40 BSC	
Lead Length	L	0.35	0.4	0.45
	K	0.2	--	--

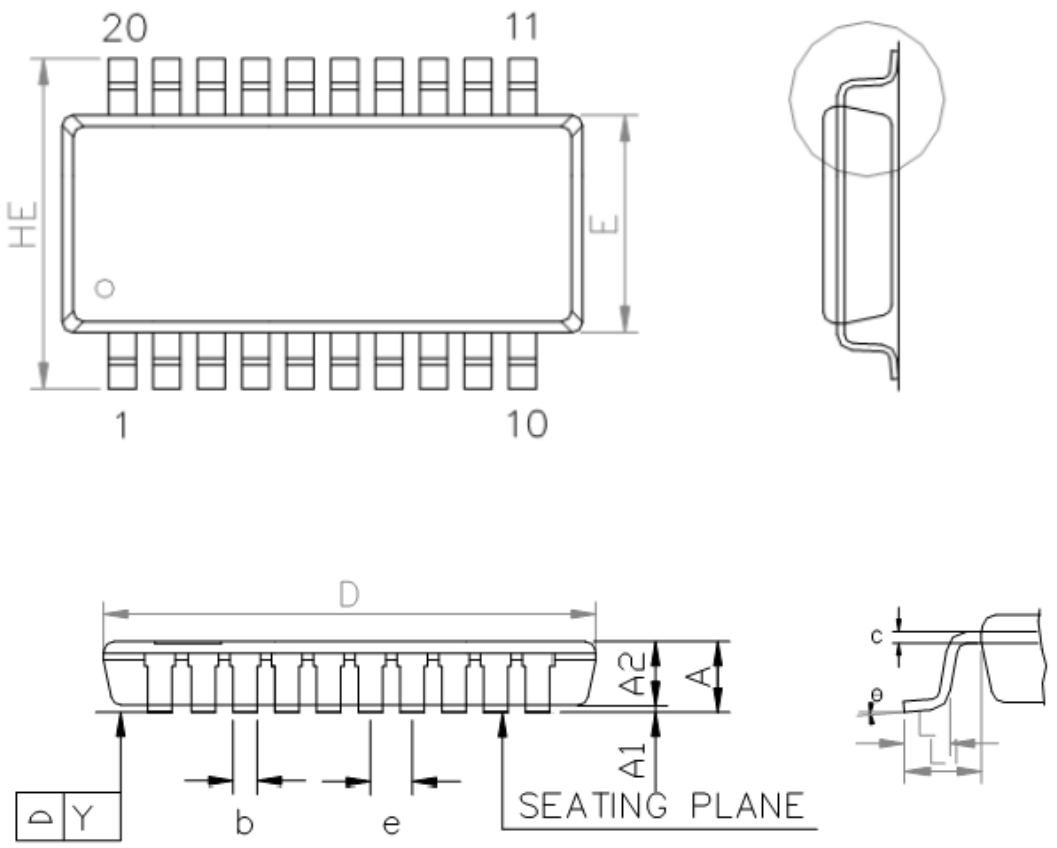
Symbol	L/F SIZE	Dimension in mm		
		MIN.	NOR.	MAX.
D2	114X114 mm	2.6	2.7	2.75
E2		2.6	2.7	2.75

8.3 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		

8.4 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		

9 REVISION HISTORY

Date	Revision	Description
2013.10.18	1.00	Initial version
2014.05.20	1.01	Supported the Mini54FHC for NuMicro Mini51 series.
2015.05.28	1.02	Updated document format
2016.07.18	1.03	1. Added Section 6.2.5 Register Protection. 2. Added Section 6.12.5.4 Internal Reference Voltage.
2020.04.21	1.04	Added notes about the hardware reference design for ICE_DAT, ICE_CLK and /RESET pins in section 4.4.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.