SN32F400 Series

USER'S MANUAL

SN32F407
SN32F405
SN32F403

SONiX 32-Bit Cortex-M0 Micro-Controller

AMENDENT HISTORY

Version	Date	Description
1.0	2024/04/10	First version released.
1.1	2024/05/15	1. Modify typing error of LVD Voltage Detector in 23.2 ELECTRICAL CHARACTERISTIC. 2. Add 15.8.3 I2C TX_BIT Changing Timing. 3. Add ELS operating note in 3.2.3 External Low-speed (ELS) Clock.
1.2	2024/09/23	1. Modify typing error in 8.5.2 ADC OVERRUN. 2. Modify typing error in 10.2 CONFIGURATION OF OPERATION. 3. Modify the figure in 22.2 SN32F400 STARTER-KIT. 4. Modify pin assignment of SN32F407F in 1.4 PIN ASSIGNMENT. 5. Add chip SN32F407J. 6. Add 25.1 QFN 48 PIN 6x6. 7. Modify typing error in 1.2 SYSTEM BLOCK DIAGRAM.
1.3	2024/09/26	1. Modify typing error of Features Selection Table in 1.1 FEATURES. 2. Add note of Features Selection Table in 1.1 FEATURES. 3. Modify typing error of ADC and COMPARATOR in 23.2 ELECTRICAL CHARACTERISTIC. 4. Add note of SN32F403J in 1.4 PIN ASSIGNMENT.
1.4	2024/10/07	1. Modify typing error in 6 PERIPHERAL FUNCTION PIN ASSIGNMENT (PFPA).
1.5	2024/10/16	1. Modify 23.2 ELECTRICAL CHARACTERISTIC. 2. Modify 26.3 MARKING EXAMPLE.
1.6	2024/12/05	1. Add application note of SPI in 14.9 SPI REGISTERS. 2. Add application note of ADC in 8.1 OVERVIEW and 23.2 ELECTRICAL CHARACTERISTIC. 3. Modify typing error of ADC in 8.11.1 ADC Management register (ADC_ADM)
1.7	2025/02/24	1. Add notice in 11.10.3 CT16Bn Prescale register (CT16Bn_PRE) (n=0,1,5) and 11.10.5 CT16Bn Count Control register (CT16Bn_CNTCTRL) (n=0,1,5).
1.8	2025/03/31	1. Add notice in 4.3.2 DEEP-SLEEP MODE
1.9	2025/04/02	1. Modify notice in 4.3.2 DEEP-SLEEP MODE
2.0	2025/05/09	1. Modify typing error in 14.9.12 SPIn ARGB CTRL register (SPIn_ARGBCTRL) (n=0). 2. Update 14.6 ARGB MODE description.

Table of Content

1 PRODUCT OVERVIEW

1.1 FEATURES

- **Memory configuration**
  32KB on-chip Flash programming memory.
  8KB SRAM.
  4KB Boot ROM

- **Operation Frequency up to 60MHz**

- **Interrupt sources**
  ARM® Cortex®-M0 built-in Nested Vectored Interrupt Controller (NVIC).

- **I/O pin configuration**
  Up to 46 General Purpose I/O (GPIO) pins with configurable pull-up/pull-down resistors.
  GPIO pins can be used as edge and level sensitive interrupt sources.
  All IO with typical 30mA driving/sinking current.

- **Programmable Watchdog Timer (WDT)**
  Programmable watchdog frequency with watchdog clock source and divider.

- **System tick timer**
  The 24-bit SysTick timer clock source is fixed to the system clock, and is intended to generate a fixed 10-ms interrupt.

- **Real-Time Clock (RTC)**

- **LVD with separate thresholds**
  Reset: 2.1V/2.5V/2.9V/3.3V/3.7V/4.1V/4.5V for VDD
  Interrupt: 2.1V/2.5V/2.9V/3.3V/3.7V/4.1V/4.5V for VDD

- **Fcpu (Instruction cycle)**
  F_{CPU} = F_{HCLK} = F_{SYSCLK}/1, F_{SYSCLK} /2, F_{SYSCLK} /4, …, F_{SYSCLK} /128.

- **Operating modes**
  Normal, Sleep, Deep-sleep

- **Cyclic Redundancy Check (CRC)**
  CRC-16
  CRC-16-CCCITT
  CRC-32

- **5-channel DMA controller**

- **PFPA**

- **Serial Wire Debug (SWD)**

- **In-Circuit Programming (ICP) supported**

- **Timer**
  3 16-bit timer support up-counting, down-counting, and center-aligned mode.
  4 sets inverse PWM with programmable dead-band
  PCLK up to 120MHz

- **Working voltage 1.8V ~ 5.5V**

- **16+6 CH 12-bit SAR ADC**
  16 external ADC input
  1 internal battery measurement
  4-level internal reference voltage source (VDD, 4.5V, 3V, 2V, 1.5V)
  22-frame ADB FIFO
  Single/Continuous mode
  Single/Multi-channel auto conversion

- **3 OPA**
  1X~32X PGA

- **4 Rail to Rail Comparators**
  Internal 8-bit DAC reference voltage source
  Programmable 255-level internal ref. voltage divider
  3 external negative inputs
  3 external positive inputs

- **FOC**
  Flux, speed, current PI controllers
  Park/ipark transformers
  Clark/iclark transformers
  SVPWM

- **ACC**
  Calculation accelerator
  Include arctangent, division, and square root.

- **Interface**
  -One I2C controllers supporting I2C-bus specification with multiple address recognition.
  -Two UART controllers with fractional baud rate generation.
  -One SPI controller.

- **System clocks**
  -External high clock: Crystal type 4MHz~25MHz
  -External low clock: Crystal type 32.768 KHz
  -Internal high clock: RC type 48MHz
  -Internal low clock: RC type 32 KHz
  -PLL allows CPU operation up to the maximum CPU rate without the need for a high-frequency crystal.
  -Clock output function which can reflect the internal high/low RC oscillator, HCLK, PLL output, and external low clock.

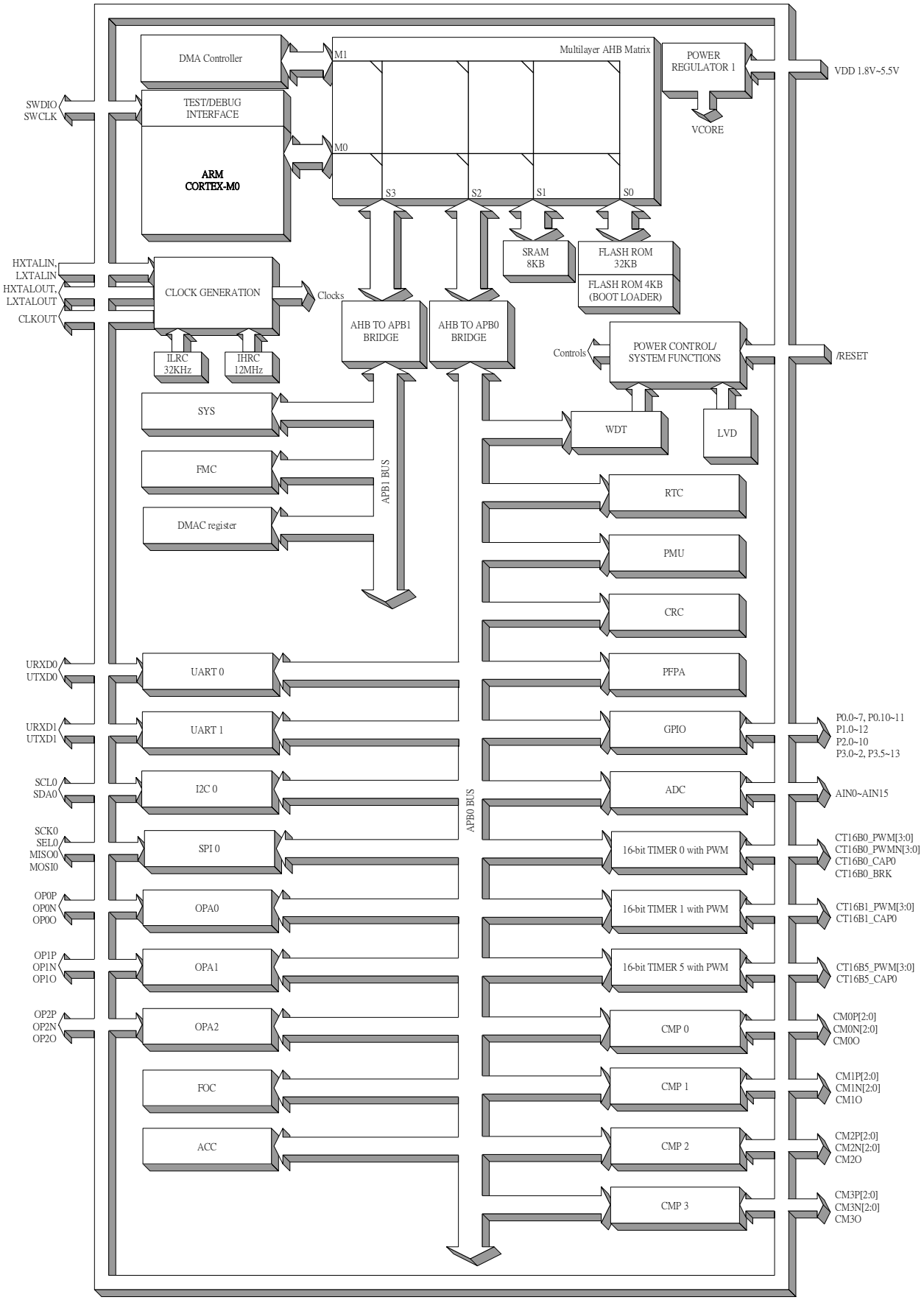
- **Package (Chip form support)**
  LQFP48/32 pin
  QFN48/32/24 pin

☞ **Features Selection Table**

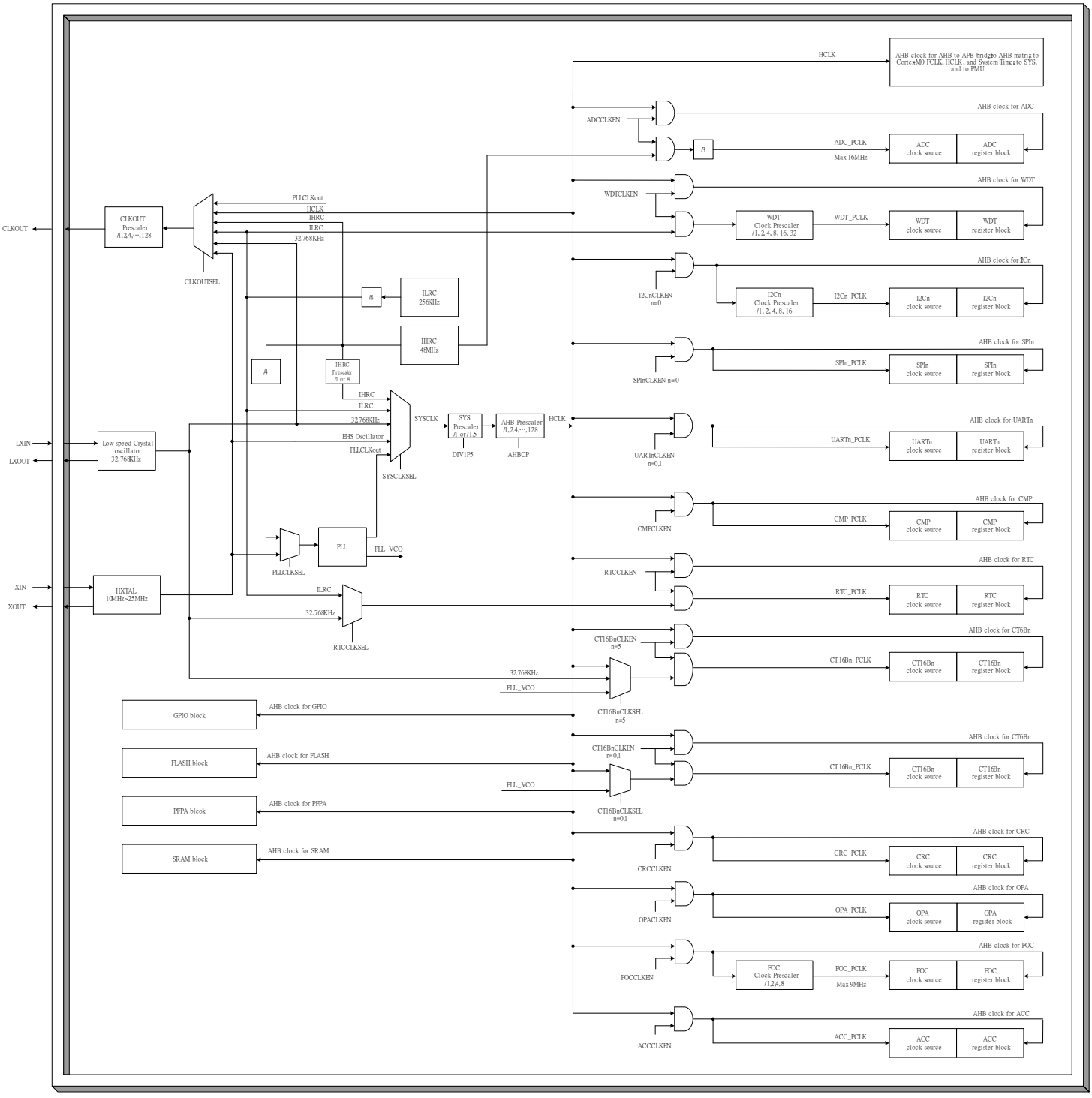
Chip	ROM (KB)	RAM (KB)	Fcpu. (Max MHz)	TIMER	UART	I2C	SPI	PWM	12-bit ADC	CMP	OPA	GPIO with Wakeup	PKG
SN32F407F/J	32	8	60	24-bit x 1 16-bit x 3	2	1	1	12+4	16+6	4	3	46	QFN48/ LQFP48
SN32F405F/J	32	8	60	24-bit x 1 16-bit x 3	2	1	1	12+4	16+6	4	3	30	QFN32/ LQFP32
SN32F403J	32	8	60	24-bit x 1 16-bit x 3	2	1	1	12+4	11+6	4	*1	23	QFN24

✸ ***Note: 1. The chip "SN32F403J" does not have OP0O and OP1O, but the OP0O and OP1O can be the internal channels of the ADC.***

1.2 SYSTEM BLOCK DIAGRAM

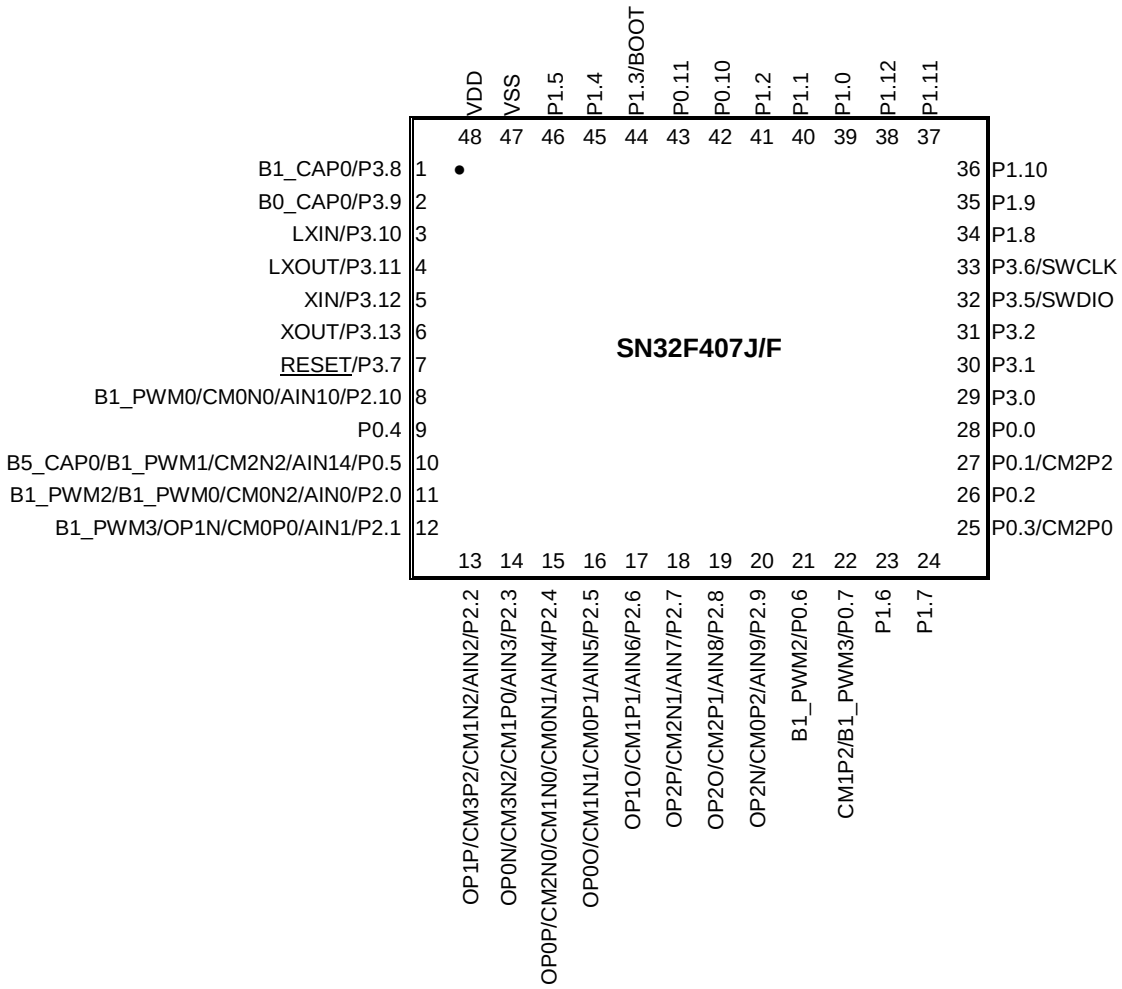


1.3 CLOCK GENERATION BLOCK DIAGRAM



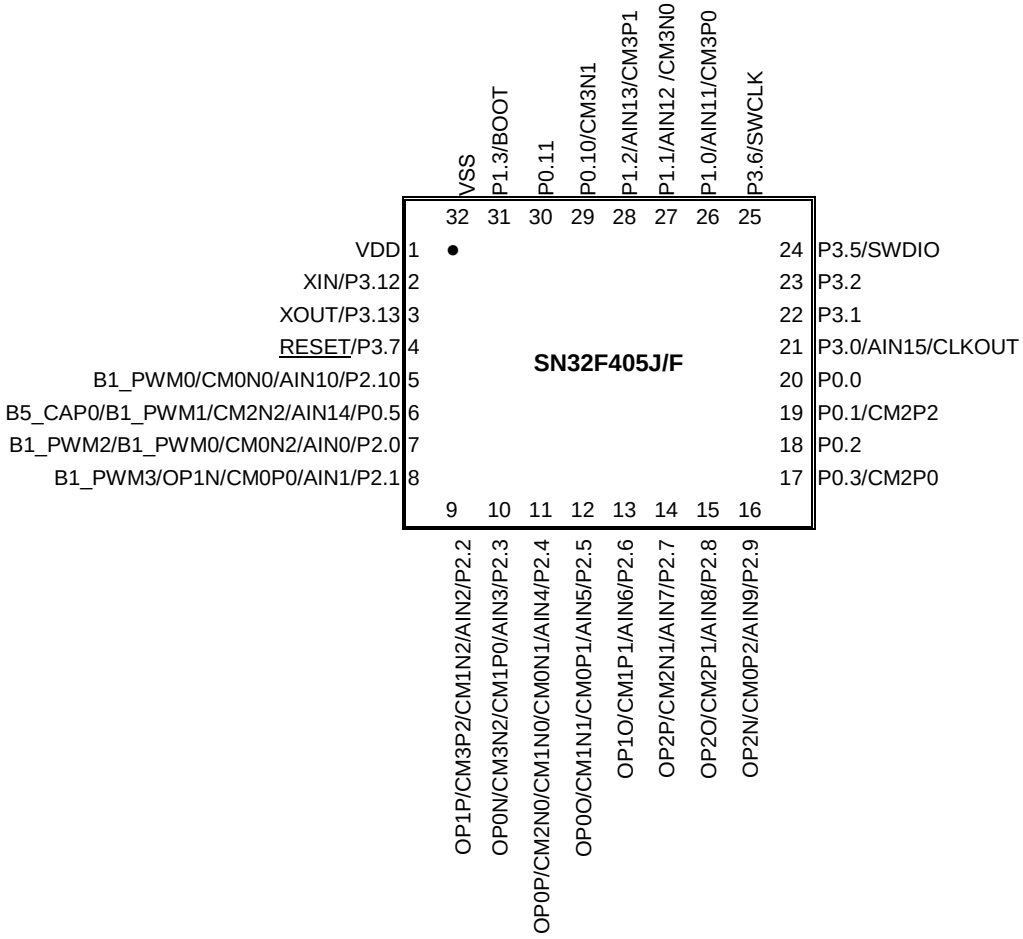
1.4 PIN ASSIGNMENT

SN32F407J/F (QFN 48 pins 6x6/LQFP 48 pins)



✸ ***Note: SONiX provide Boot loader to check the status of P1.3 (BOOT pin) during boot procedure. If BOOT pin is Low during Boot procedure, MCU will execute code in Boot loader instead of User code. We strongly recommended NOT using BOOT pin as output pin to drive the LED, otherwise, the BOOT pin status may be low during boot procedure.***

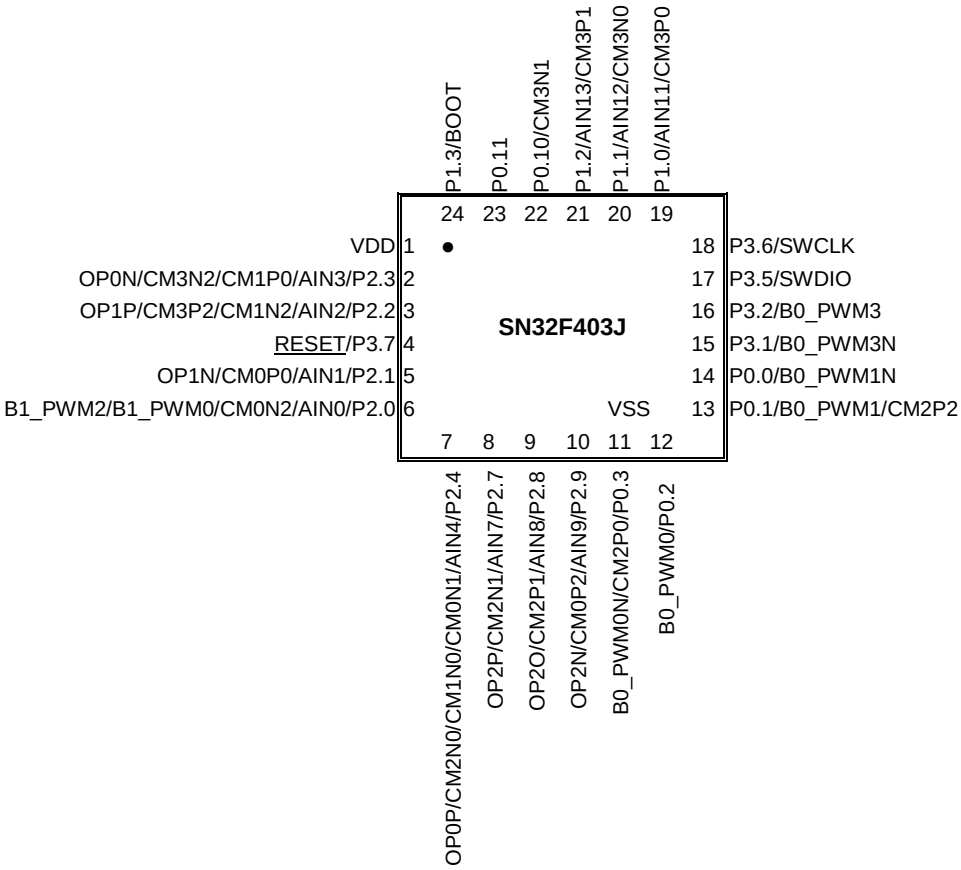
SN32F405J/F (QFN 32 pins 4x4/LQFP 32 pins)



✸ ***Note: 1. The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

2. SONiX provide Boot loader to check the status of P1.3 (BOOT pin) during boot procedure. If BOOT pin is Low during Boot procedure, MCU will execute code in Boot loader instead of User code. We strongly recommended NOT using BOOT pin as output pin to drive the LED, otherwise, the BOOT pin status may be low during boot procedure.

SN32F403J (QFN 24 pins 4x4 pins)



✸ ***Note: 1. The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

2. SONiX provide Boot loader to check the status of P1.3 (BOOT pin) during boot procedure. If BOOT pin is Low during Boot procedure, MCU will execute code in Boot loader instead of User code. We strongly recommended NOT using BOOT pin as output pin to drive the LED, otherwise, the BOOT pin status may be low during boot procedure.

3. The VSS is connected to the exposed thermal pad (EPAD).

1.5 PIN ALLOCATION TABLE

Port	UART	SPI	I2C	CT16	CMP	ADC	OPA	Other
P0.0		SCK0		B0_PWM0 B0_PWM1N B0_PWM2 B0_PWM3N B5_PWM3				
P0.1		SEL0		B0_PWM0N B0_PWM1 B0_PWM2N B0_PWM3 B0_PWM3N B5_PWM2	CM2P2			
P0.2		MISO0		B0_PWM0 B0_PWM1N B0_PWM3N B5_PWM1				
P0.3		MOSI0		B0_PWM0N B0_PWM3 B5_PWM0	CM2P0 CM2O			
P0.4								
P0.5				B0_PWM3 B0_BRK B1_PWM1 B5_CAP0	CM2N2	AIN14		
P0.6			SCL0	B1_PWM2				
P0.7			SDA0	B1_PWM3	CM1P2			
P0.10	UTXD0		SCL0	B0_BRK B1_PWM2	CM3N1			
P0.11	URXD0		SDA0	B1_PWM3				
P1.0	UTXD1	SCK0		B5_PWM2	CM2O CM3P0	AIN11		
P1.1	URXD1	MISO0	SCL0	B0_PWM0N B1_PWM0 B5_PWM0	CM3N0	AIN12		
P1.2		MOSI0	SDA0	B1_PWM1 B5_PWM1	CM3P1	AIN13		
P1.3				B5_PWM3	CM3O			BOOT
P1.4			SCL0					
P1.5			SDA0					
P1.6					CM0O			
P1.7		SEL0						
P1.8	URXD1	SEL0						
P1.9	UTXD1							
P1.10					CM1O			
P1.11					CM2O			
P1.12				B0_BRK B1_PWM0				
P2.0	URXD0			B1_PWM0 B1_PWM2	CM0N2 CM0O	AIN0		

P2.1	UTXD0			B1_PWM3 B5_PWM3	CM0P0	AIN1	OP1N	
P2.2	UTXD1			B5_PWM2	CM1N2 CM3P2 CM1O	AIN2	OP1P	
P2.3	URXD1			B5_PWM1	CM1P0 CM3N2	AIN3	OP0N	
P2.4		SEL0		B0_PWM1N B1_PWM1 B5_PWM0	CM0N1 CM1N0 CM2N0	AIN4	OP0P	
P2.5		SCK0		B0_PWM1	CM0P1 CM1N1	AIN5	OP0O	
P2.6		MISO0		B0_PWM0 B0_PWM2N B5_PWM0	CM0O CM1P1	AIN6	OP1O	
P2.7	UTXD1	MOSI0		B0_PWM0N B0_PWM2 B1_PWM2 B5_PWM1	CM1O CM2N1	AIN7	OP2P	
P2.8	URXD1			B0_PWM1N B0_PWM2 B0_PWM2N B1_PWM1 B5_PWM2	CM2P1 CM3O	AIN8	OP2O	
P2.9	UTXD0			B0_PWM1 B0_PWM2N B1_PWM3 B5_PWM3	CM0P2	AIN9	OP2N	
P2.10	URXD0			B0_BRK B1_PWM0	CM0N0	AIN10		
P3.0		SCK0		B0_PWM0	CM0O CM3O	AIN15		CLKOUT
P3.1	UTXD0	MISO0		B0_PWM1 B0_PWM3N	CM1O			
P3.2	URXD0	MOSI0		B0_PWM2 B0_PWM3				
P3.5								SWDIO
P3.6								SWCLK
P3.7								RESET
P3.8				B1_CAP0	CM2O			
P3.9				B0_CAP0				
P3.10								LXIN
P3.11								LXOUT
P3.12					CM3O			XIN
P3.13								XOUT

1.6 PIN DESCRIPTIONS

Pad NAME	TYPE	DESCRIPTION
VDD, VSS	P	Power supply input pins for digital circuit.
P0.0	I/O	**P0.0 —** General-purpose digital input/output pin.
P0.1/CM2P2	I/O	**P0.1 —** General-purpose digital input/output pin.
	I	**CM2P2 —** Positive input pin 2 of comparator 2.
P0.2	I/O	**P0.2 —** General-purpose digital input/output pin.
P0.3/CM2P0	I/O	**P0.3 —** General-purpose digital input/output pin.
	I	**CM2P0 —** Positive input pin 0 of comparator 2.
P0.4	I/O	**P0.4 —** General-purpose digital input/output pin.
P0.5/AIN14/CM2N2/ CT16B5_CAP0/CT16B0_BRK	I/O	**P0.5 —** General-purpose digital input/output pin.
	I	**AIN14 —** ADC channel input 14.
		CM2N2 — Negative input pin 2 of comparator 2.
		CT16B5_CAP0 — Capture input 0 for CT16B5.
		CT16B0_BRK — Break input for CT16B0.
P0.6	I/O	**P0.6 —** General purpose digital input/output pin.
P0.7/CM1P2	I/O	**P0.7 —** General purpose digital input/output pin.
	I	**CM1P2 —** Positive input pin 2 of comparator 1.
P0.10/CM3N1/CT16B0_BRK	I/O	**P0.10 —** General-purpose digital input/output pin.
	I	**LXIN —** External low-speed X'tal input pin.
		CM3N1 — Negative input pin 1 of comparator 3.
		CT16B0_BRK — Break input for CT16B0.
P0.11	I/O	**P0.11 —** General-purpose digital input/output pin.
P1.0/AIN11/CM3P0	I/O	**P1.0 —** General-purpose digital input/output pin.
	I	**AIN11 —** ADC channel input 11.
		CM3P0 — Positive input pin 0 of comparator 3.
P1.1/AIN12/CM3N0	I/O	**P1.1 —** General-purpose digital input/output pin.
	I	**AIN12 —** ADC channel input 12.
		CM3N0 — Negative input pin 0 of comparator 3.
P1.2/AIN13/CM3P1	I/O	**P1.2 —** General-purpose digital input/output pin.
	I	**AIN13 —** ADC channel input 13.
		CM3P1 — Positive input pin 1 of comparator 3.
P1.3~P1.11	I/O	**P1.3~P1.11 —** General-purpose digital input/output pin.

P1.12/CT16B0_BRK	I/O	**P1.12 —** General-purpose digital input/output pin.
	I	**CT16B0_BRK —** Break input for CT16B0.
P2.0/AIN0/CM0N2	I/O	**P2.0 —** General-purpose digital input/output pin.
	I	**AIN0 —** ADC channel input 0.
		CM0N2 — Negative input pin 2 of comparator 0.
P2.1/AIN1/CM0P0/OP1N	I/O	**P2.1 —** General-purpose digital input/output pin.
	I	**AIN1 —** ADC channel input 1.
		CM0P0 — Positive input pin 0 of comparator 0.
		OP1N — Negative input pin of OPA 1.
P2.2/AIN2/CM1N2/CM3P2/ OP1P	I/O	**P2.2 —** General-purpose digital input/output pin.
	I	**AIN2 —** ADC channel input 2.
		CM1N2 — Negative input pin 2 of comparator 1.
		CM3P2 — Positive input pin 2 of comparator 3.
		OP1P — Positive input pin of OPA 1.
P2.3/AIN3/CM1P0/CM3N2/ OP0N	I/O	**P2.3 —** General-purpose digital input/output pin.
	I	**AIN3 —** ADC channel input 3.
		CM1P0 — Positive input pin 0 of comparator 1.
		CM3N2 — Negative input pin 2 of comparator 3.
		OP0N — Negative input pin of OPA 0.
P2.4/AIN4/CM1N0/CM0N1/ CM2N0/OP0P	I/O	**P2.4 —** General-purpose digital input/output pin.
	I	**AIN4 —** ADC channel input 4.
		CM0N1 — Negative input pin 1 of comparator 0.
		CM1N0 — Negative input pin 0 of comparator 1.
		CM2N0 — Negative input pin 0 of comparator 2.
		OP0P — Positive input pin of OPA 0.
P2.5/AIN5/CM1N1/CM0P1/ OP0O	I/O	**P2.5 —** General-purpose digital input/output pin.
	I	**AIN5 —** ADC channel input 5.
		CM0P1 — Positive input pin 1 of comparator 0.
		CM1N1 — Negative input pin 1 of comparator 1.
	O	**OP0O —** Output pin of OPA 0.
P2.6/AIN6/CM1P1/OP1O	I/O	**P2.6 —** General-purpose digital input/output pin.
	I	**AIN6 —** ADC channel input 6.
		CM1P1 — Positive input pin 1 of comparator 1.
	O	**OP1O —** Output pin of OPA 1.

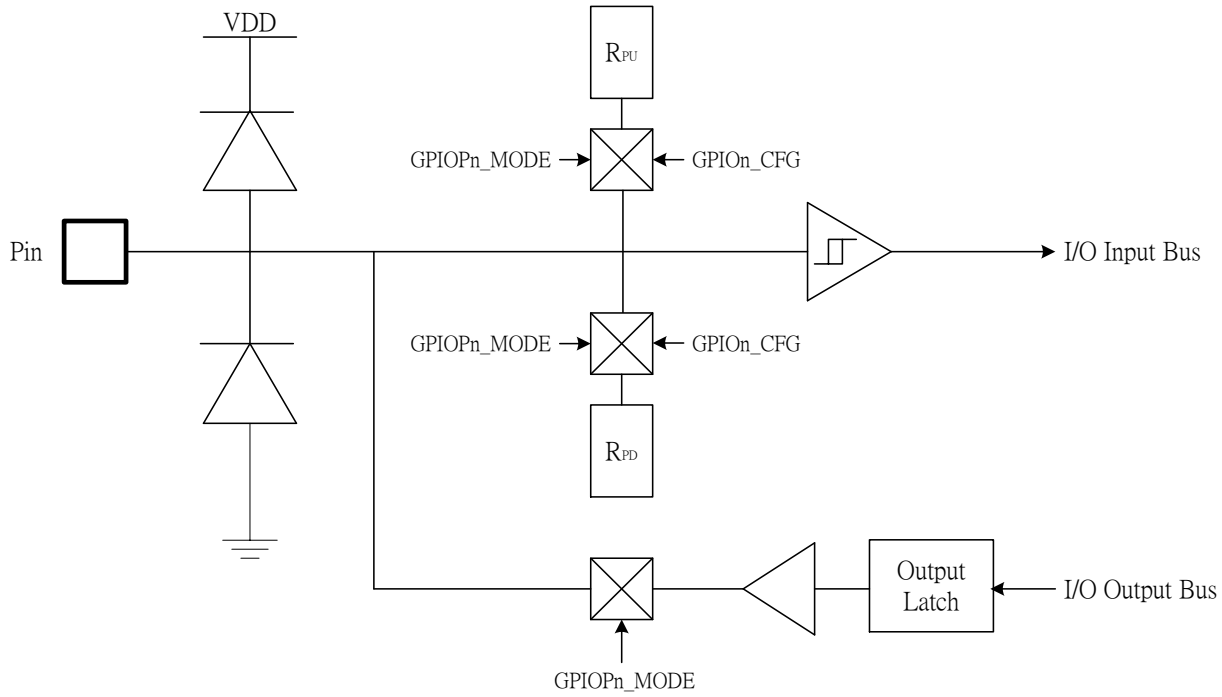
P2.7/AIN7/CM2N1/OP2P	I/O	**P2.7 —** General-purpose digital input/output pin.
	I	**AIN7 —** ADC channel input 7.
		CM2N1 — Negative input pin 1 of comparator 2.
		OP2P — Positive input pin of OPA 2.
P2.8/AIN8/CM2P1/OP2O	I/O	**P2.8 —** General-purpose digital input/output pin.
	I	**AIN8 —** ADC channel input 8.
		CM2P1 — Positive input pin 1 of comparator 2.
	O	**OP2O —** Output pin of OPA 2.
P2.9/AIN9/CM0P2/OP2N	I/O	**P2.9 —** General-purpose digital input/output pin.
	I	**AIN9 —** ADC channel input 9.
		CM0P2 — Positive input pin 2 of comparator 0.
		OP2N — Negative input pin of OPA 2.
P2.10/AIN10/CM0N0/ CT16B0_BRK	I/O	**P2.10 —** General-purpose digital input/output pin.
	I	**AIN10 —** ADC channel input 10.
		CM0N0 — Negative input pin 0 of comparator 0.
		CT16B0_BRK — Break input for CT16B0.
P3.0/AIN15/CLKOUT	I/O	**P3.0 —** General-purpose digital input/output pin.
	I	**AIN15 —** ADC channel input 15.
	O	**CLKOUT —** Clockout pin
P3.1	I/O	**P3.1 —** General-purpose digital input/output pin.
P3.2	I/O	**P3.2 —** General-purpose digital input/output pin.
P3.5/SWDIO	I/O	**P3.5 —** General-purpose digital input/output pin.
		SWDIO — Serial wire debug input/output.
P3.6/SWCLK	I/O	**P3.6 —** General-purpose digital input/output pin.
	I	**SWCLK —** Serial wire clock.
P3.7/RESET	I/O	**P3.7 —** General-purpose digital input/output pin.
	I	**RESET —** External Reset input.
P3.8~P3.9	I/O	**P3.8~P3.9 —** General-purpose digital input/output pin.
P3.10/LXIN	I/O	**P3.10 —** General-purpose digital input/output pin.
	I	**LXIN —** External low-speed X'tal input pin.
P3.11/LXOUT	I/O	**P3.11 —** General-purpose digital input/output pin.
	O	**LXOUT —** External low-speed X'tal output pin.
P3.12/XIN	I/O	**P3.12 —** General-purpose digital input/output pin.

	I	**XIN —** External high-speed X'tal input pin.
P3.13/XOUT	I/O	**P3.13 —** General-purpose digital input/output pin.
	O	**XOUT —** External high-speed X'tal output pin.

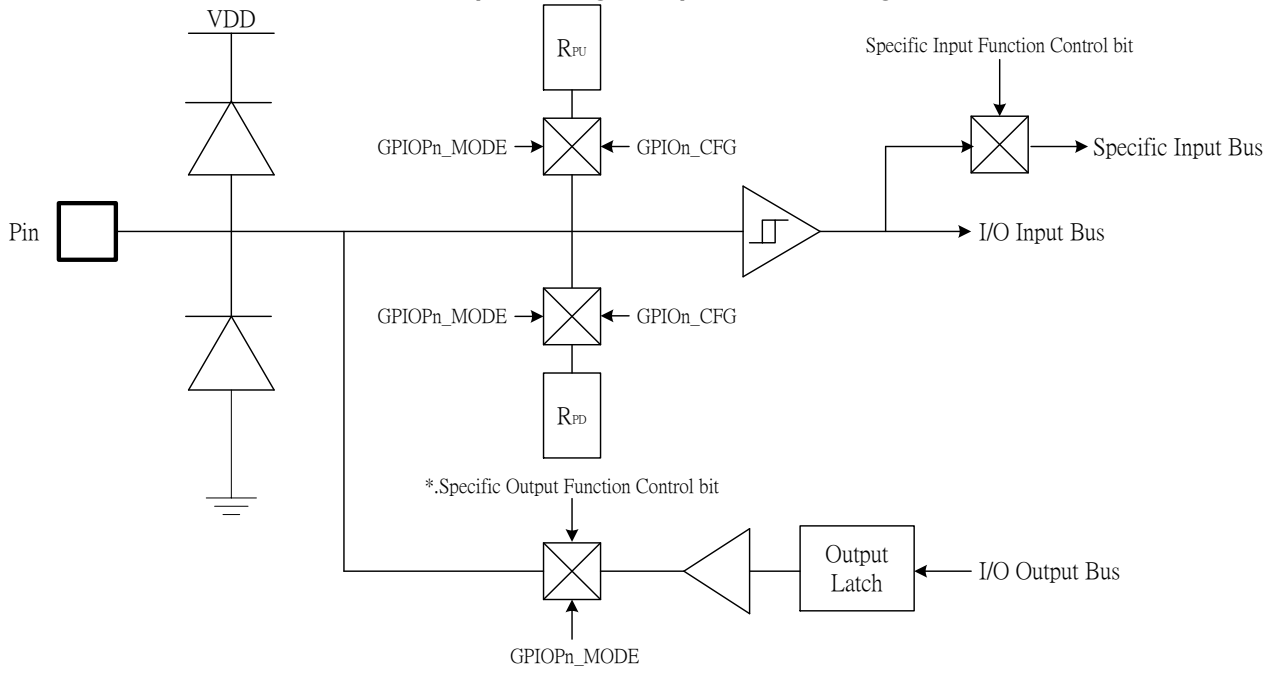
Please refer to Peripheral Function Pin Assignment (PFPA) chapters for setting of each GPIOs.

1.7 PIN CIRCUIT DIAGRAMS

- **Normal Bi-direction I/O Pin.**

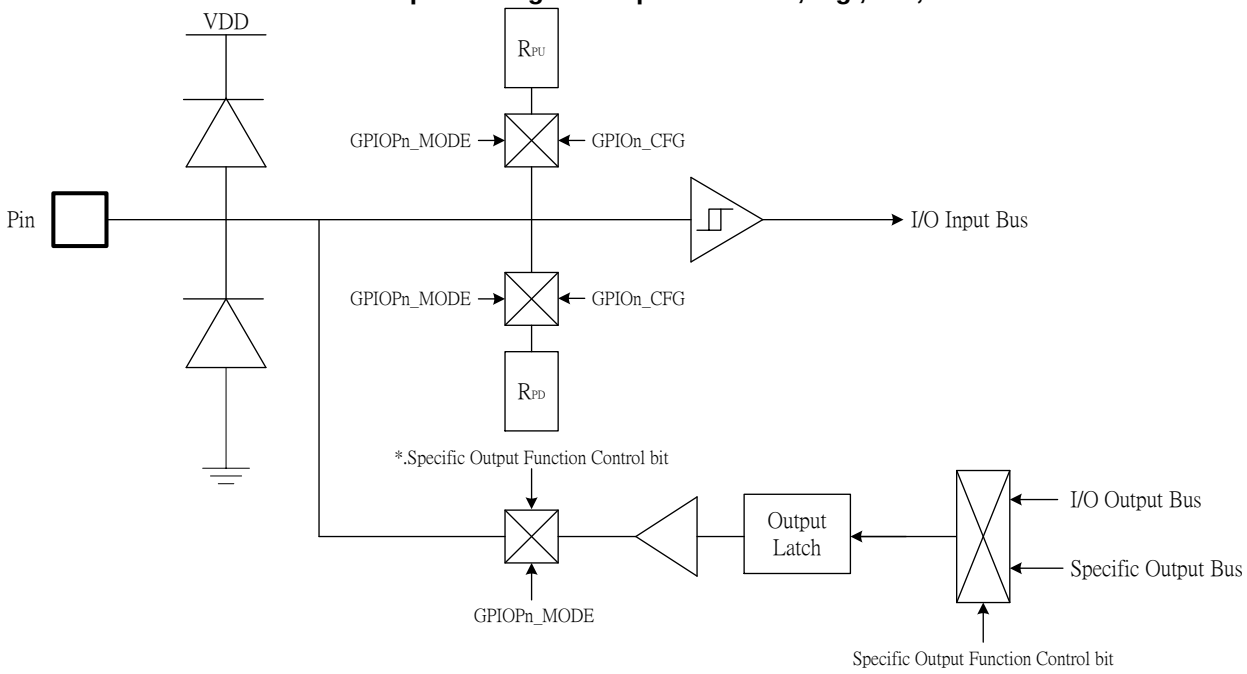


- **Bi-direction I/O Pin Shared with Specific Digital Input Function, e.g., SPI, I2C…**



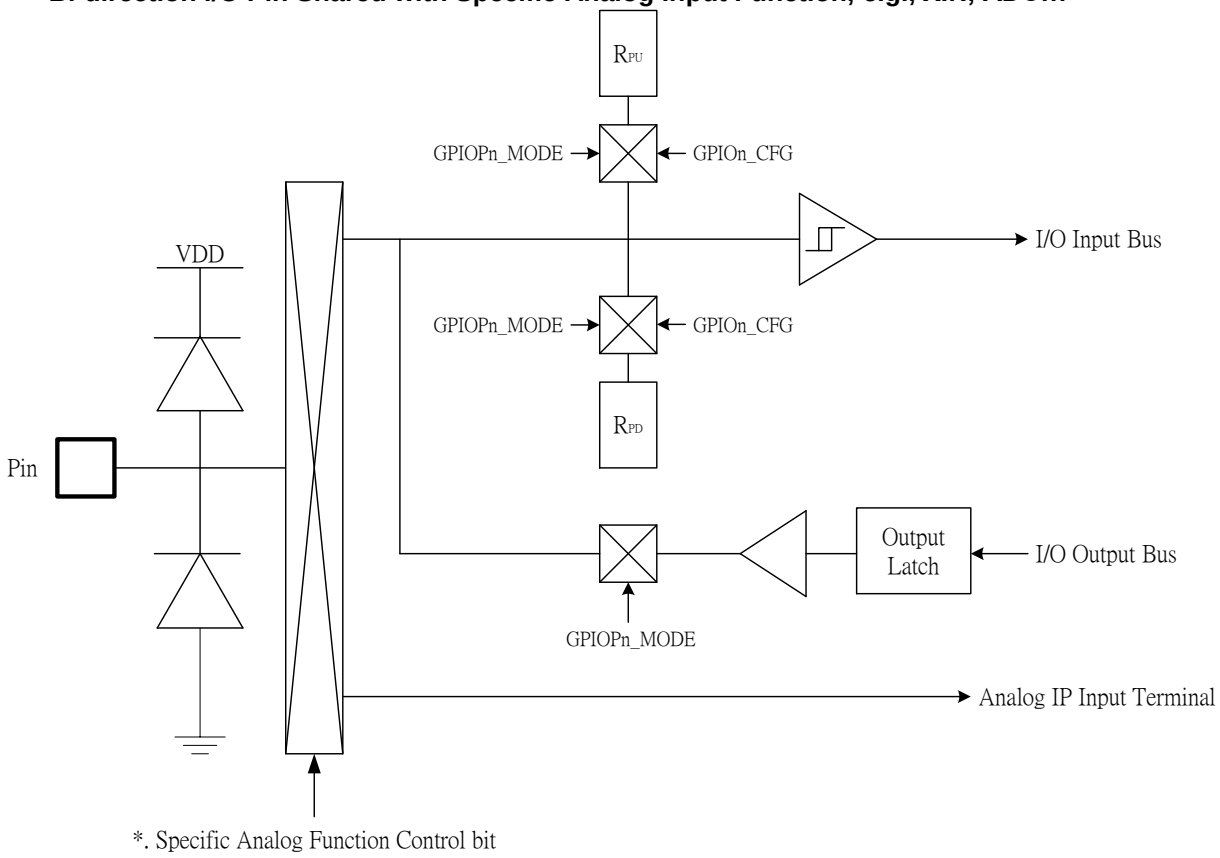
*. Some specific functions switch I/O direction directly, not through GPIOn_MODE register.

- **Bi-direction I/O Pin Shared with Specific Digital Output Function, e.g., SPI, I2C…**



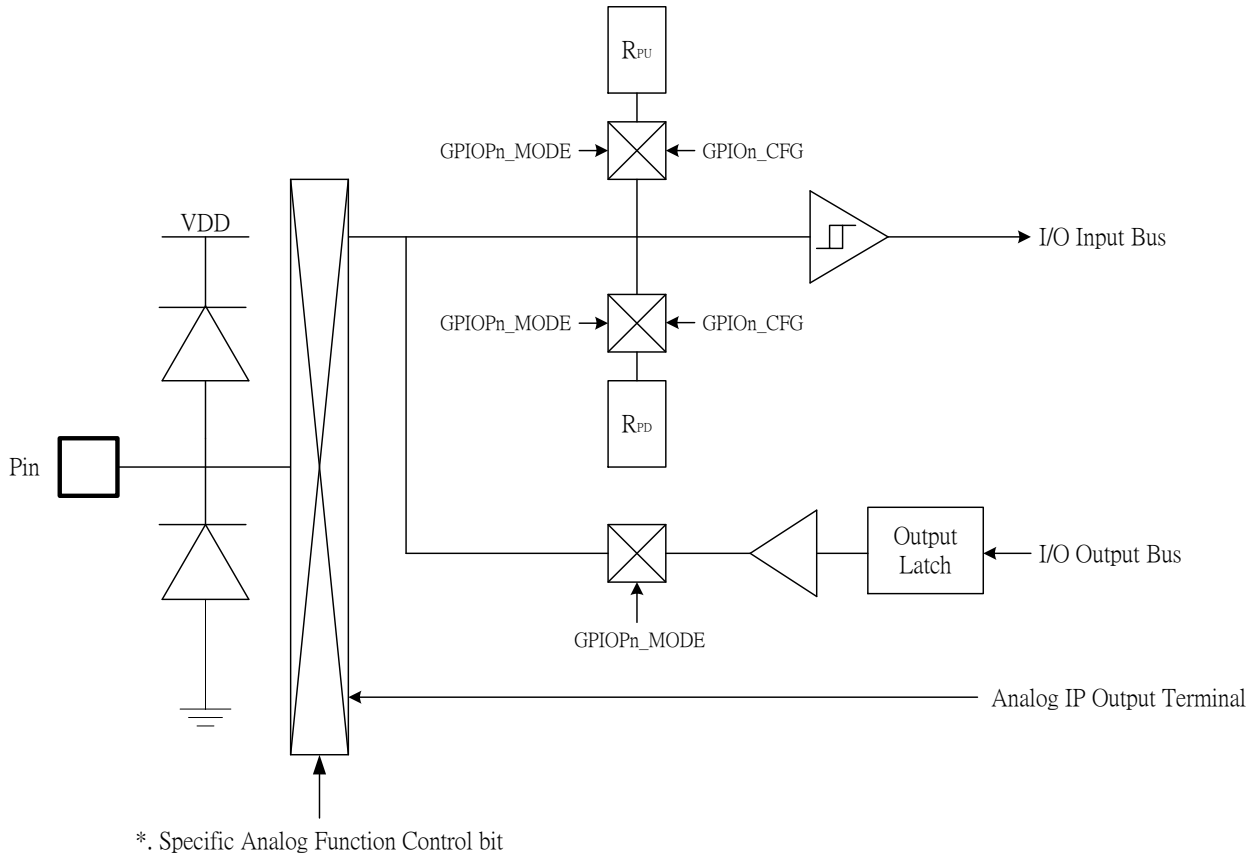
*. Some specific functions switch I/O direction directly, not through GPIOn_MODE register.

- **Bi-direction I/O Pin Shared with Specific Analog Input Function, e.g., XIN, ADC…**



*. Some specific functions switch I/O direction directly, not through GPIOn_MODE register.

- **Bi-direction I/O Pin Shared with Specific Analog Output Function, e.g., XOUT…**



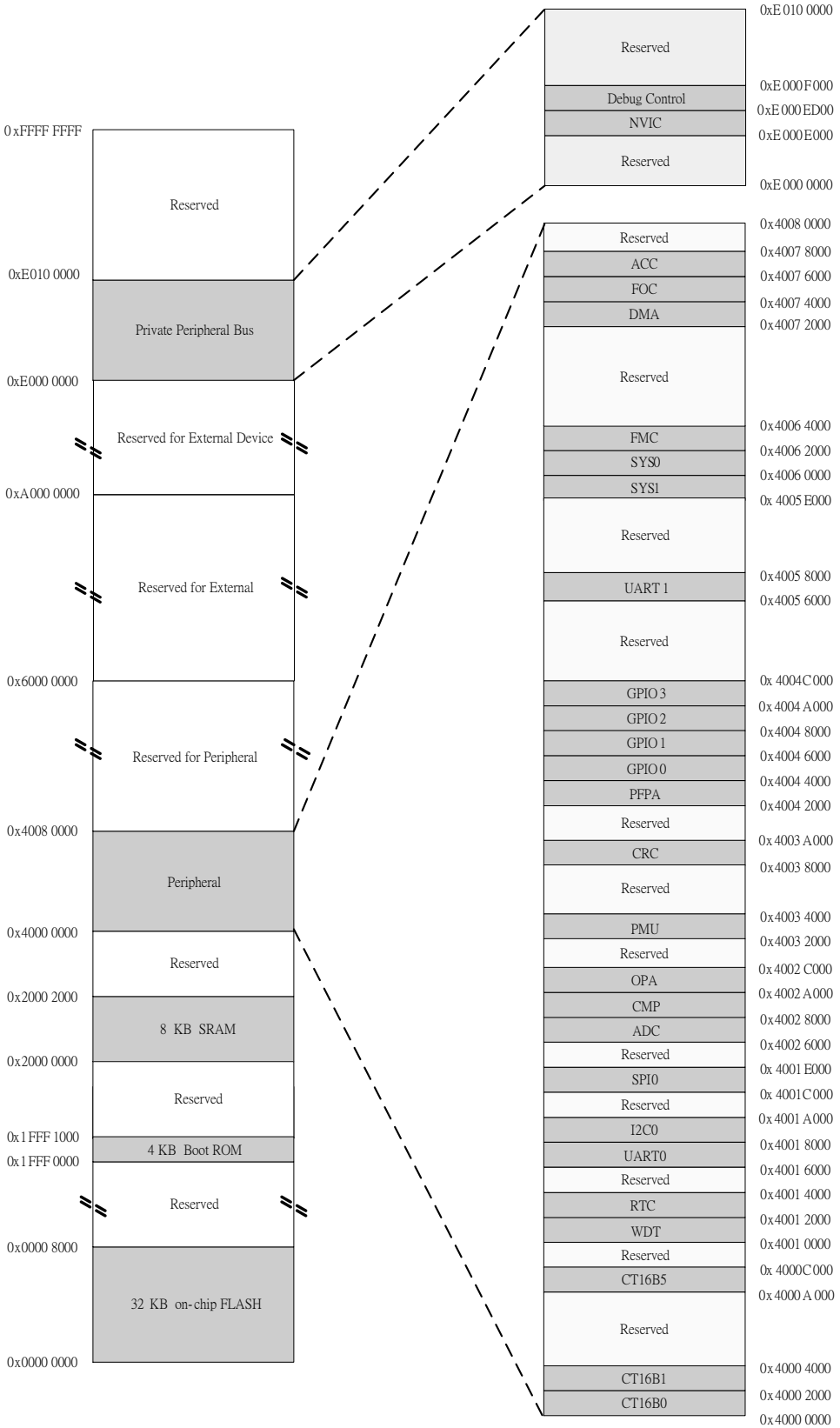
*. Some specific functions switch I/O direction directly, not through GPIOn_MODE register.

1.8 PIN CHARACTERISTICS

Port	Dir	ADC	CMP Input	OPA input	Open-Drain	Pull-up Resistor	Pull-down Resistor	Schmitt trigger (V_{IH}: 0.7*VDD V_{IL}: 0.3*VDD)	Drive Current 30mA @VDD-0.5V	Sink Current 30mA @VSS+0.5V
P0.0	I/O					V	V	V	V	V
P0.1	I/O		V			V	V	V	V	V
P0.2	I/O					V	V	V	V	V
P0.3	I/O		V			V	V	V	V	V
P0.4	I/O					V	V	V	V	V
P0.5	I/O	V	V			V	V	V	V	V
P0.6	I/O				V	V	V	V	V	V
P0.7	I/O		V		V	V	V	V	V	V
P0.10	I/O		V		V	V	V	V	V	V
P0.11	I/O				V	V	V	V	V	V
P1.0	I/O	V	V			V	V	V	V	V
P1.1	I/O	V	V		V	V	V	V	V	V
P1.2	I/O	V	V		V	V	V	V	V	V
P1.3	I/O					V	V	V	V	V
P1.4	I/O				V	V	V	V	V	V
P1.5	I/O				V	V	V	V	V	V
P1.6	I/O					V	V	V	V	V
P1.7	I/O					V	V	V	V	V
P1.8	I/O					V	V	V	V	V
P1.9	I/O					V	V	V	V	V
P1.10	I/O					V	V	V	V	V
P1.11	I/O					V	V	V	V	V
P1.12	I/O					V	V	V	V	V
P2.0	I/O	V	V			V	V	V	V	V
P2.1	I/O	V	V	V		V	V	V	V	V
P2.2	I/O	V	V	V		V	V	V	V	V
P2.3	I/O	V	V	V		V	V	V	V	V
P2.4	I/O	V	V	V		V	V	V	V	V
P2.5	I/O	V	V			V	V	V	V	V
P2.6	I/O	V	V			V	V	V	V	V
P2.7	I/O	V	V	V		V	V	V	V	V
P2.8	I/O	V	V			V	V	V	V	V
P2.9	I/O	V	V	V		V	V	V	V	V
P2.10	I/O	V	V			V	V	V	V	V
P3.0	I/O	V				V	V	V	V	V
P3.1	I/O					V	V	V	V	V
P3.2	I/O					V	V	V	V	V
P3.5	I/O					V	V	V	V	V
P3.6	I/O					V	V	V	V	V
P3.7	I/O					V	V	V	V	V
P3.8	I/O					V	V	V	V	V
P3.9	I/O					V	V	V	V	V
P3.10	I/O					V	V	V	V	V
P3.11	I/O					V	V	V	V	V
P3.12	I/O					V	V	V	V	V
P3.13	I/O					V	V	V	V	V

2 CENTRAL PROCESSOR UNIT (CPU)

2.1 MEMORY MAP



2.2 SYSTEM TICK TIMER

The SysTick timer is an integral part of the Cortex-M0. The SysTick timer is intended to generate a fixed 10-ms interrupt for use by an operating system or other system management software.

Since the SysTick timer is a part of the Cortex-M0, it facilitates porting of software by providing a standard timer that is available on Cortex-M0 based devices.

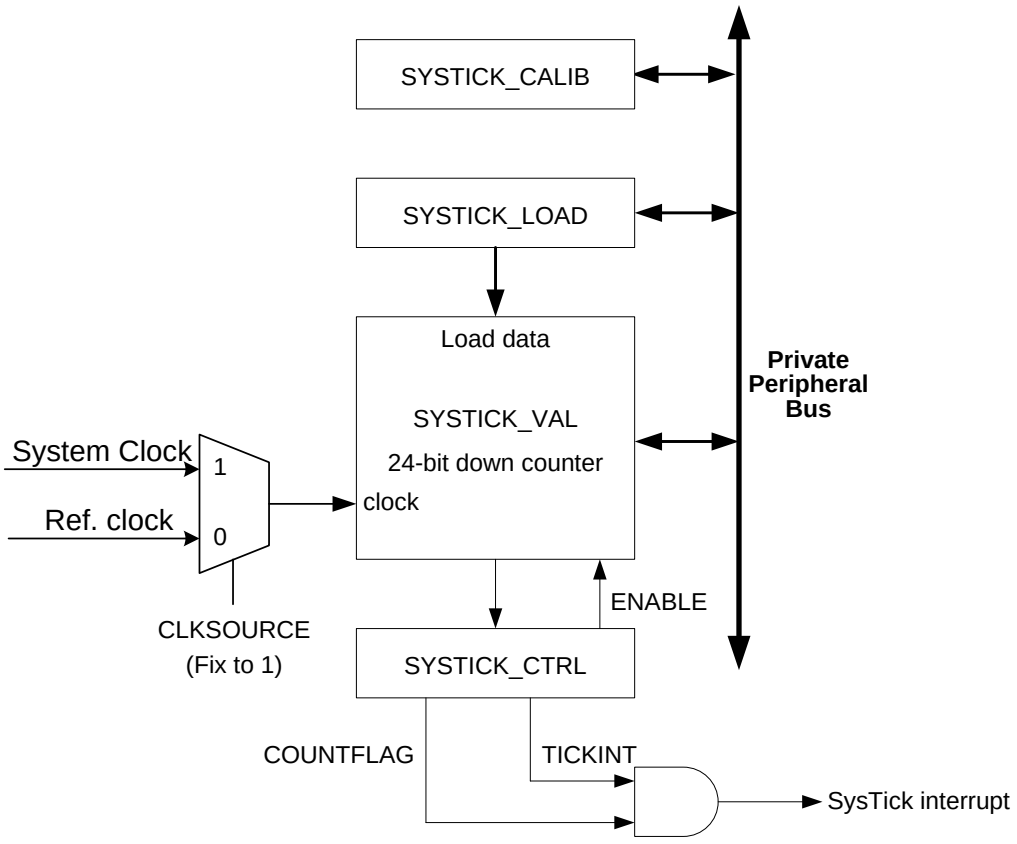
Refer to the Cortex-M0 User Guide for details.

2.2.1 OPERATION

The SysTick timer is a 24-bit timer that counts down to zero and generates an interrupt.

The intent is to provide a fixed 10-ms time interval between interrupts. The system tick timer is enabled through the SysTick control register. The system tick timer clock is fixed to the frequency of the system clock.

The block diagram of the SysTick timer:



When SysTick timer is enabled, the timer counts down from the current value (SYSTICK_VAL) to zero, reloads to the value in the SysTick Reload Value Register (SYSTICK_LOAD) on the next clock edge, then decrements on subsequent clocks. When the counter transitions to zero, the COUNTFLAG status bit is set to 1. The COUNTFLAG bit clears on reads.

✷ ***Note: When the processor is halted for debugging the counter does not decrease.***

2.2.2 SYSTICK USAGE HINTS AND TIPS

The interrupt controller clock updates the SysTick counter. Some implementations stop this clock signal for low power mode. If this happens, the SysTick counter stops.

Ensure SW uses word accesses to access the SysTick registers.

The SysTick counter reload and current value are not initialized by HW. This means the correct initialization sequence for the SysTick counter is:

1. Program the reload value in SYSTICK_LOAD register.
2. Clear the current value by writing any value to SYSTICK_VAL register.
3. Program the Control and Status (SYSTICK_CTRL) register.

2.2.3 SYSTICK REGISTERS

2.2.3.1 System Tick Timer Control and Status register (SYSTICK_CTRL)

Address: 0xE000 E010 (Refer to Cortex-M0 Spec)

Bit	Name	Description	Attribute	Reset
31:17	Reserved		**R**	**0**
16	COUNTFLAG	This flag is set when the System Tick counter counts down to 0, and is cleared by reading this register.	R/W	0
15:3	Reserved		R	0
2	CLKSOURCE	Selects the SysTick timer clock source. 0: reference clock. 1: system clock. (Fixed)	R	1
1	TICKINT	System Tick interrupt enable. 0: Disable the System Tick interrupt 1: Enable the System Tick interrupt, the interrupt is generated when the System Tick counter counts down to 0.	R/W	0
0	ENABLE	System Tick counter enable. 0: Disable 1: Enable	R/W	0

2.2.3.2 System Tick Timer Reload value register (SYSTICK_LOAD)

Address: 0xE000 E014 (Refer to Cortex-M0 Spec)

The RELOAD register is set to the value that will be loaded into the SysTick timer whenever it counts down to zero. This register is set by software as part of timer initialization. The SYSTICK_CALIB register may be read and used as the value for RELOAD if the CPU or external clock is running at the frequency intended for use with the SYSTICK_CALIB value.

The following example illustrates selecting the SysTick timer reload value to obtain a 10 ms time interval with the system clock set to 50 MHz

The SysTick clock = system clock = 50 MHz

RELOAD = (system tick clock frequency × 10 ms) −1 = (50 MHz × 10 ms) −1 = 0x0007A11F.

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	RELOAD	Value to load into the SYSTICK_VAL when the counter is enabled and when it reaches 0.	R/W	0

2.2.3.3 System Tick Timer Current Value register (SYSTICK_VAL)

Address: 0xE000 E018 (Refer to Cortex-M0 Spec)

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	CURRENT	Reading this register returns the current value of the System Tick counter. Writing any value clears the System Tick counter and the COUNTFLAG bit in SYSTICK_CTRL.	R/W	0x7E7F35

2.2.3.4 System Tick Timer Calibration Value register (SYSTICK_CALIB)

Address: 0xE000 E01C (Refer to Cortex-M0 Spec)

Bit	Name	Description	Attribute	Reset
31	NOREF	Indicates the reference clock to M0 is provided or not. 1: No reference clock provided.	R	1
30	SKEW	Indicates whether the TENMS value is exact, an inexact TENMS value can affect the suitability of SysTick as a software real time clock. 0: TENMS value is exact 1: TENMS value is inexact, or not given.	R	0
29:24	Reserved		R	0
23:0	TENMS	Reload value for 10ms timing, subject to system clock skew errors. If the value reads as zero, the calibration value is not known.	R/W	0xA71FF

2.3 NESTED VECTORED INTERRUPT CONTROLLER (NVIC)

All interrupts including the core exceptions are managed by the NVIC. NVIC has the following Features:

- The NVIC supports 32 vectored interrupts.
- 4 programmable interrupt priority levels with hardware priority level masking.
- Low-latency exception and interrupt handling.
- Efficient processing of late arriving interrupts.
- Implementation of System Control Registers
- Software interrupt generation.

2.3.1 INTERRUPT AND EXCEPTION VECTORS

Execution No.	Priority	Function	Description	Address Offset
0	-	-	Reserved	0x0000 0000
1	-3	Reset	Reset	0x0000 0004
2	-2	NMI_Handler	Non maskable interrupt.	0x0000 0008
3	-1	HardFault_Handler	All class of fault	0x0000 000C
4~10	Reserved	Reserved	Reserved	-
11	Settable	SVCCalll		0x0000 002C
12~13	Reserved	Reserved	Reserved	-
14	Settable	PendSV		0x0000 0038
15	Settable	SysTick		0x0000 003C
16	Settable	IRQ0/NDTIRQ	NDT	0x0000 0040
17	Settable	IRQ1/DMA0IRQ	DMA0	0x0000 0044
18	Settable	IRQ2/CMP3IRQ	CMP3	0x0000 0048
19	Settable	IRQ3/		0x0000 004C
20	Settable	IRQ4/		0x0000 0050
21	Settable	IRQ5/		0x0000 0054
22	Settable	IRQ6/SPI0IRQ	SPI0	0x0000 0058
23	Settable	IRQ7/		0x0000 005C
24	Settable	IRQ8/		0x0000 0060
25	Settable	IRQ9		0x0000 0064
26	Settable	IRQ10/I2C0IRQ	I2C0	0x0000 0068
27	Settable	IRQ11/		0x0000 006C
28	Settable	IRQ12/CMP2IRQ	CMP2	0x0000 0070
29	Settable	IRQ13/UART0IRQ	UART0	0x0000 0074
30	Settable	IRQ14/UART1IRQ	UART1	0x0000 0078
31	Settable	IRQ15/CT16B0IRQ	CT16B0	0x0000 007C
32	Settable	IRQ16/CT16B1IRQ	CT16B1	0x0000 0080

33	Settable	IRQ17/SQRTIRQ	SQRT	0x0000 0084
34	Settable	IRQ18/CMP1IRQ	CMP1	0x0000 0088
35	Settable	IRQ19/ATANIRQ	ATAN	0x0000 008C
36	Settable	IRQ20/DIVIRQ	DIV	0x0000 0090
37	Settable	IRQ21/CT16B5IRQ	CT16B5	0x0000 0094
38	Settable	IRQ22/FOCIRQ	FOC	0x0000 0098
39	Settable	IRQ23/RTCIRQ	RTC	0x0000 009C
40	Settable	IRQ24/ADCIRQ	ADC	0x0000 00A0
41	Settable	IRQ25/WDTIRQ	WDT	0x0000 00A4
42	Settable	IRQ26/LVDIRQ	LVD	0x0000 00A8
43	Settable	IRQ27/CMP0IRQ	CMP0	0x0000 00AC
44	Settable	IRQ28/P3IRQ	GPIO interrupt status of port 3	0x0000 00B0
45	Settable	IRQ29/P2IRQ	GPIO interrupt status of port 2	0x0000 00B4
46	Settable	IRQ30/P1IRQ	GPIO interrupt status of port 1	0x0000 00B8
47	Settable	IRQ31/P0IRQ	GPIO interrupt status of port 0	0x0000 00BC

2.3.2 NVIC REGISTERS

2.3.2.1 IRQ0~31 Interrupt Set-Enable Register (NVIC_ISER)

Address: 0xE000 E100 (Refer to Cortex-M0 Spec.)

The ISER enables interrupts, and shows the interrupts that are enabled.

Bit	Name	Description	Attribute	Reset
31:0	SETENA[31:0]	Interrupt set-enable bits. Write→ 0: No effect 1: Enable interrupt. Read→ 0: Interrupt disabled 1: Interrupt enabled.	R/W	0

2.3.2.2 IRQ0~31 Interrupt Clear-Enable Register (NVIC_ICER)

Address: 0xE000 E180 (Refer to Cortex-M0 Spec.)

The ICER disables interrupts, and shows the interrupts that are enabled.

Bit	Name	Description	Attribute	Reset
31:0	CLRENA[31:0]	Interrupt clear-enable bits. Write→ 0: No effect 1: Disable interrupt. Read→ 0: Interrupt disabled 1: Interrupt enabled.	R/W	0

2.3.2.3 IRQ0~31 Interrupt Set-Pending Register (NVIC_ISPR)

Address: 0xE000 E200 (Refer to Cortex-M0 Spec.)

The ISPR forces interrupts into the pending state, and shows the interrupts that are pending.

✸ ***Note: Writing 1 to the ISPR bit corresponding to***
1. ***an interrupt that is pending has no effect***
2. ***a disabled interrupt sets the state of that interrupt to pending.***

Bit	Name	Description	Attribute	Reset
31:0	SETPEND[31:0]	Interrupt set-pending bits. Write→ 0: No effect 1: Change interrupt state to pending Read→ 0: Interrupt is not pending 1: Interrupt is pending	R/W	0

2.3.2.4 IRQ0~31 Interrupt Clear-Pending Register (NVIC_ICPR)

Address: 0xE000 E280 (Refer to Cortex-M0 Spec.)

The ICPR removes the pending state from interrupts, and shows the interrupts that are pending.

✸ ***Note: Writing 1 to an ICPR bit does not affect the active state of the corresponding interrupt.***

Bit	Name	Description	Attribute	Reset
31:0	CLRPEND[31:0]	Interrupt clear-pending bits. Write→ 0: No effect 1: Removes pending state of an interrupt Read→ 0: Interrupt is not pending 1: Interrupt is pending	R/W	0

2.3.2.5 IRQ0~31 Interrupt Priority Register (NVIC_IPRn) (n=0~7)

Address: 0xE000 E400 + 0x4 * n (Refer to Cortex-M0 Spec.)

The interrupt priority registers provide an 8-bit priority field for each interrupt, and each register holds four priority fields. This means the number of registers is implementation-defined, and corresponds to the number of implemented interrupts.

Bit	Name	Description	Attribute	Reset
31:24	PRI_(4*n+3)	Each priority field holds a priority value, 0-192. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[31:30] of each field, bits [29:24] read as zero and ignore writes. This means writing 255 to a priority register saves value 192 to the register.	R/W	0
23:16	PRI_(4*n+2)	Each priority field holds a priority value, 0-192. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[23:22] of each field, bits [21:16] read as zero and ignore writes. This means writing 255 to a priority register saves value 192 to the register.	R/W	0
15:8	PRI_(4*n+1)	Each priority field holds a priority value, 0-192. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[15:14] of each field, bits [13:8] read as zero and ignore writes. This means writing 255 to a priority register saves value 192 to the register.	R/W	0
7:0	PRI_4*n	Each priority field holds a priority value, 0-192. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[7:6] of each field, bits [5:0] read as zero and ignore writes. This means writing 255 to a priority register saves value 192 to the register.	R/W	0

2.4 APPLICATION INTERRUPT AND RESET CONTROL (AIRC)

Address: 0xE000 ED0C (Refer to Cortex-M0 Spec)

The entire MCU, including the core, can be reset by SW by setting the SYSRESREQ bit in the AIRC register in Cortex-

M0 spec.

✷ ***Note: To write to this register, user must write 0x05FA to the VECTKEY field at the same time, otherwise the processor ignores the write.***

Bit	Name	Description	Attribute	Reset
31:16	VECTKEY	Register key. Read as unknown. Write 0x05FA to VECTKEY, otherwise the write is ignored.	R/W	0
15	ENDIANESS	Data endianness implemented 0: Little-endian 1: Big-endian	R	0
14:3	Reserved		R	0
2	SYSRESETREQ	System reset request. This bit read as 0. 0: No effect 1: Requests a system level reset.	W	0
1	VECTCLRACTIVE	Reserved for debug use. This bit read as 0. When writing to the register you must write 0 to this bit, otherwise behavior is Unpredictable.	W	0
0	Reserved		R	0

2.5 CODE OPTION TABLE

Address: 0x1FFF 2000

Bit	Name	Description	Attribute	Reset
31:16	Code Security[15:0]	Code Security 0xFFFF: CS0 0x5A5A: CS1 0xA5A5: CS2	R/W	0xFFFF
15:3	Reserved		R/W	0x1FFF
2	BOOTPINEN	Boot Pin enable 0: Disable 1: Enable	R/W	1
1	EXTRSTHWDIS	External reset HW disable bit 0: Enable (HW enables, FW can NOT control with SYS0_EXRSTCTRL) 1: Disable (FW can control with SYS0_EXRSTCTRL)	R/W	1
0	BLEN	Boot loader enable 0: Disable 1: Enable	R/W	1

2.6 UNIQUE NUMBER

The unique number is an 8-byte unique device serial number of each IC. In other words, the unique number is different and discontinuous for each IC.

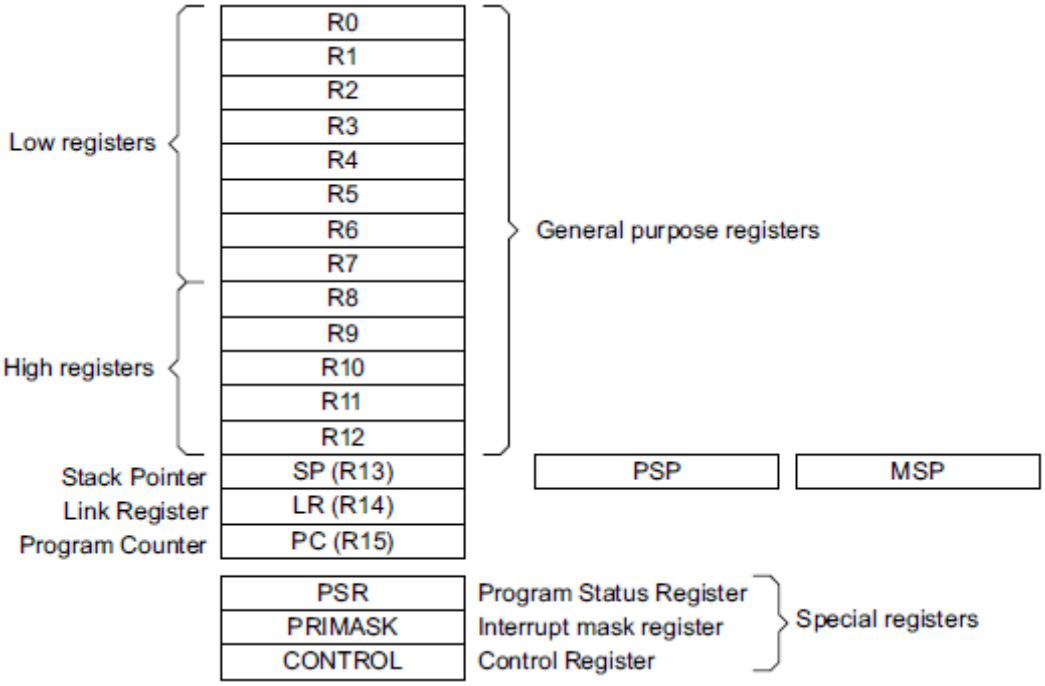
Address: 0x1FFF 2DF0

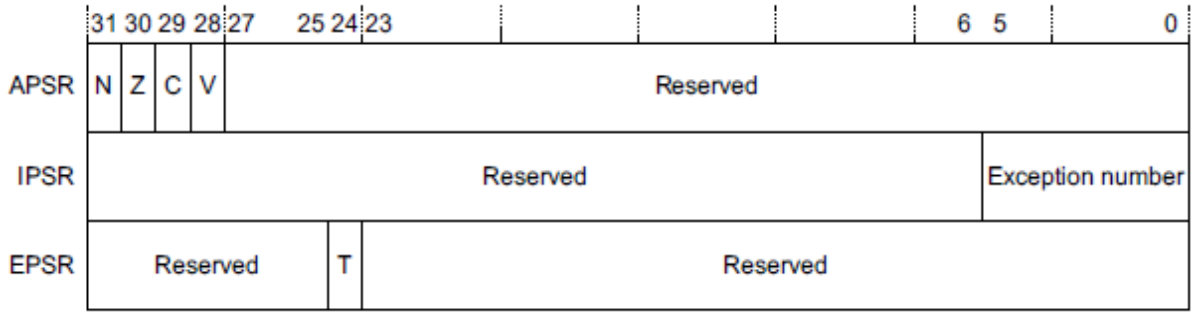
Bit	Name	Description	Attribute	Reset
31:0	L4BYTE[31:0]	Lower 4 bytes of Unique number	R	By Die

Address: 0x1FFF 2DF8

Bit	Name	Description	Attribute	Reset
31:0	H4BYTE[31:0]	High 4 bytes of Unique number	R	By Die

2.7 CORE REGISTER OVERVIEW



Register	Description (Refer to Cortex-M0 Spec)
R0~R12	General-purpose registers for data operations.
SP (R13)	The Stack Pointer (SP). In Thread mode, the CONTROL register indicates the stack pointer to use, Main Stack Pointer (MSP) or Process Stack Pointer (PSP) On reset, the processor loads the MSP with the value from address 0x00000000.
LR (R14)	The Link Register (LR). It stores the return information for subroutines, function calls, and exceptions.
PC (R15)	The Program Counter (PC). It contains the current program address. On reset, the processor loads the PC with the value of the reset vector, at address 0x00000004.
PSR	The Program Status Register (PSR) combines: • Application Program Status Register (APSR) • Interrupt Program Status Register (IPSR) • Execution Program Status Register (EPSR). These registers are mutually exclusive bit fields in the 32-bit PSR. 
PRIMASK	The PRIMASK register prevents activation of all exceptions with configurable priority.
CONTROL	The CONTROL register controls the stack used when the processor is in Thread mode.

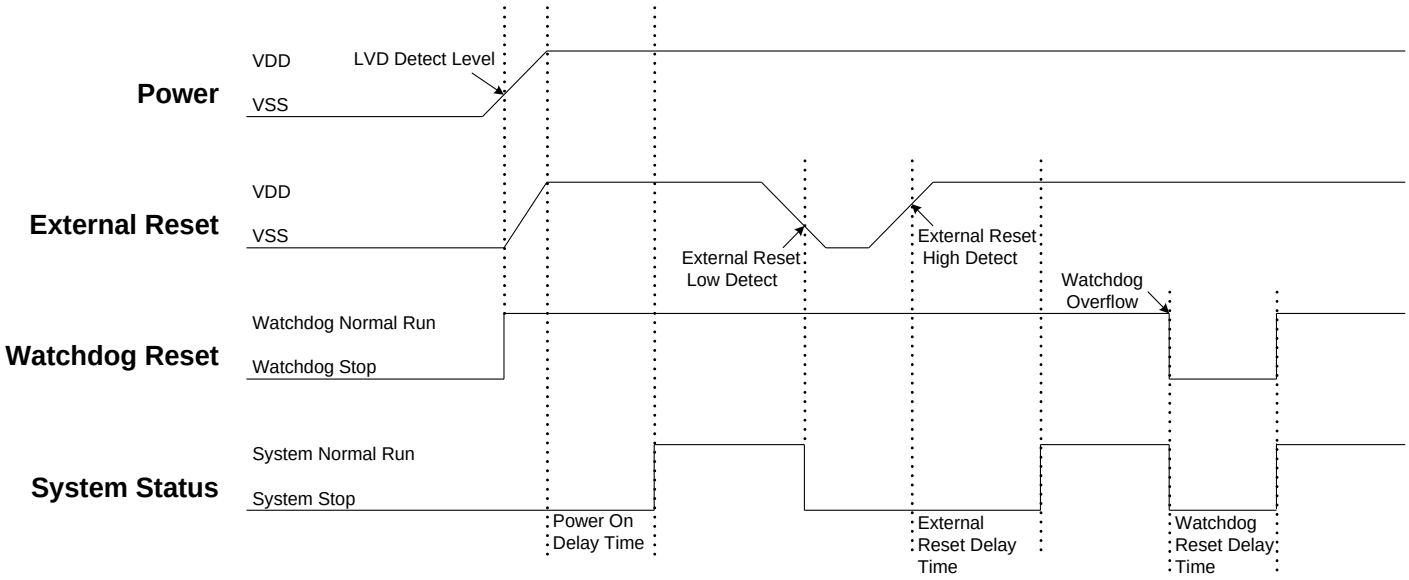
3 SYSTEM CONTROL

3.1 RESET

A system reset is generated when one of the following events occurs:
1. A low level on the RST pin (external reset).
2. Power-on reset (POR reset)
3. LVD reset
4. Watchdog Timer reset (WDT reset)
5. Software reset (SW reset)

The reset source can be identified by checking the reset flags in System Reset Status register (SYS0_RSTST). These sources act on the RST pin and it is always kept low during the delay phase. The RESET service routine vector is fixed at address 0x00000004 in the memory map. For more details, refer to Interrupt and Exception Vectors.

Finishing any reset sequence needs some time. The system provides complete procedures to make the power on reset successful. For different oscillator types, the reset time is different. That causes the VDD rise rate and start-up time of different oscillator is not fixed. RC type oscillator's start-up time is very short, but the crystal type is longer. Under client terminal application, users have to take care of the power on reset time for the master terminal requirement. The reset timing diagram is as following.



3.1.1 POWER-ON RESET (POR)

The power on reset depends on LVD operation for most power-up situations. The power supplying to system is a rising curve and needs some time to achieve the normal voltage. Power on reset sequence is as following:

- **Power-up:** System detects the power voltage up and waits for power stable.
- **External reset (only external reset pin enable):** System checks external reset pin status. If external reset pin is not high level, the system keeps reset status and waits external reset pin released.
- **System initialization:** All system registers is set as initial conditions and system is ready.
- **Oscillator warm up:** Oscillator operation is successfully and supply to system clock.
- **Program executing:** Power on sequence is finished and program executes from Boot loader.

3.1.2 WATCHDOG RESET (WDT RESET)

Watchdog reset is a system protection. In normal condition, system works well and clears watchdog timer by program. Under error condition, system is in unknown situation and watchdog can't be clear by program before watchdog timer overflow. Watchdog timer overflow occurs and the system is reset. After watchdog reset, the system restarts and returns normal mode. Watchdog reset sequence is as following.

- **Watchdog timer status:** System checks watchdog timer overflow status. If watchdog timer overflow occurs, the system is reset.
- **System initialization:** All system registers is set as initial conditions and system is ready.
- **Oscillator warm up:** Oscillator operation is successfully and supply to system clock.
- **Program executing:** Power on sequence is finished and program executes from 0x0.

Watchdog timer application note is as following.

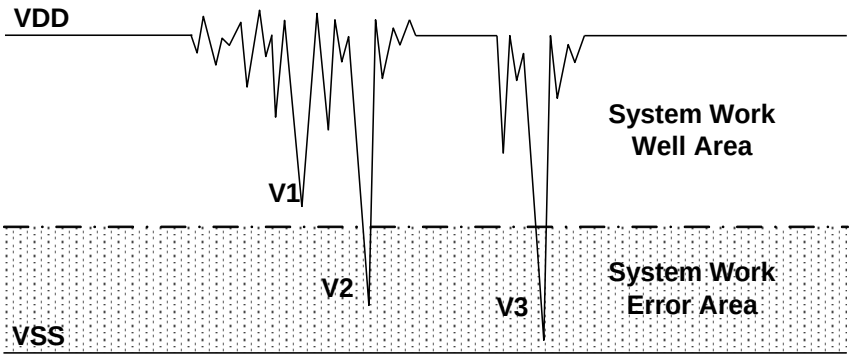
- Before clearing watchdog timer, check I/O status and check RAM contents can improve system error.
- Don't clear watchdog timer in interrupt vector and interrupt service routine. That can improve main routine fail.
- Clearing watchdog timer program is only at one part of the program. This way is the best structure to enhance the watchdog timer function.

✸ ***Note: Please refer to the "WATCHDOG TIMER" about watchdog timer detail information.***

3.1.3 BROWN-OUT RESET

3.1.3.1 BROWN OUT DESCRIPTION

The brown-out reset is a power dropping condition. The power drops from normal voltage to low voltage by external factors (e.g. EFT interference or external loading changed). The brown out reset would make the system not work well or executing program error.



Brown-Out Reset Diagram

The power dropping might through the voltage range that's the system dead-band. The dead-band means the power range can't offer the system minimum operation power requirement. The above diagram is a typical brown out reset diagram. There is a serious noise under the VDD, and VDD voltage drops very deep. There is a dotted line to separate the system working area. The above area is the system work well area. The below area is the system work error area called dead-band. V1 doesn't touch the below area and not affect the system operation. But the V2 and V3 is under the below area and may induce the system error occurrence. Let system under dead-band includes some conditions.

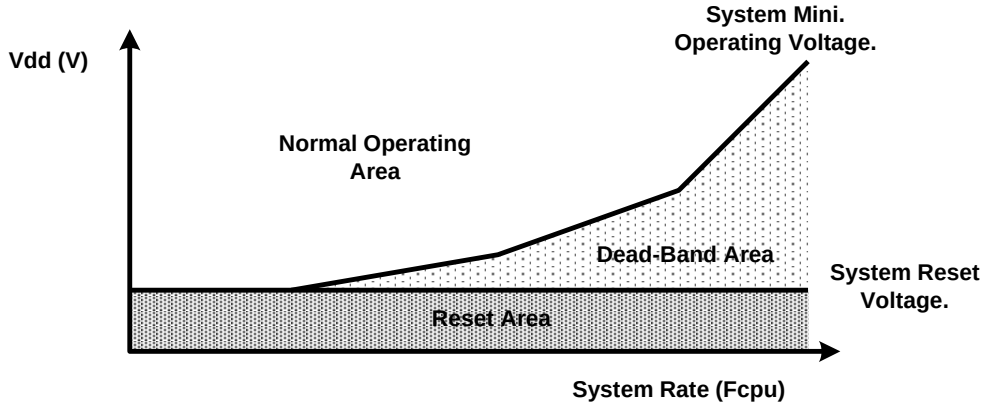
DC application:
The power source of DC application is usually using battery. When low battery condition and MCU drive any loading, the power drops and keeps in dead-band. Under the situation, the power won't drop deeper and not touch the system reset voltage. That makes the system under dead-band.

AC application:
In AC power application, the DC power is regulated from AC power source. This kind of power usually couples with AC noise that makes the DC power dirty. Or the external loading is very heavy, e.g. driving motor. The loading operating induces noise and overlaps with the DC power. VDD drops by the noise, and the system works under unstable power situation.

The power on duration and power down duration are longer in AC application. The system power on sequence protects the power on successful, but the power down situation is like DC low battery condition. When turn off the AC power, the VDD drops slowly and through the dead-band for a while.

3.1.3.2 THE SYSTEM OPERATING VOLTAGE DECSRIPTION

To improve the brown out reset needs to know the system minimum operating voltage which is depend on the system executing rate and power level. Different system executing rates have different system minimum operating voltage. The electrical characteristic section shows the system voltage to executing rate relationship.



Normally the system operation voltage area is higher than the system reset voltage to VDD, and the reset voltage is decided by LVD detect level. The system minimum operating voltage rises when the system executing rate upper even higher than system reset voltage. The dead-band definition is the system minimum operating voltage above the system reset voltage.

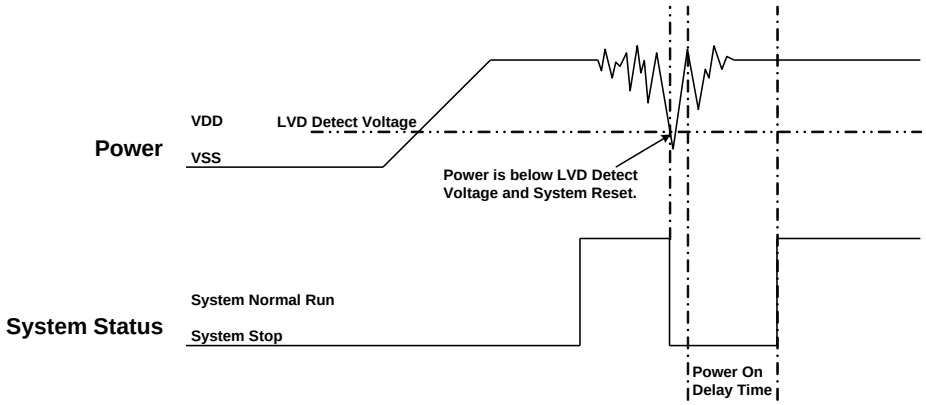
3.1.3.3 BROWN-OUT RESET IMPROVEMENT

How to improve the brown reset condition? There are some methods to improve brown out reset as following.

- **LVD reset**
- **Watchdog reset**
- **Reduce the system executing rate**
- **External reset circuit. (Zener diode reset circuit, Voltage bias reset circuit, External reset IC)**

✷ ***Note: The "Zener diode reset circuit", "Voltage bias reset circuit" and "External reset IC" can completely improve the brown out reset, DC low battery and AC slow power down conditions.***

LVD reset:



The LVD (low voltage detector) is built-in SONiX 32-bit MCU to be brown out reset protection. When the VDD drops and is below LVD detect voltage, the LVD asserts an interrupt signal to the NVIC. This signal can be enabled for interrupt in the Interrupt Enable Register in the NVIC in order to cause a CPU interrupt; if not, SW can monitor the signal by reading a dedicated status register. An additional threshold level can be selected to cause a forced reset of the chip. The LVD detect level is different by each MCU. The LVD voltage level is a point of voltage and not easy to cover all dead-band range. Using LVD to improve brown out reset is dependent on application requirement and environment. If the power variation is very deep, violent and trigger the LVD, the LVD can be the protection. If the power variation can touch the LVD detect level and make system work error, the LVD can't be the protection and need to other reset methods. More detail LVD information is in the electrical characteristic section.

Watchdog reset:
The watchdog timer is a protection to make sure the system executes well. Normally the watchdog timer would be clear at one point of program. Don't clear the watchdog timer in several addresses. The system executes normally and the watchdog won't reset system. When the system is under dead-band and the execution error, the watchdog timer can't be clear by program. The watchdog is continuously counting until overflow occurrence. The overflow signal of watchdog timer triggers the system to reset and return to normal mode after reset sequence. This method also can improve brown out reset condition and make sure the system to return normal mode.
If the system reset by watchdog and the power is still in dead-band, the system reset sequence won't be successful and the system stays in reset status until the power return to normal range.

Reduce the system executing rate:
If the system rate is fast and the dead-band exists, to reduce the system executing rate can improve the dead-band. The lower system rate is with lower minimum operating voltage. Select the power voltage that's no dead-band issue and find out the mapping system rate. Adjust the system rate to the value and the system exits the dead-band issue. This way needs to modify whole program timing to fit the application requirement.

External reset circuit:
The external reset methods also can improve brown out reset and is the complete solution. There are three external reset circuits to improve brown out reset including "Zener diode reset circuit", "Voltage bias reset circuit" and "External reset IC". These three reset structures use external reset signal and control to make sure the MCU be reset under power dropping and under dead-band. The external reset information is described in the next section.

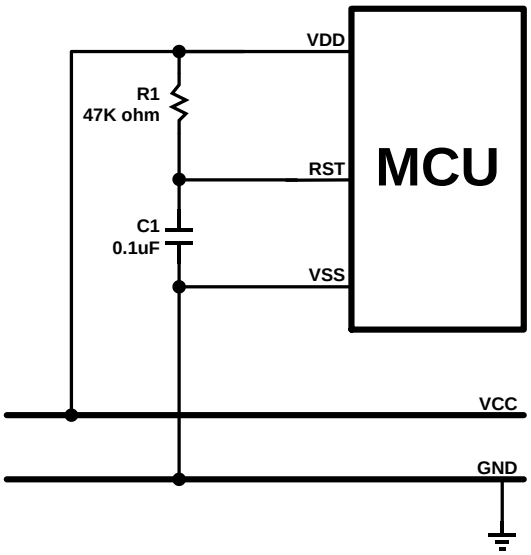
3.1.4 EXTERNAL RESET

External reset function is controlled by External RESET pin control (SYS0_EXRSTCTRL) register. Default value is 1, which means external reset function is disabled. External reset pin is Schmitt Trigger structure and low level active. The system is running when reset pin is high level voltage input. The reset pin receives the low voltage and the system is reset. The external reset operation actives in power on and normal running mode. During system power-up, the external reset pin must be high level input, or the system keeps in reset status. External reset sequence is as following.

- **External reset (only external reset pin enable):** System checks external reset pin status. If external reset pin is not high level, the system keeps reset status and waits external reset pin released.
- **System initialization:** All system registers is set as initial conditions and system is ready.
- **Oscillator warm up:** Oscillator operation is successfully and supply to system clock.
- **Program executing:** Power on sequence is finished and program executes from Boot loader.

The external reset can reset the system during power on duration, and good external reset circuit can protect the system to avoid working at unusual power condition, e.g. brown out reset in AC power application.

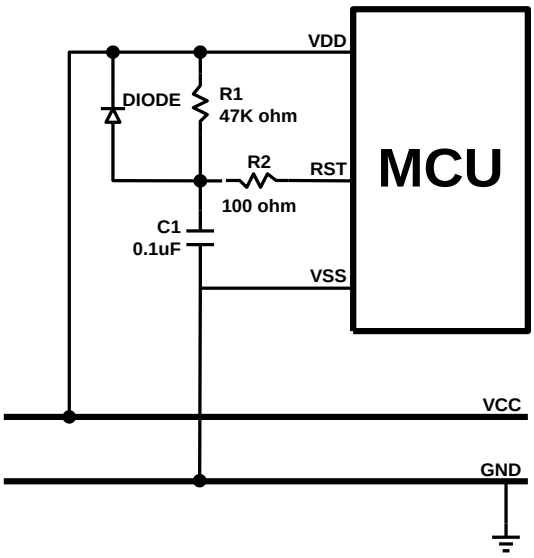
3.1.4.1 SIMPLY RC RESET CIRCUIT



This is the basic reset circuit, and only includes R1 and C1. The RC circuit operation makes a slow rising signal into reset pin as power up. The reset signal is slower than VDD power up timing, and system occurs a power on signal from the timing difference.

✸ ***Note: The reset circuit is no any protection against unusual power or brown out reset.***

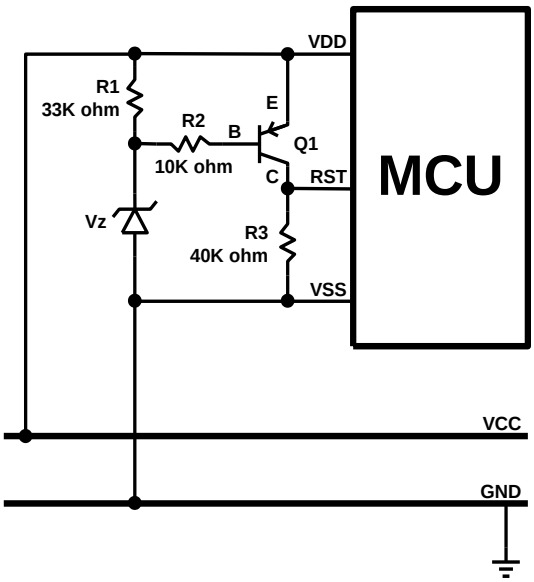
3.1.4.2 DIODE & RC RESET CIRCUIT



This is the better reset circuit. The R1 and C1 circuit operation is like the simply reset circuit to make a power on signal. The reset circuit has a simply protection against unusual power. The diode offers a power positive path to conduct higher power to VDD. It is can make reset pin voltage level to synchronize with VDD voltage. The structure can improve slight brown out reset condition.

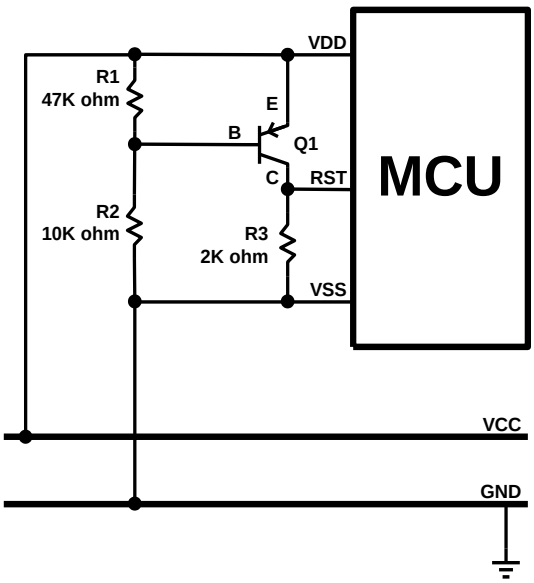
✷ ***Note: The R2 100 ohm resistor of "Simply reset circuit" and "Diode & RC reset circuit" is necessary to limit any current flowing into reset pin from external capacitor C in the event of reset pin breakdown due to Electrostatic Discharge (ESD) or Electrical Over-stress (EOS).***

3.1.4.3 ZENER DIODE RESET CIRCUIT



The Zener diode reset circuit is a simple low voltage detector and can **improve brown out reset condition completely**. Use Zener voltage to be the active level. When VDD voltage level is above "Vz + 0.7V", the C terminal of the PNP transistor outputs high voltage and MCU operates normally. When VDD is below "Vz + 0.7V", the C terminal of the PNP transistor outputs low voltage and MCU is in reset mode. Decide the reset detect voltage by Zener specification. Select the right Zener voltage to conform the application.

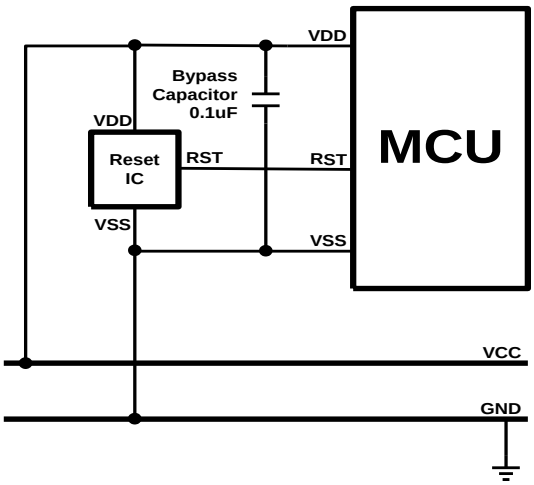
3.1.4.4 VOLTAGE BIAS RESET CIRCUIT



The voltage bias reset circuit is a low cost voltage detector and can **improve brown out reset condition completely**. The operating voltage is not accurate as Zener diode reset circuit. Use R1, R2 bias voltage to be the active level. When VDD voltage level is above or equal to “0.7V x (R1 + R2) / R1”, the C terminal of the PNP transistor outputs high voltage and MCU operates normally. When VDD is below “0.7V x (R1 + R2) / R1”, the C terminal of the PNP transistor outputs low voltage and MCU is in reset mode.
Decide the reset detect voltage by R1, R2 resistances. Select the right R1, R2 value to conform the application. In the circuit diagram condition, the MCU’s reset pin level varies with VDD voltage variation, and the differential voltage is 0.7V. If the VDD drops and the voltage lower than reset pin detect level, the system would be reset. If want to make the reset active earlier, set the R2 > R1 and the cap between VDD and C terminal voltage is larger than 0.7V. The external reset circuit is with a stable current through R1 and R2. For power consumption issue application, e.g. DC power system, the current must be considered to whole system power consumption.

✷ ***Note: Under unstable power condition as brown out reset, “Zener diode reset circuit” and “Voltage bias reset circuit” can protects system no any error occurrence as power dropping. When power drops below the reset detect voltage, the system reset would be triggered, and then system executes reset sequence. That makes sure the system work well under unstable power situation.***

3.1.4.5 EXTERNAL RESET IC



The external reset circuit also uses external reset IC to enhance MCU reset performance. This is a high cost and good effect solution. By different application and system requirement to select suitable reset IC. The reset circuit can improve all power variation.

3.1.5 SOFTWARE RESET

The entire MCU, including the core, can be reset by software by setting the SYSRESREQ bit in the AIRC (Application Interrupt and Reset Control) register in Cortex-M0 spec.

The software-initiated system reset sequence is as follows:
1. A software reset is initiated by setting the SYSRESREQ bit.
2. An internal reset is asserted.
3. The internal reset is deasserted and the MCU loads from memory the initial stack pointer, the initial program counter, and the first instruction designated by the program counter, and then begins execution.

3.2 SYSTEM CLOCK

Different clock sources can be used to drive the system clock (SYSCLK):

- 48 MHz internal high speed RC (IHRC)
- 32 KHz internal low speed RC (ILRC)
- PLL clock
- High speed external (EHS) crystal clock
- Low speed external (ELS) 32.768 KHz crystal

Each clock source can be switched on or off independently when it is not used, to optimize power consumption.

The micro-controller is a dual clock system. There are high-speed clock and low-speed clock. The high-speed clock is generated from the external oscillator & on-chip PLL circuit. The low-speed clock is generated from on-chip low-speed RC oscillator circuit (ILRC 32KHz).

3.2.1 INTERNAL RC CLOCK SOURCE

3.2.1.1 Internal High-speed RC Oscillator (IHRC)

The internal high-speed oscillator is 48MHz RC type. The accuracy is ±2.5% under commercial condition.
The IHRC can be switched on and off using the IHRCEN bit in Analog Block Control register (SYS0_ANBCTRL).

3.2.1.2 Internal Low-speed RC Oscillator (ILRC)

The system low clock source is the internal low-speed oscillator built in the micro-controller. The low-speed oscillator uses RC type oscillator circuit. The frequency is affected by the voltage and temperature of the system. In common condition, the frequency of the RC oscillator is about 32 KHz.

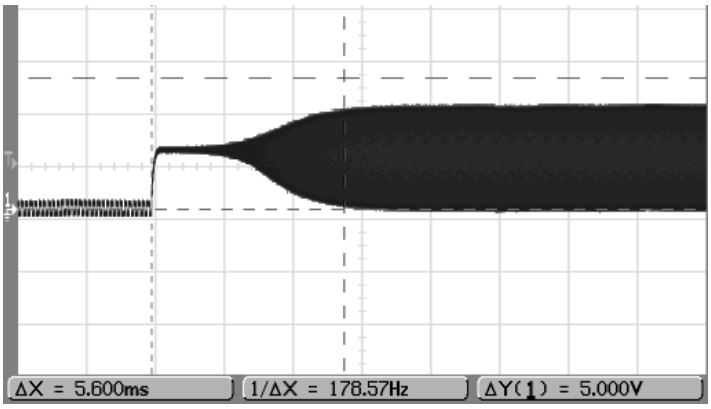
✸	***Note: The ILRC can ONLY be switched on and off by HW.***

3.2.2 EXTERNAL CLOCK SOURCE

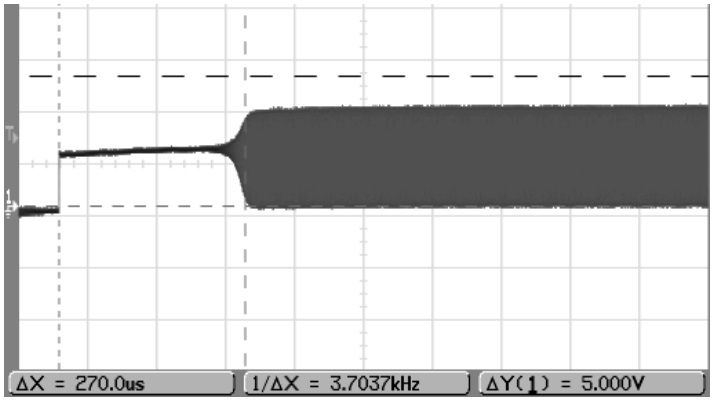
3.2.2.1 External High-speed (EHS) Clock

External high clock includes Crystal/Ceramic modules. The startup time of Crystal is longer. The oscillator start-up time decides reset time length.

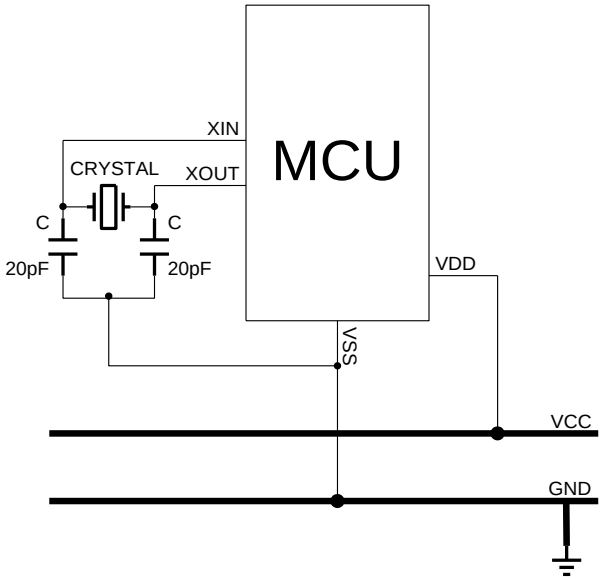
4MHz Crystal



4MHz Ceramic



Crystal/Ceramic devices are driven by XIN, XOUT pins. For high/normal/low frequency, the driving currents are different.



✷ ***Note: Connect the Crystal/Ceramic and C as near as possible to the XIN/XOUT/VSS pins of MCU.***

- **Structure:** 4MHz~25MHz EHS external crystal/ceramic resonator
- **Main Purpose:** System high clock source, RTC clock source, and PLL clock source.
- **Warm-up Time:** 2048*F_{EHS}
- **XIN/XOUT Shared Pin Selection:**

Oscillator Mode	XIN pin	XOUT pin
IHRC	GPIO	GPIO
EHS X'TAL	Crystal/Ceramic	Crystal/Ceramic

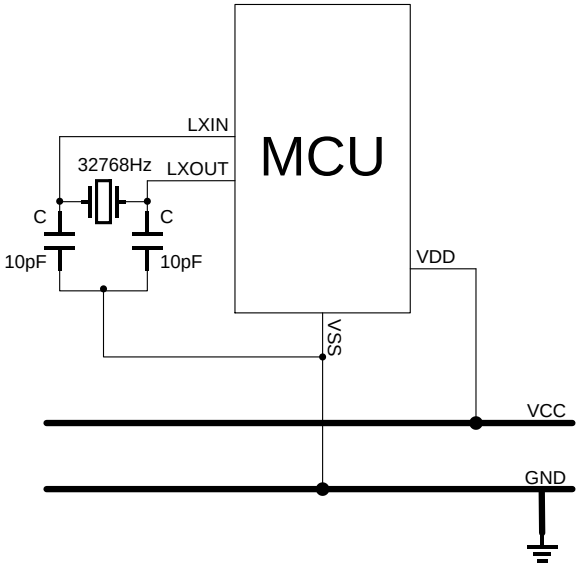
The resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and startup stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

The EHS crystal is switched on and off using the EHSEN bit in Analog Block Control register (SYS0_ANBCTRL).

3.2.3 External Low-speed (ELS) Clock

The low-speed oscillator can use 32768Hz crystal oscillator circuit.

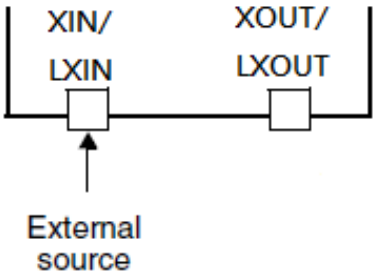
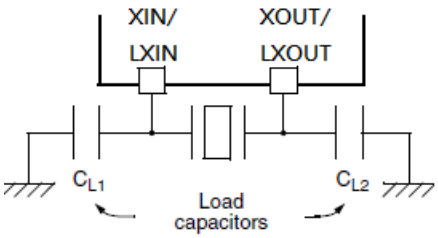
Crystal devices are driven by LXIN, LXOUT pins. The 32768 crystal and 10pF capacitor must be as near as possible to MCU. The ELS crystal is switched on and off using the ELSEN bit in Analog Block Control register (SYS0_ANBCTRL).



✸ ***Note: Connect the Crystal/Ceramic and C as near as possible to the LXIN/LXOUT/VSS pins of MCU. The capacitor between LXIN/LXOUT and VSS must be 10pF.***

✸ ***Note: When the ELS operates above 105℃ and VDD is less than 3V, a 15MΩ resistor needs to be connected in parallel to LXIN/LXOUT to ensure normal start-up.***

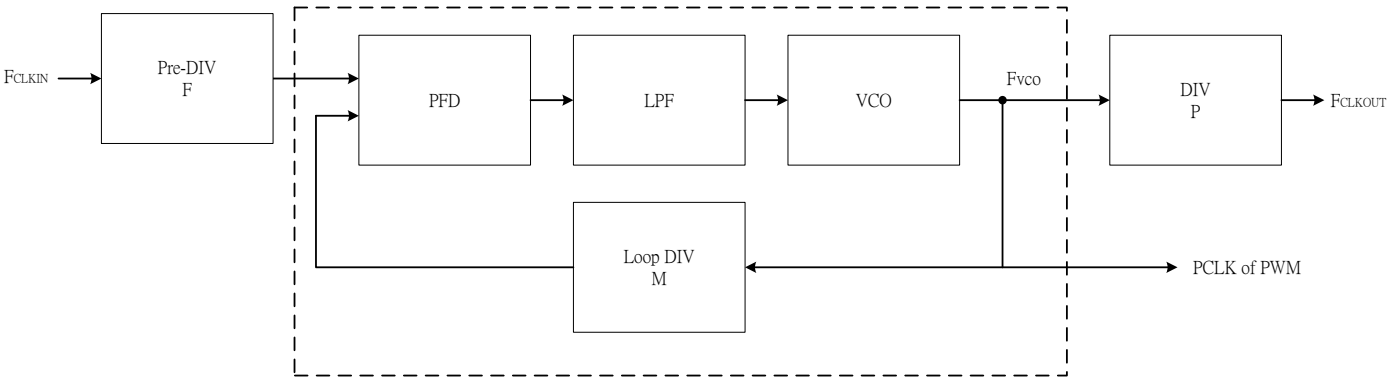
3.2.3.1 Bypass Mode

Clock Source	H/W Configuration	Description
External clock source (Bypass)		In Bypass mode, the external clock signal (square, sinus or triangle) with ~50% duty cycle must be provided to drive the XIN/LXIN pin while the XOUT/LXOUT pin should be the inverse of the input clock signal. EHS X'tal can have a frequency of up to 25 MHz Select this mode by setting EHSEN bit in Analog Block Control register (SYS0_ANBCTRL). ELS X'TAL must have a frequency of 32.768 KHz. You select this mode by setting ELSEN bit in Analog Block Control register (SYS0_ANBCTRL).
External X'TAL (EHS/ELS X'TAL)		The 4 to 25 MHz EHS X'TAL has the advantage of producing a very accurate rate on the main clock ELS X'TAL must have a frequency of 32.768 KHz.

3.2.4 PLL

SONiX 32-bit MCU uses the PLL to create the clocks for the core and peripherals. The input frequency range is 4MHz to 25MHz. The input clock is divided down and fed to the Phase-Frequency Detector (PFD). This block compares the phase and frequency of its inputs, and generates a control signal when phase and/ or frequency do not match. The loop filter filters these control signals and drives the voltage controlled oscillator (VCO), which generates the main clock and optionally two additional phases. The VCO frequency range is 96MHz to 120MHz. These clocks are divided by P by the programmable post divider to create the output clock(s). The VCO output clock is then divided by M by the programmable feedback divider to generate the feedback clock. The output signal of the phase-frequency detector is also monitored by the lock detector, to signal when the PLL has locked on to the input clock.

The PLL settling time is 100 μs.



3.2.4.1 PLL Frequency selection

The PLL frequency equations:

$$F_{VCO} = F_{CLKIN} / F * M$$

$$F_{CLKOUT} = F_{VCO} / P$$

The PLL frequency is determined by the following parameters:

- F_{CLKIN}: Frequency from the PLLCLKSEL multiplexer.
- F_{VCO}: Frequency of the Voltage Controlled Oscillator (VCO); 96 to 120 MHz
- F_{CLKOUT}: Frequency of PLL output.
- P: System PLL post divider ratio, controlled by PSEL bits in PLL control register (SYS0_PLLCTRL).
- F: System PLL front divider ratio, controlled by FSEL bits in PLL control register (SYS0_PLLCTRL).
- M: System PLL feedback divider ratio, controlled by MSEL bits in PLL control register (SYS0_PLLCTRL).

To select the appropriate values for M, P, and F, it is recommended to follow these constraints:

1. $4MHz \le F_{CLKIN} \le 25MHz$
2. $4MHz \le (F_{CLKIN} / F) \le 16MHz$
3. $96MHz \le F_{VCO} \le 120MHz$
4. $1 \le F \le 3$
5. $2 \le M \le 31$
6. P = 2 or 4
7. F_{CLKOUT} = 24MHz, 30MHz,48MHz, 50MHz, 60MHz with jitter < ±500 ps

3.2.4.2 PLL Recommended Frequency settings

F_{CLKIN} (MHz)	F[1:0] 1~3	F_{CLKIN} / F 4~16 MHz	M[4:0] 2~31	F_{VCO}(MHz)= F_{CLKIN} / F * M 96 ~ 120 MHz	PSEL	P	F_{CLKOUT}(MHz)= F_{VCO} / P
4	1	4	30	**120**	**0**	**2**	60
4	1	4	30	**120**	**1**	**4**	30
10	1	10	12	**120**	**0**	**2**	60
10	1	10	10	**100**	**0**	**2**	50
10	1	10	12	**120**	**1**	**4**	30
12	1	12	8	**96**	**0**	**2**	48
12	1	12	10	**120**	**0**	**2**	60
16	1	16	6	**96**	**0**	**2**	48
25	2	12.5	8	**100**	**0**	**2**	50

3.2.5 SYSTEM CLOCK (SYSCLK) SELECTION

After a system reset, the IHRC is selected as system clock. When a clock source is used directly or through the PLL as system clock, it is not possible to stop it.

A switch from one clock source to another occurs only if the target clock source is ready (clock stable after startup delay or PLL locked). If a clock source which is not yet ready is selected, the switch will occur when the clock source is ready.

Ready bits in SYS0_CSST register indicate which clock(s) is (are) ready and SYSCLKST bits in SYS0_CLKCFG register indicate which clock is currently used as system clock.

3.2.6 CLOCK-OUT CAPABITITY

The MCU clock output (CLKOUT) capability allows the clock to be output onto the external CLKOUT pin. The configuration registers of the corresponding GPIO port must be programmed in alternate function mode.

One of 6 clock signals can be selected as clock output:
1. HCLK
2. IHRC
3. ILRC
4. PLL clock output
5. ELS X'TAL
6. EHS X'TAL

The selection is controlled by the CLKOUTSEL bits in SYS1_AHBCLKEN register.

3.3 SYSTEM CONTROL REGISTERS 0

Base Address: 0x4006 0000

3.3.1 Analog Block Control register (SYS0_ANBCTRL)

Address Offset: 0x00

Reset value: 0x0000 0001

> ✸ ***Note: EHSEN / ELSEN / IHRCEN bit can NOT be cleared if the EHS X'tal / ELS X'tal / IHRC is selected as system clock or is selected to become the system clock.***

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
5	EHSFREQ	Frequency range (driving ability) of EHS X'TAL 0: <=4MHz 1: >4MHz	R/W	0
4	EHSEN	External high-speed clock enable 0: Disable EHS X'TAL. 1: Enable EHS X'TAL.	R/W	0
3	Reserved		R	0
2	ELSEN	External low-speed oscillator enable 0: Disable External 32.768 KHz oscillator 1: Enable External 32.768 KHz oscillator	R/W	0
1	IHRCFREQ	IHRC frequency 0: IHRC 12MHz 1: IHRC 48MHz	RW	0
0	IHRCEN	Internal high-speed clock enable 0: Disable internal 48 MHz RC oscillator. 1: Enable internal 48 MHz RC oscillator.	R/W	1

3.3.2 PLL control register (SYS0_PLLCTRL)

Address Offset: 0x04

> ✸ ***Note: PLLEN bit can NOT be cleared if the PLL is selected as system clock or is selected to become the system clock.***

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15	PLLEN	PLL enable 0: Disable 1: Enable	R/W	0
14:13	Reserved		R	0
12	PLLCLKSEL	System PLL clock source 0: IHRC 1: EHS X'TAL 4MHz~25MHz	R/W	0
11:10	Reserved		R	0
9:8	FSEL[1:0]	Front divider value F: 1~3	R/W	01b

7:6	Reserved		R	0
5	PSEL	Post divider value. 0: P=2 1: P=4	R/W	0
4:0	MSEL[4:0]	Loop divider value. M: 2~31	R/W	01000b

3.3.2.1 RECOMMEND FREQUENCY SETTING

$F_{VCO} = F_{CLKIN} / F * M$

$F_{CLKOUT} = F_{VCO} / P$

F_{CLKIN} (MHz)	**F[1:0] 1~3**	**F_{CLKIN} / F 4~16 MHz**	**M[4:0] 2~31**	**F_{VCO}(MHz)= F_{CLKIN} / F * M 96 ~ 120 MHz**	**PSEL**	**P**	**F_{CLKOUT}(MHz)= F_{VCO} / P**
4	1	4	30	120	0	2	60
4	1	4	30	120	1	4	30
10	1	10	12	120	0	2	60
10	1	10	10	100	0	2	50
10	1	10	12	120	1	4	30
12	1	12	8	96	0	2	48
12	1	12	10	120	0	2	60
16	1	16	6	96	0	2	48
25	2	12.5	8	100	0	2	50

3.3.3 Clock Source Status register (SYS0_CSST)

Address Offset: 0x08

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6	PLLRDY	PLL clock ready flag 0: PLL unlocked 1: PLL locked	R	0
5	Reserved		R	0
4	EHSRDY	External high-speed clock ready flag 0: EHS oscillator not ready 1: EHS oscillator ready	R	0
3	Reserved		R	0
2	ELSRDY	External low-speed clock ready flag 0: ELS oscillator not ready 1: ELS oscillator ready	R	0
1	Reserved		R	0
0	IHRCRDY	IHRC ready flag 0: IHRC not ready 1: IHRC ready	R	1

3.3.4 System Clock Configuration register (SYS0_CLKCFG)

Address Offset: 0x0C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:4	SYSCLKST[2:0]	System clock switch status Set and cleared by HW to indicate which clock source is used as system clock. 000: IHRC is used as system clock 001: ILRC is used as system clock 010: EHS X'TAL is used as system clock 011: ELS X'TAL is used as system clock 100: PLL is used as system clock Other: Reserved	R	0
3	Reserved		R	0
2:0	SYSCLKSEL[2:0]	System clock switch Set and cleared by SW. 000: IHRC 001: ILRC 010: EHS X'TAL 011: ELS X'TAL 100: PLL output Other: Reserved	R/W	0

3.3.5 AHB Clock Prescale register (SYS0_AHBCP)

Address Offset: 0x10

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	DIV1P5	SYSCLK prescaler 0: SYSCLK = SYSCLK clock source / 1 1: SYSCLK = SYSCLK clock source / 1.5	R/W	0
2:0	AHBPRE[3:0]	AHB clock source prescale value 000: SYSCLK / 1 001: SYSCLK / 2 010: SYSCLK / 4 011: SYSCLK / 8 100: SYSCLK / 16 101: SYSCLK / 32 110: SYSCLK / 64 111: SYSCLK / 128	R/W	0

3.3.6 System Reset Status register (SYS0_RSTST)

Address Offset: 0x14

This register contains the reset source except DPDWAKEUP reset, since the MODE bits in PMU_CTRL register had presented this case.

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	PORRSTF	POR reset flag Set by HW when a POR reset occurs. 0: Read→No POR reset occurred Write→Clear this bit 1: POR reset occurred.	R/W	1
3	EXTRSTF	External reset flag Set by HW when a reset from the RESET pin occurs. 0: Read→No reset from RESET pin occurred Write→Clear this bit	R/W	0

		1: Reset from RESET pin occurred.		
2	LVDRSTF	LVD reset flag Set by HW when a LVD reset occurs. 0: Read→No LVD reset occurred Write→Clear this bit 1: LVD reset occurred.	R/W	0
1	WDTRSTF	WDT reset flag Set by HW when a WDT reset occurs. 0: Read→No watchdog reset occurred Write→Clear this bit 1: Watchdog reset occurred.	R/W	0
0	SWRSTF	Software reset flag Set by HW when a software reset occurs. 0: Read→No software reset occurred Write→Clear this bit 1: Software reset occurred.	R/W	1

3.3.7 LVD Control register (SYS0_LVDCTRL)

Address Offset: 0x18

The LVD control register selects six separate threshold values for generating a LVD interrupt to the NVIC or LVD reset.

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15	LVDEN	LVD enable 0: Disable 1: Enable	R/W	0
14	LVDRSTEN	LVD Reset enable 0: Disable 1: Enable	R/W	0
13:7	Reserved		R	0
6:4	LVDINTLVL[2:0]	LVD interrupt level 0: Reserved for system 1: LVD interrupt threshold is 2.10V 2: LVD interrupt threshold is 2.50V 3: LVD interrupt threshold is 2.90V 4: LVD interrupt threshold is 3.30V 5: LVD interrupt threshold is 3.70V 6: LVD interrupt threshold is 4.10V 7: LVD interrupt threshold is 4.50V	R/W	0
3	Reserved		R	0
2:0	LVDRSTLVL[2:0]	LVD reset level 0: Reserved for system 1: LVD interrupt threshold is 2.10V 2: LVD interrupt threshold is 2.50V 3: LVD interrupt threshold is 2.90V 4: LVD interrupt threshold is 3.30V 5: LVD interrupt threshold is 3.70V 6: LVD interrupt threshold is 4.10V 7: LVD interrupt threshold is 4.50V	R/W	0

3.3.8 External RESET Pin Control register (SYS0_EXRSTCTRL)

Address Offset: 0x1C

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	RESETDIS	External RESET pin disable bit. 0: Enable external RESET pin. (P3.7 acts as RESET pin) 1: Disable. (P3.7 acts as GPIO pin)	R/W	1

3.3.9 SWD Pin Control register (SYS0_SWDCTRL)

Address Offset: 0x20

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	SWDDIS	SWD pin disable bit. 0: Enable SWD pin. (P3.5 acts as SWDIO pin, P3.6 acts as SWCLK pin) 1: Disable. (P3.5 and P3.6 act as GPIO pins)	R/W	0

3.3.10 Interrupt Vector Table Mapping register (SYS0_IVTM)

Address Offset: 0x24

This register decides whether the ARM interrupt vector table is mapping to Boot ROM or User ROM

Bit	Name	Description	Attribute	Reset
31:16	IVTMKEY[15:0]	IVTM register key. Read as 0. Behaviour of writing to this register is ignored unless writing 0xA5A5 to IVTMKEY at the same time.	W	0
15:2	Reserved		R	0
1:0	IVTM[1:0]	Interrupt table mapping selection 00: Map to Boot ROM 01: Map to User ROM Other: Reserved	R/W	By BLEN in code option

3.3.11 Anti-EFT Ability Control register (SYS0_ANTIEFT)

Address Offset: 0x30

This register decides the HW anti-EFT ability.

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
3:0	AEFT[2:0]	Hold clock time 0: No 1: Level 5 2: Level 6 3: Level 7 4: Level 8 (Strongest) 9: Level 1 10: Level 2 11: Level 3 12: Level 4 Other: Reserved	R/W	000

3.3.12 IHRC Frequency Adjustment register (SYS0_IHRCADJ)

Address Offset: 0x34

Total 32 counts available for IHRC frequency adjusting and each step varies between +1% from -1%.

Bit	Name	Description	Attribute	Reset
31:16	SYSKEY[15:0]	System register key Read as 0. Behaviour of writing to this register is ignored unless writing 0xA5A5 to SYSKEY at the same time.	W	0
15:9	Reserved		R	0
8:4	ADJ[4:0]	IHRC frequency adjusting bits.	R/W	0
3:2	Reserved		R	0
1	DIR	IHRC frequency adjusting direction bit 0: Positive 1: Negative	R/W	0
0	ADJEN	IHRC frequency adjustment enable bit 0: Disable 1: Enable	R/W	0

3.3.13 CT16Bn Clock Source Select register (SYS0_CT_CLKSEL)

Address Offset: 0x44

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:5	CLKSEL5[1:0]	CT16B5 PCLK source 0: HCLK 1: PLL_VCO 2: LXTAL 3: ILRC	R/W	0
4:2	Reserved		R	0
1	CLKSEL1	CT16B1 PCLK source 0: HCLK 1: PLL_VCO	R/W	0
0	CLKSEL0	CT16B0 PCLK source 0: HCLK 1: PLL_VCO	R/W	0

3.4 SYSTEM CONTROL REGISTERS 1

Base Address: 0x4005 E000

3.4.1 AHB Clock Enable register (SYS1_AHBCLKEN)

Address Offset: 0x00

The SYS_AHBCLKEN register enables the AHB clock to individual system and peripheral blocks.

✷ ***Note:***
1. ***When the clock is disabled, the peripheral register values may not be readable by SW and the value returned is always 0x0.***
2. ***HW will replace GPIO with CLKOUT function directly if CLKOUTSEL is Not 0.***
3. ***User shall disable the AHB clock for individual peripheral to decrease power consumption by demand.***

Bit	Name	Description	Attribute	Reset
31	Reserved		R	0
30:28	CLKOUTSEL[2:0]	Clock output source 000: Disable 001: ILRC clock 010: ELS clock 011: Reserved 100: HCLK 101: IHRC clock (48MHz) 110: EHS clock 111: PLL clock output	R/W	0
27	CRCCLKEN	Enable clock for CRC. 0: Disable 1: Enable	R/W	0
26	FLSAHEN	Enable Flash clock during sleep mode 0: Disable during sleep mode 1: Enable during sleep mode	R	0
25	SRAMEN	Enable SRAM clock during sleep mode 0: Disable during sleep mode 1: Enable during sleep mode		
24	WDTCLKEN	Enable clock for WDT. 0: Reserved 1: Enable	R	1
23	RTCCLKEN	Enable clock for RTC. 0: Disable 1: Enable	R/W	0
22	Reserved		R	0
21	I2C0CLKEN	Enable clock for I2C0. 0: Disable 1: Enable	R/W	0
20	Reserved		R	0
19	ACCCLKEN	Enable AHB clock for ACC 0: Disable 1: Enable	R/W	0
18	FOCCLKEN	Enable AHB clock for FOC 0: Disable 1: Enable	R/W	0
17	UART1CLKEN	Enable clock for UART1. 0: Disable 1: Enable	R/W	0
16	UART0CLKEN	Enable clock for UART0. 0: Disable 1: Enable	R/W	0

15	Reserved		R	0
14	CMPCLKEN	Enable clock for CMP. 0: Disable 1: Enable	R/W	0
13	Reserved		R	0
12	SPI0CLKEN	Enable clock for SPI0. 0: Disable 1: Enable	R/W	0
11	ADCCLKEN	Enable clock for ADC. 0: Disable 1: Enable	R/W	0
10	CT16B5CLKEN	Enable clock for CT16B5. 0: Disable 1: Enable	R/W	0
9:7	Reserved		R	0
6	CT16B1CLKEN	Enable clock for CT16B1. 0: Disable 1: Enable	R/W	0
5	CT16B0CLKEN	Enable clock for CT16B0. 0: Disable 1: Enable	R/W	0
4:3	Reserved		R	0
2	DMA0CLKEN	Enable clock for DMA0. 0: Disable 1: Enable	R/W	0
1	Reserved		R	0
0	OPACLKEN	Enable clock for OPA. 0: Disable 1: Enable	R/W	0

3.4.2 APB Clock Prescale register 1 (SYS1_APBCP1)

Address Offset: 0x08

✸ ***Note: Must reset the corresponding peripheral with SYS1_PRST register after changing the prescale value.***

Bit	Name	Description	Attribute	Reset
31	Reserved		R	0
30:28	CLKOUTPRE[2:0]	Clock-out source prescaler 000: Clock-out source / 1 001: Clock-out source / 2 010: Clock-out source / 4 011: Clock-out source / 8 100: Clock-out source / 16 101: Clock-out source / 32 110: Clock-out source / 64 111: Clock-out source / 128	R/W	0
27:23	Reserved		R	0
22:20	WDTPRE[2:0]	WDT clock source prescaler 000: WDT_PCLK = WDT clock source / 1 001: WDT_PCLK = WDT clock source / 2 010: WDT_PCLK = WDT clock source / 4 011: WDT_PCLK = WDT clock source / 8 100: WDT_PCLK = WDT clock source / 16 101: WDT_PCLK = WDT clock source / 32 Other: Reserved	R/W	0

19:11	Reserved		R	0
10:8	I2C0PRE[2:0]	I2C0 clock source prescaler 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved	R/W	0
7:0	Reserved		R	0

3.4.3 Peripheral Reset register (SYS1_PRST)

Address Offset: 0x10

All bits are cleared by HW automatically after setting as “1”.

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R	0
28	OPARST	OPA reset 0: No effect 1: Reset OPA	R/W	0
27	DMA0RST	DMA0 reset 0: No effect 1: Reset DMA0	R/W	0
26	CRCRST	CRC reset 0: No effect 1: Reset CRC	R/W	0
25	Reserved		R	0
24	WDTRST	WDT reset 0: No effect 1: Reset WDT	R/W	0
23	RTCRST	RTC reset 0: No effect 1: Reset RTC	R/W	0
22	Reserved		R	0
21	I2C0RST	I2C0 reset 0: No effect 1: Reset I2C0	R/W	0
20	Reserved		R	0
19	ACCRST	ACC reset 0: No effect 1: Reset ACC	R/W	0
18	FOCRST	FOC reset 0: No effect 1: Reset FOC	R/W	0
17	UART1RST	UART1 reset 0: No effect 1: Reset UART1	R/W	0
16	UART0RST	UART0 reset 0: No effect 1: Reset UART0	R/W	0
15	Reserved		R	0
14	CMPRST	CMP reset 0: No effect 1: Reset CMP	R/W	0
13	Reserved		R	0
12	SPI0RST	SPI0 reset 0: No effect 1: Reset SPI0	R/W	0
11	ADCRST	ADC reset 0: No effect 1: Reset ADC	R/W	0

10	CT16B5RST	CT16B5 reset 0: No effect 1: Reset CT16B5	R/W	0
9:7	Reserved		R	0
6	CT16B1RST	CT16B1 reset 0: No effect 1: Reset CT16B1	R/W	0
5	CT16B0RST	CT16B0 reset 0: No effect 1: Reset CT16B0	R/W	0
4	Reserved		R	0
3	GPIOP3RST	GPIO port 3 reset 0: No effect 1: Reset GPIO port 3	R/W	0
2	GPIOP2RST	GPIO port 2 reset 0: No effect 1: Reset GPIO port 2	R/W	0
1	GPIOP1RST	GPIO port 1 reset 0: No effect 1: Reset GPIO port 1	R/W	0
0	GPIOP0RST	GPIO port 0 reset 0: No effect 1: Reset GPIO port 0	R/W	0

3.4.4 ILRC Frequency Calibration register (SYS1_ILRCFC)

Address Offset: 0x14

Bit	Name	Description	Attribute	Reset
31	CALEN	Auto-calibration function enable 0: Disable 1: Enable	R/W	0
30	CALDONE	Calibration done 0: Not done 1: Done	R/W	0
29:24	Reserved		R	0
23:16	FRQCMD[7:0]	Frequency calibration command 0x4B: Execute ILRC auto-trim function 0xB4: Load ILRC trim value Other: Reserved	W	0
15:9	Reserved		R	0
8:0	LFRQ[8:0]	ILRC trim value	R/W	0x16B

4 SYSTEM OPERATION MODE

4.1 OVERVIEW

The chip builds in 3 operating modes for difference clock rate and power saving reason. These modes control oscillators, op-code operation and analog peripheral devices' operation.

- Normal mode
- Sleep mode
- Deep sleep mode

4.2 NORMAL MODE

In Normal mode, the ARM Cortex-M0 core, memories, and peripherals are clocked by the system clock. The SYS1_AHBCLKEN register controls which peripherals are running.

Selected peripherals have individual peripheral clocks with their own clock dividers in addition to the system clock. The peripheral clocks can be disabled respectively.

The power to various analog blocks (IHRC, EHS X'TAL, ELS X'TAL, PLL, Flash, LVD, ADC) can be controlled at any time individually through the enable bit of all blocks.

4.3 LOW-POWER MODES

There are 2 special modes of processor power reduction: Sleep mode, and Deep-sleep mode. The PMU_CTRL register controls which mode is desired.

The CPU clock rate may also be controlled as needed by changing clock sources, re-configuring PLL values, and/or altering the system clock divider value. This allows a trade-off of power versus processing speed based on application requirements.

Run-time power control allows disable the clocks to individual on-chip peripherals, allowing fine tuning of power consumption by eliminating all dynamic power use in any peripherals that are not required for the application. Selected peripherals have their own clock divider for power control.

✸ ***Note:***

1. ***The debug mode is not supported in Deep-sleep mode.***
2. ***The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes.*** **Strongly recommended to set these pins as input pull-up.**

4.3.1 SLEEP MODE

In Sleep mode, the system clock to the ARM Cortex-M0 core is stopped and execution of instructions is suspended.

Peripheral functions, if selected to be clocked in SYS1_AHBCLKEN register, continue operation during Sleep mode and may generate interrupts to cause the processor to resume execution. Sleep mode eliminates dynamic power used by the processor itself, memory systems and related controllers, and internal buses.

The power state of the analog blocks (IHRC, EHS X'TAL, ELS X'TAL, PLL, Flash, LVD, ADC) is determined by the enable bit of all blocks.

The processor state and registers, peripheral registers, and internal SRAM values are maintained and the logic levels of the pins remain static.

The pins of port 1 and port 2 configured as ADC input channel must be set as input mode, inactive (no pull-down/pull-up resistor enabled, Schmitt trigger disabled, Data register keeps low) with GPIO1_MODE, GPIO1_CFG, GPIO2_MODE and GPIO2_CFG register by program to avoid current leakage.

Wake up the chip from Sleep mode by an interrupt occurs.

The RESET pin has kept functionality in Sleep mode.

The Sleep mode is entered by using the following steps:
1. Write 4 to PMU_CTRL register.
2. Execute ARM Cortex-M0 WFI instruction.

4.3.2 DEEP-SLEEP MODE

In Deep-sleep mode, the system clock to the ARM Cortex-M0 core is stopped, and execution of instructions is suspended.

The clock to the peripheral functions are stopped because the power state of oscillators are powered down, the clock source are stopped, except RTC low speed clock source (ELS X'tal, ILRC) if used.

✷ ***Note: User SHALL decide to power down low speed clock source (ELS X'TAL, ILRC oscillator) or not if RTC is enabled.***

The processor state and registers, peripheral registers, and internal SRAM values are maintained and the logic levels of the pins remain static.

All GPIO pins are served as wakeup pins. The user must program the GPIO registers for each pin to set the appropriate edge polarity for the corresponding wakeup event, only edge sensitive is supported to wakeup MCU. The system will exit Deep-sleep mode when GPIO indicates a GPIOn interrupt to the ARM core. Furthermore, the interrupts corresponding to each input must be enabled in the NVIC.

✷ ***Note: Strongly recommended to set the GPIO wake-up condition to both edge sensing.***

The RESET pin has kept functionality in Deep-sleep mode.

The Deep-sleep mode is entered by using the following steps:
1. Write 2 to PMU_CTRL register.
2. Execute ARM WFI instruction.

The advantage of the Deep-sleep mode is that can power down clock generating blocks such as oscillators and PLL, thereby gaining far greater dynamic power savings over Sleep mode. In addition, the Flash can be powered down in Deep-sleep mode resulting in savings in static leakage power, however at the expense of longer wake-up times for the Flash memory.

4.4 WAKEUP

4.4.1 OVERVIEW

Under low power mode, program doesn't execute. The wakeup trigger can wake the system up to normal mode. The wakeup function builds in interrupt operation and trigger system executing interrupt service routine as system wakeup occurrence.

- The wakeup trigger sources of the Sleep mode are all interrupts and the RESET pin.
- The wakeup trigger sources of the Deep-sleep mode are the GPIO interrupt, RTC interrupt, CT16B5 interrupt, and the RESET pin.

4.4.2 WAKEUP TIME

When the system is in Sleep mode, the high clock is enabled or disabled by F/W. If the high clock stops and MCU is waken up from Sleep mode, MCU waits for 2048 external high-speed oscillator clocks and 32 internal high-speed oscillator clocks as the wakeup time to stable the oscillator circuit. After the wakeup time, the system goes into the normal mode.

- ***Note: Wakeup from Sleep mode spends NO wakeup time if the clock doesn't stop.***

When the system is in Deep-sleep mode, the high clock will stop. When MCU is waken up from Deep-sleep mode, MCU waits for IHRC warm up time (10us plus 32T*IHRC) if HCLK is IHRC, or EHS Xtal warm up time (3.2ms plus 2048T*EHS) if HCLK is EHS Xtal. After the wakeup time, the system goes into the normal mode.

The value of the external high clock oscillator wakeup time from Deep sleep mode is as the following.

The total Wakeup time of EHS X'tal = 3.2ms + 1/F_{EHS} * 2048 (sec)

- Example: F_{EHS}=20MHz, the wakeup time is as the following.

The total Wakeup time = 3.2ms + 1/F_{EHS} * 2048 = 3.3ms (F_{EHS} = 20MHz)

The value of the IHRC wakeup time from Deep sleep mode is as the following.

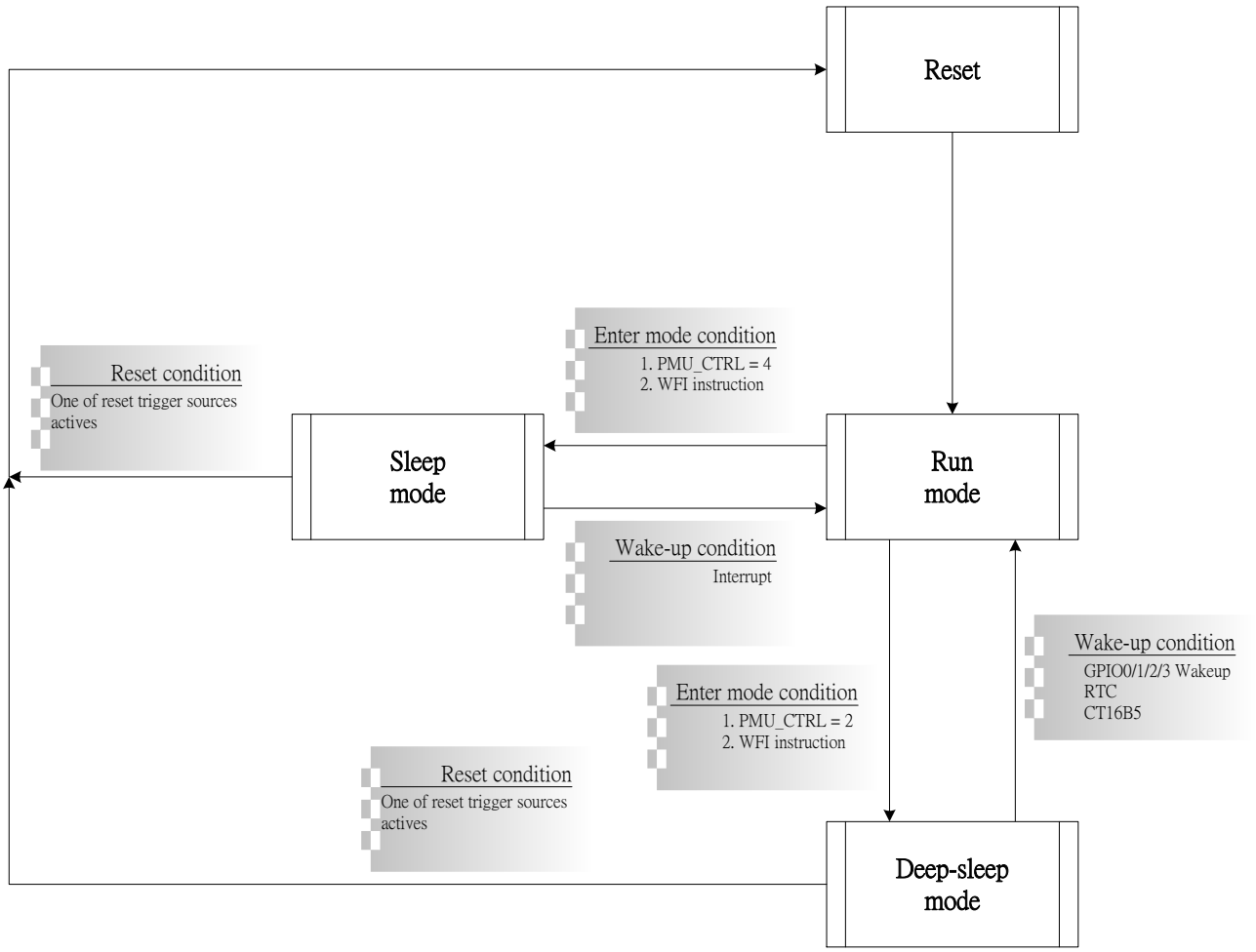
The total Wakeup time of IHRC = 10us + 1/F_{IHRC} * 32 (sec)

- Example: F_{IHRC}=12MHz, the wakeup time is as the following.

The total Wakeup time = 10us + 1/F_{IHRC} * 32 = 12.67 us (F_{IHRC} = 12MHz)

- ***Note: The high clock start-up time is depended on the VDD and oscillator type of high clock.***

4.5 STATE MACHINE OF PMU



4.6 OPERATION MODE COMPARSION TABLE

Operation Mode	Normal Mode	Low-Power Mode	
		Sleep Mode	Deep-Sleep Mode
IHRC	By IHRCEN		Disable
ILRC	ON		Enable if used as clock source of RTC/CT16B5
EHS X'TAL	By EHSEN		Disable
ELS X'TAL	By ELSEN		Enable if used as clock source of RTC/CT16B5
PLL	By PLLEN		Disable
Cortex-M0	Running	Stop	Stop
Flash ROM	Enable	By FLASHEN bit for DMA0	Disable
RAM	Enable	By SRAMEN bit for DMA0	Maintain
ADC	By ADENB		Disable
LVD	By LVDEN		Disable
RTC	By RTCEN		By RTCEN
CMP	By CMnEN & CMnMODE		Disable
OPA	By OPnEN		Disable
DMA	By DMAnEN		Disable
Peripherals	By Enable bit of each peripheral		Disable HCLK
IO status	-	Maintained	Maintained
Wakeup Source	N/A	All interrupts, RESET pin	GPIO0/1/2/3 interrupt, CT16B5 interrupt, RTC interrupt, RESET pin

4.7 PMU REGISTERS

Base Address: 0x4003 2000

4.7.1 Power Control register (PMU_CTRL)

Address Offset: 0x40

The power control register selects whether one of the ARM Cortex-M0 controlled power-down modes (Sleep mode or Deep-sleep mode) is entered and provides the flags for Sleep or Deep-sleep modes respectively.

✸ ***Note: The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.***

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2:0	MODE[2:0]	Low power mode selection 010: WFI instruction will make MCU enter Deep-sleep mode. 100: WFI instruction will make MCU enter Sleep mode. Other: Normal mode	R/W	0

5 GENERAL PURPOSE I/O PORT (GPIO)

5.1 OVERVIEW

Digital ports can be configured input/output by SW

- Each individual port pin can serve as external interrupt input pin.
- Interrupts can be configured on single falling or rising edges and on both edges.
- The I/O configuration registers control the electrical characteristics of the pads.
- Internal pull-up/pull-down resistor.
- Most of the I/O pins are mixed with analog pins and special function pins.

5.2 GPIO MODE

All GPIO pins are inputs and floating by default. The MODE bits in the GPIOn_CFG (n=0,1,2,3) register allow the selection of on-chip pull-up or pull-down resistors for each pin.

5.3 GPIO REGISTERS

Base Address: 0x4004 4000 (GPIO 0)
0x4004 6000 (GPIO 1)
0x4004 8000 (GPIO 2)
0x4004 A000 (GPIO 3)

5.3.1 GPIO Port n Data register (GPIOn_DATA) (n=0,1,2,3)

Address offset: 0x00

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:0	DATA[13:0]	Input data (read) or output data (write) for Pn.x	R/W	0

5.3.2 GPIO Port n Mode register (GPIOn_MODE) (n=0,1,2,3)

Address offset: 0x04

✸ ***Note: HW will switch I/O Mode directly when Specific function (Peripheral, ADC) is enabled, not through GPIOn_MODE register.***

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:0	MODE[13:0]	Selects pin x as input or output (x = 0 to 13) 0: Pn.x is configured as input 1: Pn.x is configured as output	R/W	0

5.3.3 GPIO Port n Configuration register (GPIOn_CFG) (n=0,1,2,3)

Address offset: 0x08

Reset value: 0x00AAAAAA

✸ ***Note: HW will switch I/O Mode directly when Specific function (Peripheral, ADC) is enabled, not through GPIOn_MODE register.***

Bit	Name	Description	Attribute	Reset
31:28	Reserved		R	0
27:26	CFG13[1:0]	Configuration of Pn.13 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
25:24	CFG12[1:0]	Configuration of Pn.12 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
23:22	CFG11[1:0]	Configuration of Pn.11 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b

21:20	CFG10[1:0]	Configuration of Pn.10 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
19:18	CFG9[1:0]	Configuration of Pn.9 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
17:16	CFG8[1:0]	Configuration of Pn.8 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
15:14	CFG7[1:0]	Configuration of Pn.7 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
13:12	CFG6[1:0]	Configuration of Pn.6 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
11:10	CFG5[1:0]	Configuration of Pn.5 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
9:8	CFG4[1:0]	Configuration of Pn.4 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
7:6	CFG3[1:0]	Configuration of Pn.3 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
5:4	CFG2[1:0]	Configuration of Pn.2 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
3:2	CFG1[1:0]	Configuration of Pn.1 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b
1:0	CFG0[1:0]	Configuration of Pn.0 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/up resistor enabled, Schmitt trigger enabled). 11: Inactive (no pull-down/up resistor enabled, Schmitt trigger disabled, data register keeps low)	R/W	10b

5.3.4 GPIO Port n Interrupt Sense register (GPIOn_IS) (n=0,1,2,3)

Address offset: 0x0C

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:0	IS[13:0]	Selects interrupt on pin x as level or edge sensitive (x = 0 to 13). 0: Interrupt on Pn.x is configured as edge sensitive. 1: Interrupt on Pn.x is configured as event sensitive.	R/W	0

5.3.5 GPIO Port n Interrupt Both-edge Sense register (GPIOn_IBS) (n=0,1,2,3)

Address offset: 0x10

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:0	IBS[13:0]	Selects interrupt on Pn.x to be triggered on both edges (x = 0 to 13). 0: Interrupt on Pn.x is controlled through register GPIOn_IEV. 1: Both edges on Pn.x trigger an interrupt.	R/W	0

5.3.6 GPIO Port n Interrupt Event register (GPIOn_IEV) (n=0,1,2,3)

Address offset: 0x14

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:0	IEV[13:0]	Selects interrupt on pin x to be triggered rising or falling edges (x = 0 to 13). 0: Depending on setting in register GPIOn_IS, Rising edges or HIGH level on Pn.x trigger an interrupt. 1: Depending on setting in register GPIOn_IS, Falling edges or LOW level on Pn.x trigger an interrupt.	R/W	0

5.3.7 GPIO Port n Interrupt Enable register (GPIOn_IE) (n=0,1,2,3)

Address offset: 0x18

Bits set to HIGH in the GPIOn_IE register allow the corresponding pins to trigger their individual interrupts. Clearing a bit disables interrupt triggering on that pin.

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:0	IE[13:0]	Selects interrupt on pin x to be enabled (x = 0 to 13). 0: Disable Interrupt on Pn.x 1: Enable Interrupt on Pn.x	R/W	0

5.3.8 GPIO Port n Raw Interrupt Status register (GPIOn_RIS) (n=0,1,2,3)

Address offset: 0x1C

This register indicates the status for GPIO control raw interrupts. A GPIO interrupt is sent to the interrupt controller if the corresponding bit in GPIOn_IE register is set.

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:0	IF[13:0]	GPIO raw interrupt flag (x = 0 to 13). 0: No interrupt on Pn.x 1: Interrupt requirements met on Pn.x.	R	0

5.3.9 GPIO Port n Interrupt Clear register (GPIOn_IC) (n=0,1,2,3)

Address offset: 0x20

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:0	IC[13:0]	Selects interrupt flag on pin x to be cleared (x = 0 to 13). 0: No effect 1: Clear interrupt flag on Pn.x	W	0

5.3.10 GPIO Port n Bits Set Operation register (GPIOn_BSET) (n=0,1,2,3)

Address offset: 0x24

In order for SW to set GPIO bits without affecting any other pins in a single write operation, the GPIO bit is set if the corresponding bit in the GPIOn_BSET register is set.

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:0	BSET[13:0]	Bit Set enable (x = 0 to 13) 0: No effect on Pn.x 1: Set Pn.x to “1”	W	0

5.3.11 GPIO Port n Bits Clear Operation register (GPIOn_BCLR) (n=0,1,2,3)

Address offset: 0x28

In order for SW to clear GPIO bits without affecting any other pins in a single write operation, the GPIO bit is cleared if the corresponding bit in this register is set.

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:0	BCLR[13:0]	Bit clear enable (x = 0 to 13) 0: No effect on Pn.x 1: Clear Pn.x.	W	0

6 PERIPHERAL FUNCTION PIN ASSIGNMENT (PFPA)

6.1 OVERVIEW

PFPA registers are used to provide flexible assignment of digital peripheral functions to desired external pins of different packages.

6.2 FEATURES

- Flexible assignment of digital peripheral functions to desired pins.
- Supported functions are UART, I2C, SPI, and PWM.

6.3 PIN ASSIGNMENT LIST

Peripheral	Pin Name	PA0	PA1	PA2	PA3
UART0	URXD0	P0.11	P2.0	P3.2	P2.10
	UTXD0	P0.10	P2.1	P3.1	P2.9
UART1	URXD1	P1.8	P2.3	P2.8	P1.1
	UTXD1	P1.9	P2.2	P2.7	P1.0
SPI0	SCK0	P0.0	P2.5	P1.0	P3.0
	SEL0	P0.1	P2.4	P1.8	P1.7
	MISO0	P0.2	P2.6	P1.1	P3.1
	MOSI0	P0.3	P2.7	P1.2	P3.2
I2C0	SCL0	P0.6	P1.4	P0.10	P1.1
	SDA0	P0.7	P1.5	P0.11	P1.2
CT16B0	PWM0	P0.2	P3.0	P2.6	P0.0
	PWM0N	P0.3	P0.1	P2.7	P1.1
	PWM1	P2.9	P3.1	P0.1	P2.5
	PWM1N	P0.2	P0.0	P2.8	P2.4
	PWM2	P2.7	P3.2	P0.0	P2.8
	PWM2N	P2.8	P2.6	P2.9	P0.1
	PWM3	P0.1	P0.5	P0.3	P3.2
	PWM3N	P0.0	P0.1	P0.2	P3.1
	BRK	P0.5	P0.10	P1.12	P2.10
CT16B1	PWM0	P2.10	P1.1	P1.12	P2.0
	PWM1	P0.5	P1.2	P2.8	P2.4
	PWM2	P0.6	P0.10	P2.0	P2.7
	PWM3	P0.7	P0.11	P2.1	P2.9
CT16B5	PWM0	P2.6	P2.4	P1.1	P0.3
	PWM1	P2.7	P2.3	P1.2	P0.2
	PWM2	P2.8	P2.2	P1.0	P0.1

	PWM3	P2.9	P2.1	P1.3	P0.0
CMP	CM0O	P2.0	P2.6	P1.6	P3.0
	CM1O	P2.2	P2.7	P1.10	P3.1
	CM2O	P1.0	P0.3	P3.8	P1.11
	CM3O	P3.0	P2.8	P3.12	P1.3

6.4 PFPA REGISTERS

Base Address: 0x4004 2000

6.4.1 PFPA for CT16B0 register (PFPA_CT16B0)

Address offset: 0x00

Bit	Name	Description	Attribute	Reset
31:30	BRK	CT16B0_BRK assigned pin 0: P0.5 1: P0.10 2: P1.12 3: P2.10	RW	0
29:16	Reserved		R	0
15:14	PWM3N	CT16B0_PWM3N assigned pin 0: P0.0 1: P0.1 2: P0.2 3: P3.1	RW	0
13:12	PWM3	CT16B0_PWM3 assigned pin 0: P0.1 1: P0.5 2: P0.3 3: P3.2	RW	0
11:10	PWM2N	CT16B0_PWM2N assigned pin 0: P2.8 1: P2.6 2: P2.9 3: P0.1	RW	0
9:8	PWM2	CT16B0_PWM2 assigned pin 0: P2.7 1: P3.2 2: P0.0 3: P2.8	RW	0
7:6	PWM1N	CT16B0_PWM1N assigned pin 0: P0.2 1: P0.0 2: P2.8 3: P2.4	RW	0
5:4	PWM1	CT16B0_PWM1 assigned pin 0: P2.9 1: P3.1 2: P0.1 3: P2.5	RW	0
3:2	PWM0N	CT16B0_PWM0N assigned pin 0: P0.3 1: P0.1 2: P2.7 3: P1.1	RW	0
1:0	PWM0	CT16B0_PWM0 assigned pin 0: P0.2 1: P3.0 2: P2.6 3: P0.0	RW	0

6.4.2 PFPA for CT16B1 register (PFPA_CT16B1)

Address offset: 0x04

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:6	PWM3	CT16B1_PWM3 assigned pin 0: P0.7 1: P0.11 2: P2.1 3: P2.9	R/W	0
5:4	PWM2	CT16B1_PWM2 assigned pin 0: P0.6 1: P0.10 2: P2.0 3: P2.7	R/W	0
3:2	PWM1	CT16B1_PWM1 assigned pin 0: P0.5 1: P1.2 2: P2.8 3: P2.4	R/W	0
1:0	PWM0	CT16B1_PWM0 assigned pin 0: P2.10 1: P1.1 2: P1.12 3: P2.0	R/W	0

6.4.3 PFPA for CT16B5 register (PFPA_CT16B5)

Address offset: 0x14

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:6	PWM3	CT16B5_PWM3 assigned pin 0: P2.9 1: P2.1 2: P1.3 3: P0.0	R/W	0
5:4	PWM2	CT16B5_PWM2 assigned pin 0: P2.8 1: P2.2 2: P1.0 3: P0.1	R/W	0
3:2	PWM1	CT16B5_PWM1 assigned pin 0: P2.7 1: P2.3 2: P1.2 3: P0.2	R/W	0
1:0	PWM0	CT16B5_PWM0 assigned pin 0: P2.6 1: P2.4 2: P1.1 3: P0.3	R/W	0

6.4.4 PFPA for UART0 register (PFPA_UART0)

Address offset: 0x20

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3:2	URXD0	URXD0 assigned pin 0: P0.11 1: P2.0 2: P3.2 3: P2.10	R/W	0
1:0	UTXD0	UTXD0 assigned pin 0: P0.10 1: P2.1 2: P3.1 3: P2.9	R/W	0

6.4.5 PFPA for UART1 register (PFPA_UART1)

Address offset: 0x24

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3:2	URXD1	URXD1 assigned pin 0: P1.8 1: P2.3 2: P2.8 3: P1.1	R/W	0
1:0	UTXD1	UTXD1 assigned pin 0: P1.9 1: P2.2 2: P2.7 3: P1.0	R/W	0

6.4.6 PFPA for I2C0 register (PFPA_I2C0)

Address offset: 0x40

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3:2	SCL0	SCL0 assigned pin 0: P0.6 1: P1.4 2: P0.10 3: P1.1	R/W	0
1:0	SDA0	SDA0 assigned pin 0: P0.7 1: P1.5 2: P0.11 3: P1.2	R/W	0

6.4.7 PFPA for SPI0 register (PFPA_SPI0)

Address offset: 0x50

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:6	SEL0	SEL0 assigned pin 0: P0.1 1: P2.4 2: P1.8 3: P1.7	R/W	0
5:4	SCK0	SCK0 assigned pin 0: P0.0 1: P2.5 2: P1.0 3: P3.0	R/W	0
3:2	MOSI0	MOSI0 assigned pin 0: P0.3 1: P2.7 2: P1.2 3: P3.2	R/W	0
1:0	MISO0	MISO0 assigned pin 0: P0.2 1: P2.6 2: P1.1 3: P3.1	R/W	0

6.4.8 PFPA for CMP register (PFPA_CMP)

Address offset: 0x90

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:6	CM3O	CM3O assigned pin 0: P3.0 1: P2.8 2: P3.12 3: P1.3	R/W	0
5:4	CM2O	CM2O assigned pin 0: P1.0 1: P0.3 2: P3.8 3: P1.11	R/W	0
3:2	CM1O	CM1O assigned pin 0: P2.2 1: P2.7 2: P1.10 3: P3.1	R/W	0
1:0	CM0O	CM0O assigned pin 0: P2.0 1: P2.6 2: P1.6 3: P3.0	R/W	0

7 DMA

7.1 OVERVIEW

Direct memory access (DMA) is used in order to provide high-speed data transfer between peripherals and memory as well as memory to memory. Data can be quickly moved by DMA without any CPU actions. This keeps CPU resources free for other operations.

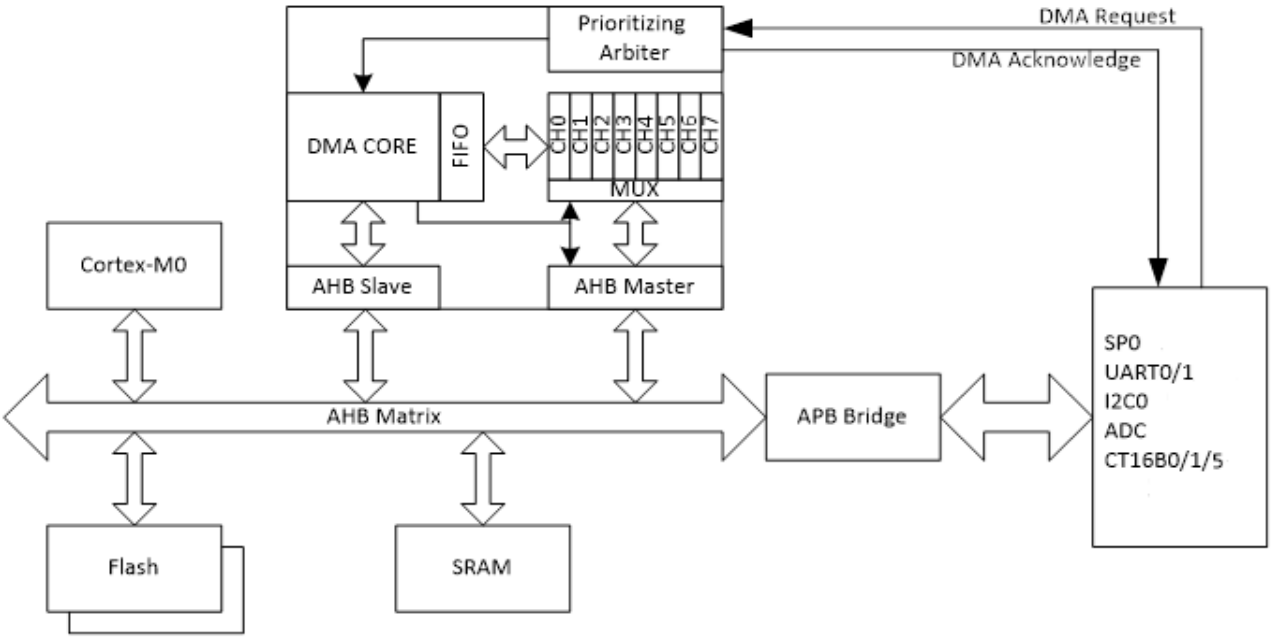
The DMA controller has up to 5 channels, each dedicated to managing memory access requests from one or more peripherals. It has an arbiter for handling the priority between DMA requests.

7.2 FEATURES

- Up to 5 independently configurable channels (requests) on DMA
- Each channel is connected to dedicated hardware DMA requests, software trigger is also supported on each channel. This configuration is done by software.
- Priorities between requests from the DMA channels are software programmable (4 levels consisting of very high, high, medium, low) or hardware in case of equality (round robin scheme)
- Independent source and destination transfer size (byte, half word, word), emulating packing and unpacking. Source/destination addresses must be aligned on the data size.
- Support for circular buffer management
- 2 event flags (DMA Transfer complete, and DMA Transfer Abort) logically ORed together in a single interrupt request for each channel
- Memory-to-memory transfer
- Peripheral-to-memory and memory-to-peripheral, and peripheral-to-peripheral transfers
- Access to Flash, SRAM, APB and AHB peripherals as source and destination
- Programmable number of data to be transferred: up to 4M

7.3 FUNCTION DESCRIPTIONS

The block diagram is shown in the following figure.



The DMA controller performs direct memory transfer by sharing the system bus with the Cortex®-M0 core. The DMA request may stop the CPU access to the system bus for some bus cycles, when the CPU and DMA are targeting the

same destination (memory or peripheral). The bus matrix implements round-robin scheduling, thus ensuring at least half of the system bus bandwidth (both to memory and peripheral) for the CPU.

7.3.1 DMA Transactions

After an event, the peripheral sends a request signal to the DMA Controller. The DMA controller serves the request depending on the channel priorities. As soon as the DMA Controller accesses the peripheral, an Acknowledge is sent to the peripheral by the DMA Controller. The peripheral releases its request as soon as it gets the Acknowledge from the DMA Controller. Once the request is de-asserted by the peripheral, the DMA Controller release the Acknowledge. If there are more requests, the peripheral can initiate the next transaction.

In summary, each DMA transfer consists of three operations:

• The loading of data from the peripheral data register or a location in memory addressed through an internal current peripheral/memory address register. The start address used for the first transfer is the base peripheral/memory address programmed in the Cn_SRCADDR register.

• The storage of the data loaded to the peripheral data register or a location in memory addressed through an internal current peripheral/memory address register. The start address used for the first transfer is the base peripheral/memory address programmed in the Cn_DSTADDR register.

• The post-decrementing of the Cn_SIZE register, which contains the number of transactions that have still to be performed.

7.3.2 Arbiter

The arbiter manages the channel requests based on their priority and launches the peripheral/memory access sequences.

The priorities are managed in two stages:

• Software: each channel priority can be configured in the Cn_CSR register. There are four levels:

3 : Very high priority

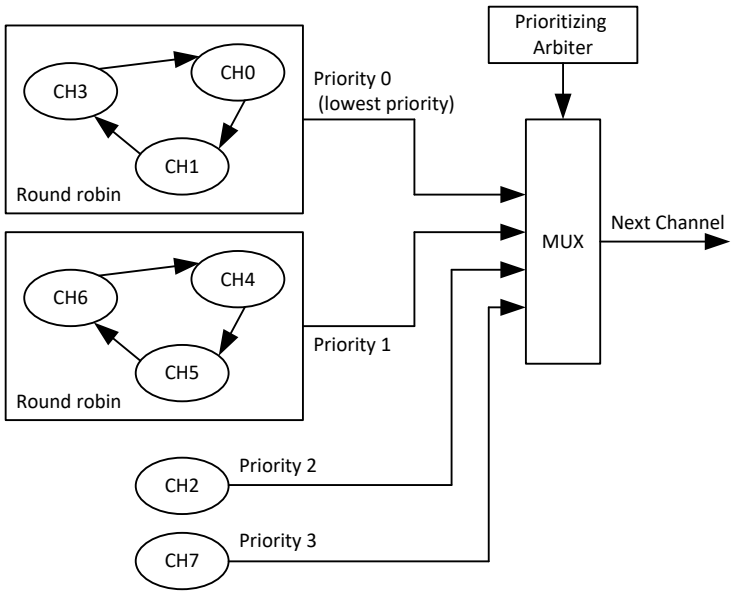
2 : High priority

1 : Medium priority

0 : Low priority

• Hardware: If the channels have the same priority level, the arbitration will then be based on the round robin scheme.

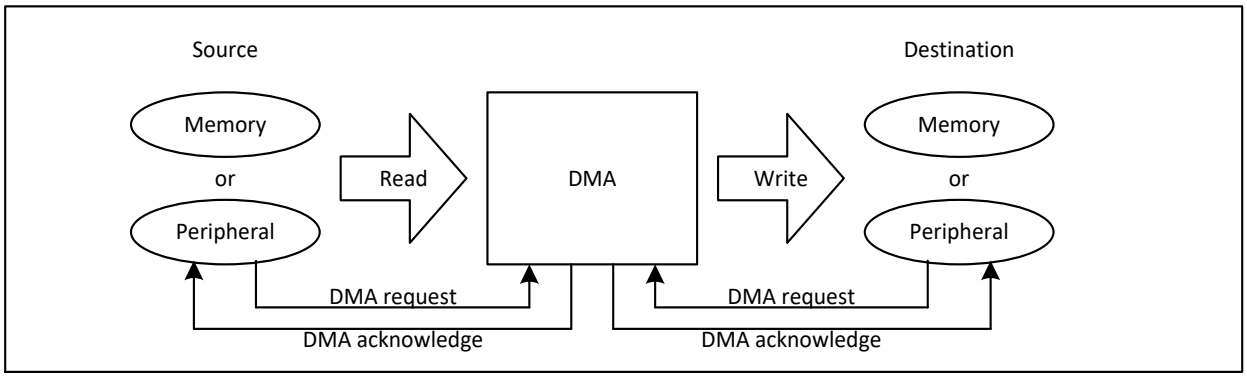
For example, set channel 0, 1 and 3 are priority 0 level (lowest priority), channel 4, 5 and 6 are priority 1 level (medium priority), channel 2 is priority 2 level (high priority), and channel 7 is priority 3 level (very high priority). Since priority level: 3 > 2 > 1 > 0, therefore, channel 7 is the first operating channel. Next is channel 2. Then there is the priority 1 group, and finally the priority 0 group. Arbitration in the priority 0 group and priority 1 group will be based on the round-robin scheme.

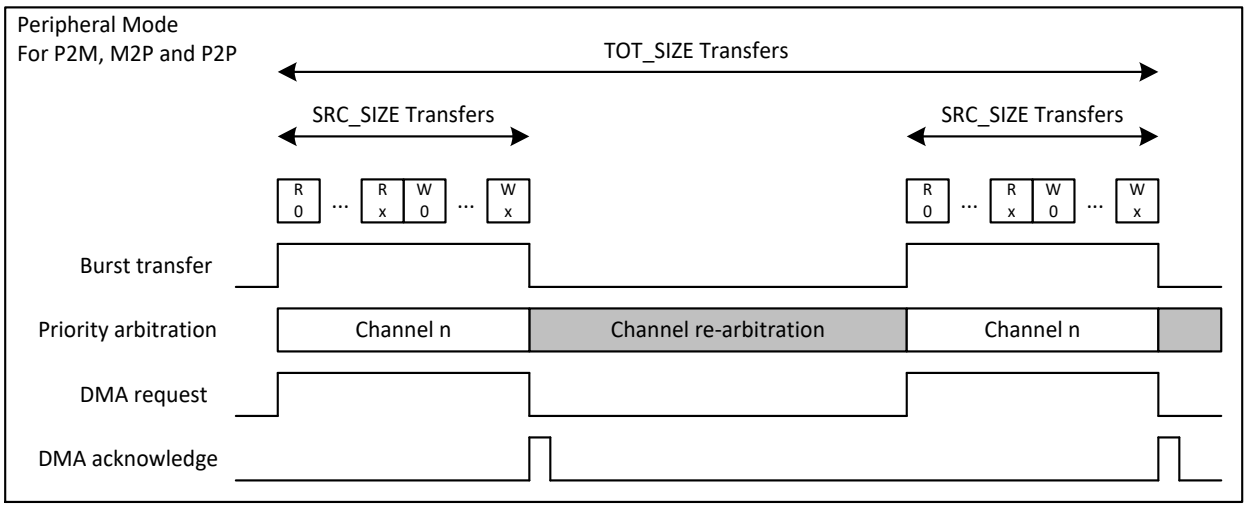


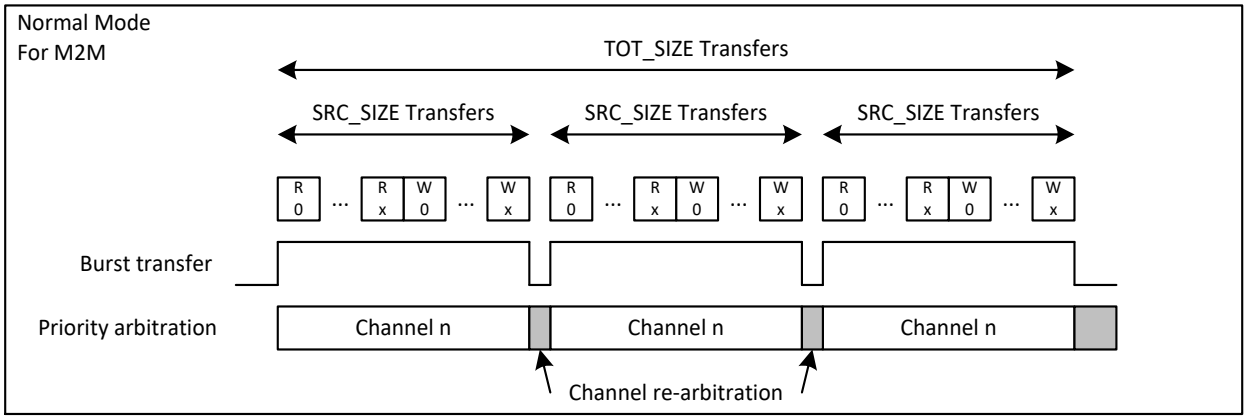
7.3.3 Peripheral Mode

The DMA peripheral mode can be enabled by setting bit 7 of Channel Control Register (Cn_CSR). If the DMA peripheral mode of channel n is enabled, after channel n wins the arbitration, the DMA controller will wait for the peripheral request to be asserted before starting the DMA transfer. Each time the peripheral request is asserted, the controller transfers units equal to SRC_SIZE. When SRC_SIZE transfer is completed, the DMA controller re-arbitrates among all DMA requests. After TOT_SIZE transfers have been done, the DMA controller asserts TC flag when TC_MSK is 0.

✸ ***Note: If only the MODE bit is set, the DMA will still ignore any requests. Before the DMA controller receives the request, SRC_HE or DST_HE must be set. If SRC_HE is 1, the DMA controller will wait for the source request is asserted. If DST_HE is 1, the DMA controller will wait for the destination request is asserted. If SRC_HE and DST_HE are both 1, the DMA controller will wait for both requests to be asserted.***







7.3.4 DMA Channels

7.3.4.1 Programmable Data Sizes

Each channel can handle DMA transfer between a peripheral register located at a fixed address and a memory address. The amount of data to be transferred (up to 4M) is programmable. The register which contains the amount of data items to be transferred is decremented after each transaction.

Transfer data sizes of the peripheral and memory are fully programmable through the TOT_SIZE bits in the Cn_SIZE register.

7.3.4.2 Programmable Burst Sizes

It indicates the number of transfers existing before the DMA re-arbitrates among the enabled channels. The number of bytes to be transferred for one burst depends on this source burst size and the source transfer width. For example, if the source burst size is 64 (bits[18:16] are set as 101) and source transfer width is 16 bits (bits[13:11] are set as 001), the total number of bytes for this burst transfer will be 128 (64 *2).

(Burst size * SRC_WIDTH) must be equal to or larger than DST_WIDTH. So, the following settings are not allowed:
Burst size = 1, source width = 8, destination width = 16 or
Burst size = 1, source width = 8, destination width = 32 or
Burst size = 1, source width = 16, destination width = 32
Burst sizes of the peripheral and memory are fully programmable through the SRC_SIZE bits in the Cn_CSR register.

✷ ***Note: The setting value of the burst size must consider the size of the peripheral FIFO. The burst data cann't be larger than the FIFO size of the peripheral.***

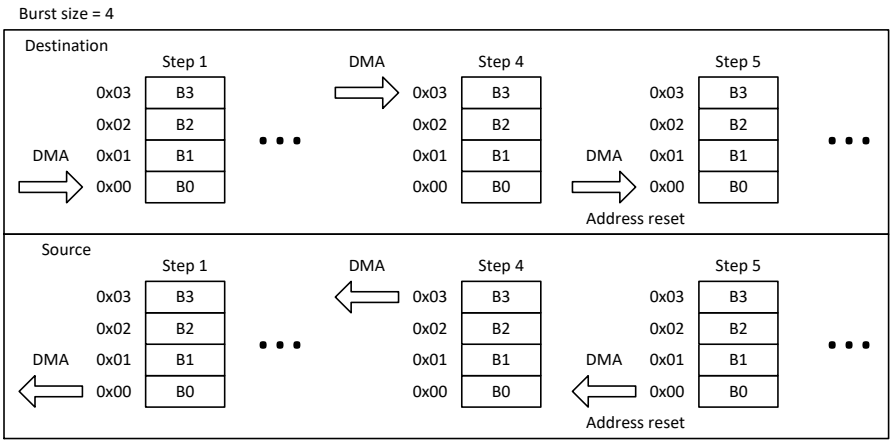
7.3.4.3 Pointer Incrementation/Decrementation

Peripheral and memory pointers can optionally be automatically post-incremented/decrementation after each transaction depending on the SRCAD_CTL and DSTAD_CTL bits in the Cn_CSR register. If incremented mode is enabled, the address of the next transfer will be the address of the previous one incremented by 1. If decremented mode is enabled, the address of the next transfer will be the address of the previous one decremented by 1. The first transfer address is the one programmed in the Cn_SRCADDR/Cn_DSTADDR registers. During transfer operations, these registers will change by current address. The current transfer addresses are accessible by software.

If the number of data items to be transferred has reached zero, the DMA request will not be serviced after the last transfer. Then the channel enablement will be cancelled.

7.3.4.4 Incremental Cyclic Mode

Incremental cycle mode is an extension of incremental mode. Through the SRCAD_CTL and DSTAD_CTL bits in the Cn_CSR register, peripherals and memory pointers can be selected as incremental cyclic mode. The usage is similar to incremental mode, the difference is that the address will be reset automatically. After the DMA completes a burst, the address will return to the original address. This mode can be used to update the register group within a limited range. e.g. CT16Bn MR0~3 register



7.3.4.5 Memory-to-Memory (M2M) Mode

The DMA channels can work without being triggered by a request from a peripheral.

If the MODE bit in the Cn_CSR register is cleared, then the channel initiates transfers as soon as it is enabled by software by setting the Enable bit (DMAEN) in the Cn_CSR register. The transfer stops once the Cn_SIZE register reaches zero.

7.3.4.6 Memory-to-Peripheral (M2P) Mode

The DMA channels works with being triggered by a request from the destination peripheral.

If the DST_HE bit in the Cn_CFG register and the MODE bit in the Cn_CSR register are set, then the channel will not starts transfer immediately. DMA will wait destination request. When destination request is issued, the DMA will transfer a burst of data. The transfer stops once the Cn_SIZE register reaches zero.

7.3.4.7 Peripheral-to-Memory (P2M) Mode

The DMA channels works with being triggered by a request from the source peripheral.

If the SRC_HE bit in the Cn_CFG register and the MODE bit in the Cn_CSR register are set, then the channel will not starts transfer immediately. DMA will wait source request. When source request is issued, the DMA will transfer a burst of data. The transfer stops once the Cn_SIZE register reaches zero.

7.3.4.8 Peripheral-to-Peripheral (P2P) Mode

The DMA channels works with being triggered by two requests from the source peripheral and destination peripheral.

If the SRC_HE bit in the Cn_CFG register, the DST_HE bit in the Cn_CFG register and the MODE bit in the Cn_CSR register are set, then the channel will not starts transfer immediately. DMA will wait source request AND destination request. When both the source request and the target request are issued, the DMA will transfer a burst of data. The transfer stops once the Cn_SIZE register reaches zero.

7.3.4.9 Channel Configuration Precedure

The following sequence should be followed to configure a DMA channel n (where n is the channel number).

1. Set the DMACEN bit in the MCSR register to 1 for enable the DMA module.
2. Set the source address in the Cn_SRCADDR register. The data will be moved from this address to the destination after the peripheral event (peripheral mode). Memory-to-memory mode works without any request.
3. Set the destination address in the Cn_DSTADDR register. The data will be written to this memory after the peripheral event (peripheral mode). Memory-to-memory mode works without any request.
4. Configure the total number of data to be transferred in the Cn_SIZE register. After each peripheral event, this value will be decremented.
5. Configure the burst transfer size in the SRC_SIZE bit in Cn_SIZE register. The number of bytes to be transferred for one burst depends on this source burst size and the source transfer width.
6. Configure the channel priority using the CHPRI[1:0] bits in the Cn_CSR register
7. Configure source/destination address control mode (SRCAD_CTL and DSTAD_CTL bits) and data width (SRC_WIDTH and DST_WIDTH bits) in the Cn_CSR register
8. Configure source/destination request source using the SRC_RS[3:0] bits and DST_RS[3:0] bits in the Cn_CFG register. If the operation mode is not peripheral mode, the request source will be ignored.
9. If the source request is required for DMA transfer, please configure the SRC_HE bit in the Cn_CFG register to peripheral mode
10. If the destination request is required for DMA transfer, please configure the DST_HE bit in the Cn_CFG register to peripheral mode

11. If DMA transfer requires source request or target request, please configure the MODE bit in the Cn_CSR register to peripheral mode. If the DMA mode is "memory to memory" mode, it can work without any request. Please configure the MODE bit in the Cn_CSR register to normal mode
12. The peripheral must be configured with DMA related settings. Please refer to the chapter on peripheral functions.
13. Activate the channel by setting the CH_EN bit in the Cn_CSR register. If the MODE bit is the normal mode, the DMA transfer starts immediately.

As soon as the channel is enabled, it can serve any DMA request from the peripheral connected on the channel.

At the end of the transfer, the Transfer Complete Flag (TC) and Transfer Complete interrupt Flag (INT_TC) are set and an interrupt is generated if the Transfer Complete Interrupt Mask bit (INT_TC_MSK) is cleared.

7.3.5 Data Transfer Mapping Rule

When SRC_WIDTH and DST_WIDTH are not equal, the DMA performs some data alignments as described in Table: Programmable data width & endian behavior (Burst size >= 4).

If source transfer width < destination transfer width, DMA will pack source input data. For example, if source transfer width = 8-bit, destination transfer width = 32-bit, then DMA will pack 4 sets of 8-bit source data and transfer 1 set of 32-bit data to destination.

If source transfer width > destination transfer width, DMA will unpack source input data. For example, if source transfer width = 32-bit, destination transfer width = 8-bit, then DMA will unpack source 32-bit data and transfer 4 sets of 8-bit data to destination.

The address fix rule is the same as increment.

✸ ***Note:***

1. ***Do not set SRCAD_CTL to 1 (decrement source address) when the pack function works; otherwise DMA will have a wrong action.***
2. ***Do not set SRC_SIZE to 0 (burst size = 1) when the pack function works; otherwise DMA will have a wrong action.***

7.3.5.1 Burst Data is Enough for Packet Operation

Table: Programmable data width & endian behavior (Burst size >= 4).

Source			Destination		
Width	Source content Data @Address		Width	Destination content Data @Address	
Address Control	Increment	Decrement	Address Control	Increment	Decrement
8	0x34 @N+1 0x12 @N	0x12 @N 0x34 @N-1	8	0x34 @M+1 0x12 @M	0x12 @M 0x34 @M-1
8	0x78 @N+3 0x56 @N+2 0x34 @N+1 0x12 @N	Don't supply pack	16	0x7856 @N+2 0x3412 @M	0x3412 @M 0x7856 @M-2
8	0x78 @N+3 0x56 @N+2 0x34 @N+1 0x12 @N	Don't supply pack	32	0x78563412 @M	0x78563412 @M
16	0x7856 @N+2 0x3412 @N	0x3412 @N 0x7856 @N-2	8	0x78 @M+3 0x56 @M+2 0x34 @M+1 0x12 @M	0x34 @M 0x12 @M-1 0x78 @M-2 0x56 @M-3

16	0x7856 @N+2 0x3412 @N	0x3412 @N 0x7856 @N-2	16	0x7856 @M+2 0x3412 @M	0x3412 @M 0x7856 @M-2
16	0xF0CD @N+6 0xAB89 @N+4 0x7856 @N+2 0x3412 @N	Don't supply pack	32	0xF0CDAB89 @M+4 0x78563412 @M	0x78563412 @M 0xF0CDAB89 @M-4
32	0x78563412 @N	0x78563412 @N	8	0x78 @M+3 0x56 @M+2 0x34 @M+1 0x12 @M	0x78 @M 0x56 @M-1 0x32 @M-2 0x12 @M-3
32	0x78563412 @N	0x78563412 @N	16	0x7856 @M+2 0x3412 @M	0x7856 @M 0x3412 @M-2
32	0x78563412 @N	0x78563412 @N	32	0x78563412 @M	0x78563412 @M

7.3.5.2 Burst Data is Not Enough for Packet Operation

When the packing function works, burst data is not enough to pack. DMA will fill the vacancies with 0 to satisfy the packing operation.

Table: Programmable data width & endian behavior (Burst size = 1).

Source			Destination		
Width	Source content Data @Address		Width	Destination content Data @Address	
Address Control	Increment	Decrement	Address Control	Increment	Decrement
8	0x34 @N+1 0x12 @N	0x12 @N 0x34 @N-1	8	0x34 @M+1 0x12 @M	0x12 @M 0x34 @M-1
8	0x78 @N+3 0x56 @N+2 0x34 @N+1 0x12 @N	Don't supply pack	16	0xRR78 @N+6 0xRR56 @M+4 0xRR34 @N+2 0xRR12 @M	0xRR12 @M 0xRR34 @N-2 0xRR56 @M-4 0xRR78 @N-6
8	0x78 @N+3 0x56 @N+2 0x34 @N+1 0x12 @N	Don't supply pack	32	0xRRRRRR78 @M+6 0xRRRRRR56 @M+4 0xRRRRRR34 @M+2 0xRRRRRR12 @M	0xRRRRRR12 @M 0xRRRRRR34 @M-2 0xRRRRRR56 @M-4 0xRRRRRR78 @M-6
16	0x7856 @N+2 0x3412 @N	0x3412 @N 0x7856 @N-2	8	0x78 @M+3 0x56 @M+2 0x34 @M+1 0x12 @M	0x34 @M 0x12 @M-1 0x78 @M-2 0x56 @M-3
16	0x7856 @N+2 0x3412 @N	0x3412 @N 0x7856 @N-2	16	0x7856 @M+2 0x3412 @M	0x3412 @M 0x7856 @M-2
16	0xF0CD @N+6 0xAB89 @N+4 0x7856 @N+2 0x3412 @N	Don't supply pack	32	0xRRRRF0CD @M+0xC 0xRRRRAB89 @M+8 0xRRRR7856 @M+4 0xRRRR3412 @M	0xRRRR3412 @M 0xRRRR7856 @M-4 0xRRRRAB89 @M-8 0xRRRRF0CD @M-0xC
32	0x78563412 @N	0x78563412 @N	8	0x78 @M+3 0x56 @M+2 0x34 @M+1 0x12 @M	0x78 @M 0x56 @M-1 0x32 @M-2 0x12 @M-3
32	0x78563412 @N	0x78563412 @N	16	0x7856 @M+2 0x3412 @M	0x7856 @M 0x3412 @M-2
32	0x78563412 @N	0x78563412 @N	32	0x78563412 @M	0x78563412 @M

✷ ***Note: RR is random value which is the last value of DMA buffer.***

7.3.5.3 Pack Function is Not Allowed to Decrement Type

When the pack function works, the data packing sequence needs to increment the address. If the address is decremented, the data packing will be wrong. The data will be copied into the pack. The packaging function will not work.

Source		Destination		
Width	Source content	Width	Destination content Data @Address	
Address Control	Decrement	Address Control	Increment	Decrement
8	0x12 @N 0x34 @N-1 0x56 @N-2 0x78 @N-3	16	0x7878 @M+6 0x5656 @M+4 0x3434 @M+2 0x1212 @M	0x1212 @M 0x3434 @M-2 0x5656 @M-4 0x7878 @M-6
8	0x12 @N 0x34 @N-1 0x56 @N-2 0x78 @N-3	32	0x78787878 @M+0xC 0x56565656 @M+8 0x34343434 @M+4 0x12121212 @M	0x12121212 @M 0x34343434 @M-4 0x56565656 @M-8 0x78787878 @M-0xC
16	0x3412 @N 0x7856 @N-2 0xAB89 @N-4 0xF0CD @N-6	32	0xF0CDF0CD @M+0xC 0xAB89AB89 @M+8 0x78567856 @M+4 0x34123412 @M	0x34123412 @M 0x78567856 @M-4 0xAB89AB89 @M-8 0xF0CDF0CD @M-0xC

7.3.6 Abort Operation When Channel is Active

During the transfer, if the software sets the abort bit (bit 15 of Channel Control Register (Cn_CSR), after finishing burst transfers, the DMA controller will set the ABT bit (bits 23:16 of Error/Abort status register (ERR_ABT) and terminate the DMA transfer at once. Then, if INT_ABT_MSK (bit 2 of Cn_CFG register) is 0, the DMA controller asserts INT_ABT. If an abort event occurs, the channel enablement will be cancelled, and TOT_SIZE remains in the final state. If the channel is enabled again, the remaining transfers will continue. If you want to restart a new transfer event, first turn off the peripheral function of the source or destination. Make sure to cancel the last transfer request to avoid erroneous transfer.

7.3.7 Interrupt

DMA interrupts are caused by the following three conditions: transfer completion, abort and error. The interrupt flag is located in the INT register, which combines these three states (INT_TC, INT_ABT, INT_ERR). If the INT flag is not cleared, the interrupt will continue to occur.

7.3.8 DMA Request Selection Table

The hardware requests from the peripherals. Request source selected by SRC_RS bits and DST_RS bits (CT16Bn, ADC, SPI, I2C and UARTn). After setting up the DMA controller, you must configure DMA related settings for the peripheral. Please refer to the chapter on peripheral functions.

The selection list for each channel has the same configuration, so all supported peripherals can be used. However you must be careful about the direction of DMA transfer. For example, the source can only choose RX, and the destination can only choose TX to avoid DMA transfer error.

SRC_RS/DST_RS Table

SRC_RS[5:0] DST_RS[5:0]	Peripheral function	DMA Request	SRC_RS[5:0] DST_RS[5:0]	Peripheral function	DMA Request
0	ADC	Source	25	CT16B1 MR1	Source/Destination
1	Reserve		26	CT16B1 MR2	Source/Destination
2	SPI0 TX	Destination	27	CT16B1 MR3	Source/Destination

3	SPI0 RX	Source	28		Source
4		Destination	29		Source/Destination
5		Source	30		Source/Destination
6	UART0 TX	Destination	31		Source/Destination
7	UART0 RX	Source	32		Source/Destination
8	UART1 TX	Destination	33		Source/Destination
9	UART1 RX	Source	34		Source
10		Destination	35		Source/Destination
11		Source	36		Source/Destination
12		Destination	37		Source/Destination
13		Source	38		Source
14	I2C0	Source/Destination	39		Source/Destination
15		Source/Destination	40		Source/Destination
16	CT16B0 CAP0	Source	41		Source/Destination
17	CT16B0 MR9	Source/Destination	42	CT16B5 CAP0	Source
18	CT16B0 MR0	Source/Destination	43	CT16B5 MR9	Source/Destination
19	CT16B0 MR1	Source/Destination	44	CT16B5 MR0	Source/Destination
20	CT16B0 MR2	Source/Destination	45	CT16B5 MR1	Source/Destination
21	CT16B0 MR3	Source/Destination	46	CT16B5 MR2	Source/Destination
22	CT16B1 CAP0	Source	47	CT16B5 MR3	Source/Destination
23	CT16B1 MR9	Source/Destination	48		Source/Destination
24	CT16B1 MR0	Source/Destination	49		Source/Destination
			50~63	Reserve	

7.4 SLEEP MDOE

DMA supports data transfer in sleep mode. The memory or peripheral function needs to enable the interface clock. The control bits are shown in the following table. After enabling, DMA is allowed to access data. Other functions do not support DMA operation in sleep mode to reduce power consumption.

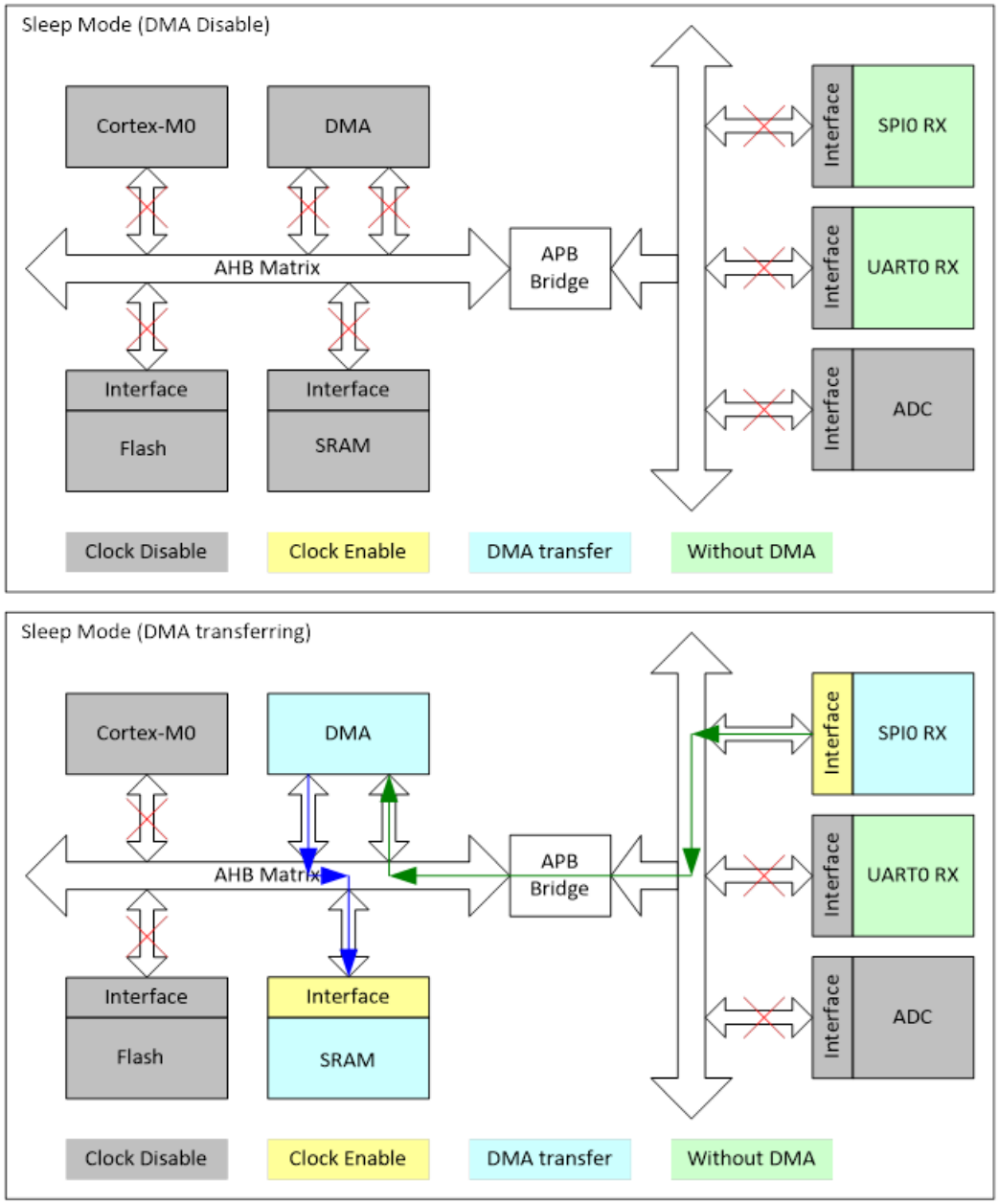


Table. Clock enable during sleep mode

DMA Source/Destination	Clock enable during sleep mode	
	Register	Control bit
Flash 32KB (Memory)	SYS1_AHBCLKEN (0x4005E000)	FLASHEN (bit 26)
SRAM 8KB (Memory)	SYS1_AHBCLKEN (0x4005E000)	SRAMEN (bit 25)
ADC	ADC_DMA (0x40026050)	DMA_EN (bit 22)
SPIn RX/TX (n = 0)	SPIn_DMA (0x4001C050)	RX_DMA_EN (bit 30) or TX_DMA_EN (bit 31)
UARTn RX/TX (n = 0,1)	UARTn_DMA (0x40016050, 0x40056050,)	DMA_EN (bit 31)

I2Cn RX/TX (n = 0)	I2Cn_DMA (0x40018050)	RX_DMA_EN (bit 30) or TX_DMA_EN (bit 31)
CT16Bn CAP0 (n = 0,1,5) CT16Bn MR9 (n = 0,1,5) CT16Bn MR0 (n = 0,1,5) CT16Bn MR1 (n = 0,1,5) CT16Bn MR2 (n = 0,1,5) CT16Bn MR3 (n = 0,1,5)	CT16Bn_DMA (n = 0,1,5) (0x400000D0, 0x400020D0, 0x4000A0D0)	DMA_CAP0 (bit 4) DMA_MR9 (bit 5) DMA_MR0 (bit 0) DMA_MR1 (bit 1) DMA_MR2 (bit 2) DMA_MR3 (bit 3)

7.5 DMA REGISTERS

Base Address: 0x4007 2000 (DMA0)

Register Address Offset					
Channel n	**0**	**1**	**2**	**3**	**4**
DMA_INT	0x00				
DMA_INT_TC	0x04				
DMA_INT_TC_CLR	0x08				
DMA_INT_ABT	0x0C				
DMA_INT_ABT_CLR	0x10				
DMA_TC	0x14				
DMA_ABT	0x18				
DMA_CH_EN	0x1C				
DMA_CH_BUSY	0x20				
DMA_MCSR	0x24				
DMA_Cn_CSR	0x100	0x120	0x140	0x160	0x180
DMA_Cn_CFG	0x104	0x124	0x144	0x164	0x184
DMA_Cn_SRCADDR	0x108	0x128	0x148	0x168	0x188
DMA_Cn_DSTADDR	0x10C	0x12C	0x14C	0x16C	0x18C
DMA_Cn_SIZE	0x114	0x134	0x154	0x174	0x194

7.5.1 DMA n Interrupt Status register (DMAn_INT)

Address Offset: 0x00

This register is used to keep the result of INTn = INT_ABTn | INT_TCn where “n” is channel number n.

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	INT4	The result of INT_ABT4 | INT_TC4 0: Channel 4 has no pending interrupt 1: Channel 4 has a pending interrupt	R	0
3	INT3	The result of INT_ABT3 | INT_TC3 0: Channel 3 has no pending interrupt 1: Channel 3 has a pending interrupt	R	0
2	INT2	The result of INT_ABT2 | INT_TC2 0: Channel 2 has no pending interrupt 1: Channel 2 has a pending interrupt	R	0
1	INT1	The result of INT_ABT1 | INT_TC1 0: Channel 1 has no pending interrupt 1: Channel 1 has a pending interrupt	R	0
0	INT0	The result of INT_ABT0 | INT_TC0 0: Channel 0 has no pending interrupt 1: Channel 0 has a pending interrupt	R	0

7.5.2 DMA n Transfer Complete Interrupt Status register (DMAn_INT_TC) (n=0)

Address Offset: 0x04

This register shows the status of the DMA transfer complete interrupts after masking. The mask bit of these interrupts is bit 0 (INT_TC_MSK) of Channel Configuration Register (Cn_CFG). If this mask bit is set, the content of this register is always 0 no matter there exists a pending DMA transfer complete interrupt or not.

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	INT_TC4	Status of the DMA terminal count interrupts after masking 0: Channel 4 has no pending interrupt 1: Channel 4 has a pending interrupt	R	0
3	INT_TC3	Status of the DMA terminal count interrupts after masking 0: Channel 3 has no pending interrupt 1: Channel 3 has a pending interrupt	R	0
2	INT_TC2	Status of the DMA terminal count interrupts after masking 0: Channel 2 has no pending interrupt 1: Channel 2 has a pending interrupt	R	0
1	INT_TC1	Status of the DMA terminal count interrupts after masking 0: Channel 1 has no pending interrupt 1: Channel 1 has a pending interrupt	R	0
0	INT_TC0	Status of the DMA terminal count interrupts after masking 0: Channel 0 has no pending interrupt 1: Channel 0 has a pending interrupt	R	0

7.5.3 DMA n Transfer Complete Interrupt Status Clear register (DMAn_INT_TC_CLR) (n=0)

Address Offset: 0x08

This register shows the status of the DMA transfer complete interrupts after masking. The mask bit of these interrupts is bit 0 (INT_TC_MSK) of Channel Configuration Register (Cn_CFG). If this mask bit is set, the content of this register is always 0 no matter there exists a pending DMA transfer complete interrupt or not.

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	INT_TC_CLR4	0: No effect 1: Clear the INT_TC4 and TC4 status	W	0
3	INT_TC_CLR3	0: No effect 1: Clear the INT_TC3 and TC3 status	W	0
2	INT_TC_CLR2	0: No effect 1: Clear the INT_TC2 and TC2 status	W	0
1	INT_TC_CLR1	0: No effect 1: Clear the INT_TC1 and TC1 status	W	0
0	INT_TC_CLR0	0: No effect 1: Clear the INT_TC0 and TC0 status	W	0

7.5.4 DMA n Abort Interrupt Status register (DMAn_INT_ABT) (n=0)

Address offset: 0x0C

INT_ABT is the status of the DMA abort interrupts after masking. The mask bit of these interrupts is bit 2 (INT_ABT_MSK) of channel configuration register (Cn_CFG). If this mask bit is set, the content of INT_ABT[n] of this register is always 0 no matter there exists a pending DMA abort interrupt or not.

Bit	Name	Description	Attribute	Reset
31:21	Reserved		R	0
20	INT_ABT4	Status of the DMA abort interrupts after masking 0: Channel 4 has no pending interrupt 1: Channel 4 has a pending interrupt	R	0

19	INT_ABT3	Status of the DMA abort interrupts after masking 0: Channel 3 has no pending interrupt 1: Channel 3 has a pending interrupt	R	0
18	INT_ABT2	Status of the DMA abort interrupts after masking 0: Channel 2 has no pending interrupt 1: Channel 2 has a pending interrupt	R	0
17	INT_ABT1	Status of the DMA abort interrupts after masking 0: Channel 1 has no pending interrupt 1: Channel 1 has a pending interrupt	R	0
16	INT_ABT0	Status of the DMA abort interrupts after masking 0: Channel 0 has no pending interrupt 1: Channel 0 has a pending interrupt	R	0
15:0	Reserved		R	0

7.5.5 DMA n Abort Interrupt Status Clear register (DMAn_INT_ABT_CLR) (n=0)

Address offset: 0x10

Bit	Name	Description	Attribute	Reset
31:21	Reserved		R	0
20	INT_ABT_CLR4	0: No effect 1: Clear the INT_ABT4 and ABT4 status	W	0
19	INT_ABT_CLR3	0: No effect 1: Clear the INT_ABT3 and ABT3 status	W	0
18	INT_ABT_CLR2	0: No effect 1: Clear the INT_ABT2 and ABT2 status	W	0
17	INT_ABT_CLR1	0: No effect 1: Clear the INT_ABT1 and ABT1 status	W	0
16	INT_ABT_CLR0	0: No effect 1: Clear the INT_ABT0 and ABT0 status	W	0
15:0	Reserved		R	0

7.5.6 DMA n Transfer Complete Status register (DMAn_TC) (n=0)

Address Offset: 0x14

This register shows the status of the DMA terminal count after masking. The mask bit for the DMA terminal count is bit 31 (TC_MSK) of channel control register (Cn_CSR). If this mask bit of Cn_CSR is set, the TC[n] of this register is always 0 no matter a DMA terminal count happens or not.

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	TC4	Status of the DMA terminal count 0: Channel 4 has no terminal count status 1: Channel 4 has a terminal count status	R	0
3	TC3	Status of the DMA terminal count 0: Channel 3 has no terminal count status 1: Channel 3 has a terminal count status	R	0
2	TC2	Status of the DMA terminal count 0: Channel 2 has no terminal count status 1: Channel 2 has a terminal count status	R	0
1	TC1	Status of the DMA terminal count 0: Channel 1 has no terminal count status 1: Channel 1 has a terminal count status	R	0
0	TC0	Status of the DMA terminal count 0: Channel 0 has no terminal count status 1: Channel 0 has a terminal count status	R	0

7.5.7 DMA n Abort Status register (DMAn_ABT) (n=0)

Address Offset: 0x18

ABT is the status of the DMA abort. If the ABT bit (bit 15 of Channel Control Register (Cn_CSR) is set, the DMA controller will stop the current DMA transfer and set ABT[n] to 1. Then, if INT_ABT_MSK is not set, the DMA controller will set INT_ABT[n] to 1 and assert interrupt.

Bit	Name	Description	Attribute	Reset
31:21	Reserved		R	0
20	ABT4	Status of the DMA abort 0: Channel 4 has no abort status 1: Channel 4 has an abort status	R	0
19	ABT3	Status of the DMA abort 0: Channel 3 has no abort status 1: Channel 3 has an abort status	R	0
18	ABT2	Status of the DMA abort 0: Channel 2 has no abort status 1: Channel 2 has an abort status	R	0
17	ABT1	Status of the DMA abort 0: Channel 1 has no abort status 1: Channel 1 has an abort status	R	0
16	ABT0	Status of the DMA abort 0: Channel 0 has no abort status 1: Channel 0 has an abort status	R	0
15:0	Reserved		R	0

7.5.8 DMA n Channel Enable Status register (DMAn_CH_EN) (n=0)

Address Offset: 0x1C

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	CH_EN4	Status of the channel 4 CH_EN bit of C4_CSR register 0: CH_EN = 0 1: CH_EN = 1	R	0
3	CH_EN3	Status of the channel 3 CH_EN bit of C3_CSR register 0: CH_EN = 0 1: CH_EN = 1	R	0
2	CH_EN2	Status of the channel 2 CH_EN bit of C2_CSR register 0: CH_EN = 0 1: CH_EN = 1	R	0
1	CH_EN1	Status of the channel 1 CH_EN bit of C1_CSR register 0: CH_EN = 0 1: CH_EN = 1	R	0
0	CH_EN0	Status of the channel 0 CH_EN bit of C0_CSR register 0: CH_EN = 0 1: CH_EN = 1	R	0

7.5.9 DMA n Channel Busy Status register (DMAn_CH_BUSY) (n=0)

Address Offset: 0x20

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	CH_BUSY4	Status of the channel 4 BUSY bit of C4_CFG register 0: BUSY = 0 1: BUSY = 1	R	0
3	CH_ BUSY3	Status of the channel 3 BUSY bit of C3_CFG register 0: BUSY = 0 1: BUSY = 1	R	0
2	CH_ BUSY2	Status of the channel 2 BUSY bit of C2_CFG register 0: BUSY = 0 1: BUSY = 1	R	0
1	CH_ BUSY1	Status of the channel 1 BUSY bit of C1_CFG register 0: BUSY = 0 1: BUSY = 1	R	0

0	CH_ BUSY0	Status of the channel 0 BUSY bit of C0_CFG register 0: BUSY = 0 1: BUSY = 1	R	0

7.5.10 DMA n Main Configuration Status register (DMAn_MCSR) (n=0)

Address Offset: 0x24

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1	M0ENDIAN	AHB Master 0 endian configuration 0: Little-endian 1: Big-endian	R/W	0
0	DMACEN	DMA controller enable 0: Disable 1: Enable	R/W	0

7.5.11 DMA n Channel m Control register (DMAn_Cm_CSR) (n=0/m=0,1,2,3,4)

Address Offset: 0x100, 0x120, 0x140, 0x160, 0x180

Source burst size just indicates the number of transfers existing before the DMA re-arbitrates among the enabled channels. The number of bytes to be transferred for one burst depends on this source burst size and the source transfer width. For example, if the source burst size is 64 (bits[18:16] are set as 101) and source transfer width is 16 bits (bits[13:11] are set as 001), the total number of bytes for this burst transfer will be 128 (=64 *2).

(Burst size * SRC_WIDTH) must be equal to or larger than DST_WIDTH. So, the following settings are not allowed:
Burst size = 1, source width = 8, destination width = 16 or
Burst size = 1, source width = 8, destination width = 32 or
Burst size = 1, source width = 16, destination width = 32

If source transfer width < destination transfer width, DMA will pack source input data. For example, if source transfer width = 8-bit, destination transfer width = 32-bit, then DMA will pack 4 sets of 8-bit source data and transfer 1 set of 32-bit data to destination.

If source transfer width > destination transfer width, DMA will unpack source input data. For example, if source transfer width = 32-bit, destination transfer width = 8-bit, then DMA will unpack source 32-bit data and transfer 4 sets of 8-bit data to destination.

✷ ***Note:***

1. ***SRC_WIDTH shall be equal to DST_WIDTH in Incremental cyclic mode.***
2. ***All other bits of this register will remain the same when writing 1 to ABT bit, do NOT program this bit and the other bits of this register simultaneously.***
3. ***Do not set SRCAD_CTL to 1 (decrement source address) when the pack function works; otherwise DMA will have a wrong action.***

Bit	Name	Description	Attribute	Reset
31	TC_MSK	Terminal count status mask for current transaction 0: When terminal count happens, TC status register will be set 1: When terminal count happens, TC status register will not be set	R/W	0
30:24	Reserved		R	0
23:22	CHPRI	Channel priority level 0: Lowest priority 1: 3rd high priority 2: 2nd high priority 3: Highest priority	R/W	0
21:19	Reserved		R	0
18:16	SRC_ SIZE	Source burst size selection 0: Burst size = 1	R/W	0

		1: Burst size = 4 2: Burst size = 8 3: Burst size = 16 4: Burst size = 32 5: Burst size = 64 6: Burst size = 128 7: Burst size = 256		
15	ABT	Transaction abort 1: Stop the current transfer, then set the ABT[n] bit of Error/Abort Status register and assert DMAINT interrupt if INT_ABT_MST = 0.	W	0
14:13	Reserved		R	0
12:11	SRC_WIDTH	Source transfer width The hardware automatically packs and unpacks the data as required. 0: Transfer width is 8 bits 1: Transfer width is 16 bits 2: Transfer width is 32 bits Other: Reserved	R/W	0x2
10	Reserved		R	0
9:8	DST_WIDTH	Destination transfer width The hardware automatically packs and unpacks the data as required. 0: Transfer width is 8 bits 1: Transfer width is 16 bits 2: Transfer width is 32 bits Other: Reserved	R/W	0x2
7	MODE	Hardware handshake mode 0: Normal mode 1: Peripheral mode	R/W	0
6:5	SRCADCTL	Source Address Control 0: Incremental source address 1: Decremental source address 2: Fixed source address 3: Incremental cyclic mode	R/W	0
4:3	DSTAD_CTL	Destination Address Control 0: Incremental destination address 1: Decremental destination address 2: Fixed destination address 3: Incremental cyclic mode	R/W	0
2:1	Reserved		R	0
0	CH_EN	Channel m Enable 0: Disable 1: Enable	R/W	0

7.5.12 DMA n Channel m Configuration register (DMAn_Cm_CFG) (n=0/m=0,1,2,3,4)

Address Offset: 0x104, 0x124, 0x144, 0x164, 0x184

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R	0
29:24	DST_RS[5:0]	Destination DMA request selection refer to SRC_RS/DST_RS Table It specifies which DMA request as the destination request, and is used only when DMA Peripheral Mode is enabled. 0: ADC 2: SPI0 TX 3: SPI0 RX 6: UART0 TX 7: UART0 RX 8: UART1 TX 9: UART1 RX 14: I2C0 16: CT16B0 CAP0 17: CT16B0 MR9 18: CT16B0 MR0 19: CT16B0 MR1 20: CT16B0 MR2	R/W	0

		21: CT16B0 MR3 22: CT16B1 CAP0 23: CT16B1 MR9 24: CT16B1 MR0 25: CT16B1 MR1 26: CT16B1 MR2 27: CT16B1 MR3 42: CT16B5 CAP0 43: CT16B5 MR9 44: CT16B5 MR0 45: CT16B5 MR1 46: CT16B5 MR2 47: CT16B5 MR3 Other: Reserved		
23:22	Reserved	Reserved	R	0
21:16	SRC_RS[5:0]	Source DMA request selection refer to SRC_RS/DST_RS Table It specifies which DMA request as the source req, and is used only when DMA Hardware Handshake Mode is enabled. 0: ADC 2: SPI0 TX 3: SPI0 RX 6: UART0 TX 7: UART0 RX 8: UART1 TX 9: UART1 RX 14: I2C0 16: CT16B0 CAP0 17: CT16B0 MR9 18: CT16B0 MR0 19: CT16B0 MR1 20: CT16B0 MR2 21: CT16B0 MR3 22: CT16B1 CAP0 23: CT16B1 MR9 24: CT16B1 MR0 25: CT16B1 MR1 26: CT16B1 MR2 27: CT16B1 MR3 42: CT16B5 CAP0 43: CT16B5 MR9 44: CT16B5 MR0 45: CT16B5 MR1 46: CT16B5 MR2 47: CT16B5 MR3 Other: Reserved	R/W	0
15:14	Reserved	Reserved	R	0
13	DST_HE	Destination Peripheral mode enable When the destination hardware handshake (peripheral mode) is disabled, DMA will start transfer data without waiting the destination request. This bit is only valid when DMA is in the Peripheral Mode. 0: Disable 1: Enable	R/W	1
12:9	Reserved	Reserved	R	0
8	BUSY	The DMA channel is busy 0: The DMA channel is not Busy 1: The DMA channel is Busy	R	0
7	SRC_HE	Source Peripheral mode enable When the source hardware handshake (peripheral mode) is disabled, DMA will start transfer data without waiting the source request. This bit is only valid when DMA is in the Peripheral Mode. 0: Disable 1: Enable	R/W	1
6:3	Reserved	Reserved	R	0
2	INT_ABT_MSK	Channel abort interrupt mask 0: No mask interrupt 1: Mask interrupt	R/W	1
1	Reserved	Reserved	R/W	1
0	INT_TC_MSK	Channel terminal count interrupt mask 0: No mask interrupt 1: Mask interrupt	R/W	1

7.5.12.1 SRC_RS/DST_RS Table

SRC_RS[5:0] DST_RS[5:0]	Peripheral function	DMA Request	SRC_RS[5:0] DST_RS[5:0]	Peripheral function	DMA Request
0	ADC	Source	25	CT16B1 MR1	Source/Destination
1	Reserve		26	CT16B1 MR2	Source/Destination
2	SPI0 TX	Destination	27	CT16B1 MR3	Source/Destination
3	SPI0 RX	Source	28	Reserve	
4	Reserve		29	Reserve	
5	Reserve		30	Reserve	
6	UART0 TX	Destination	31	Reserve	
7	UART0 RX	Source	32	Reserve	
8	UART1 TX	Destination	33	Reserve	
9	UART1 RX	Source	34	Reserve	
10	Reserve		35	Reserve	
11	Reserve		36	Reserve	
12	Reserve		37	Reserve	
13	Reserve		38	Reserve	
14	I2C0	Source/Destination	39	Reserve	
15	Reserve		40	Reserve	
16	CT16B0 CAP0	Source	41	Reserve	
17	CT16B0 MR9	Source/Destination	42	CT16B5 CAP0	Source
18	CT16B0 MR0	Source/Destination	43	CT16B5 MR9	Source/Destination
19	CT16B0 MR1	Source/Destination	44	CT16B5 MR0	Source/Destination
20	CT16B0 MR2	Source/Destination	45	CT16B5 MR1	Source/Destination
21	CT16B0 MR3	Source/Destination	46	CT16B5 MR2	Source/Destination
22	CT16B1 CAP0	Source	47	CT16B5 MR3	Source/Destination
23	CT16B1 MR9	Source/Destination	48~63	Reserve	
24	CT16B1 MR0	Source/Destination			

7.5.13 DMA n Channel m Channel Source Address register (DMAn_Cm_SRCADDR) (n=0/m=0,1,2,3,4)

Address Offset: 0x108, 0x128, 0x148, 0x168, 0x188

Bit	Name	Description	Attribute	Reset
31:0	SRCADDR	Source starting address When the DMA transaction is done, its value changes to the DMA source ending address.	R/W	0

7.5.14 DMA n Channel m Destination Address register (DMAn_Cm_DSTADDR) (n=0/=0,1,2,3,4)

Address Offset: 0x10C, 0x12C, 0x14C, 0x16C, 0x18C

Bit	Name	Description	Attribute	Reset
31:0	DSTADDR	Destination starting address When the DMA transaction is done, its value changes to the DMA destination ending address.	R/W	0

7.5.15 DMA n Channel m Transfer Size register (DMAn_Cm_SIZE) (n=0/=0,1,2,3,4)

Address Offset: 0x114, 0x134, 0x154, 0x174, 0x194

The transfer unit depends on the source width, for example:
SRC_WIDTH = 000, unit: 8-bit
SRC_WIDTH = 001, unit: 16-bit
SRC_WIDTH = 010, unit: 32-bit
SRC_WIDTH = 011, unit: 64-bit

Bit	Name	Description	Attribute	Reset
31:22	Reserved		R	0
21:0	TOT_SIZE	Total transfer size When the DMA transaction is done, its value changes to 0.	R/W	0

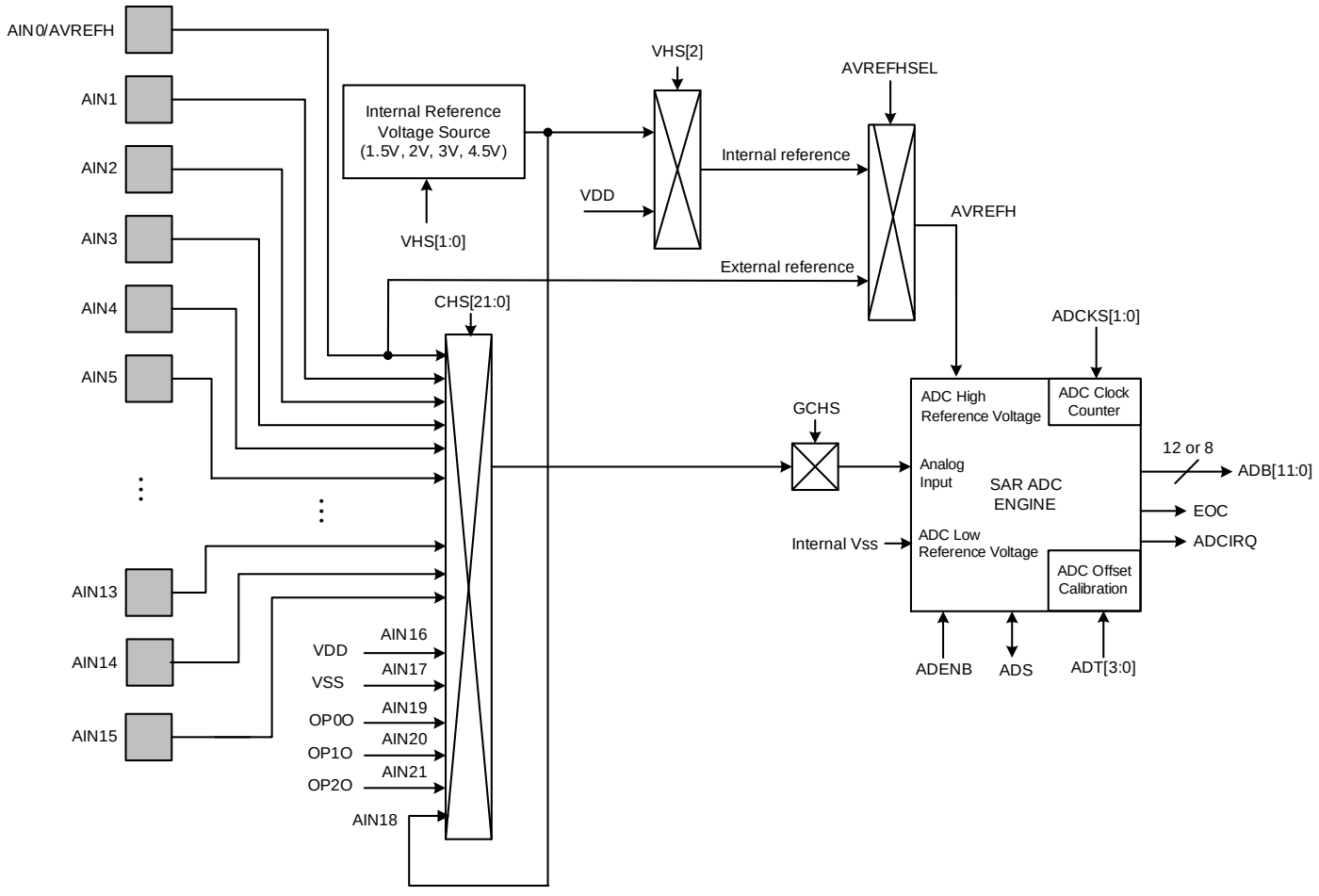
8 16+6 CHANNEL ANALOG TO DIGITAL CONVERTOR (ADC)

8.1 OVERVIEW

This analog to digital converter (ADC) has 16 external input sources, one internal channel to internal voltage source (1.5V/2V/3V/4.5V), one internal channel to VDD, one internal channel to VSS, one internal channel to OPA0 output, one internal channel to OPA1 output, and one internal channel to OPA2 output, with up to 4096-step resolution to transfer analog signal into 12-bits digital data. The ADC resolution can be selected 8-bit or 12-bit through ADLEN bit in ADR register. The ADC converting rate can be selected by ADCKS[1:0] bits. These two parameters decide the ADC converting time. The ADC is capable of maximum 1Msps converting rate with 16MHz clock. The ADC also has built-in calibration circuit. The calibration cycle must be run at least once before starting A/D conversion. During calibration cycle, ADS bit is ignored.

The sequence of ADC operation is to select input source CHS[21:0] bits, CH bit, and SCMODE bit at first, then set GCHS and ADS bit to “1” to start conversion. When the conversion is complete, the ADC circuit will set EOC bit to “1” and final value output in ADB register. When ADC is enabled (ADENB=1) and global channel is enabled (GCHS=1), the ADC shared pins transfers to ADC purpose and disable GPIO function and disable pull-up/pull-down resistor by HW automatically. When ADC is disabled or global channel is disabled, the ADC pins returns to GPIO last status including pull-up/pull-down resistor. Use CHS[14:0] to select AIN pin and GCHS enables global ADC channel, the analog signal inputs to ADC engine.

The AIN16 is internal 1.5V, 2V, 3V or 4.5V input channel, there is no any input pin from outside. In this time ADC reference voltage must be internal VDD and External voltage, not internal 1.5V, 2V, 3V, or 4.5V. AIN16 can be a good battery detector for battery system. To select appropriate internal AVREFH level and compare value, a high performance and cheaper low battery detector is built in the system.



✷ ***Note:***

1. ***ADC_PCLK shall be less than 16MHz.***
2. ***The analog input level must be between the AVREFH and AVREFL.***
3. ***The AVREFH level must be between the AVDD and AVREFL + 1.5V.***
4. ***ADC programming notice***
   - ***Set ADC input pins as input mode and inactive (no pull-down/ pull-up resistor enabled, Schmitt trigger disabled, Data register keeps low) with GPIO1_MODE, GPIO1_CFG, GPIO2_MODE and GPIO2_CFG register by program***
   - ***Disable ADC (set ADENB = "0") before enter low-power (Sleep/Deep-sleep) mode to save power consumption.***
   - ***Delay 100us after enable ADC (set ADENB = "1") to wait ADC circuit ready for conversion.***
   - ***The calibration cycle must be run at least once before starting A/D conversion. During calibration cycle, ADS bit is ignored.***
5. ***The resolution of ADC would be only 8-bit when HCLK = EHS.***

8.2 FEATURES

1. Sampling rate up to 1Msps
2. Internal voltage reference
3. 22-frame FIFO for ADB
4. Self-calibration.
5. Programmable sampling time
6. Support auto-conversion mode
7. ADB watchdog

8.3 ADC CONVERTING TIME

The ADC converting time is from ADS=1 (Start to ADC convert) to EOC=1 (End of ADC convert). The converting time duration is depended on ADC resolution and ADC clock rate.

ADC clock source is controlled by ADCKS[2:0] bits. The ADC converting time affects ADC performance. If input high rate analog signal, it is necessary to select a high ADC converting rate. If the ADC converting time is slower than analog signal variation rate, the ADC result would be error. So to select a correct ADC clock rate and ADC resolution to decide a right ADC converting rate is very important.

12-bit ADC conversion time = 1/ADC MLCK * 16 sec

ADCKS [2:0]	MCLK ADC Clock	ADC_PCLK = 12 MHz		ADC_PCLK = Max 16 MHz	
		ADC Conversion Time (us)	ADC Conversion Rate (KHz)	ADC Conversion Time (us)	ADC Conversion Rate (KHz)
000	ADC_PCLK	1.33	750	1	1000
001	ADC_PCLK/2	2.67	375	2	500
010	ADC_PCLK/4	5.33	187.5	4	250
011	ADC_PCLK/8	10.67	93.75	8	125
100	ADC_PCLK/16	21.33	46.88	16	62.5
101	ADC_PCLK/32	42.67	23.44	32	31.25

8.4 ADC OFFSET CALIBRATION

The calibration must be run at least one time before starting A/D conversion. The internal calibration register will keep the value even when ADC is disabled. It removes the offset error which may vary from chip to chip due to process variation. The calibration is initiated by software by setting ACS to 1. ACS bit remains 1 during all the calibration sequence. It is then cleared by hardware as soon the calibration completes. The internal analog calibration is kept even the ADC is disabled. When the ADC operating conditions change, it is recommended to re-run a calibration cycle.

8.5 ADC CONTROL NOTICE

8.5.1 ADC SIGNAL

The ADC high reference voltage includes internal VDD/4.5V/3V/2V/1.5V or external reference voltage source from P2.0/AVREFH pin controlled by AVREFHSEL bit. The ADC low reference voltage is ground.

ADC reference voltage range limitation is **"(ADC high reference voltage – low reference voltage) ≧ 1.5V"**. ADC low reference voltage is Vss = 0V. So **ADC high reference voltage range is 1.5V~Vdd**. The range is ADC external high reference voltage range.

- **ADC Internal Low Reference Voltage = 0V**
- **ADC Internal High Reference Voltage = VDD/4.5V/3V/2V/1.5V (AVREFHSEL=0)**
- **ADC External High Reference Voltage = 1.5V~VDD (AVREFHSEL =1)**

ADC sampled input signal voltage must be from ADC low reference voltage to ADC high reference. If the ADC input signal voltage is over the range, the ADC converting result is error (full scale or zero).

- **ADC Low Reference Voltage (VSS) ≦ ADC Sampled Input Voltage ≦ ADC High Reference Voltage**

8.5.2 ADC OVERRUN

The ADC has a 22-frame data buffer. The ADC_ADB register is the top layer of the ADC data FIFO, and contains the oldest A/D converter result.

The overrun flag (OVRIF) indicates a data overrun event, when the converted data was not read in time before the data from a new conversion is available. The OVRIF bit is set in the ADC_RIS register if the ADB FIFO remains unread data at the time when a new conversion completes.

The ADC interrupt can be generated if the OVRIE bit is set in ADC_IE register. When an overrun condition occurs, the ADC keeps operating and can continue to convert unless the software decides to stop and reset the sequence by setting the ADSTOP bit.

Configure whether the data is preserved or overwritten when an overrun event occurs by programming the OVRMOD bit in ADC_ADM register.

- OVRMOD=0: An overrun event preserves the data register from being overwritten, the old data is maintained, the new conversion is discarded.
- OVRMOD=1: The data register is overwritten with the last conversion result and the previous unread data is lost.

8.5.3 ADC PROGRAM

The first step of ADC execution is to setup ADC configuration. The ADC program setup sequence and notices are as following.

- **Step 1**: If the ADC high reference voltage is from external voltage source, set AVREFHSEL to 1. The ADC external high reference voltage inputs from P2.0 pin. It is necessary to set P2.0 as input mode without pull-up resistor.
- **Step 2:** Set ADENB to 1 to enable ADC. ***When ADENB is enabled, the system must be delay 100us to be the ADC warm-up time by program, and then set ADS to do ADC converting. The 100us delay time is necessary after ADENB setting (not ADS setting), or the ADC converting result would be error.*** Normally, the ADENB is set one time when the system under normal run condition, and do the delay time only one time.
- **Step 3:** Start ADC offset calibration by setting ACS = 1 and wait for ACS bit to be cleared. It means that the calibration

is completed. Set CALIVALENB = 1 to enable offset calibration. The ADC must be calibrated once after MCU power on or the ADC converting result would be error.

- **Step 4:** Select the ADC input pins by CHS[14:0], set CH and SCMODE bits, and set ADC input pins as input mode and inactive (no pull-down/ pull-up resistor enabled, Schmitt trigger disabled, Data register keep low) with GPIO1_MODE, GPIO1_CFG, GPIO2_MODE and GPIO2_CFG register by program.
- **Step 5:** Start to execute ADC conversion by setting ADS = 1.
- **Step 6:** Wait the end of ADC converting through checking EOC = 1 or ADCIF = 1. If ADC interrupt function is enabled, the program executes ADC interrupt service when ADC interrupt occurrence. ***ADS is cleared when the end of ADC converting automatically. EOC bit indicates ADC processing status immediately and is cleared when ADS = 1. Users needn't to clear it by program.***

8.5.4 ADC PIN CONFIGURATION

ADC input pins are shared with P2.0~P2.6 and P1.0~P1.4 digital I/O pins. ADC channel selection is through CHS[14:0] bits in ADC_CONVCTRL register.

Connect an analog signal to COMS digital input pin, especially, the analog signal level is about 1/2 VDD will cause extra current leakage. In the power down mode, the above leakage current will be a big problem. Unfortunately, if users connect more than one analog input signal to AINx will encounter above current leakage situation.

The P2.0/AIN0 can be ADC external high reference voltage input pin when AVREFHSEL =1. In the condition, P2.0 GPIO mode must be set as input mode and inactive (no pull-down/up resistor enabled, Schmitt trigger disabled) with GPIO2_MODE and GPIO2_CFG register by program.

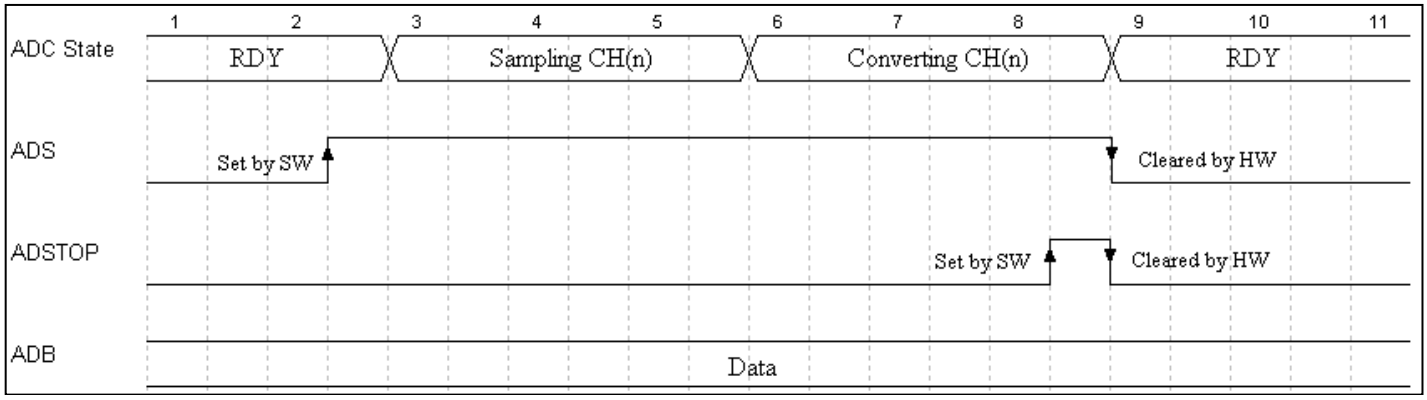
Only one pin of Port 2 can be configured as ADC input in the same time. The pins of Port 1 and Port 2 configured as ADC input channel must be set as input mode, inactive (no pull-down/pull-up resistor enabled, Schmitt trigger disabled, Data register keeps low) with GPIO1_MODE, GPIO1_CFG, GPIO2_MODE and GPIO2_CFG register by program to avoid current leakage.

> ✷ ***Note: The GPIO mode of ADC input channels used must be set as input mode and inactive (no pull-down/pull-up resistor enabled, Schmitt trigger disabled, Data register keeps low) with GPIO1_MODE, GPIO1_CFG, GPIO2_MODE and GPIO2_CFG register by program.***

8.6 ADC CONVERSION MODES

The ADC can either converts a single channel or a sequence of multiple channels, determined by CH bit in ADC_CONVCTRL register. The ADC will convert only one channel selected by CHS[14:0] bits in ADC_CONVCTRL register if CH bit is 0, or convert a sequence of multiple channels if CH bit is 1. The multiple channels are selected by CHS[14:0] bits in ADC_CONVCTRL register, and each analog input channel has a dedicated selection bit.

Besides, the ADC supports either Single mode or Continuous mode, determined by SCMODE bit in ADC_CONVCTRL register. The ADSTOP bit can be set to 1 to stop the continuous mode.



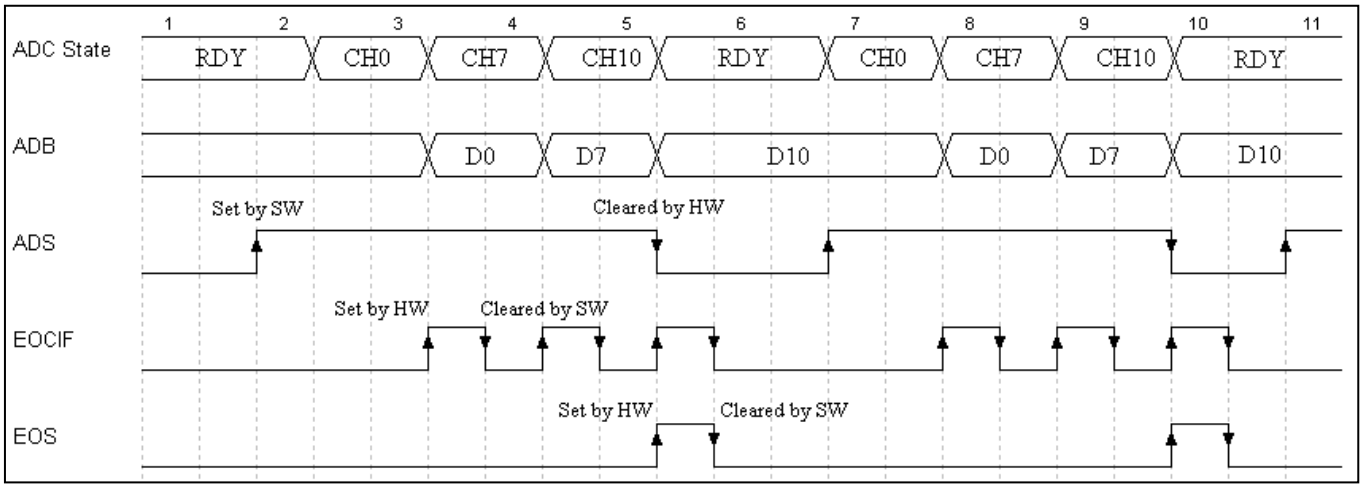
8.6.1 Single Mode

Clear SCMODE bit and set ADS bit to 1 to start the ADC conversion in Single mode. The selected channels will be converted once per trigger.

When each conversion is complete, the converted data are stored in the 16-frame ADB FIFO. The EOCIF[14:0] bit of the converted channels will be set individually to indicate the end of the conversion, and the ADC interrupt is generated if the related EOCIE[14:0] bits are set.

If a sequence of multiple channels is selected, and the conversion of all channels of the sequence is complete, the EOSIF bit will be set to indicate the end of sequence, and the ADC interrupt is generated if the EOSIE bit is set.

Then the ADC stops until a new external trigger event occurs or the ADSTART bit is set again.

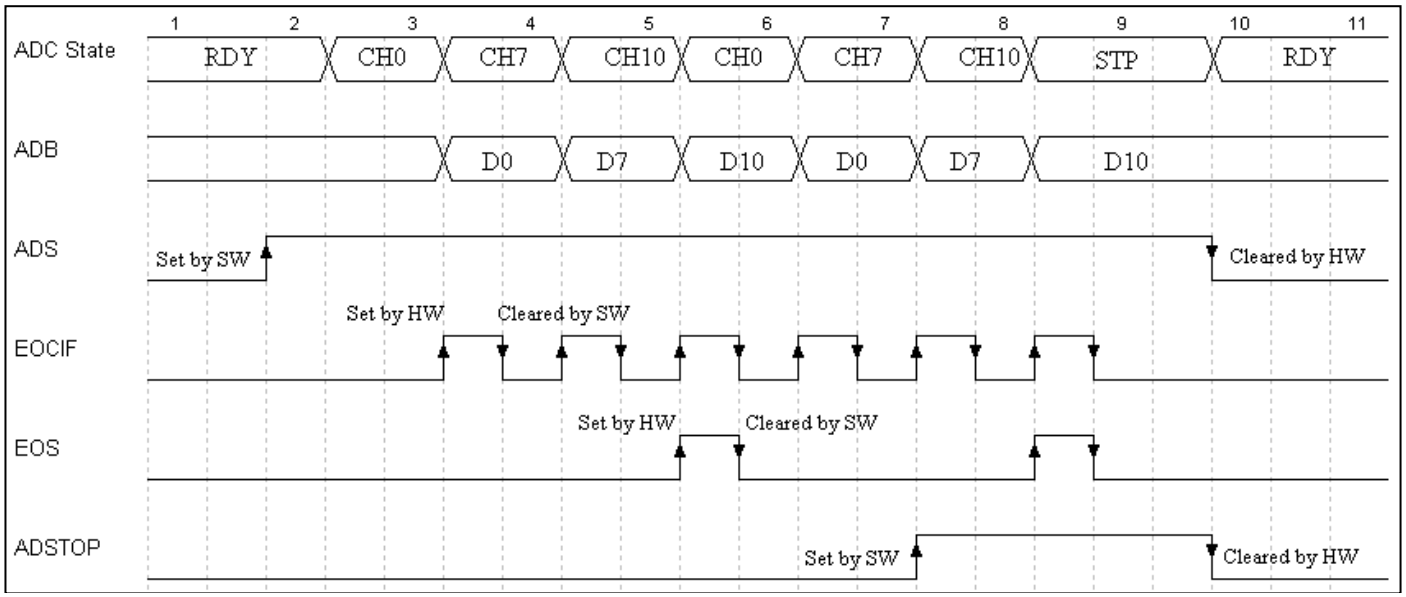


8.6.2 Continuous Mode

Set both SCMODE bit and ADS bit to 1 to start the ADC conversion in Continuous mode. The ADC will convert either one channel or a sequence of multiple channels once and then automatically re-starts and continuously performs the same sequence of conversions until the ADSTOP bit is set to 1 to stop the continuous mode.

When each conversion is complete, the converted data are stored in the 16-frame ADB FIFO. The EOCIF[14:0] bit of the converted channels will be set individually to indicate the end of the conversion, and the ADC interrupt is generated if the related EOCIE[14:0] bits are set.

If a sequence of multiple channels is selected, and the conversion of all channels of the sequence is complete, the EOSIF bit will be set to indicate the end of sequence, and the ADC interrupt is generated if the EOSIE bit is set.



8.7 ADC WATCHDOG WINDOW

The ADC window watchdog (AWW) is enabled by setting AWWEN bit to 1, it will monitor whether the measured data is inside or outside of the AWW high threshold value and low threshold value in ADC_AWW register. The AWWIF bit is set to 1 to notify the system when a desired condition is matched, and the ADC interrupt is generated if the AWWIE bit is set.

- AWDWINDOW[1:0] = 00b: ADB[11:0] > HT[11:0]

ADC_AWWTH	ADB[11:0]	AWWIF
	0x3FF	AWWIF=1
	…	
	0x081	
HT[11:0]=0x080	0x080	AWWIF=0
	0x07F	
	…	
LT[11:0]=0x000	0x000	

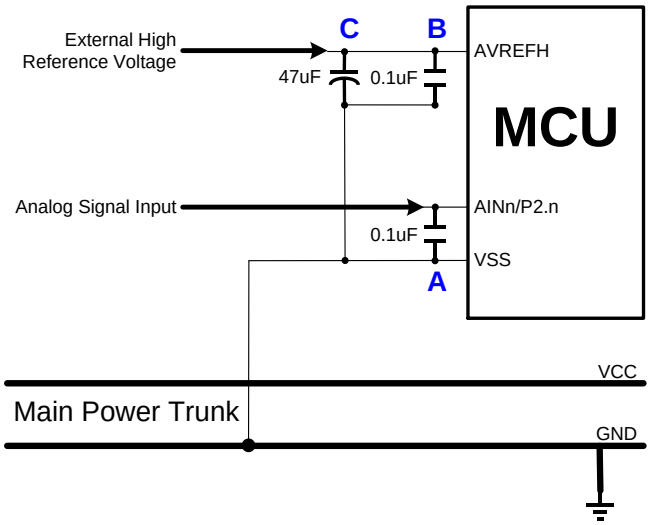
- AWDWINDOW[1:0] = 10b : ADB[11:0] < LT[11:0]

ADC_AWWTH	ADB[11:0]	AWWIF
HT[11:0]=0x3FF	0x3FF	AWWIF=0
	…	
	0x041	
LT11:0]=0x040	0x040	
	0x03F	AWWIF=1
	…	
	0x000	

- AWDWINDOW[1:0] = 01b: LT[11:0] < ADB[11:0] < HT[11:0]

ADC_AWWTH	ADB[11:0]	AWWIF
	0x3FF	AWWIF=0
	…	
	0x081	
HT[11:0]=0x80	0x080	
	0x07F	AWWIF=1
	…	
	0x041	
LT[11:0]=0x040	0x040	AWWIF=0
	0x03F	
	…	
	0x000	

8.8 ADC CIRCUIT



The analog signal is inputted to ADC input pin “AINn/P2.n”. The ADC input signal must be through a 0.1uF capacitor “A”. The 0.1uF capacitor is set between ADC input pin and VSS pin, and must be on the side of the ADC input pin as possible. Don’t connect the capacitor’s ground pin to ground plain directly, and must be through VSS pin. The capacitor can reduce the power noise effective coupled with the analog signal.

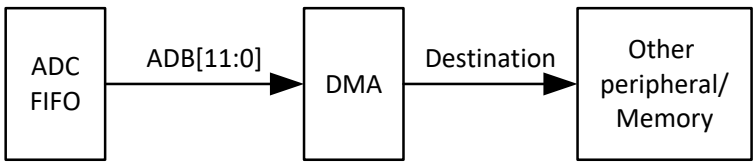
If the ADC high reference voltage is from external voltage source, the external high reference is connected to AVREFH pin (P2.0). The external high reference source must be through a 47uF ”C” capacitor first, and then 0.1uF capacitor “B”. These capacitors are set between AVREFH pin and VSS pin, and must be on the side of the AVREFH pin as possible. Don’t connect the capacitor’s ground pin to ground plain directly, and must be through VSS pin.

8.9 ADC Conversion Trigger Source

The ADC conversion can be triggered by SW or external trigger source. The SW triggers ADC conversion through ADS bit. External trigger sources are interrupt request of timer. They are CT16B0/1/5 MR0~MR9 interrupt request. The rising edge of the external trigger source can trigger ADC conversion, therefore, when the timer interrupt is issued, an ADC conversion will occur.

8.10 DMA Mode

The ADC DMA mode is to use DMA engine to move data out ADC. Before the DMA transfer start, DMA engine must be set up first. In ADC DMA Mode, DMA receive data from ADC and send to other peripherals or memories.

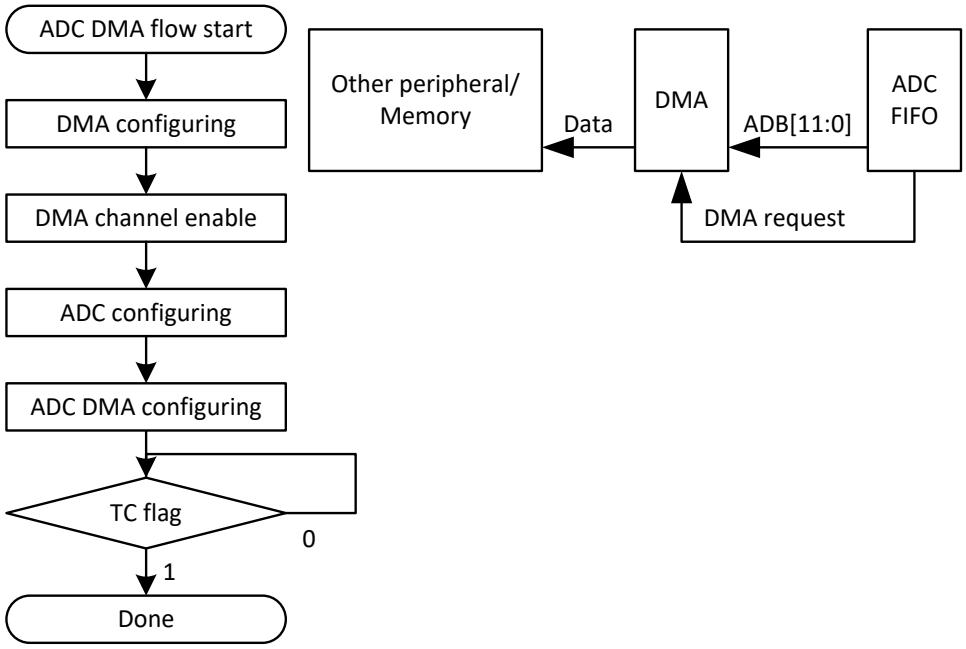


✸ ***Note:***

1. ***The burst data can NOT be larger than the FIFO size of ADC.***
2. ***ADC FIFO threshold level = burst size – 1***

8.10.1 Flow Chart

The following register programming flow chart used for ADC DMA mode. This example will turn on DMA Channel to move data from ADC data register to other peripherals or SRAM. When ADC FIFO is greater than the threshold, DMA request is issued.



8.10.2 DMA Configuration Recommendations

If data in ADC FIFO > ADC FIFO threshold, the ADC FIFO request will be issued.

DMAC setting for ADC data buffer (12-bit)				Recommend setting for ADC register (Assume ADC FIFO depth == 17 x 13-bit)	
Cn_CSR **SRC_WIDTH**	Cn_CSR **SRC_SIZE**	Cn_SIZE **TOT_SIZE**	Cn_CSR **DST_WIDTH**	**DMA_FIFO_TH**	DMA_SIZE (half-word unit)
0x1 (half-word)	0x0 (burst=1)	N (half-word)	0x1 (half-word) 0x0 (byte)	0x0	N
0x1 (half-word)	0x1 (burst=4)	N (half-word)	0x2 (word) 0x1 (half word) 0x0 (byte)	0x3	N
0x1 (half-word)	0x2 (burst=8)	N (half-word)	0x2 (word) 0x1 (half word) 0x0 (byte)	0x7	N
0x1 (half-word)	0x3 (burst=16)	N (half-word)	0x2 (word) 0x1 (half word) 0x0 (byte)	0xF	N

8.11 ADC REGISTERS

Base Address: 0x4002 6000

8.11.1 ADC Management register (ADC_ADM)

Address Offset: 0x00

✸ ***Note:***

1. ***When ADC is enabled (ADENB=1) and global channel is enabled (GCHS=1), the ADC shared pins transfers to ADC purpose and disable GPIO function and disable pull-up/pull-down resistor by HW automatically, the P2.n/AINn's digital I/O function including pull-up is isolated.***
2. ***When ADC is disabled (ADENB=0) or global channel is disabled (GCHS=0) , the ADC pins returns to last GPIO status.***
3. ***If P2.0 is used as external reference voltage input pin, users should set P2.0 as input mode without pull-up.***
4. ***GAINSEL bit shall be set when ADC sample rate is more than 250KHz, or higher resolution is needed when using internal reference voltage, P2.0 becomes VREF pin, and MUST connect 0.1uF capacitor to VSS.***
5. ***If ADC internal voltage reference, CMP internal voltage reference, or DAC internal voltage reference is used, ADC_ADM2 SHALL be set to 0x3.***

Bit	Name	Description	Attribute	Reset
31:22	Reserved		R	0
21	GAINSEL	VIREF gain select bit 0: Low-gain 1: High-gain (Shall be selected when ADC sample rate is more than 250KHz, or higher resolution is needed when using internal reference voltage, P2.0 becomes VREF pin, and MUST connect 0.1uF capacitor to VSS)	R/W	0
20:19	Reserved		R	0
18	FIFORST	ADB FIFO Reset bit 0: No effect 1: Reset ADB FIFO (Data in FIFO is cleared). This bit will be cleared by HW automatically.	W	0
17	OVRMODE	Overrun mode select bit 0: Preserve (The old data is preserved when the overrun occurs) 1: Overwrite (The ADC data FIFO is overwritten with the latest data)	R/W	0
16	ADSTOP	ADC stop control bit. 0: No effect 1: Stop ADC converting. ***ADSTOP is cleared when the end of ADC converting is stopped automatically.***	R/W	0
15:13	VHS[2:0]	Internal reference voltage level selection. 000: Internal 2.0V as ADC internal reference high voltage 001: Internal 3.0V as ADC internal reference high voltage 010: Internal 4.5V as ADC internal reference high voltage 011: Internal 1.5V as ADC internal reference high voltage 100: VDD as ADC internal reference voltage, Internal 2.0V as AIN18 101: VDD as ADC internal reference voltage, Internal 3.0V as AIN18 110: VDD as ADC internal reference voltage, Internal 4.5V as AIN18 111: VDD as ADC internal reference voltage, Internal 1.5V as AIN18	R/W	000b
12	AVREFHSEL	ADC high reference voltage source select bit 0: Internal reference voltage. (P2.0 is GPIO or AIN0 pin) 1: Enable external reference voltage from P2.0	R/W	0
11	ADENB	ADC Enable bit. **In power saving mode, disable ADC to reduce power consumption.** 0: Disable 1: Enable	R/W	0
10:8	ADCKS[2:0]	ADC Clock (MCLK) source divider 000: ADC_PCLK / 1 001: ADC_PCLK / 2 010: ADC_PCLK / 4 011: ADC_PCLK / 8	R/W	0

		100: ADC_PCLK / 16 101: ADC_PCLK / 32 Other: Reversed		
7	ADLEN	ADC resolution control bit. 0: 8-bit ADC 1:12-bit ADC	R/W	1
6	ADS	ADC start control bit. 0: ADC converting stops. 1: Start to execute ADC converting. ***ADS is cleared when the end of ADC converting automatically.***	R/W	0
5	EOC	ADC status bit. ***Indicates ADC processing status immediately and is cleared when ADS = 1.*** 0: ADC is progressing (converting or calibrating). 1: End of converting and reset ADS bit.	R	0
4	GCHS	ADC global channel select bit. 0: Disable AIN channel 1: Enable AIN channel	R/W	0
3:0	Reserved		R	0

8.11.2 ADC Data register (ADC_ADB)

Address Offset: 0x04

The ADC has a 22-frame data buffer. This register is the top layer of the ADC data FIFO, and contains the oldest A/D converter result.

✸ ***Note: The initial value of ADC buffer (ADB) after reset is unknown.***

Bit	Name	Description	Attribute	Reset
31:13	Reserved		R	0
12	FIRST	First ADB of Multiple channels 0: Not first ADB of Multiple channels 1: First ADB of Multiple channels	R	0
11:0	ADB[11:0]	ADB11~ADB4 bits for 8-bit ADC ADB11~ADB0 bits for 12-bit ADC	R	0

The AINx input voltage vs. ADB's output data

AIN n	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
0/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	0
1/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	1
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
4094/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	0
4095/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	1

8.11.3 ADC Interrupt Enable register (ADC_IE)

Address offset: 0x0C

This register allows control over which A/D channels generate an interrupt when a conversion is complete. For example, it may be desirable to use some A/D channels to monitor sensors by continuously performing conversions on them. The most recent results are read by the application program whenever they are needed. In this case, an interrupt is not desirable at the end of each conversion for some A/D channels.

Bit	Name	Description	Attribute	Reset
31:28	Reserved		R	0
27	EOCALIE	End of calibration interrupt enable bit 0: Disable 1: Enable	R/W	0
26	OVRIE	Overrun raw interrupt enable bit 0: Disable	R/W	0

		1: Enable		
25	AWWIE	ADC Window WDT interrupt enable bit 0: Disable 1: Enable	R/W	0
24	EOSIE	End of sequence interrupt enable bit 0: Disable 1: Enable	R/W	0
23:22	Reserved		R	0
21:0	EOCIE[21:0]	End of conversion interrupt enable bit 0: Disable 1: Enable	R/W	0

8.11.4 ADC Raw Interrupt Status register (ADC_RIS)

Address offset: 0x10

Bit	Name	Description	Attribute	Reset
31:28	Reserved		R	0
27	EOCALIF	End of calibration raw interrupt flag 0: No interrupt requirements met 1: Interrupt requirements met	R	0
26	OVRIF	Overrun raw interrupt flag 0: No interrupt requirements met 1: Interrupt requirements met	R	0
25	AWWIF	ADC window WDT raw interrupt flag 0: No interrupt requirements met 1: Interrupt requirements (ADB matches AWW condition) met	R	0
24	EOSIF	End of sequence raw interrupt flag 0: No sequence of conversion is completed 1: Interrupt requirements (The multiple channels selected by CHS[14:0] bits are all converted) met	R	0
23:22	Reserved		R	0
21:0	EOCIF[21:0]	End of conversion interrupt flag. 0: No conversion is completed on AINx 1: Interrupt requirements (AINx finishes conversion) met on AINx	R	0

8.11.5 ADC Raw Interrupt Status register (ADC_IC)

Address offset: 0x18

Bit	Name	Description	Attribute	Reset
31:28	Reserved		R	0
27	EOCALIC	End of calibration interrupt flag clear bit 0: No effect 1: Clear EOCALIF bit	W	0
26	OVRIC	Overrun raw interrupt flag clear bit 0: No effect. 1: Clear OVRIF bit	W	0
25	AWWIC	ADC window WDT raw interrupt flag clear bit 0: No effect. 1: Clear AWWIF bit	W	0
24	EOSIC	End of sequence raw interrupt flag clear bit 0: No effect. 1: Clear EOSIF bit	W	0
23:22	Reserved		R	0
21:0	EOCIC[21:0]	End of conversion raw interrupt flag clear bit. 0: No effect. 1: Clear related EOCIF[x] bit	W	0

8.11.6 ADC Convert Control register (ADC_CONVCTRL)

Address offset: 0x1C

✸ ***Note:***

1. ADC can't be stopped in Continuous mode when only 1 channel is enabled and CH bit is 1 (multiple channel).

2. Recommended to stop ADC when EOSIF is set if CH bit is 1 (multiple channel).

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R	0
29:26	ADSTRI[3:0]	ADC start conversion trigger source selection 0: ADS bit (Software mode) 1: CT16B0 MR0IF rising edge 2: CT16B0 MR1IF rising edge 3: CT16B0 MR2IF rising edge 4: CT16B0 MR3IF rising edge 5: CT16B0 MR9IF rising edge 6: CT16B1 MR9IF rising edge 10: CT16B5 MR9IF rising edge Other: Reserved	R/W	0
25	SCMODE	Single/Continuous mode select bit 0: Single mode 1: Continuous mode	R/W	0
24	CH	Converting channel number 0: Single channel 1: Multiple channels	R/W	0
23:22	Reserved		R	0
21:0	CHS[21:0]	AINx (AIN0~AIN15=External channel, AIN16=VDD, AIN17=VSS, AIN18=Internal reference voltage, AIN19=OP0O, AIN20=OP1O, AIN21=OP2O) selection bits 0: Not selected to convert 1: AINx selected to convert	R/W	0

8.11.7 ADC Window Watchdog register (ADC_AWW)

Address offset: 0x20

Bit	Name	Description	Attribute	Reset
31:27	Reserved		R	0
26:25	WINDOW[1:0]	AWW window select bits 00: ADB > AWDHT 01: AWDLT < ADB < AWDHT 10: ADB < AWDLT 11: Reserved	R/W	0
24	AWWENB	ADC Window Watchdog enable bit 0: Disable 1: Enable	R/W	0
23:22	Reserved		R	0
21:0	AWWCHS[21:0]	AWW channel selection bits 0: Not selected 1: AINx selected to be monitored	R/W	0

8.11.8 ADC Window Watchdog Threshold register (ADC_AWWTH)

Address offset: 0x24

Bit	Name	Description	Attribute	Reset
31:28	Reserved		R	0
27:16	HT[11:0]	AWW window High threshold value	R/W	0
15:12	Reserved		R	0
11:0	LT[11:0]	AWW window Low threshold value	R/W	0

8.11.9 ADC Management register 1 (ADC_ADM1)

Address offset: 0x30

✸ ***Note:***

1. The calibration must be run at least one time before starting A/D conversion.

2. CALIVALENB SHALL be set 1.

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
4	ADCCR	ADC Conversion current bit → IP ADCCR 0: ADC clock up to 16MHz 1: ADC clock up to 4MHz, reduce the power consumption	R/W	0
3	ACS	ADC Calibration start bit 0: ADC calibration stops. 1: ADC calibration starts, and is cleared in the end of calibration automatically.	R/W	0
2	CALIVALENB	ADC calibration value enable bit 0: ADC conversion without calibration value 1: ADC conversion with calibration value	R/W	0
1:0	Reserved		R	0

8.11.10 ADC Calibration Status register (ADC_CALIST)

Address offset: 0x34

Bi	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	CAL_ERR	ADC calibration error 0: No error 1: Error occurs in ADC calibration period.	R	0

8.11.11 ADC DMA Mode register (ADC_DMA)

Address offset: 0x50

Bi	Name	Description	Attribute	Reset
31:27	DMA_FIFO_TH	ADC FIFO Threshold level 0: RX FIFO threshold level = 0 1: RX FIFO threshold level = 1 2: RX FIFO threshold level = 2 3: RX FIFO threshold level = 3 4: RX FIFO threshold level = 4 5: RX FIFO threshold level = 5 6: RX FIFO threshold level = 6 7: RX FIFO threshold level = 7 8: RX FIFO threshold level = 8 9: RX FIFO threshold level = 9 10: RX FIFO threshold level = 10 11: RX FIFO threshold level = 11 12: RX FIFO threshold level = 12 13: RX FIFO threshold level = 13 14: RX FIFO threshold level = 14 15: RX FIFO threshold level = 15 16: RX FIFO threshold level = 16 (FIFO Full) Other: Reserved	RW	0
26:23	Reserved		R	0
22	DMA_EN	ADC DMA mode enable 0: Disable 1: Enable	RW	0
21:0	DMA_SIZE	Total DMA transfer size	RW	0

9 RAIL TO RAIL ANALOG COMPARATOR

9.1 OVERVIEW

The micro-controller builds in 4 sets comparators (CMP0/1/2/3) with stopping PWM pulse generator function. The comparators are Rail-to-Rail structure. That means the input/output voltage is real from Vdd~Vss. When the positive input voltage is greater than the negative input voltage, the comparator output is high. When the positive input voltage is smaller than the negative input voltage, the comparator output is low.

The CMnOUT and CMnIF bits indicate the comparator result. The CMnOUT shows the comparator result immediately, but the CMnIF only indicates the event of the comparator result. The event condition is controlled by CMnG bit and includes rising edge (CMnOUT changes from low to high), falling edge (CMnOUT changes from high to low). The CMnIF = 1 condition makes the comparator interrupt service executed when CMnIE (comparator interrupt enable control bit) is set.

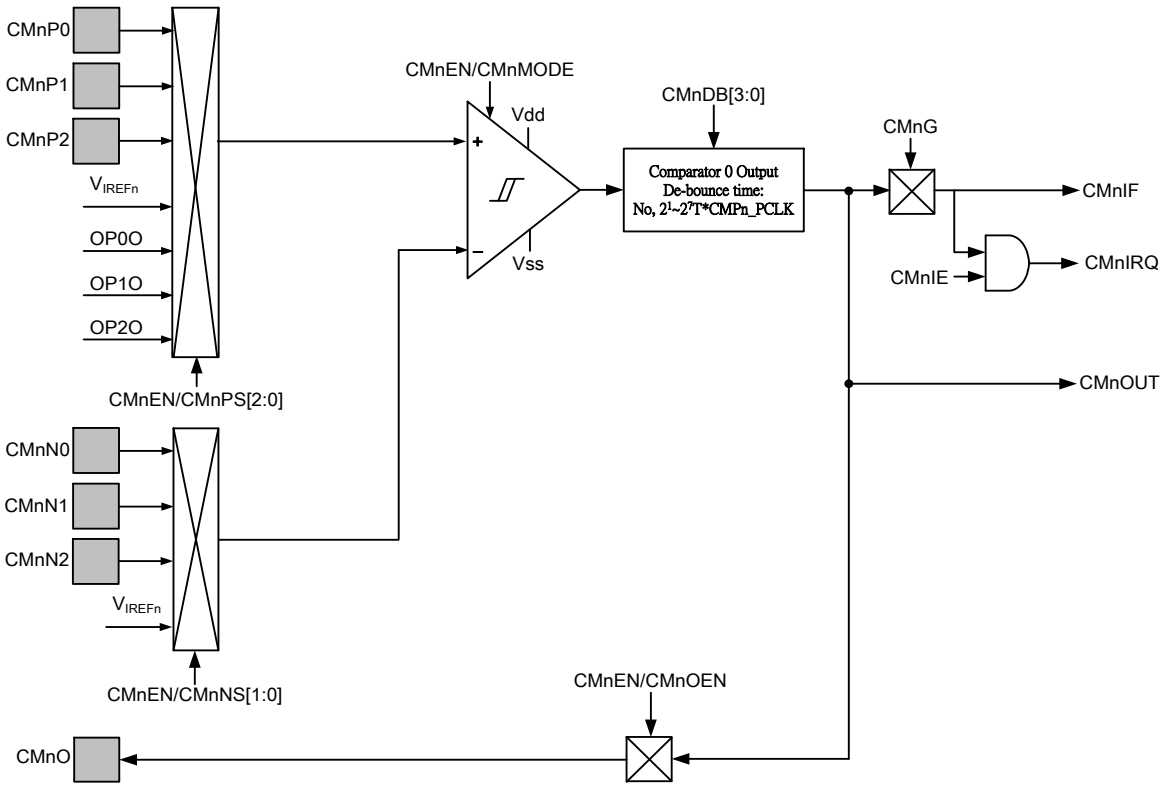
The comparator builds in two 8-bit DAC. DAC0 is used to provide the reference voltage either to CMP0/1/2 negative input or to CMP0/1/2 positive input, and is controlled by CMPIREFEN bit. DAC1 is used to provide the reference voltage either to CMP3 negative input or to CMP3 positive input and controlled by CMPIREF1EN bit.

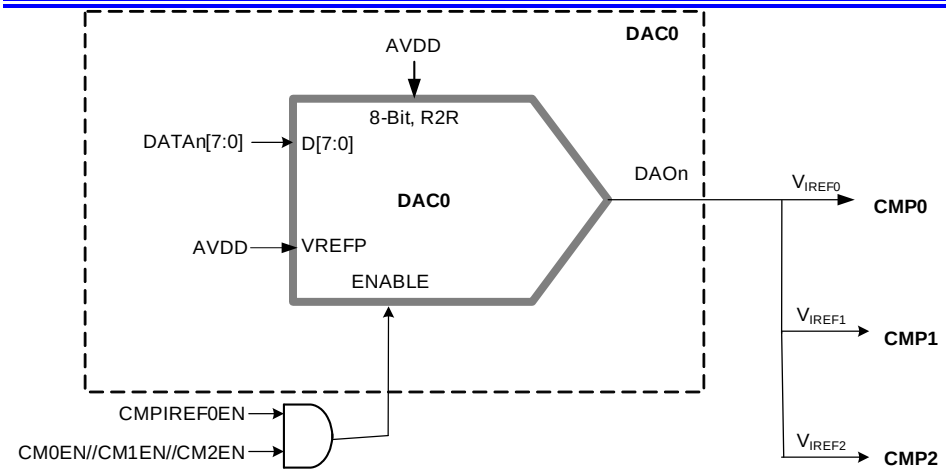
Besides, the comparator output signal is through a de-bounce circuit to filter comparator transient status. The de-bounce time is controlled by CMnDB[2:0] bits.

- CMP0/1/2

The comparator negative input terminal is controlled by CMnNS[1:0]. When CMnNS[1:0] = 00b~10b, the comparator negative input is from external voltage source through CMnN0, CMnN1, or CMnN2 pin. When CMnNS[1:0] = 11b, the comparator negative voltage source is from internal DAC0 and CMnN0/1/2 pins are GPIO function.

The comparator positive input terminal is controlled by CMnPS[2:0]. When CMnPS[1:0] = 000b, the comparator positive input is from DAC0 and CMnP0/1/2 pins are GPIO function. When CMnPS[1:0] = 001b~011b, the comparator positive input is from CMnP0, CMnP1, or CMnP2 pin. When CMnPS[1:0] = 100b~110b, the comparator positive input is from internal OPA output OP0O, OP1O, or OP2O, and CMnP0/1/2 pins are GPIO function.

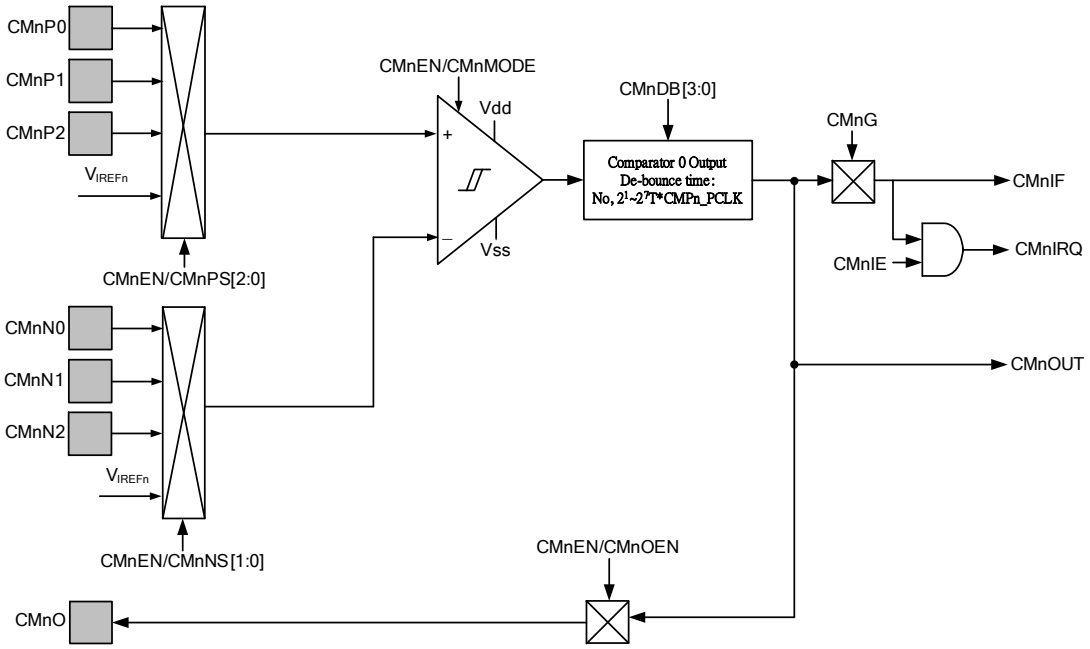


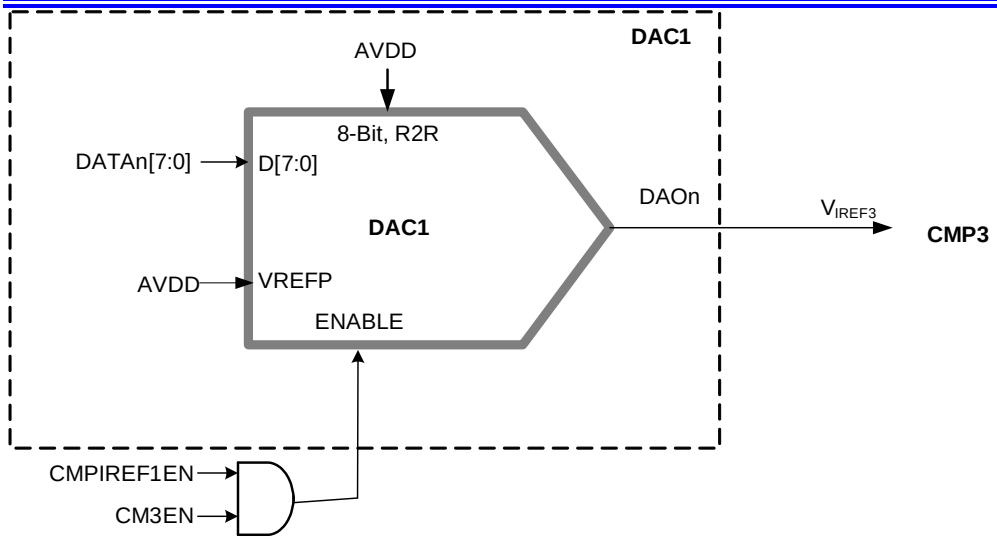


- CMP3

The comparator negative input terminal is controlled by CM3NS[1:0]. When CM3NS[1:0] = 00b~10b, the comparator negative input is from external voltage source through CM3N0, CM3N1, or CM3N2 pin. When CM3NS[1:0] = 11b, the comparator negative voltage source is from internal DAC1 and CM3N0/1/2 pins are GPIO function.

The comparator positive input terminal is controlled by CM3PS[2:0]. When CM3PS[1:0] = 000b, the comparator positive input is from DAC1 and CM3P0/1/2 pins are GPIO function. When CM3PS[2:0] = 001b~011b, the comparator positive input is from CM3P0, CM3P1, or CM3P2 pin.





The main purposes of comparator are as following.

- **Normal comparator function:** General comparator mode compares the two tensions of positive input terminal and negative input terminal.
- **Interrupt function:** When comparator output edge direction equals to edge selection, the CMnIRQ actives and the system points program counter to interrupt vector to do interrupt sequence when CMnIE (comparator interrupt enable control bit) is set.

9.2 NORMAL COMPARATOR MODE

9.2.1 COMPARATOR ENABLE

The comparator pins are shared with GPIO controlled by CMnEN bit.

CMnOEN controls Comparator output connected to GPIO or not. When CMnOEN=1, Comparator output terminal is connected to GPIO pins and isolate GPIO function. When CMnOEN=0, comparator output status can be read through CMnOUT flag and CMnO pin is GPIO mode.

- CMP0/1/2

When CMnEN=1 and CMnNS[1:0] = 00b~10b, the comparator negative voltage source is from external voltage source through CMnN0, CMnN1, or CMnN2 pin. When CMnNS[1:0] = 11b, the comparator negative voltage is from DAC0 and CMnN0/1/2 pins are GPIO function. When CMnPS[1:0] = 001b~011b, the comparator positive input is from CMnP0, CMnP1, or CMnP2 pin. When CMnPS[1:0] = 000b, the comparator positive voltage is from DAC0 and CMnP0/1/2 pins are GPIO function. When CMnPS[1:0] = 100b~110b, the comparator positive input is from internal OPA output OP0O, OP1O, or OP2O, and CMnP0/1/2 pins are GPIO function.

- CMP3

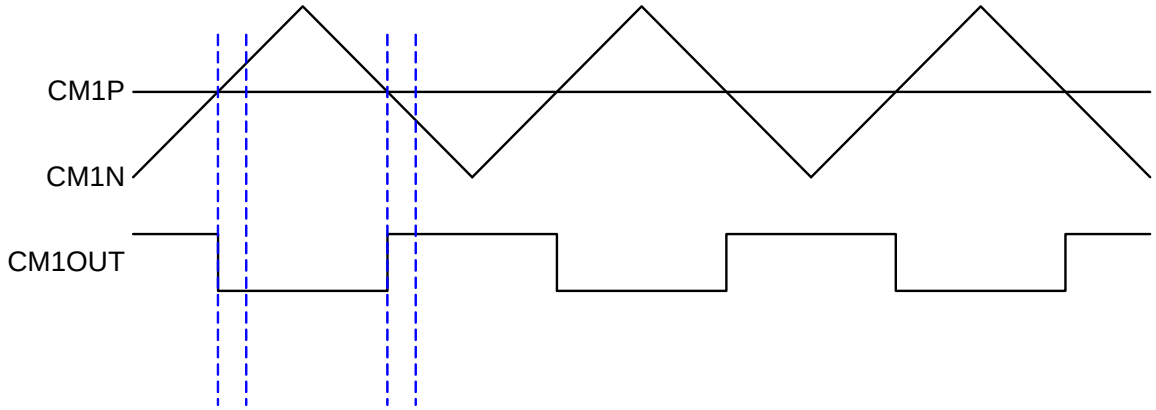
When CM3EN=1 and CM3NS[1:0] = 00b~10b, the comparator negative voltage source is from external voltage source through CM3N0, CM3N1, or CM3N2 pin. When CM3NS[1:0] = 11b, the comparator negative positive voltage is from DAC1 and CM3N0/1/2 pins are GPIO function. When CM3PS[2:0] = 001b~011b, the comparator positive input is from CM3P0, CM3P1, or CM3P2 pin. When CM3PS[2:0] = 000b, the comparator positive input is from DAC1 and CM3P0/1/2 pins are GPIO function.

9.2.2 CMnOUT, CMnG AND CMnIF

The CMnOUT and CMnIF bits indicate the comparator result. The CMnOUT shows the comparator result immediately, but the CMnIF only indicates the event condition of the comparator result.

Comparator n compares positive terminal's voltage and negative terminal's voltage, and then output result to output pin. When V+ > V-, comparator outputs high status. When V+ < V-, comparator outputs low status.

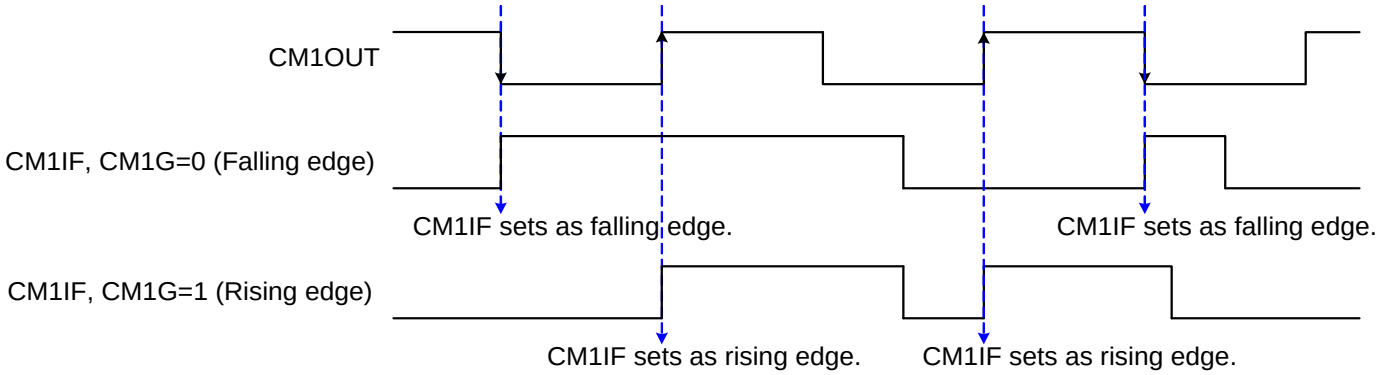
Take Comparator 1 as example,



The event condition is controlled by register and includes rising edge (CMnOUT changes from low to high), or falling edge (CMnOUT changes from high to low), controlled by CMnG bit. When CMnG= 0, the comparator n interrupt trigger direction is falling edge. When CMnG= 1, the comparator n interrupt trigger direction is rising edge. When comparator output edge event occurs and equal CMnG condition, CMnIF flag is issued. If CMnIE= 1, program counter points to interrupt vector to execute interrupt service routine.

✸ ***Note: CMnOUT is comparator raw output without latch. It varies depend on the comparator process result. But the CMnIF is latch comparator output result. It must be cleared by program.***

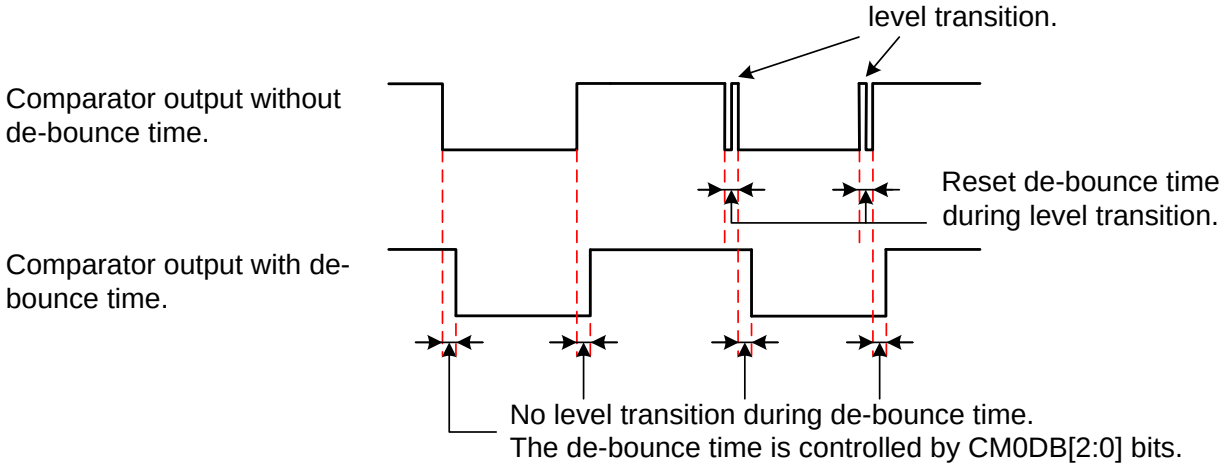
Take Comparator 1 as example,



***. CM1IF is cleared by program.**

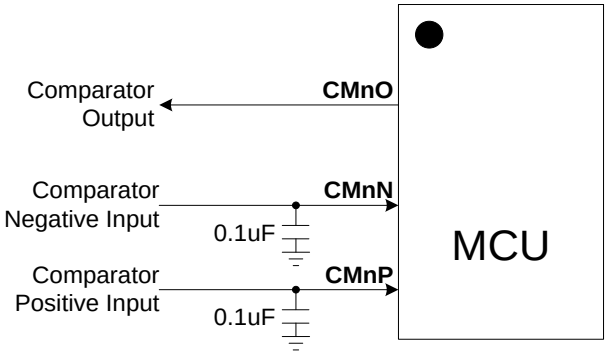
9.2.3 COMPARATOR OUTPUT DEBOUNCE TIME CONTROL

The critical condition is comparator positive voltage equal to comparator negative voltage, and the voltage range is decided comparator offset parameter of input common mode. In the voltage range, the comparator output signal is unstable and keeps oscillating until the differential voltage exits the range. In the condition, the comparator flag (CM0IF) latches the first exchanging and issue the status, but the status is a transient, not a stable condition. So, the comparator builds in a filter to de-bounce the transient condition. The comparator output signal is through a de-bounce circuit to filter comparator transient status. The de-bounce time is controlled by CM0DB[2:0] bits that means the comparator minimum response time is 2*CMP_PCLK, 4*CMP_PCLK, 8*CMP_PCLK, …, 128*CMP_PCLK, or no de-bounce. The de-bounce time depends on the signal slew rate and selected by program.



9.3 COMPARATOR APPLICATION NOTICE

The comparator is to compares the positive voltage and negative voltage to output result. The positive and negative sources are analog signal. In hardware application circuit, the comparator input pins must be connected a 0.1uF comparator to reduce power noise and make the input signal more stable. The application circuit is as following.



9.4 COMPARATOR Registers

Base Address: 0x4002 8000 (CMP)

9.4.1 CMP Control register (CMP_CTRL)

Address Offset: 0x00

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R	0
29	CM1G	CMP1 interrupt trigger direction control bit. 0: Falling edge trigger (CMP1 output status is from high to low as V_+ < V-) 1: Rising edge trigger (CMP1 output status is from low to high as V+ > V-)	R/W	0
28	Reserved		R	0
27	CM1OEN	CMP1 output pin control bit. 0: Disable (CM1O is GPIO mode) 1: Enable	R/W	0
26:22	Reserved		R	0
21:20	CM1NS[1:0]	CMP1 negative input pin selection bit 00: CM1N0 is comparator negative input pin, and isolate GPIO function 01: CM1N1 is comparator negative input pin, and isolate GPIO function 10: CM1N2 is comparator negative input pin, and isolate GPIO function 11: V_{IREF1}. CM1N0/CM1N1/CM1N2 pins are GPIO mode.	R/W	0
19:17	CM1PS[2:0]	CMP1 Positive input selection bits 000: V_{IREF1}. CM1P0/CM1P1/CM1P2 pins are GPIO mode 001: CM1P0 010: CM1P1 011: CM1P2 100: OP0O 101: OP1O 110: OP2O Other: Reserved	R/W	0
16	CM1EN	CMP1 enable bit. 0: Disable (CM1P0/1/2, CM1N0/1/2, CM1O are GPIO mode) 1: Enable	R/W	0
15:14	Reserved		R	0
13	CM0G	CMP0 interrupt trigger direction control bit. 0: Falling edge trigger (CMP0 output status is from high to low as V+ < V-) 1: Rising edge trigger (CMP0 output status is from low to high as V+ > V-)	R/W	0
12	Reserved		R	0
11	CM0OEN	CMP0 output pin control bits 0: Disable (CM0O is GPIO mode) 1: Enable	R/W	0
10:6	Reserved		R	0
5:4	CM0NS[1:0]	CMP0 negative input selection bits 00: CM0N0 is comparator negative input pin, and isolate GPIO function 01: CM0N1 is comparator negative input pin, and isolate GPIO function 10: CM0N2 is comparator negative input pin, and isolate GPIO function 11: V_{IREF0}. CM0N0/CM0N1/CM0N2 pins are GPIO mode.	R/W	00b
3:1	CM0PS[2:0]	CMP0 Positive input selection bits 000: V_{IREF0}. CM0P0/CM0P1/CM0P2 pins are GPIO mode 001: CM0P0 010: CM0P1 011: CM0P2 100: OP0O 101: OP1O 110: OP2O Other: Reserved	R/W	00b
0	CM0EN	CMP0 enable bit. 0: Disable (CM0P0/1/2, CM0N0/1/2, CM0O are GPIO mode) 1: Enable	R/W	0

9.4.2 CMP Control register 1 (CMP_CTRL1)

Address Offset: 0x04

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R	0
29	CM3G	CMP3 interrupt trigger direction control bit 0: Falling edge trigger (CMP3 output status is from high to low as V+ < V-) 1: Rising edge trigger (CMP3 output status is from low to high as V+ > V-)	R/W	0
28	Reserved	Reserved	R	0
27	CM3OEN	CMP3 Output pin control bits 0: Disable (CM3O is GPIO mode) 1: Enable	R/W	0
26:22	Reserved	Reserved	R	0
21:20	CM3NS[1:0]	CMP3 Negative input pin 0: CM3N0 1: CM3N1 2: CM3N2 3: V_{IREF3}. CM3N0/CM3N1/CM3N2 pins are GPIO mode	R/W	0
19:17	CM3PS[2:0]	CMP3 Positive input selection bits 0: V_{IREF3}. CM3P0/CM3P1/CM3P2 pins are GPIO mode 1: CM3P0 2: CM3P1 3: CM3P2 Other: Reserved	R/W	0
16	CM3EN	CMP3 Enable bit 0: Disable (CM3P0/1/2, CM3N0/1/2, CM3O are GPIO mode) 1: Enable	R/W	0
15:14	Reserved	Reserved	R	0
13	CM2G	CMP2 interrupt trigger direction control bit 0: Falling edge trigger (CMP2 output status is from high to low as V+ < V-) 1: Rising edge trigger (CMP2 output status is from low to high as V+ > V-)	R/W	0
12	Reserved	Reserved	R	0
11	CM2OEN	CMP2 Output pin control bits 0: Disable (CM2O is GPIO mode) 1: Enable	R/W	0
10:6	Reserved	Reserved	R	0
5:4	CM2NS[1:0]	CMP2 Negative input pin 0: CM2N0 1: CM2N1 2: CM2N2 3: V_{IREF2}. CM2N0/CM2N1/CM2N2 pins are GPIO mode	R/W	00b
3:1	CM2PS[2:0]	CMP2 Positive input selection bits 0: V_{IREF2}. CM2P0/CM2P1/CM2P2 pins are GPIO mode 1: CM2P0 2: CM2P1 3: CM2P2 4: OP0O 5: OP1O 6: OP2O Other: Reserved	R/W	00b
0	CM2EN	CMP2 Enable bit 0: Disable (CM2P0/1/2, CM2N0/1/2, CM2O are GPIO mode) 1: Enable	R/W	0

9.4.3 CMP Internal Reference Voltage Source register (CMP_VIREF)

Address Offset: 0x08

Bit	Name	Description	Attribute	Reset
31:28	Reserved		R	0
27:20	DATA1[7:0]	8-bit DAC1 setup date bits	R/W	0

19:17	Reserved		R	0
16	CMP1REF1EN	CMP3 internal reference voltage (VIREF3) enable 0: Disable 1: Enable	R/W	0
15:12	Reserved		R	0
11:4	DATA0[7:0]	8-bit DAC0 setup date bits	R/W	0
3:1	Reserved		R	0
0	CMPIREFEN	CMP0/1/2 internal reference voltage ($V_{IREF0/1/2}$) enable 0: Disable 1: Enable	R/W	0

9.4.4 CMP Output Status register (CMP_OS)

Address Offset: 0x0C

✸ ***Note: CMnOUT is comparator raw output without latch. It varies depend on the comparator process result. But the CMnIF is latch comparator output result. It must be cleared by program.***

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CM3OUT	CMP3 output flag bit. 0: $V_{3+} < V_{3-}$ 1: $V_{3+} > V_{3-}$	R	0
2	CM2OUT	CMP2 output flag bit. 0: $V_{2+} < V_{2-}$ 1: $V_{2+} > V_{2-}$	R	0
1	CM1OUT	CMP1 output flag bit. 0: $V_{1+} < V_{1-}$ 1: $V_{1+} > V_{1-}$	R	0
0	CM0OUT	CMP0 output flag bit. 0: $V_{0+} < V_{0-}$ 1: $V_{0+} > V_{0-}$	R	0

9.4.5 CMP Interrupt Enable register (CMP_IE)

Address Offset: 0x10

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CM3IE	CMP3 interrupt enable control bit. 0: Disable 1: Enable	R/W	0
2	CM2IE	CMP2 interrupt enable control bit. 0: Disable 1: Enable	R/W	0
1	CM1IE	CMP1 interrupt enable control bit. 0: Disable 1: Enable	R/W	0
0	CM0IE	CMP0 interrupt enable control bit. 0: Disable 1: Enable	R/W	0

9.4.6 CMP Raw Interrupt Status register (CMP_RIS)

Address offset: 0x14

This register indicates the status for comparator raw interrupts. A CMP0/1 interrupt is sent to the interrupt controller if the corresponding CMnIE bit is set.

✸ ***Note: CMnOUT is comparator raw output without latch. It varies depend on the comparator process result. But the CMnIF is latch comparator output result. It must be cleared by program.***

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CM3IF	CMP3 raw interrupt flag 0: No interrupt on CMP3 1: Interrupt requirements met on CMP3	R	0
2	CM2IF	CMP2 raw interrupt flag 0: No interrupt on CMP2 1: Interrupt requirements met on CMP2	R	0
1	CM1IF	CMP1 raw interrupt flag 0: No interrupt on CMP1 1: Interrupt requirements met on CMP1	R	0
0	CM0IF	CMP0 raw interrupt flag 0: No interrupt on CMP0 1: Interrupt requirements met on CMP0	R	0

9.4.7 CMP Interrupt Clear register (CMP_IC)

Address offset: 0x18

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CM3IC	Selects interrupt flag on CMP3 to be cleared 0: No effect 1: Clear interrupt flag on CMP3	W	0
2	CM2IC	Selects interrupt flag on CMP2 to be cleared 0: No effect 1: Clear interrupt flag on CMP2	W	0
1	CM1IC	Selects interrupt flag on CMP1 to be cleared 0: No effect 1: Clear interrupt flag on CMP1	W	0
0	CM0IC	Selects interrupt flag on CMP0 to be cleared 0: No effect 1: Clear interrupt flag on CMP0	W	0

9.4.8 CMP Output Debounce register (CMP_DB)

Address offset: 0x1C

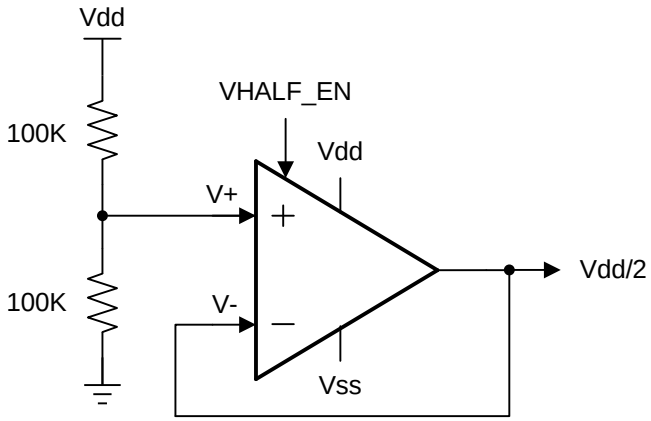
Bit	Name	Description	Attribute	Reset
31:15	Reserved		R	0
14:12	CM3DB[2:0]	Count for CMP3 output debounce time 000: No CMP3 output debounce time 001~111: CMP3 output debounce time = $2^{CM3DB[2:0]}$ * CMP3_PCLK cycle	W/R	000b
11	Reserved		R	0
10:8	CM2DB[2:0]	Count for CMP2 output debounce time 000: No CMP2 output debounce time 001~111: CMP2 output debounce time = $2^{CM2DB[2:0]}$ * CMP2_PCLK cycle	W/R	000b

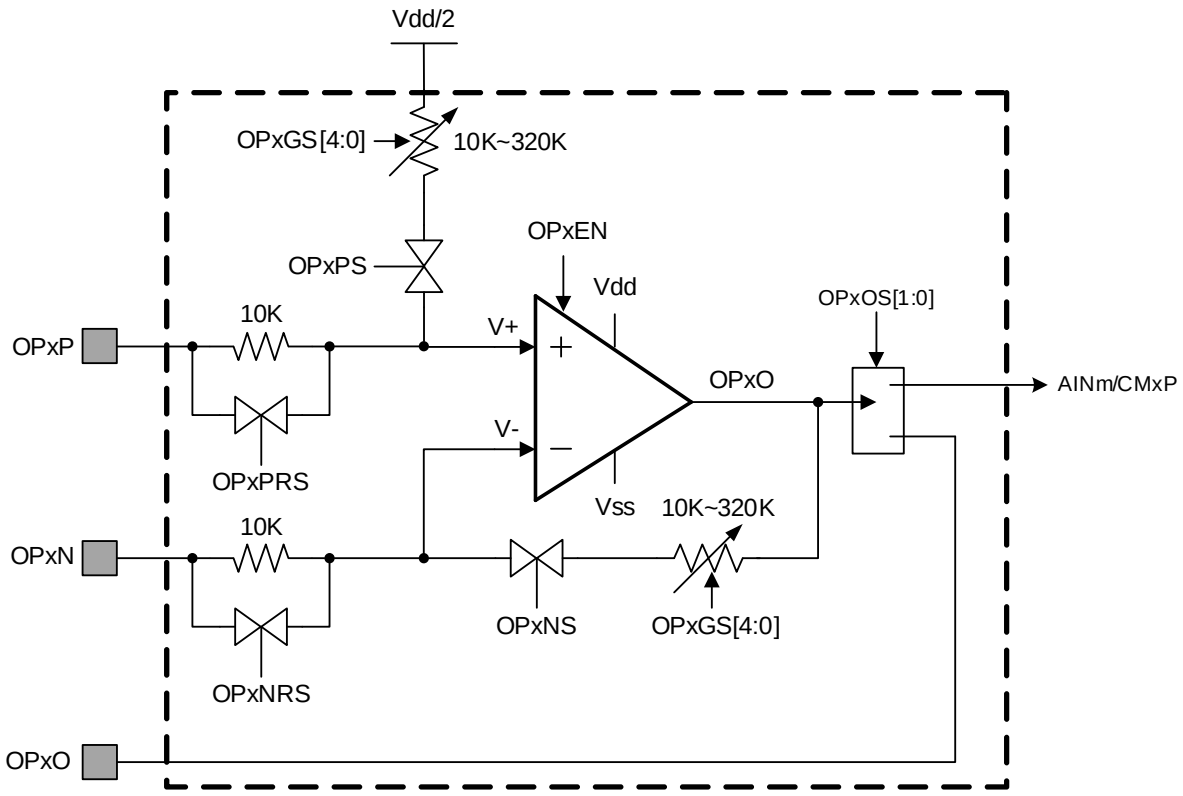
7	Reserved		R	0
6:4	CM1DB[2:0]	Count for CMP1 output debounce time 000: No CMP1 output debounce time 001~111: CMP1 output debounce time = $2^{CM1DB[2:0]}$ * CMP1_PCLK cycle	W/R	000b
3	Reserved		R	0
2:0	CM0DB[2:0]	Count for CMP0 output debounce time 000: No CMP0 output debounce time 001~111: CMP0 output debounce time = $2^{CM0DB[2:0]}$ * CMP0_PCLK cycle	W/R	000b

10 OPERATIONAL-AMPLIFIER (OPA)

10.1 OVERVIEW

The microcontroller builds in three operational amplifiers (OP0, OP1, and OP2) with programmable gain 1x~32x controlled by OPxGS[4:0] bits. The OP-Amp power range is VSS~VDD. OP-Amp input signal and output voltage are within the voltage range. The input of the PGA operates from rail to rail, and the output of the PGA is rail to rail for sufficiently high load resistance. The OP-Amp output pin can be programmable to connect with ADC input channel (AIN19, AIN20, and AIN21) and comparator positive input channel (CM0P, CM1P, and CM2P).





10.2 CONFIGURATION OF OPERATION

MODE	OPxPRS	OPxNRS	OPxPS	OPxNS	OPxOS[1:0]	OPxGS[4:0]	VHALF_EN	OPxEN	OPxP	OPxN	OPxO
Disable	-	-	-	-	-	-	0	0	GPIO mode	GPIO mode	GPIO mode
OPA mode	1	1	0	0	0	-	0	1	Follow OPA mode setting		
PGA mode 0	0	0	1	1	1	0~31	1	1	Follow PGA mode setting		GPIO mode
PGA mode 1	1	1	1	1	1	0~31	1	1	Follow PGA mode setting		GPIO mode

10.3 OPA REGISTERS

Base Address: 0x4002 A000 (OPA)

10.3.1 OPA Control register (OPA_CTRL)

Address Offset: 0x00

Bit	Name	Description	Attribute	Reset
31:19	Reserved		R	0
18:17	OP2OS	OPA2 output select bits 0: OP2O 1: ADC AIN21 and CM2P Other: Reserved	R/W	0
16	OP2EN	OPA2 enable bit 0: Disable 1: Enable	R/W	0
15:11	Reserved		R	0
10:9	OP1OS	OPA1 output select bits 0: OP1O 1: ADC AIN20 and CM1P Other: Reserved	R/W	0
8	OP1EN	OPA1 enable bit 0: Disable 1: Enable	R/W	0
7:3	Reserved		R	0
2:1	OP0OS	OPA0 output select bits 0: OP0O 1: ADC AIN19 and CM0P Other: Reserved	R/W	0
0	OP0EN	OPA0 enable bit 0: Disable 1: Enable	R/W	0

10.3.2 OPA PGA Control register (OPA_PGACTRL)

Address offset: 0x04

Bit	Name	Description	Attribute	Reset
31	VHALF_EN	VDD/2 bias enable bit 0: Disable 1: Enable	R/W	0
30:21	Reserved		R	0
20:16	OP1GS	OPA1 gain select bits Gain = (OP1GS[4:0] + 1)X =1X~32X	R/W	0
15	OP1NRS	OPA1 negative input resistor select bit 0: 10K Ohm 1: Short	R/W	0
14	OP1PRS	OPA1 positive input resistor select bit 0: 10K Ohm 1: Short	R/W	0
13	OP1NS	OPA1 negative input feedback enable bit 0: Disable 1: Enable	R/W	0
12	OP1PS	OPA1 positive input bias enable bit 0: Disable 1: Enable	R/W	0
11:9	Reserved		R	0
8:4	OP0GS	OPA0 gain select bits Gain = (OP0GS[4:0] + 1)X =1X~32X	R/W	0

3	OP0NRS	OPA0 negative input resistor select bit 0: 10K Ohm 1: Short	R/W	0
2	OP0PRS	OPA0 positive input resistor select bit 0: 10K Ohm 1: Short	R/W	0
1	OP0NS	OPA0 negative input feedback enable bit 0: Disable 1: Enable	R/W	0
0	OP0PS	OPA0 positive input bias enable bit 0: Disable 1: Enable	R/W	0

10.3.3 OPA PGA Control register 2 (OPA_PGACTRL2)

Address offset: 0x08

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0
8:4	OP2GS	OPA2 gain select bits Gain = (OP2GS[4:0] + 1)X =1X~32X	R/W	0
3	OP2NRS	OPA2 negative input resistor select bit 0: 10K Ohm 1: Short	R/W	0
2	OP2PRS	OPA2 positive input resistor select bit 0: 10K Ohm 1: Short	R/W	0
1	OP2NS	OPA2 negative input feedback enable bit 0: Disable 1: Enable	R/W	0
0	OP2PS	OPA2 positive input bias enable bit 0: Disable 1: Enable	R/W	0

11 16-BIT TIMER WITH CAPTURE FUNCTION

11.1 OVERVIEW

The microcontroller builds in three 16-bit timers (CT16B0/1/5). Each Counter/timer is designed to count cycles of the peripheral clock (PCLK) or an externally supplied clock and can optionally generate interrupts or perform other actions at specified timer values based on four match registers. Each counter/timer also includes one capture input to trap the timer value when an input signal transitions, optionally generating an interrupt.

In PWM mode, up to 12 match and a global match registers can be used to provide a single-edge controlled PWM output on the match output pins.

	CT16B0	CT16B1	CT16B5
Clock Source	HCLK PLL_VCO		HCLK PLL_VCO ELS
Counter Mode	Up/Down/ Center-aligned counting	Up/Down/ Center-aligned counting	Up/Down/ Center-aligned counting
PWM	PWM0/1/2/3+ PWM0N/1N/2N/3N	PWM0/1/2/3	PWM0/1/2/3
PCLK	max 120MHz	max 120MHz	max 120MHz

11.2 FEATURES

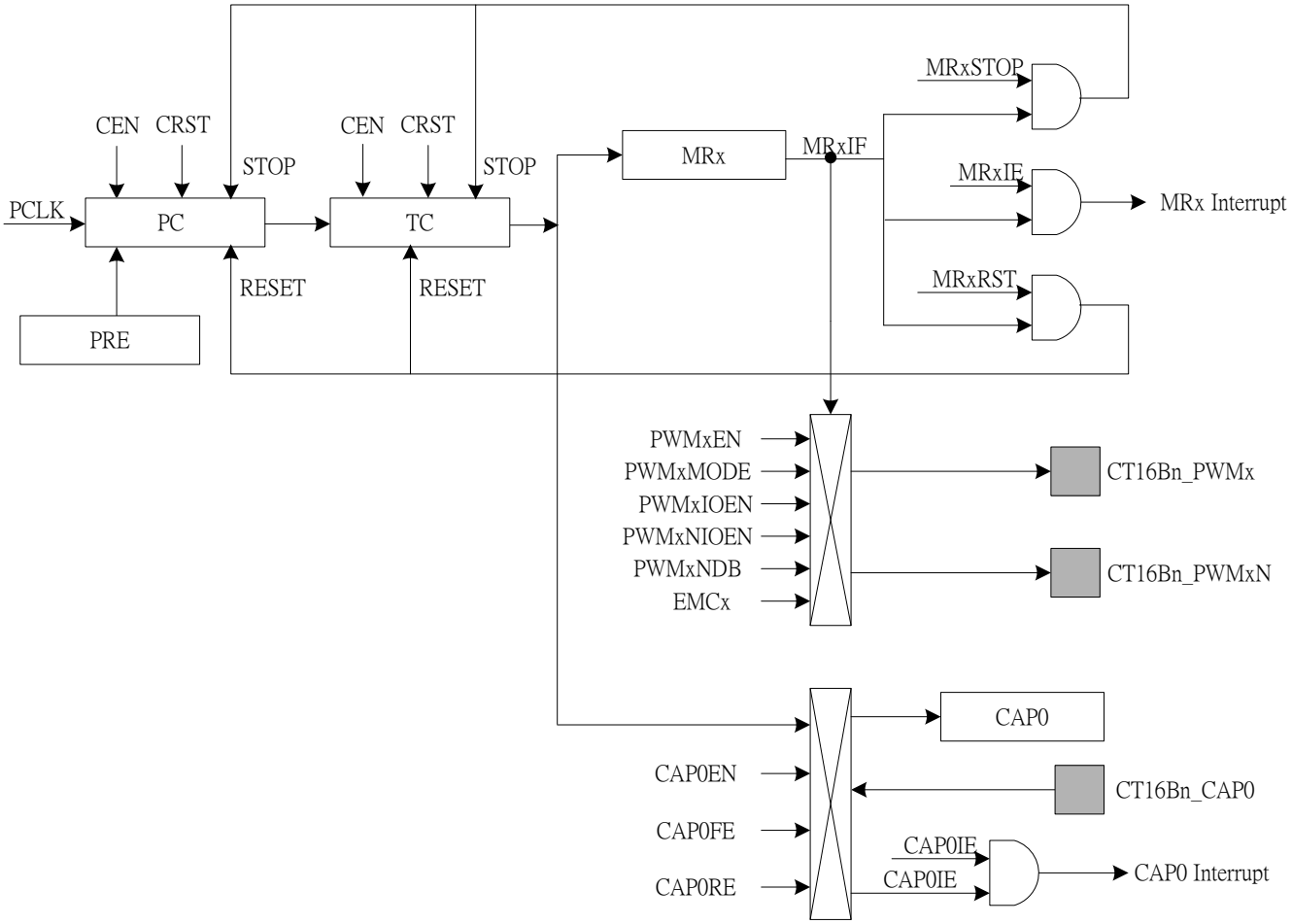
- Three 16-bit counter/timers with a programmable 8-bit prescaler.
- Counter or timer operation
- Three 16-bit capture channels that can take a snapshot of the timer value when an input signal transitions. A capture event may also optionally generate an interrupt.
- The timer and prescaler may be configured to be cleared on a designated capture event. This feature permits easy pulse-width measurement by clearing the timer on the leading edge of an input pulse and capturing the timer value on the trailing edge.
- Six 16-bit match registers that allow:
  - Continuous operation with optional interrupt generation on match.
  - Stop timer on match with optional interrupt generation.
  - Reset timer on match with optional interrupt generation.
- Up to 3 (CT16B0, CT16B1, CT16B5) PWM outputs corresponding to match registers with the following capabilities:
  - Set LOW on match.
  - Set HIGH on match.
  - Toggle on match.
  - Do nothing on match.
- For CT16B0, up to 4 complete inverse waveform of the PWM signals, and builds in programmable dead-band function.
- For CT16B0, the PWM can be stopped and the PWM duty can be decreased by comparator output.
- Supply DMA transfer

11.3 PIN DESCRIPTION

Pin Name	Type	Description	GPIO Configuration

CT16Bn_CAP0	I	Capture channel input 0	Depends on GPIOn_CFG
CT16Bn_PWMx	O	Output channel x of Match/PWM output.	
CT16Bn_PWMxN	O	Inverse Output channel of Match/PWMx output.	
CT16Bn_BRK	I	Break signal input	Depends on GPIOn_CFG

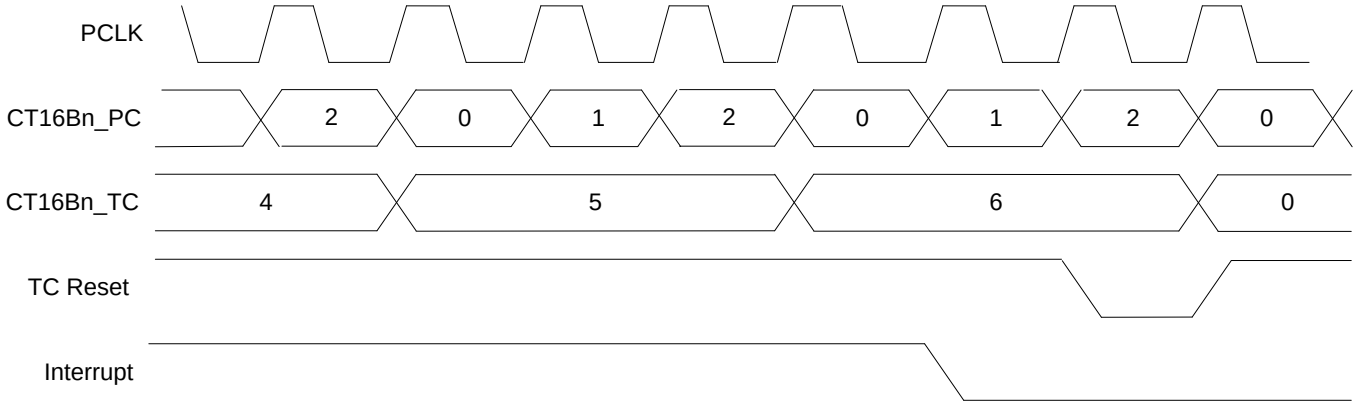
11.4 BLOCK DIAGRAM



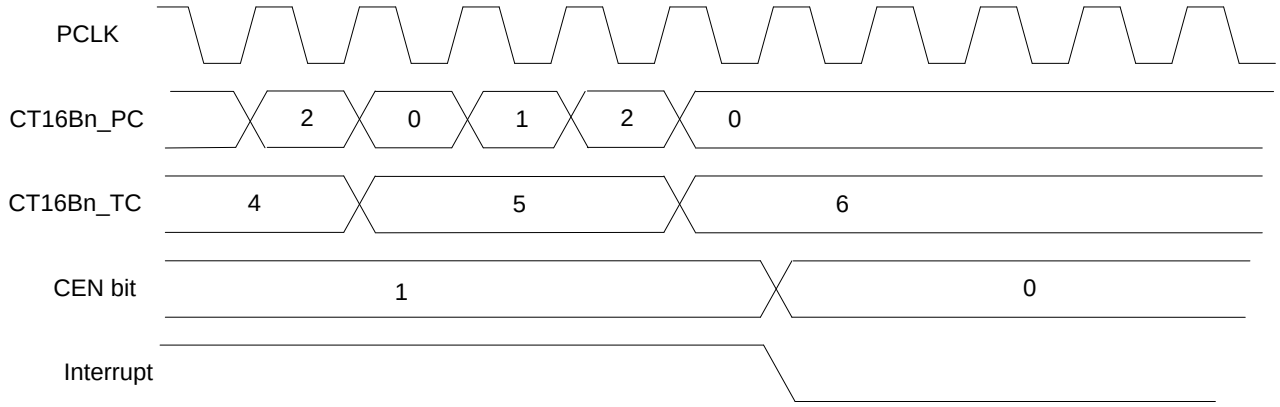
11.5 TIMER OPERATION

11.5.1 Edge-aligned Up-counting Mode

The following figure shows a timer configured to reset the count (TC) and generate an interrupt on match in Edge-aligned up-counting mode. The CT16Bn_PRE register is set to 2, and the CT16Bn_MRx register is set to 6. At the end of the timer cycle where the match occurs, the timer count is reset. This gives a full length cycle to the match value. The interrupt indicating that a match occurred is generated in the next clock after the timer reached the match value.



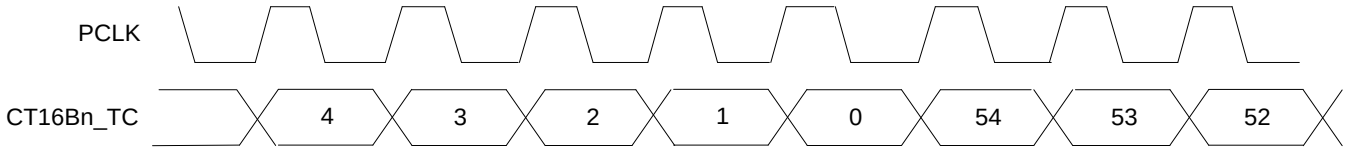
The following figure shows a timer configured to stop and generate an interrupt on match in Edge-aligned up-counting mode. The CT16Bn_PRE register is set to 2, and the CT16Bn_MRx register is set to 6. In the next clock after the timer reaches the match value, the CEN bit in CT16Bn_TMRCTRL register is cleared, and the interrupt indicating that a match occurred is generated.



11.5.2 Edge-aligned Down-counting Mode

The timer count TC[15:0] will be reset to the value of CT16Bn_MR9 after resetting counter or TC reaches 0. Besides, TC is blocked while the value of CT16Bn_MR9 is zero.

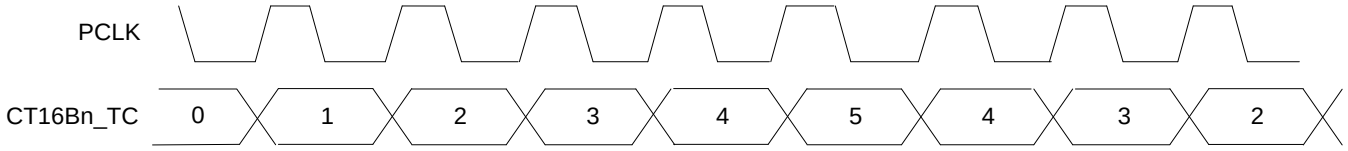
The following figure shows a timer configured to reset the count in Edge-aligned down-counting mode. The CT16Bn_PRE register is set to 0, and the CT16Bn_MR9 register is set to 54. After TC reaches 0, the timer count is reset and loaded from the value of CT16Bn_MR9.



11.5.3 Center-aligned Counting Mode

In Center-aligned counting mode, TC counts up from 0 to the value of CT16Bn_MR9, and then counts down to 0 alternatively. Besides, TC is blocked while the value of CT16Bn_MR9 is zero.

The following figure shows a timer in Center-aligned counting mode. The CT16Bn_PRE register is set to 0, and the CT16Bn_MR9 register is set to 5.



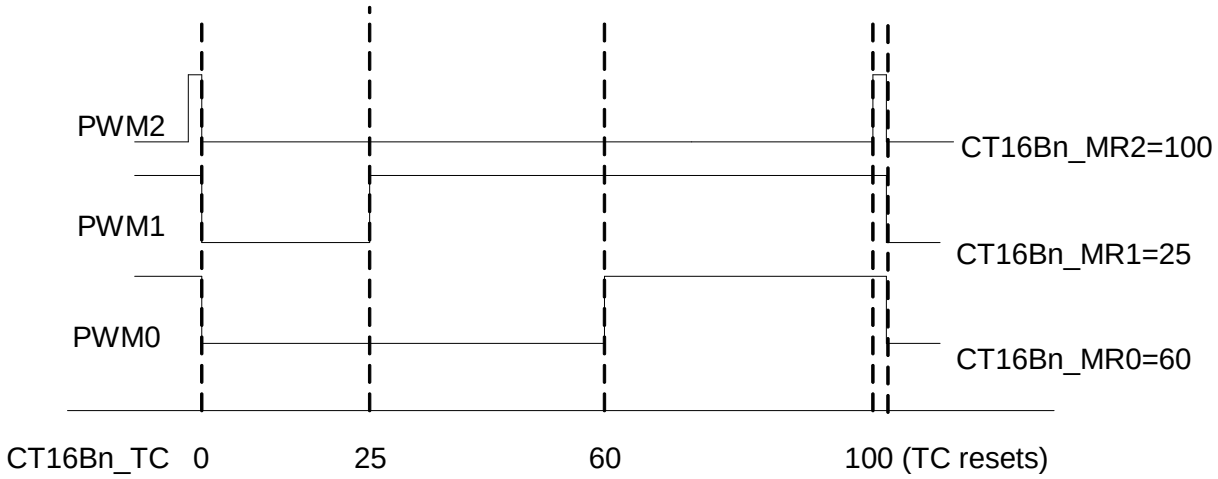
11.6 PWM

11.6.1 PWM Mode 1

- PWMn is 0 when TC<MRn during Up-counting period
- PWMn is 0 when TC≤MRn during Down-counting period

Take Edge-aligned up-counting Mode as example,

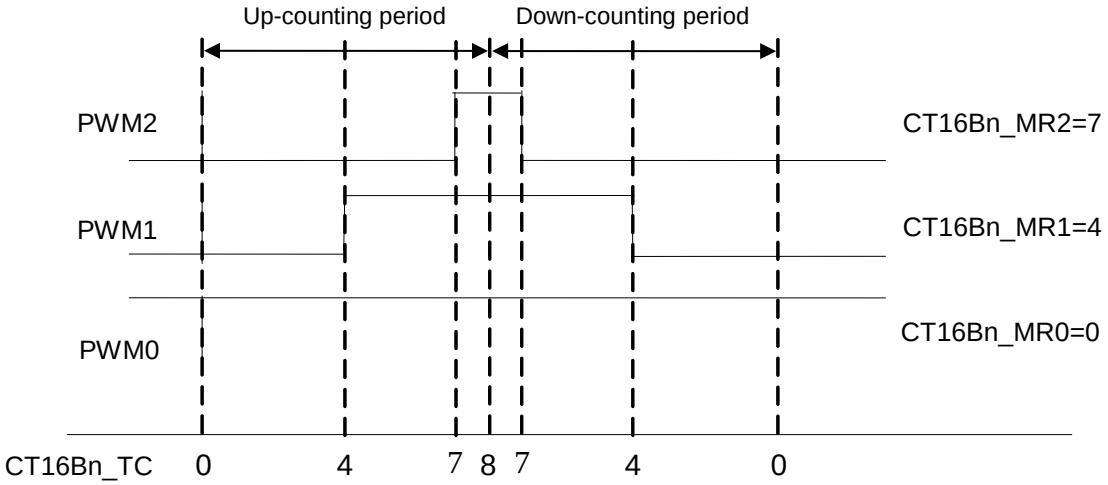
0. All single edge controlled PWM outputs go LOW at the beginning of each PWM cycle (timer is set to zero) unless their match value in CT16Bn_MR0~3 registers is equal to zero.
1. Each PWM output will go HIGH when its match value is reached. If no match occurs, the PWM output remains continuously LOW.
2. If a match value larger than the PWM cycle length is written to the CT16Bn_MR0~3 registers, and the PWM signal is HIGH already, then the PWM signal will be cleared on the next start of the next PWM cycle.
3. If a match register contains the same value as the timer reset value (the PWM cycle length), then the PWM output will be reset to LOW on the next clock tick. Therefore, the PWM output will always consist of a one clock tick wide positive pulse with a period determined by the PWM cycle length.
4. If a match register is set to zero, then the PWM output will go HIGH the first time the timer goes back to zero and will stay HIGH continuously.



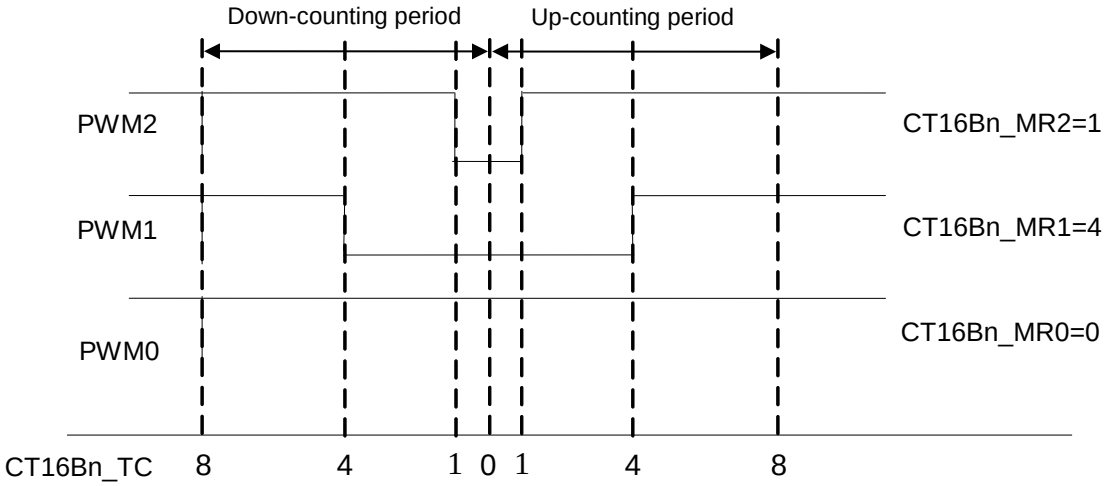
✸ ***Note: When the match outputs are selected to perform as PWM outputs, the timer reset (MRnRST) and timer stop (MRnSTOP) bits in CT16Bn_MCTRL register must be set to zero except for the match register setting the PWM cycle length. For this register, set the MRnR bit to one to enable the timer reset when the timer value matches the value of the corresponding match register.***

The following figure shows the PWM mode 1 wave form in Center-aligned counting mode.

Case1: The CT16Bn_PRE register is set to 0, the CT16Bn_MR9 register is set to 8, the CT16Bn_MR2 register is set to 7, the CT16Bn_MR1 register is set to 4, and the CT16Bn_MR0 register is set to 0.



Case 2: The CT16Bn_PRE register is set to 0, the CT16Bn_MR9 register is set to 8, the CT16Bn_MR2 register is set to 1, the CT16Bn_MR1 register is set to 4, and the CT16Bn_MR0 register is set to 0.



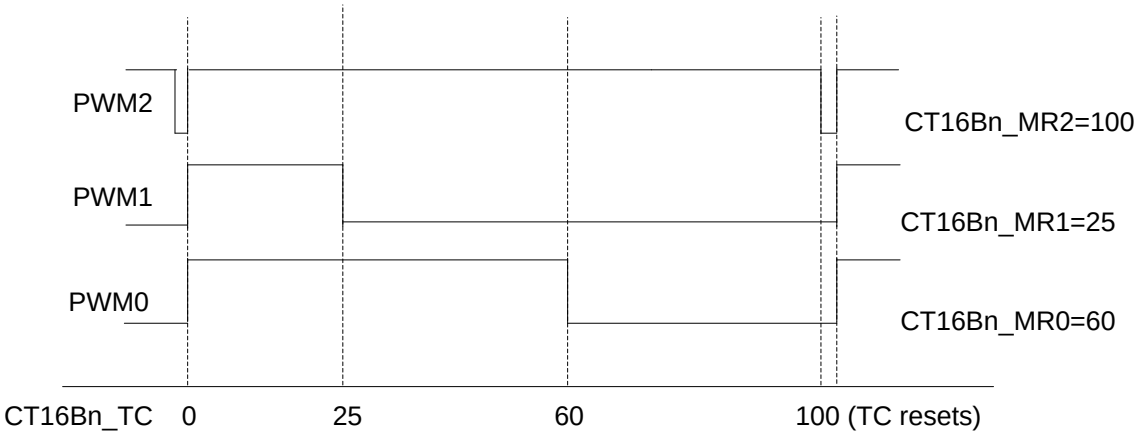
11.6.2 PWM Mode 2

- PWMn is 1 when TC<MRn during Up-counting period
- PWMn is 1 when TC≤MRn during Down-counting period
- Not support in Center-aligned counting mode

Take Edge-aligned up-counting Mode as example,

1. All single edge controlled PWM outputs go HIGH at the beginning of each PWM cycle (timer is set to zero) unless their match value in CT16Bn_MR0~3 registers is equal to zero.
2. Each PWM output will go LOW when its match value is reached. If no match occurs, the PWM output remains continuously HIGH.
3. If a match value larger than the PWM cycle length is written to the CT16Bn_MR0~3 registers, and the PWM signal is LOW already, then the PWM signal will go HIGH on the next start of the next PWM cycle.
4. If a match register contains the same value as the timer reset value (the PWM cycle length), then the PWM output will be reset to HIGH on the next clock tick. Therefore, the PWM output will always consist of a one clock tick wide low pulse with a period determined by the PWM cycle length.

5. If a match register is set to zero, then the PWM output will go LOW the first time the timer goes back to zero and will stay LOW continuously.



✸ ***Note: When the match outputs are selected to perform as PWM outputs, the timer reset (MRnRST) and timer stop (MRnSTOP) bits in CT16Bn_MCTRL register must be set to zero except for the match register setting the PWM cycle length. For this register, set the MRnR bit to one to enable the timer reset when the timer value matches the value of the corresponding match register.***

11.7 INVERSE PWM OUTPUT WITH DEAD-BAND PERIOD

The CT16B0_PWMm builds in inverse output function controlled by PWMmNIOEN[1:0] bits in CT16Bn_PWMCTRL register.

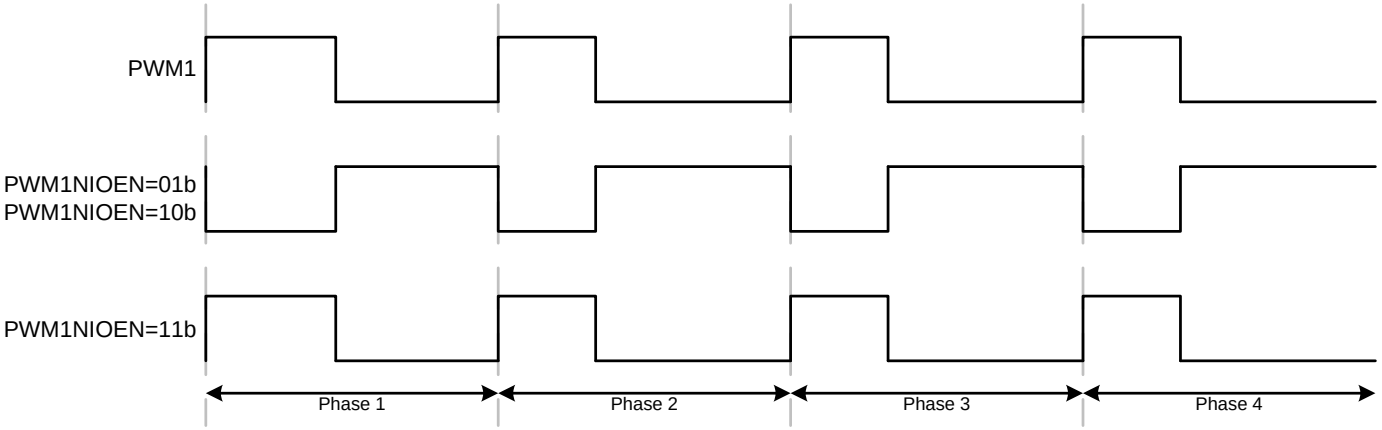
- When PWMmNIOEN[1:0] = 00b, PWMmN pin is GPIO mode.
- When PWMmNIOEN[1:0] = 01b, PWMmN pin changes to inverse PWM output pin which outputs the inverse PWM signal of PWMm with dead-band period, but same High signal during dead-band period.
- When PWMmNIOEN[1:0] = 10b, PWMmN pin changes to inverse PWM output pin which outputs the inverse PWMm signal with dead-band period, but same Low signal during dead-band period.
- When PWMmNIOEN[1:0] = 11b, PWMmN pin changes to non-inverse PWM output pin which outputs the same PWMm signal with dead-band period.

The dead-band period is symmetrical at left-right terminal of PWM high pulse width, and the PWM dead-band period is controlled by CT16Bn_PWMmNDB register. This register is only usable when PWMmNIOEN[1:0] ≠ 00b, and the PWM dead-band function is disabled when the value of this register is 0.

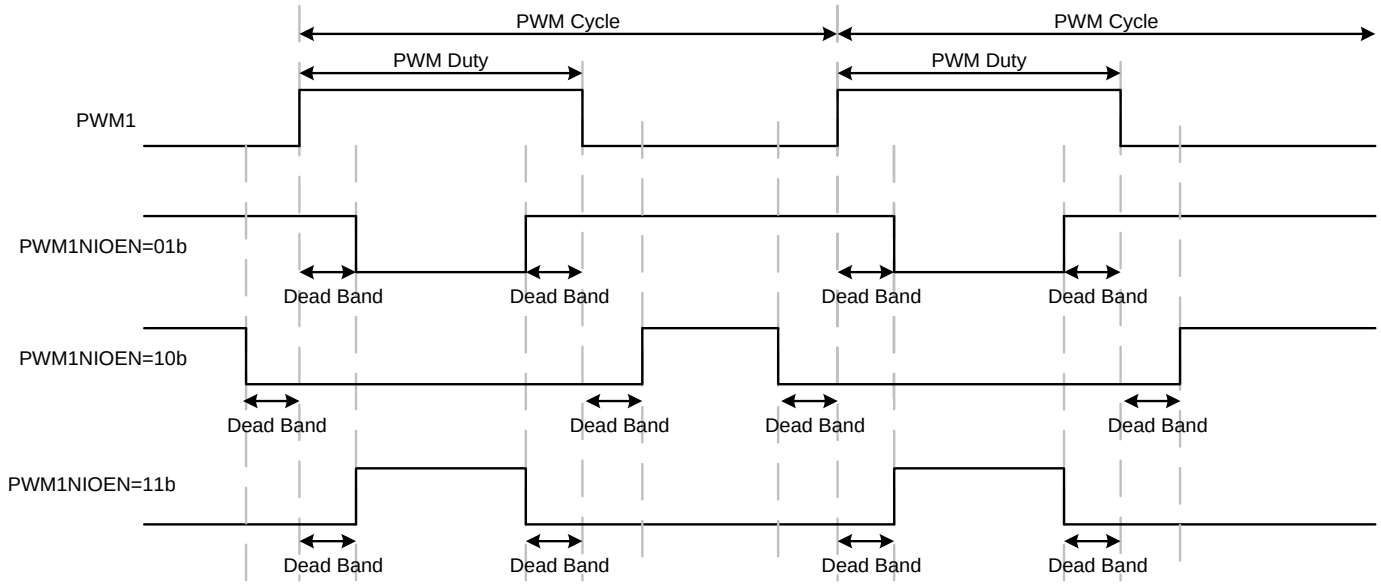
If DB = 1, the dead-band period is set as 1*CT16Bn_PCLK*(PR+1), 1*CT16Bn_PCLK*(PR+1) dead-band is in the left side of PWM high pulse, and the other side also includes one dead-band duration, so the total dead-band period is 2*CT16Bn_PCLK*(PR+1).

To take care the PWM high pulse width with dead-bane function is necessary. Recommend the dead-band period less than PWM high pulse width, or the PWM high pulse width disappears.

The PWMmN waveform without dead-band:



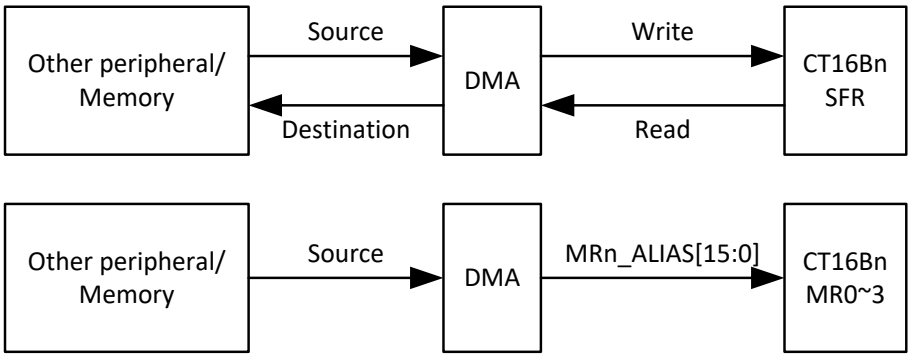
The PWMmN waveform with dead-band:



- ***Note: If the bead-band period is longer than PWM duty, the PWMmN is no output.***
- ***Note: When the dead-band function is enabled in Center-aligned mode, and MR9RST=1, CT16Bn_PWMxN will always output "0"***
- ***Note: When the dead-band function is enabled***
  - ***System will reset TC refer to MR9RST ONLY in Up-counting mode***
  - ***In Down counting mode, TC[15:0] will be reloaded from CT16Bn_MR9 after resetting counter***
  - ***System will reset TC refer to MR9RST ONLY in Center-aligned mode***

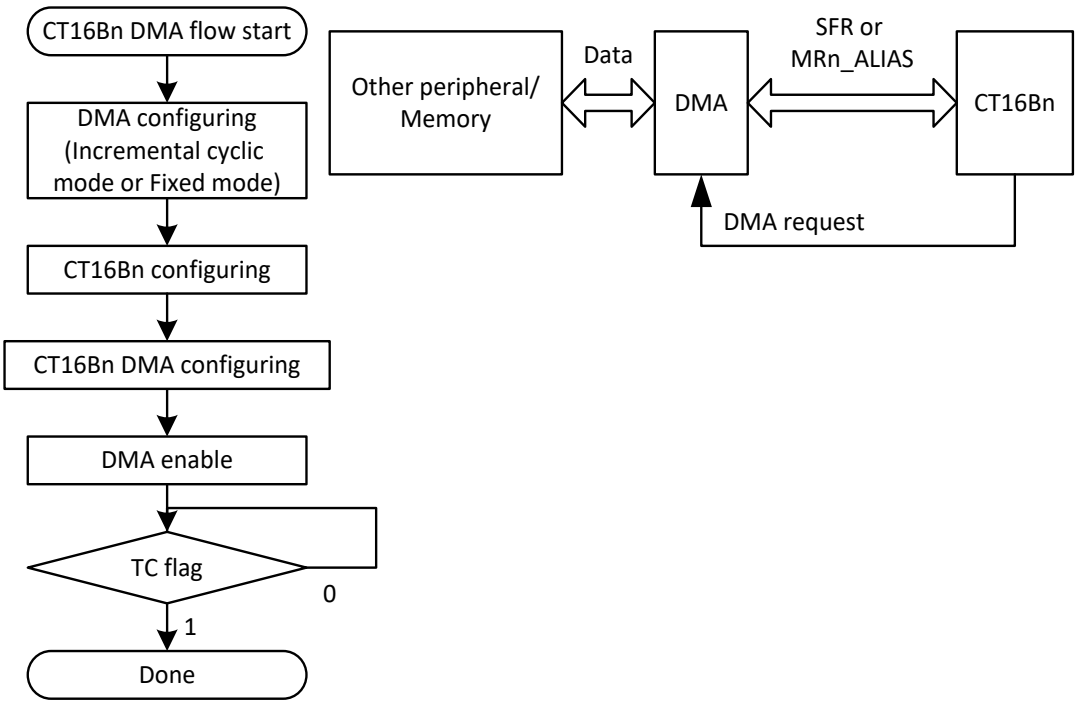
11.8 DMA MODE

The DMA mode is to use DMA engine to move data in CT16Bn. Before the DMA transfer start, DMA engine must be set up first. In CT16Bn DMA write Mode, DMA gets data from other peripherals or memories to CT16Bn register. In CT16Bn DMA read Mode, DMA receive data from CT16Bn register and send to other peripherals or memories.



11.8.1 Flow Chart

The following register programming flow is used for CT16Bn DMA mode. In this flow FW check DMA channel transfer finish then flow can go to the end. This example will turn on DMA Channel to move data from other peripherals or memory to CT16Bn register or MRn_ALIAS. When the CT16Bn IRQ is issued, the DMA request is issued.



11.8.2 DMA Configuration Recommendations

Read single register (e.g., CAP0[15:0])

DMAC setting for CT16Bn data buffer (16-bit)			
Cn_CSR SRC_WIDTH	Cn_CSR SRC_SIZE	Cn_SIZE TOT_SIZE	Cn_CSR DST_WIDTH
0x1 (half-word)	0x0 (burst=1)	N (half-word)	0x1 (half-word)

Write single register (e.g., MR9[15:0])

DMAC setting for CT16Bn data buffer (16-bit)			
Cn_CSR SRC_WIDTH	Cn_CSR SRC_SIZE	Cn_SIZE TOT_SIZE	Cn_CSR DST_WIDTH
0x1 (half-word)	0x0 (burst=1)	N (half-word)	0x1 (half-word)

Write MRn_ALIAS[15:0] (CT16B0/1/5 MR0~3)

DMAC setting for CT16Bn MRn_ALIAS[15:0]			
Cn_CSR SRC_WIDTH	Cn_CSR SRC_SIZE	Cn_SIZE TOT_SIZE	Cn_CSR DST_WIDTH
0x1 (half-word)	0x1 (burst=4)	N (half-word)	0x1 (half-word)
0x0 (byte)	0x2 (burst=8)	N (byte)	0x1 (half-word)

DMA update MR flow for CT16B0/CT16B1/CT16B5

1. Assign a DMA channel
2. Set SrcAddr, DstAddr = CT16Bn_BASE + 0xD4
3. Set TOT_SIZE
4. Set DST_RS, DST_HE=1
5. Set SRC_WIDTH=001, DST_WIDTH=001
6. Set SRC_SIZE=001 (burst 4)
7. Set SRCAD_CTL=00, DSTCAD_CTL=11
8. Set CH_EN to Enable channel
9. Set DMA mode register (CT16n_DMA)

11.9 CT16B0 BREAK FUNCTION

Besides normal PWM, CT16B0 build in a special function for motor application (break PWM pulse generator output signal, overcurrent protection). The special function is to trigger PWM pulse generator breaking output when break condition occurs. The break condition can is selected by BRKSEL bits to select comparator output, break pin (BRK) and noise detect.

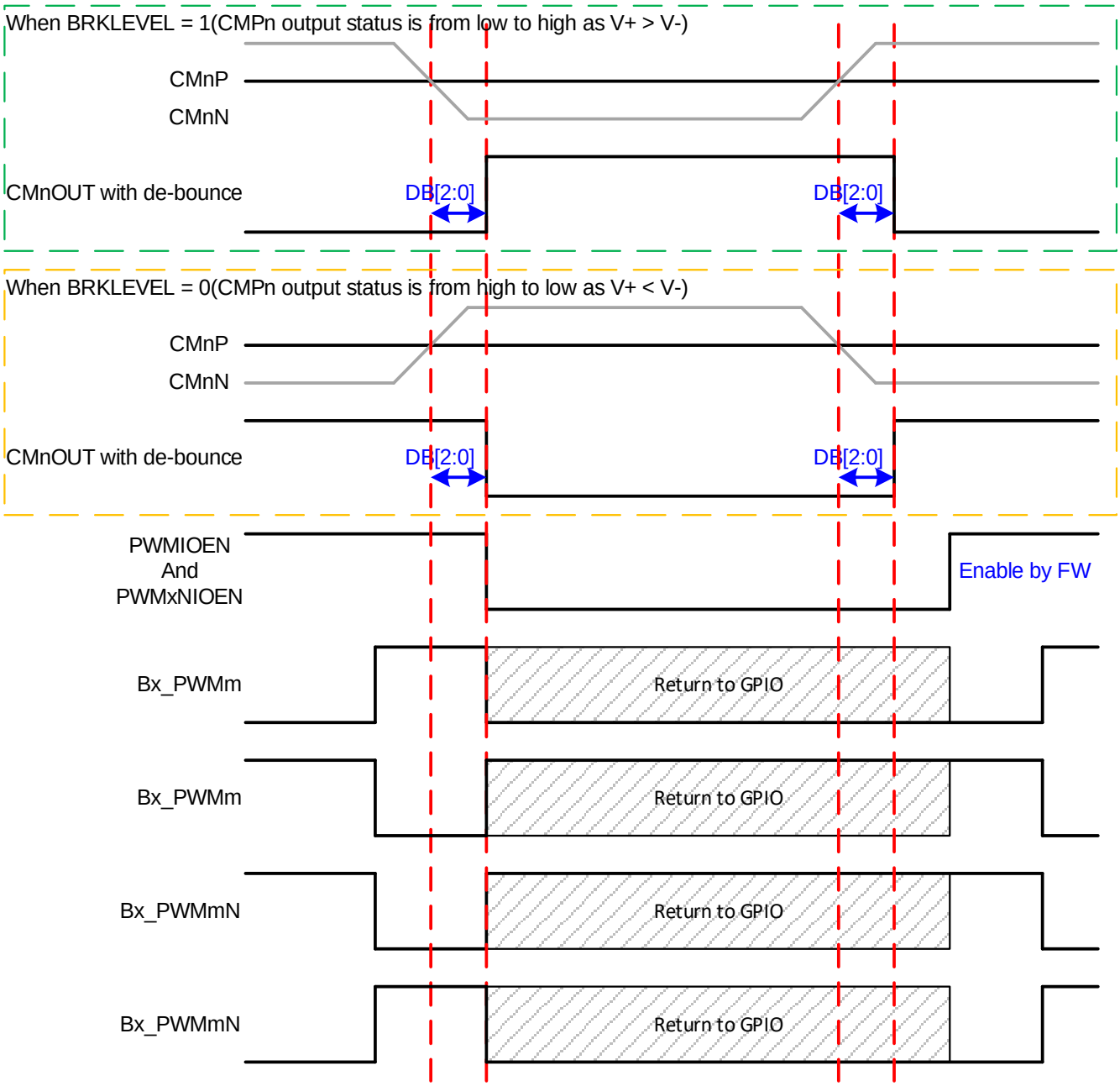
When a break condition occurs, PWM channels will be disabled to switch to GPIO mode. Before the break function is enabled, GPIO register of the PWM channel pins must be set.

The break pin signal is through a de-bounce circuit to filter comparator transient status. The de-bounce time is controlled by BRKDB[2:0] bits.

When break condition occurs, the BRKIF is set, and makes the interrupt service executed when BRKIE bit is set. Besides, PWM0IOEN~PWM3IOEN, and PWM0NIOEN~PWM3NIOEN will become 00b (GPIO mode).

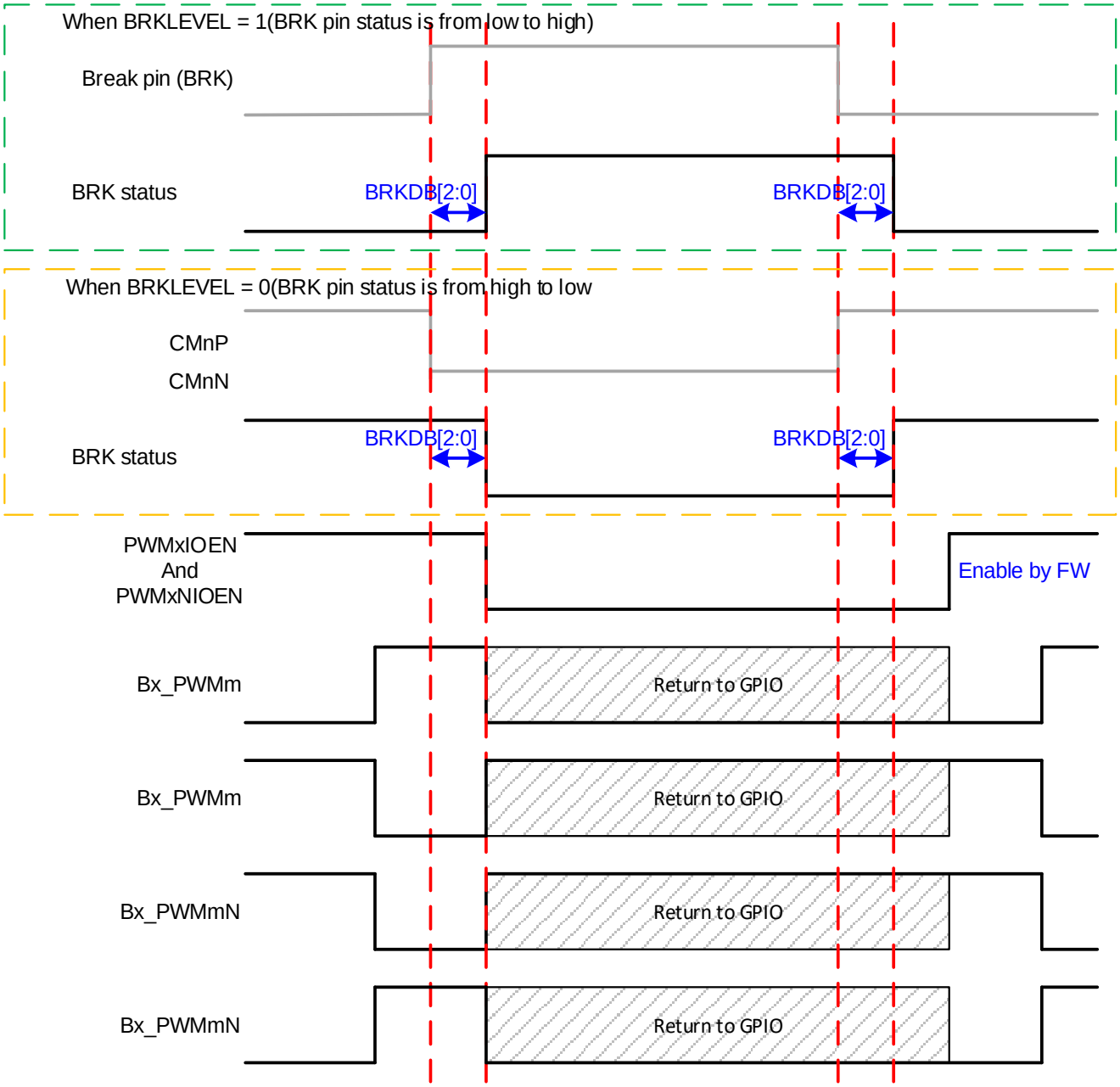
11.9.1 Break by Comparator n Output

If break source is comparator output, the break level can be selected as V+>V- or V->V+ by BRKLEVEL bit. The transient status of the comparator output signal can be filtered through a de-bounce circuit, the de-bounce time is controlled by CMnDB[2:0] bits.



11.9.2 Break by Break Pin

If break source is CT16Bx_BRK pin, the break level can be selected as low or high by BRKLEVEL bit.


When BRKLEVEL = 1(BRK pin status is from low to high)
Break pin (BRK)
BRK status
BRKDB[2:0]
BRKDB[2:0]
When BRKLEVEL = 0(BRK pin status is from high to low
CMnP
CMnN
BRK status
BRKDB[2:0]
BRKDB[2:0]
PWMxIOEN
And
PWMxNIOEN
Enable by FW
Bx_PWMm
Return to GPIO
Bx_PWMm
Return to GPIO
Bx_PWMmN
Return to GPIO
Bx_PWMmN
Return to GPIO

11.10 CT16Bn REGISTERS

Base Address: 0x4000 0000 (CT16B0)
0x4000 2000 (CT16B1)
0x4000 6000 (CT16B5)

✸ ***Note: The register below shall be set before the CEN bit is be set.***
CM[2:0], CLKSEL, TC[15:0], PC[7:0], CTM[1:0], PWMnMODE[1:0], DB[9:0], LOADCTRL, BRKCTRL, MCTRL (MRnIE/MRnRST/MRnSTOP), MR9[15:0].

11.10.1 CT16Bn Timer Control register (CT16Bn_TMRCTRL) (n=0,1,5)

Address Offset: 0x00

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:4	CM[2:0]	Counting mode selection 000: Edge-aligned Up-counting mode 001: Edge-aligned Down-counting mode 010: Center-aligned mode 1. The match interrupt flag is set during the down-counting period 100: Center-aligned mode 2. The match interrupt flag is set during the up-counting period 110: Center-aligned mode 3. The match interrupt flag is set during both up-counting and down-counting period Other: Reserved	R/W	000b
3:2	Reserved		R	0
1	CRST	Counter Reset. 0: Disable counter reset. 1: Timer Counter and the Prescale Counter are synchronously reset on the next positive edge of PCLK. This is cleared by HW when the counter reset operation finishes.	R/W	0
0	CEN	Counter Enable 0: Disable Counter. 1: Enable Timer Counter and Prescale Counter for counting.	R/W	0

11.10.2 CT16Bn Timer Counter register (CT16Bn_TC) (n=0,1,5)

Address Offset: 0x04

In Edge-aligned up-counting mode (CM[2:0]=000b), unless it is reset before reaching its upper limit, the TC will count up to the value 0x0000FFFF and then wrap back to the value 0x00000000. This event does not cause an interrupt, but a Match register can be used to detect an overflow if needed.

In Edge-aligned down-counting mode (CM[2:0]=001b) , the TC[15:0] should be reset to the value of CT16Bn_MR9 after resetting counter (SW set CRST to 1).

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	TC[15:0]	Timer Counter	R/W	0

11.10.3 CT16Bn Prescale register (CT16Bn_PRE) (n=0,1,5)

Address Offset: 0x08

✸ ***Note: If Counter mode is selected in the CNTCTRL register, PRE[7:0] must be set to 0x0.***

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PR[7:0]	Prescale max value.	R/W	0

11.10.4CT16Bn Prescale Counter register (CT16Bn_PC) (n=0,1,5)

Address Offset: 0x0C

The 8-bit Prescale Counter controls division of PCLK by some constant value before it is applied to the Timer Counter. This allows control of the relationship between the resolution of the timer and the maximum time before the timer overflows. The Prescale Counter is incremented on every PCLK. When it reaches the value stored in the Prescale Register, the Timer Counter is incremented, and the Prescale Counter is reset on the next PCLK. This causes the TC to increment on every PCLK when PR = 0, every 2 PCLKs when PR = 1, etc.

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PC[7:0]	Prescale Counter	R/W	0

11.10.5CT16Bn Count Control register (CT16Bn_CNTCTRL) (n=0,1,5)

Address Offset: 0x10

This register is used to select between Timer and Counter mode, and in Counter mode to select the pin and edges for counting.

When Counter Mode is chosen as a mode of operation, the CAP input (selected by the CIS bits) is sampled on every rising edge of the PCLK clock. After comparing two consecutive samples of this CAP input, one of the following four events is recognized: rising edge, falling edge, either of edges or no changes in the level of the selected CAP input. Only if the identified event occurs, and the event corresponds to the one selected by CTM bits in this register, will the Timer Counter register be incremented.

Effective processing of the externally supplied clock to the counter has some limitations. Since two successive rising edges of the PCLK clock are used to identify only one edge on the CAP selected input, the frequency of the CAP input cannot exceed one half of the PCLK clock. Consequently, the duration of the HIGH/LOW levels on the same CAP input in this case cannot be shorter than 1/ (2 x PCLK).

✸ ***Note: If Counter mode is selected in the CNTCTRL register, bit 2~0 of Capture Control (CAPCTRL) register and bit 7 to 0 of Prescale (PRE) register must be set to 0x0.***

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1:0	CTM[1:0]	Counter/Timer Mode. This field selects which rising PCLK edges can increment Timer's Prescale Counter (PC), or clear PC and increment Timer Counter (TC). 00: Timer Mode: every rising PCLK edge 01: Counter Mode: TC is incremented on rising edges on the CAP0 input selected by CIS bits. 10: Counter Mode: TC is incremented on falling edges on the CAP0 input selected by CIS bits. 11: Counter Mode: TC is incremented on both edges on the CAP0 input selected by CIS bits.	R/W	0

11.10.6 CT16Bn Match Control register (CT16Bn_MCTRL) (n=0,1,5)

Address Offset: 0x14

- ***Note: When the dead-band function is enabled in Center-aligned mode, and MR9RST=1, CT16Bn_PWMxN will always output "0"***
- ***Note: When the dead-band function is enabled***
  - ***System will reset TC refer to MR9RST ONLY in Up-counting mode***
  - ***In Down counting mode, TC[15:0] will be reloaded from CT16Bn_MR9 after resetting counter***
  - ***System will reset TC refer to MR9RST ONLY in Center-aligned mode***

Bit	Name	Description	Attribute	Reset
31	MR9STOP	TC and PC will stop and CEN bit will be cleared if MR9 matches TC. 0: Disable 1: Enable	R/W	0
30	MR9RST	Enable reset TC when MR9 matches TC. 0: Disable 1: Enable	R/W	0
29	MR9IE	Enable generating an interrupt based on CM[2:0] when MR9 matches the value in the TC. 0: Disable 1: Enable	R/W	0
28:12	Reserved		R	0
11	MR3STOP	TC and PC will stop and CEN bit will be cleared if MR3 matches TC. 0: Disable 1: Enable	R/W	0
10	MR3RST	Enable reset TC when MR3 matches TC. 0: Disable 1: Enable	R/W	0
9	MR3IE	Enable generating an interrupt based on CM[2:0] when MR3 matches the value in the TC. 0: Disable 1: Enable	R/W	0
8	MR2STOP	TC and PC will stop and CEN bit will be cleared if MR2 matches TC. 0: Disable 1: Enable	R/W	0
7	MR2RST	Enable reset TC when MR2 matches TC. 0: Disable 1: Enable	R/W	0
6	MR2IE	Enable generating an interrupt based on CM[2:0] when MR2 matches the value in the TC. 0: Disable 1: Enable	R/W	0
5	MR1STOP	TC and PC will stop and CEN bit will be cleared if MR1 matches TC. 0: Disable 1: Enable	R/W	0
4	MR1RST	Enable reset TC when MR1 matches TC. 0: Disable 1: Enable	R/W	0
3	MR1IE	Enable generating an interrupt based on CM[2:0] when MR1 matches the value in the TC. 0: Disable 1: Enable	R/W	0
2	MR0STOP	TC and PC will stop and CEN bit will be cleared if MR0 matches TC. 0: Disable 1: Enable	R/W	0
1	MR0RST	Enable reset TC when MR0 matches TC. 0: Disable 1: Enable	R/W	0
0	MR0IE	Enable generating an interrupt based on CM[2:0] when MR0 matches the value in the TC. 0: Disable 1: Enable	R/W	0

11.10.7CT16Bn Match register 0~3 (CT16Bn_MR0~3) (n=0,1,5)

Address Offset: 0x18, 0x1C, 0x20, 0x24

The Match register values are continuously compared to the Timer Counter (TC) value. When the two values are equal, actions can be triggered automatically. The action possibilities are to generate an interrupt, reset the Timer Counter, or stop the timer. Actions are controlled by the settings in the CT16Bn_MCTRL register.

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	MR[15:0]	Timer counter match value	R/W	0

11.10.8 CT16Bn Capture Control register (CT16Bn_CAPCTRL) (n=0,1,5)

Address Offset: 0x28

The Capture Control register is used to control whether the Capture register is loaded with the value in the Counter/timer when the capture event occurs, and whether an interrupt is generated by the capture event. Setting both the rising and falling bits at the same time is a valid configuration, resulting in a capture event for both edges.

✷ ***Note: HW will switch I/O Configuration directly when CAP0EN=1.***

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CAP0EN	Capture 0 function enable bit 0: Disable 1: Enable.	R/W	0
2	CAP0IE	Interrupt on CT16Bn_CAP0 event: a CAP0 load due to a CT16Bn_CAP0 event will generate an interrupt. 0: Disable 1: Enable	R/W	0
1	CAP0FE	Capture on CT16Bn_CAP0 falling edge: a sequence of 1 then 0 on CT16Bn_CAP0 will cause CAP0 to be loaded with the contents of TC. 0: Disable 1: Enable	R/W	0
0	CAP0RE	Capture on CT16Bn_CAP0 rising edge: a sequence of 0 then 1 on CT16Bn_CAP0 will cause CAP0 to be loaded with the contents of TC. 0: Disable 1: Enable	R/W	0

11.10.9 CT16Bn Capture 0 register (CT16Bn_CAP0) (n=0,1,5)

Address Offset: 0x2C

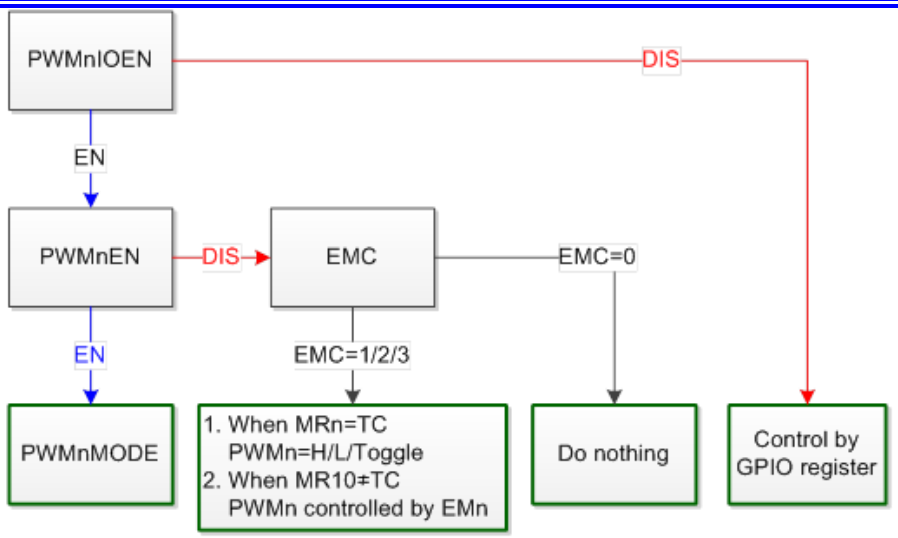
Each Capture register is associated with a device pin and may be loaded with the counter/timer value when a specified event occurs on that pin. The settings in the Capture Control register determine whether the capture function is enabled, and whether a capture event happens on the rising edge of the associated pin, the falling edge, or on both edges.

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	CAP0[15:0]	Timer counter capture value	R	0

11.10.10 CT16Bn External Match register (CT16Bn_EM) (n=0,1,5)

Address Offset: 0x30

The External Match register provides both control and status of CT16B0_PWM[2:0]. If the match outputs are configured as PWM output, the function of the external match registers is determined by the PWM rules.



✸ ***Note: Set EMCx[1:0] and then set EMx[1:0] if the external match function is used.***

Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11:10	EMC3[1:0]	Determines the functionality of CT16Bn_PWM3 00: Do Nothing. 01: CT16Bn_PWM3 pin is LOW 10: CT16Bn_PWM3 pin is HIGH 11: Toggle CT16Bn_PWM3 pin	R/W	0
9:8	EMC2[1:0]	Determines the functionality of CT16Bn_PWM2. 00: Do Nothing. 01: CT16Bn_PWM2 pin is LOW 10: CT16Bn_PWM2 pin is HIGH 11: Toggle CT16Bn_PWM2 pin	R/W	0
7:6	EMC1[1:0]	Determines the functionality of CT16Bn_PWM1. 00: Do Nothing. 01: CT16Bn_PWM1 pin is LOW 10: CT16Bn_PWM1 pin is HIGH. 11: Toggle CT16Bn_PWM1	R/W	0
5:4	EMC0[1:0]	Determines the functionality of CT16Bn_PWM0. 00: Do Nothing. 01: CT16Bn_PWM0 pin is LOW 10: CT16Bn_PWM0 pin is HIGH 11: Toggle CT16Bn_PWM0	R/W	0
3	EM3	When EMC3≠00b and MR3≠TC, this bit will drive the state of CT16Bn_PWM3 output.	R/W	0
2	EM2	When EMC2≠00b and MR2≠TC, this bit will drive the state of CT16Bn_PWM2 output.	R/W	0
1	EM1	When EMC1≠00b and MR1≠TC, this bit will drive the state of CT16Bn_PWM1 output.	R/W	0
0	EM0	When EMC0≠00b and MR0≠TC, this bit will drive the state of CT16Bn_PWM0 output.	R/W	0

11.10.11 CT16Bn PWM Control register (CT16Bn_PWMCTRL) (n=0)

Address Offset: 0x34

The PWM Control register is used to configure the match outputs as PWM outputs. Each match output can be in-dependently set to perform either as PWM output or as match output whose function is controlled by CT16Bn_EM register.

For CT16B0, a maximum of 4 single edge controlled PWM outputs can be selected via the CT16Bn_PWMCTRL[3:0] bits. One additional match register determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output

is set to HIGH. The timer is reset by the match register that is configured to set the PWM cycle length. When the timer is reset to zero, all currently HIGH match outputs configured as PWM outputs are cleared.

The Capture Control register is used to control whether the Capture register is loaded with the value in the Counter/timer when the capture event occurs, and whether an interrupt is generated by the capture event. Setting both the rising and falling bits at the same time is a valid configuration, resulting in a capture event for both edges.

- ***Note: The value of MRx shall be between the value of DB and MR9 if PWMnNIOEN is 2.***
- ***Note: PWMnMODE can be 2 (Forced to 0) or 3 (Forced to 1) only when PWMnNIOEN is 0.***

Bit	Name	Description	Attribute	Reset
31:30	PWM3NIOEN[1:0]	CT16Bn_PWM3N/GPIO selection bit 00: CT16Bn_PWM3N pin act as GPIO 01: CT16Bn_PWM3N pin outputs the inverse signal with dead-band of CT16Bn_PWM3, but same High signal during dead-band period. 10: CT16Bn_PWM3N pin outputs the inverse signal with dead-band of CT16Bn_PWM3, but same Low signal during dead-band period. 11: CT16Bn_PWM3N pin outputs the same signal with dead-band of CT16Bn_PWM3.	R/W	0
29:28	PWM2NIOEN[1:0]	CT16Bn_PWM2N/GPIO selection bit 00: CT16Bn_PWM2N pin act as GPIO 01: CT16Bn_PWM2N pin outputs the inverse signal with dead-band of CT16Bn_PWM2, but same High signal during dead-band period. 10: CT16Bn_PWM2N pin outputs the inverse signal with dead-band of CT16Bn_PWM2, but same Low signal during dead-band period. 11: CT16Bn_PWM2N pin outputs the same signal with dead-band of CT16Bn_PWM2.	R/W	0
27:26	PWM1NIOEN[1:0]	CT16Bn_PWM1N/GPIO selection bit 00: CT16Bn_PWM1N pin act as GPIO 01: CT16Bn_PWM1N pin outputs the inverse signal with dead-band of CT16Bn_PWM1, but same High signal during dead-band period. 10: CT16Bn_PWM1N pin outputs the inverse signal with dead-band of CT16Bn_PWM1, but same Low signal during dead-band period. 11: CT16Bn_PWM1N pin outputs the same signal with dead-band of CT16Bn_PWM1.	R/W	0
25:24	PWM0NIOEN[1:0]	CT16Bn_PWM0N/GPIO selection bit 00: CT16Bn_PWM0N pin act as GPIO 01: CT16Bn_PWM0N pin outputs the inverse signal with dead-band of CT16Bn_PWM0, but same High signal during dead-band period. 10: CT16Bn_PWM0N pin outputs the inverse signal with dead-band of CT16Bn_PWM0, but same Low signal during dead-band period. 11: CT16Bn_PWM0N pin outputs the same signal with dead-band of CT16Bn_PWM0.	R/W	0
23	PWM3IOEN	CT16Bn_PWM3/GPIO selection bit 0: CT16Bn_PWM3 pin act as GPIO 1: CT16Bn_PWM3 pin act as match output, and output signal depends on PWM3EN bit.	R/W	0
22	PWM2IOEN	CT16Bn_PWM2/GPIO selection bit 0: CT16Bn_PWM2 pin act as GPIO 1: CT16Bn_PWM2 pin act as match output, and output signal depends on PWM2EN bit.	R/W	0
21	PWM1IOEN	CT16Bn_PWM1/GPIO selection bit 0: CT16Bn_PWM1 pin act as GPIO 1: CT16Bn_PWM1 pin act as match output, and output signal depends on PWM1EN bit.	R/W	0
20	PWM0IOEN	CT16Bn_PWM0/GPIO selection bit 0: CT16Bn_PWM0 pin act as GPIO 1: CT16Bn_PWM0 pin act as match output, and output signal depends on PWM0EN bit.	R/W	0
19:12	Reserved		R	0
11:10	PWM3MODE[1:0]	PWM3 output mode 00: PWM mode 1 PWM3 is 0 when TC<MR3 during Up-counting period PWM3 is 0 when TC≤MR3 during Down-counting period 01: PWM mode 2	R/W	0

		PWM3 is 1 when TC<MR3 during Up-counting period PWM3 is 1 when TC≤MR3 during Down-counting period 10: PWM3 is forced to 0. 11: PWM3 is forced to 1.		
9:8	PWM2MODE[1:0]	PWM2 output mode 00: PWM mode 1 01: PWM mode 2 10: Forced to 0. 11: Forced to 1.	R/W	0
7:6	PWM1MODE[1:0]	PWM1 output mode 00: PWM mode 1 01: PWM mode 2 10: Forced to 0. 11: Forced to 1.	R/W	0
5:4	PWM0MODE[1:0]	PWM0 output mode 00: PWM mode 1 01: PWM mode 2 10: Forced to 0. 11: Forced to 1.	R/W	0
3	PWM3EN	PWM3 enable 0: CT16Bn_PWM2 is controlled by EMC3. 1: PWM mode is enabled for CT16Bn_PWM3	R/W	0
2	PWM2EN	PWM2 enable 0: CT16Bn_PWM2 is controlled by EMC2. 1: PWM mode is enabled for CT16Bn_PWM2.	R/W	0
1	PWM1EN	PWM1 enable 0: CT16Bn_PWM1 is controlled by EMC1. 1: PWM mode is enabled for CT16Bn_PWM1.	R/W	0
0	PWM0EN	PWM0 enable 0: CT16Bn_PWM0 is controlled by EMC0. 1: PWM mode is enabled for CT16Bn_PWM0.	R/W	0

11.10.12 CT16Bn PWM Control register (CT16Bn_PWMCTRL) (n=1,5)

Address Offset: 0x34

The PWM Control register is used to configure the match outputs as PWM outputs. Each match output can be in-dependently set to perform either as PWM output or as match output whose function is controlled by CT16Bn_EM register.

For CT16B0, CT16B1, and CT16B5, a maximum of 4single edge controlled PWM outputs can be selected via the CT16Bn_PWMCTRL[3:0] bits. One additional match register determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output is set to HIGH. The timer is reset by the match register that is configured to set the PWM cycle length. When the timer is reset to zero, all currently HIGH match outputs configured as PWM outputs are cleared.

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23	PWM3IOEN	CT16Bn_PWM3/GPIO selection bit 0: CT16Bn_PWM3 pin act as GPIO 1: CT16Bn_PWM3 pin act as match output, and output signal depends on PWM2EN bit.	R/W	0
22	PWM2IOEN	CT16Bn_PWM2/GPIO selection bit 0: CT16Bn_PWM2 pin act as GPIO 1: CT16Bn_PWM2 pin act as match output, and output signal depends on PWM2EN bit.	R/W	0
21	PWM1IOEN	CT16Bn_PWM1/GPIO selection bit 0: CT16Bn_PWM1 pin act as GPIO 1: CT16Bn_PWM1 pin act as match output, and output signal depends on PWM1EN bit.	R/W	0
20	PWM0IOEN	CT16Bn_PWM0/GPIO selection bit 0: CT16Bn_PWM0 pin act as GPIO 1: CT16Bn_PWM0 pin act as match output, and output signal depends on PWM0EN bit.	R/W	0
19:12	Reserved		R	0
11:10	PWM3MODE[1:0]	PWM3 output mode 00: PWM mode 1 01: PWM mode 2 10: Forced to 0	R/W	0

		11: Forced to 1		
9:8	PWM2MODE[1:0]	PWM2 output mode 00: PWM mode 1 01: PWM mode 2 10: Forced to 0 11: Forced to 1	R/W	0
7:6	PWM1MODE[1:0]	PWM1 output mode 00: PWM mode 1 01: PWM mode 2 10: Forced to 0 11: Forced to 1	R/W	0
5:4	PWM0MODE[1:0]	PWM0 output mode 00: PWM mode 1 01: PWM mode 2 10: Forced to 0 11: Forced to 1	R/W	0
3	PWM3EN	PWM3 enable 0: CT16Bn_PWM3 is controlled by EMC3 1: PWM mode is enabled for CT16Bn_PWM3.	R/W	0
2	PWM2EN	PWM2 enable 0: CT16Bn_PWM2 is controlled by EMC2. 1: PWM mode is enabled for CT16Bn_PWM2.	R/W	0
1	PWM1EN	PWM1 enable 0: CT16Bn_PWM1 is controlled by EMC1. 1: PWM mode is enabled for CT16Bn_PWM1.	R/W	0
0	PWM0EN	PWM0 enable 0: CT16Bn_PWM0 is controlled by EMC0. 1: PWM mode is enabled for CT16Bn_PWM0.	R/W	0

11.10.13 CT16Bn Timer Raw Interrupt Status register (CT16Bn_RIS) (n=0,1,5)

Address Offset: 0x38

This register indicates the raw status for Timer/PWM interrupts. A Timer/PWM interrupt is sent to the interrupt controller if the corresponding bit in the CT16Bn_IE register is set.

Bit	Name	Description	Attribute	Reset
31	BRKIF	Break interrupt flag (Only CT16B0) 0: No break condition occurs 1: Break condition occurs	R	0
30:6	Reserved		R	0
5	MR9IF	Interrupt flag for match channel 9. 0: No interrupt on match channel 9 1: Interrupt requirements met on match channel 9.	R	0
4	CAP0IF	Interrupt flag for capture channel 0. 0: No interrupt on CAP0 1: Interrupt requirements met on CAP0.	R	0
3	MR3IF	Interrupt flag for match channel 3. 0: No interrupt on match channel 3 1: Interrupt requirements met on match channel 3.	R	0
2	MR2IF	Interrupt flag for match channel 2. 0: No interrupt on match channel 2 1: Interrupt requirements met on match channel 2.	R	0
1	MR1IF	Interrupt flag for match channel 1. 0: No interrupt on match channel 1 1: Interrupt requirements met on match channel 1.	R	0
0	MR0IF	Interrupt flag for match channel 0. 0: No interrupt on match channel 0 1: Interrupt requirements met on match channel 0.	R	0

11.10.14 CT16Bn Timer Interrupt Clear register (CT16Bn_IC) (n=0,1,5)

Address Offset: 0x3C

Bit	Name	Description	Attribute	Reset

31	BRKIC	Break interrupt flag (Only CT16B0) 0: No break condition occurs 1: Break condition occurs	W	0
30:6	Reserved		R	0
5	MR9IC	0: No effect 1: Clear MR9IF bit	W	0
4	CAP0IC	0: No effect 1: Clear CAP0IF bit	W	0
3	MR3IC	0: No effect 1: Clear MR3IF bit	W	0
2	MR2IC	0: No effect 1: Clear MR2IF bit	W	0
1	MR1IC	0: No effect 1: Clear MR1IF bit	W	0
0	MR0IC	0: No effect 1: Clear MR0IF bit	W	0

11.10.15 CT16Bn Timer Match register 9 (CT16Bn_MR9) (n=0,1,5)

Address Offset: 0x40

The Match register value is continuously compared to the Timer Counter (TC) value. When the two values are equal, actions can be triggered automatically. The action possibilities are to reset the Timer Counter, or stop the timer. Actions are controlled by the settings in the CT16Bn_MCTRL[31:30].

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	MR[15:0]	Timer counter match value	R/W	0

11.10.16 CT16Bn PWMmN Dead-band Period register (CT16Bn_PWMmNDB) (n=0)

Address Offset: 0x44 (m=0), 0x48 (m=1), 0x4C (m=2), 0x50 (m=3)

The PWMmNDB register is used to configure the dead-band period of the PWMmN outputs, and is only usable when PWMmNIOEN[1:0] ≠00b

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9:0	DB[9:0]	Count of PWMmN output dead-band period time PWMmN output dead-band period time = DB * CT16Bn_PCLK * (PR+1) cycle	R/W	0

11.10.17 CT16Bn PWM Load Mode Control register (CT16Bn_LOADCTRL) (n=0,1,5)

Address Offset: 0x74

The LOADCTRL register is used to configure the load mode of MRx specifically when TC is equal to 0 and TC is equal to MR9.

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:6	LOAD_MR3	MR3 load mode selection in center-aligned mode 0: No effect 1: Load MR3 value at TC=0 and MR3 value is unchanged at TC=MR9 2: MR3 value is unchanged at TC=0 and Load MR3 value at TC=MR9 3: Load MR3 value at TC=0 and TC=MR9	R/W	1

5:4	LOAD_MR2	MR2 load mode selection in center-aligned mode 0: No effect 1: Load MR2 value at TC=0 and MR2 value is unchanged at TC=MR9 2: MR2 value is unchanged at TC=0 and Load MR0 value at TC=MR9 3: Load MR2 value at TC=0 and TC=MR9	R/W	1
3:2	LOAD_MR1	MR1 load mode selection in center-aligned mode 0: No effect 1: Load MR1 value at TC=0 and MR1 value is unchanged at TC=MR9 2: MR1 value is unchanged at TC=0 and Load MR1 value at TC=MR9 3: Load MR1 value at TC=0 and TC=MR9 Other: Reserved	R/W	1
1:0	LOAD_MR0	MR0 load mode selection in center-aligned mode 0: No effect 1: Load MR0 value at TC=0 and MR0 value is unchanged at TC=MR9 2: MR0 value is unchanged at TC=0 and Load MR0 value at TC=MR9 3: Load MR0 value at TC=0 and TC=MR9	R/W	1

11.10.18 CT16Bn DMA Mode register (CT16Bn_DMA) (n=0,1,5)

Address Offset: 0xD0

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
5	DMA_MR9	MR9 DMA request active enable 0: Disable. MR9 DMA request cannot issue. 1: Enable. MR9 DMA request can issue.	R/W	0
4	DMA_CAP0	CAP0 DMA request active enable 0: Disable. CAP0 DMA request cannot issue. 1: Enable. CAP0 DMA request can issue.	R/W	0
3	DMA_MR3	MR3 DMA request active enable 0: Disable. MR3 DMA request cannot issue. 1: Enable. MR3 DMA request can issue.	R/W	0
2	DMA_MR2	MR2 DMA request active enable 0: Disable. MR2 DMA request cannot issue. 1: Enable. MR2 DMA request can issue.	R/W	0
1	DMA_MR1	MR1 DMA request active enable 0: Disable. MR1 DMA request cannot issue. 1: Enable. MR1 DMA request can issue.	R/W	0
0	DMA_MR0	MR0 DMA request active enable 0: Disable. MR0 DMA request cannot issue. 1: Enable. MR0 DMA request can issue.	R/W	0

11.10.19 CT16Bn DMA MRm Alias register 1 (CT16Bn_DMAMRA1) (n=0,1,5)

Address Offset: 0xD4

Bit	Name	Description	Attribute	Reset
31:16	MR1_ALIAS	MR1 alias for DMA access	W	0
15:0	MR0_ALIAS	MR0 alias for DMA access	W	0

11.10.20 CT16Bn DMA MRm Alias register 1 (CT16Bn_DMAMRA2) (n=0,1,5)

Address Offset: 0xD8

Bit	Name	Description	Attribute	Reset
31:16	MR3_ALIAS	MR3 alias for DMA access	W	0

15:0	MR2_ALIAS	MR2 alias for DMA access	W	0

11.10.21 CT16Bn Break Function Control register (CT16Bn_BRKCTRL) (n=0)

Address Offset: 0xE0

Bit	Name	Description	Attribute	Reset
31	Reserved		R	0
30	BRKIE	Enable generating an interrupt when BRKIF = 1 0: Disable 1: Enable	R/W	0
29:7	Reserved		R	0
6:4	BRKDB	Break pin (BRK) debounce time 0: Debounce time=(1~3)T*CT16Bn_PCLK 1: Debounce time=(2~4)T*CT16Bn_PCLK 2: Debounce time=(4~6)T*CT16Bn_PCLK 3: Debounce time=(8~10T)*CT16Bn_PCLK 4: Debounce time=(16~18T)*CT16Bn_PCLK 5: Debounce time=(32~34T)*CT16Bn_PCLK 6: Debounce time=(64~66T)*CT16Bn_PCLK 7: Debounce time=(128~130T)*CT16Bn_PCLK	R/W	0
3	BRKLEVEL	The trigger level of PWM channels break function 0: Low level (CMP V- > V+) 1: High level (CMP V+ > V-)	R/W	0
2:0	BRKSEL	The trigger source selection of PWM channels break function 0: Disable 1: CMP0 output 2: CMP1 output 3: CMP2 output 4: Break pin (BRK) Other: Reserved	R/W	0

12 WATCHDOG TIMER (WDT)

12.1 OVERVIEW

The purpose of the Watchdog is to reset the MCU within a reasonable amount of time if it enters an erroneous state. When enabled, the Watchdog will generate a system reset or interrupt if the user program fails to "feed" (or reload) the Watchdog within a predetermined amount of time.

The Watchdog consists of a divide by 128 fixed pre-scaler and an 8-bit counter. The clock is fed to the timer via a pre-scaler. The timer decrements when clocked. The minimum value from which the counter decrements is 0x01. Hence the minimum Watchdog interval is ($T_{WDT\_PCLK} \times 128 \times 1$) and the maximum Watchdog interval is ($T_{WDT\_PCLK} \times 128 \times 256$).

The Watchdog should be used in the following manner:

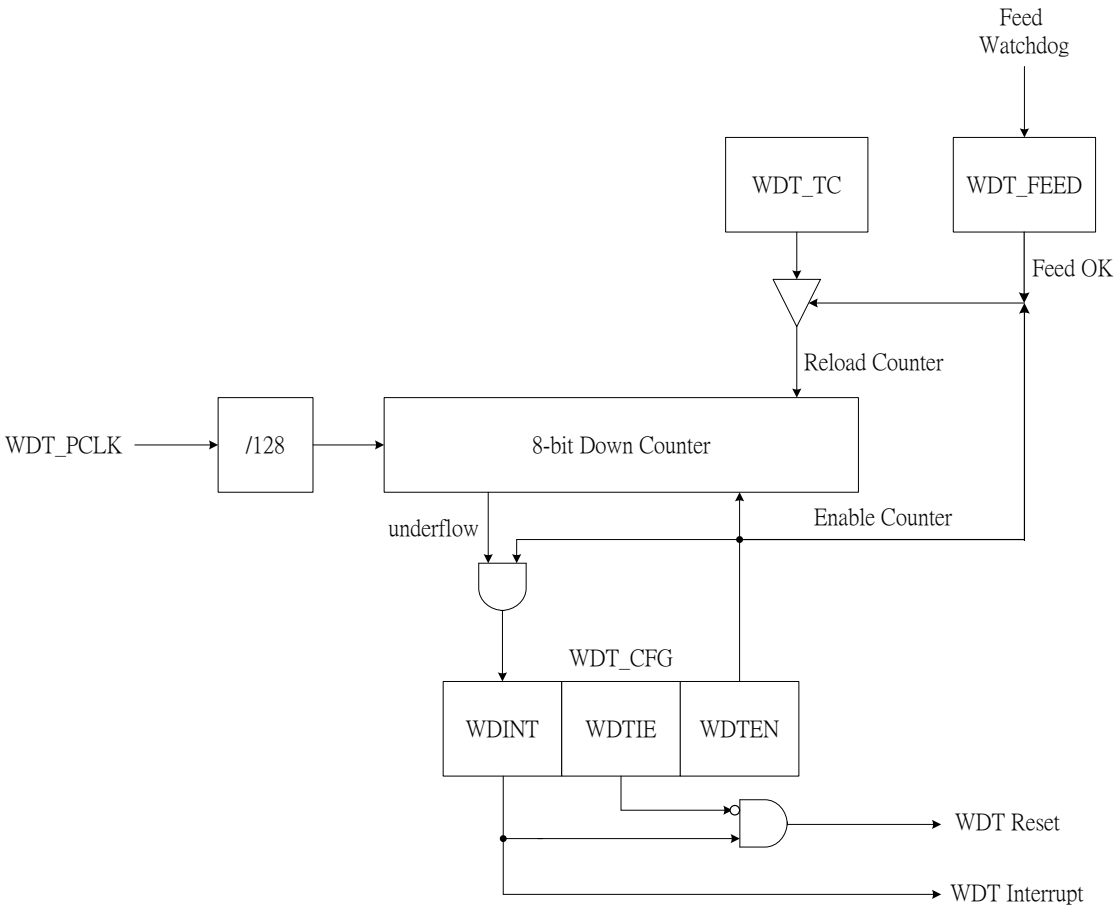
1. Set the prescale value for the watchdog clock with WDTPRE bits in APB Clock Prescale register 1 (SYS1_APBCP1) register.
2. Set the Watchdog timer constant reload value in WDT_TC register.
3. Enable the Watchdog and setup the Watchdog timer operating mode in WDT_CFG register.
4. The Watchdog should be fed again by writing 0x55AA to WDT_FEED register before the Watchdog counter underflows to prevent reset or interrupt.

When the watchdog is started by setting the WDTEN in WDT_CFG register, the time constant value is loaded in the watchdog counter and the counter starts counting down. When the Watchdog is in the reset mode and the counter underflows, the CPU will be reset, loading the stack pointer and program counter from the vector table as in the case of external reset. Whenever the value 0x55AA is written in WDT_FEED register, the WDT_TC value is reloaded in the watchdog counter and the watchdog reset or interrupt is prevented.

The watchdog timer block uses two clocks: HCLK and WDT_PCLK. HCLK is used for the AHB accesses to the watchdog registers and is derived from the system clock. The WDT_PCLK is used for the watchdog timer counting.

Watchdog reset or interrupt will occur any time the watchdog is running and has an operating clock source.

12.2 BLOCK DIAGRAM



12.3 WDT REGISTERS

Base Address: 0x4001 0000

12.3.1 Watchdog Configuration register (WDT_CFG)

Address Offset: 0x00

The WDT_CFG register controls the operation of the Watchdog through the combination of WDTEN and WDTIE bits.
This register indicates the raw status for Watchdog Timer interrupts. A WDT interrupt is sent to the interrupt controller if both the WDINT bit and the WDTIE bit are set.

Bit	Name	Description	Attribute	Reset
31:16	WDKEY	Watchdog register key. Read as 0. When writing to the register you must write 0x5AFA to WDKEY, otherwise behavior of writing to the register is ignored.	W	0
15:3	Reserved		R	0
2	WDTINT	Watchdog interrupt flag 0: Read→Watchdog does not cause an interrupt. Writ→ Clear this flag. SW shall feed Watchdog before clearing. 1: Watchdog timeout and causes an interrupt (Only when WDTIE =1).	R/W	0
1	WDTIE	Watchdog interrupt enable 0: Watchdog timeout will cause a chip reset. (Watchdog reset mode) Watchdog counter underflow will reset the MCU, and will clear the WDINT flag. 1: Watchdog timeout will cause an interrupt. (Watchdog interrupt mode)	R/W	0
0	WDTEN	Watchdog enable 0: Disable 1: Enable. When enable the watchdog, the WDT_TC value is loaded in the watchdog counter.	R/W	0

12.3.2 Watchdog Timer Constant register (WDT_TC)

Address Offset: 0x08

The WDT_TC register determines the time-out value. Every time a feed sequence occurs the WDT_TC content is reloaded in to the Watchdog timer. It's an 8-bit counter. Thus the time-out interval is T_{WDT_PCLK} × 128 x 1 ~ T_{WDT_PCLK} × 128 x 256.

Watchdog overflow time = (31.25us x 1) x 128 x 1 ~ (31.25us x 32) x 128 x 256
= 4ms ~ 32768ms

Bit	Name	Description	Attribute	Reset
31:16	WDKEY	Watchdog register key. Read as 0. When writing to the register you must write 0x5AFA to WDKEY, otherwise behavior of writing to the register is ignored.	W	0
15:8	Reserved		R	0
7:0	TC[7:0]	Watchdog timer constant reload value = TC[7:0]+1 0000 0000 : Timer constant = 1 0000 0001 : Timer constant = 2 ……. ……. 1111 1110 : Timer constant = 255 1111 1111 : Timer constant = 256	R/W	0xFF

12.3.3 Watchdog Feed register (WDT_FEED)

Address Offset: 0x0C

Bit	Name	Description	Attribute	Reset
31:16	WDKEY	Watchdog register key. Read as 0. When writing to the register you must write 0x5AFA to WDKEY, otherwise behavior of writing to the register is ignored.	W	0
15:0	FV[15:0]	Feed value (Read as 0x0) 0x55AA: The watchdog is fed, and the WDT_TC value is reloaded in the watchdog counter.	W	0

13 REAL-TIME CLOCK (RTC)

13.1 OVERVIEW

The RTC is an independent timer. The RTC provides a set of continuously running counters which can be used to provide a clock-calendar function with suitable software.
The counter values can be written to set the current time/date of the system.

13.2 FEATURES

- Programmable prescale value: division factor up to 2^{20}
- 32-bit programmable counter for long-term measurement
- The RTC clock source could be any of the following:
  - ELS X'TAL
  - ILRC
- Reset sources of the RTC Core (Prescale value, Counter and Divider):
  - "Cold" boot
- One dedicated enabled interrupt
  - Seconds interrupt: generating a periodic interrupt signal with a programmable period length (up to 1 second).

13.3 FUNCTIONAL DESCRIPTION

13.3.1 INTRODUCTION

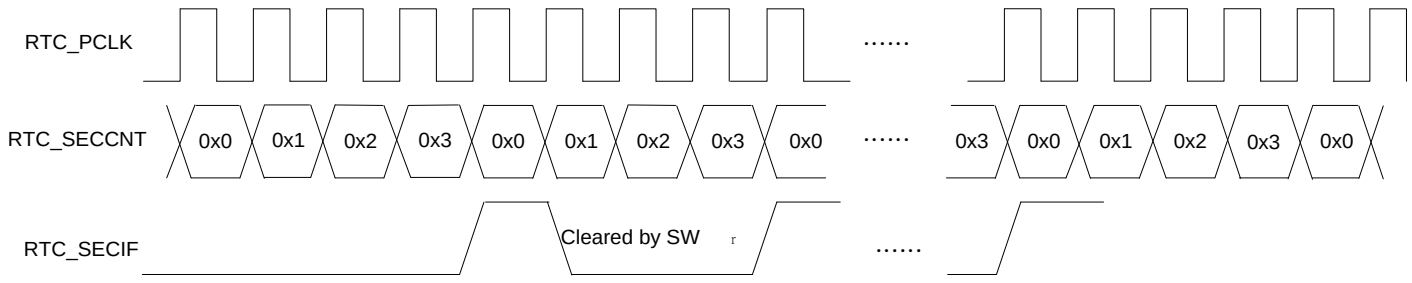
RTC core includes a 20-bit preload value (RTC SECCNTV). Every clock period, the RTC generates an interrupt (Second Interrupt) if it is enabled in RTC_IE register.

13.3.2 RTC FLAG ASSERTION

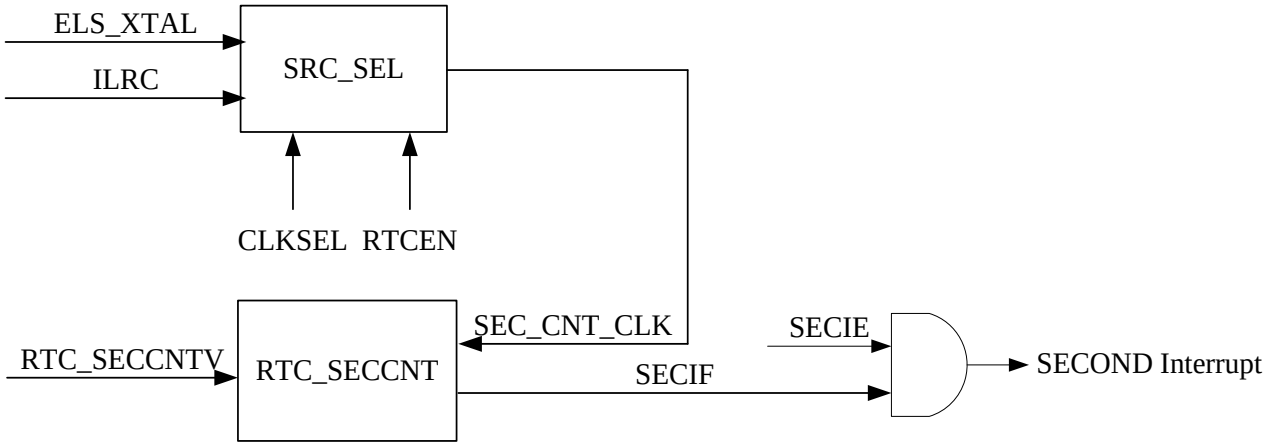
The RTC Second interrupt flag (SECIF) is asserted on each RTC Core clock cycle before the update of the RTC Counter.

13.3.3 RTC OPERATION

The following figure shows the RTC waveform when it is configured with RTC_SECCNTV=3.



13.4 BLOCK DIAGRAM



13.5 RTC REGISTERS

Base Address: 0x4001 2000

13.5.1 RTC Control register (RTC_CTRL)

Address offset: 0x00

✸ ***Note: RTCEN bit shall be set at last!***

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	RTCEN	RTC enable bit 0: Disable 1: Enable. Reset SEC_CNT.	R/W	0

13.5.2 RTC Clock Source Select register (RTC_CLKS)

Address offset: 0x04

✸ ***Note: SW shall disable RTC (RTCEN=0) when changing the value of this register.***

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	CLKSEL	RTC clock source selection. HW will reset SEC_CNT when changing the value. 0: ILRC 1: ELS X'TAL	R/W	0

13.5.3 RTC Interrupt Enable register (RTC_IE)

Address offset: 0x08

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	SECIE	Second interrupt enable 0: Disable 1: Enable	R/W	0

13.5.4 RTC Raw Interrupt Status register (RTC_RIS)

Address offset: 0x0C

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	SECIF	Second interrupt flag This bit is set by HW when SEC_CNT=SEC_CNTV. An interrupt is generated if SECIE=1. 0: Second flag condition not met. 1: Second flag condition met.	R	0

13.5.5 RTC Interrupt Clear register (RTC_IC)

Address offset: 0x10

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	SECIC	0: No effect 1: Clear SECIF bit	W	0

13.5.6 RTC Second Counter Reload Value register (RTC_SECCNTV)

Address offset: 0x14

Bit	Name	Description	Attribute	Reset
31:20	Reserved		R	0
19:0	SECCNTV[19:0]	RTC second counter reload value. Update this register will reset RTC_SECCNT register. The zero value is not recommended, and will be replaced with default value (0x8000) by HW.	R/W	0x8000

13.5.7 RTC Second Count register (RTC_SECCNT)

Address offset: 0x18

The RTC core has one 32-bit programmable counter, and this register keeps the current counting value of this counter.

Bit	Name	Description	Attribute	Reset
31:0	SECCNT[31:0]	RTC second counter The current value of the RTC counter.	R	0

14 SPI

14.1 OVERVIEW

The SPI is a Synchronous Serial Port controller capable of operation on a SPI, and 4-wire SSI bus. It can interact with multiple masters and slaves on the bus. Only a single master and a single slave can communicate on the bus during a given data transfer. Data transfers are in principle full duplex, with frames of 4 to 16 bits of data flowing from the master to the slave and from the slave to the master. In practice it is often the case that only one of these data flows carries meaningful data.

14.2 FEATURES

- Compatible with Motorola SPI bus.
- Synchronous Serial Communication.
- Supports master or slave operation.
- 8-frame FIFO for both transmitter and receiver.
- 4-bit to 16-bit frame.
- Maximum SPI speed of 30 Mbps (master) or 16 Mbps (slave)
- Data transfer format is from MSB or LSB controlled by register.
- The start phase of data sampling location selection is 1st-phase or 2nd-phase controlled register.
- ARGB Mode
- Supply DMA transfer

14.3 PIN DESCRIPTION

Pin Name	Type	Description	GPIO Configuration
SCKn	O	SPI Serial clock (Master)	
	I	SPI Serial clock (Slave)	Depends on GPIOn_CFG
SELn	O	SPI Slave Select/SSI Frame Sync (Master)	
	I	SPI Slave Select (Slave)	Depends on GPIOn_CFG
MISOn	I	Master In Slave Out (Master)	Depends on GPIOn_CFG
	O	Master In Slave Out (Slave)	
MOSIn	O	Master Out Slave In (Master)	
	I	Master Out Slave In (Slave)	Depends on GPIOn_CFG

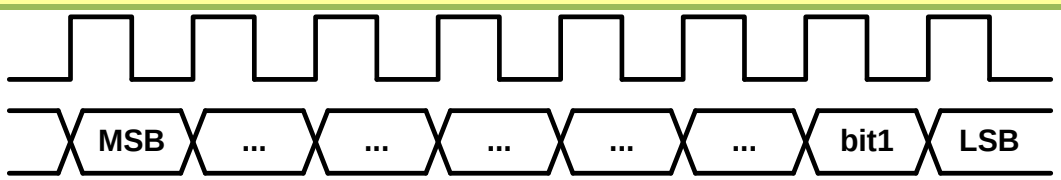
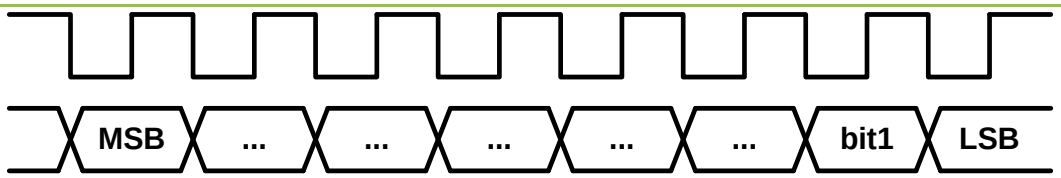
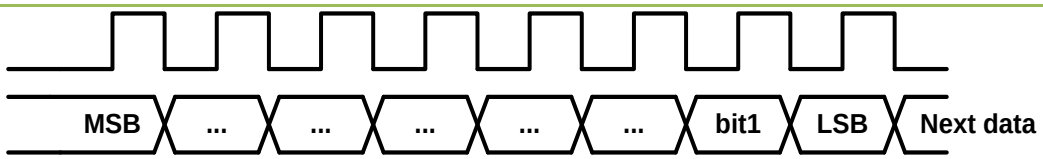
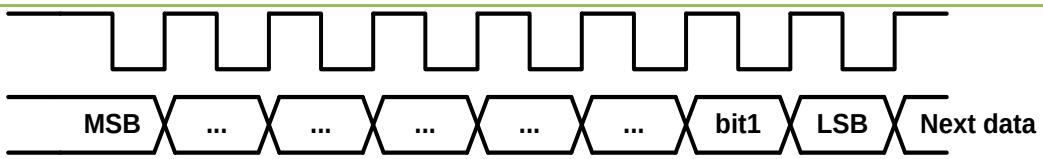
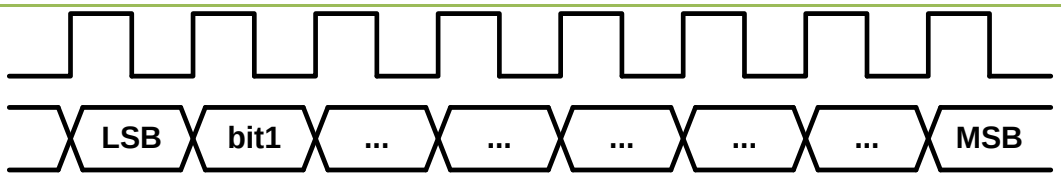
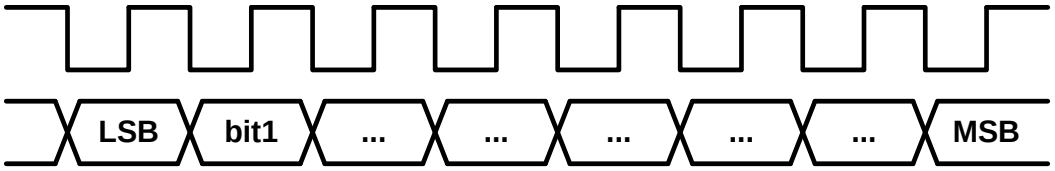
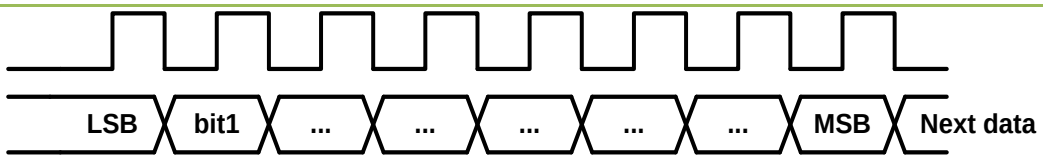
14.4 INTERFACE DESCRIPTION

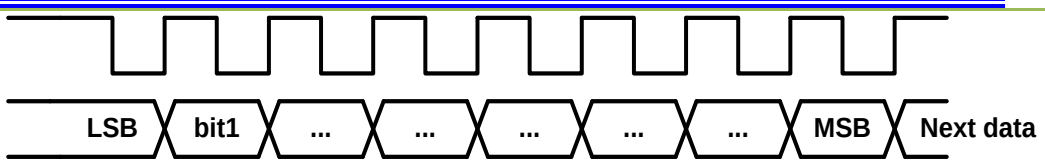
14.4.1 SPI

The SPI interface is a 4-wire interface where the SEL signal behaves as a slave select. The main feature of the SPI format is that the inactive state and phase of the SCK signal are programmable through the CPOL and CPHA bits in SPIn_CTRL1 register.

When the “CPOL” clock polarity control bit is LOW, it produces a steady state low value on the SCK pin. If the CPOL clock polarity control bit is HIGH, a steady state high value is placed on the CLK pin when data is not being transferred. The “CPHA” clock phase bit controls the phase of the clock on which data is sampled. When CPHA=1, the SCK first edge is for data transition, and receive and transmit data is at SCK 2nd edge. When CPHA=0, the 1st bit is fixed already, and the SCK first edge is to receive and transmit data.

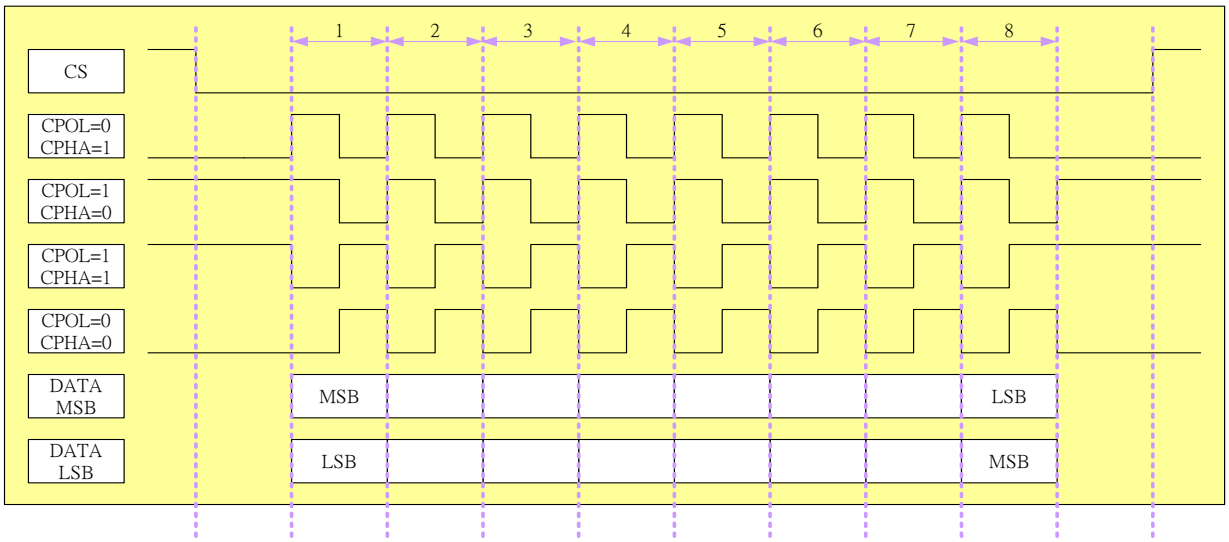
The SPI data transfer timing as following figure:

MLSB	CPOL	CPHA	SCK Idle Status	Diagrams
0	0	1	Low	
0	1	1	High	
0	0	0	Low	
0	1	0	High	
1	0	1	Low	
1	1	1	High	
1	0	0	Low	

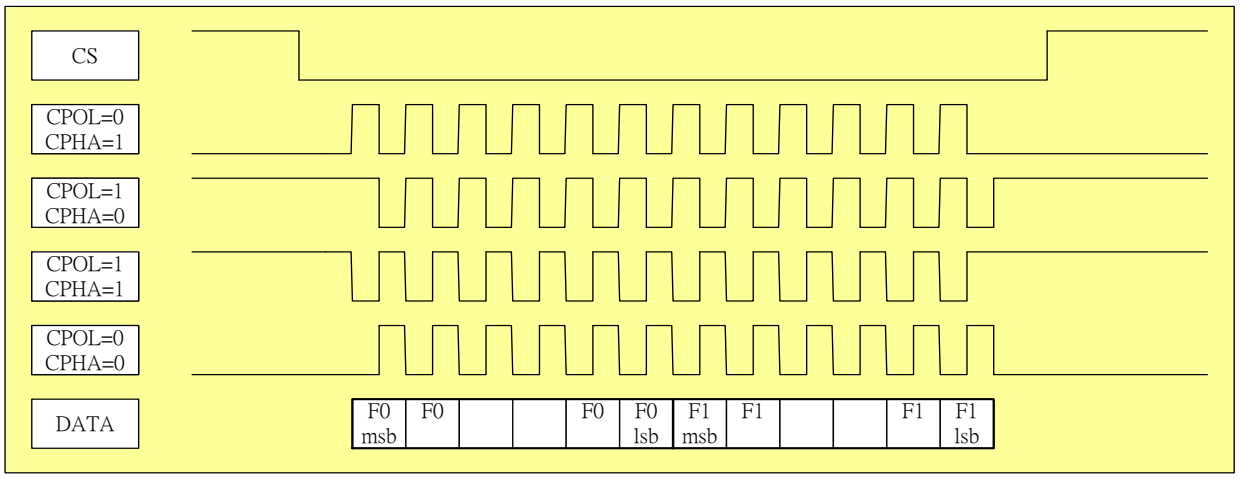
1	1	0	High	

14.4.2 COMMUNICATION FLOW

14.4.2.1 SINGLE-FRAME



14.4.2.2 MULTI-FRAME

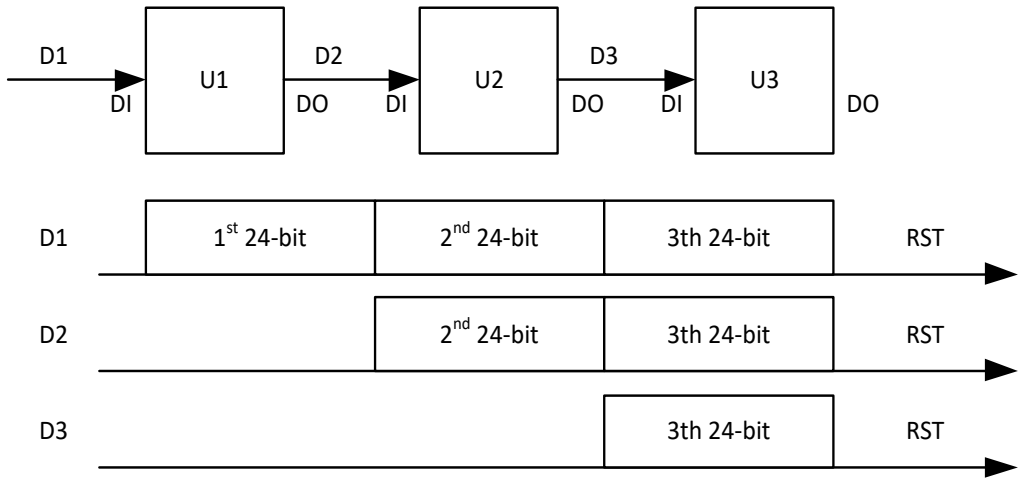


14.5 AUTO-SEL

The Auto-SEL function is disabled (SELDIS = 1) by default, HW does NOT control SELn pin at all, and SELn pin is GPIO. If Auto-SEL function is enabled (SELDIS = 0), SPI HW controls the SELn activity, and SELn is assigned by PFPA_SPI register.

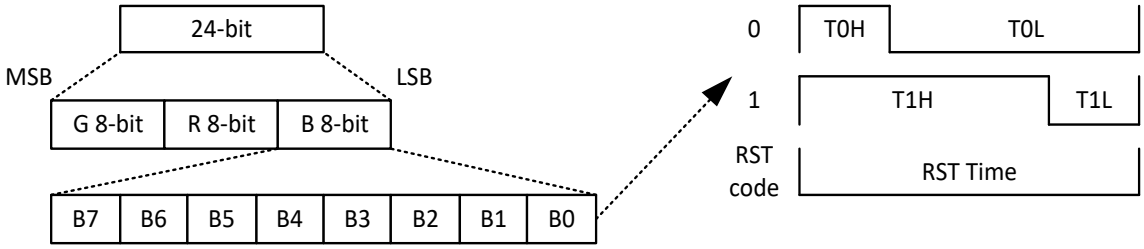
14.6 ARGB MODE

ARGB lighting control is a single-wire control method, so the parameters of each light are updated through serial communication.



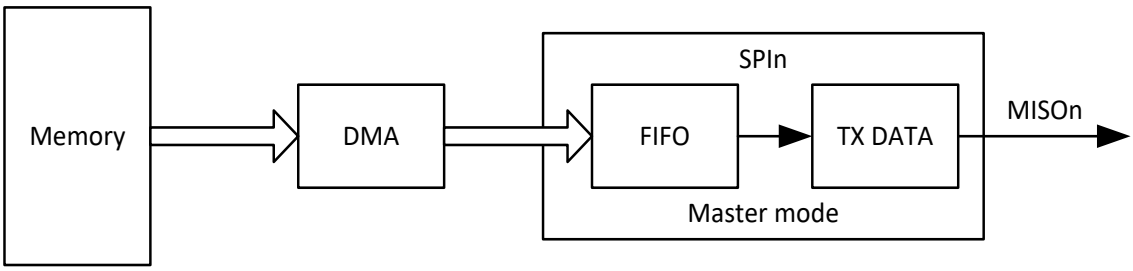
14.6.1 Master Mode MOSI

Using SPI Master mode output, the waveform change of each ARGB code bit is determined by the TX data. When the ARGBM bit is 1, the SCK, SEL, and MISO functions can be turned off, and the corresponding pins can be returned to GPIO mode to save pin resources.



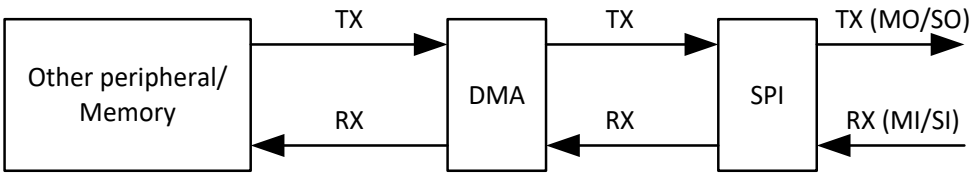
14.6.2 DMA Transfer

ARGB update codes can be continuously output via DMA lookup table.



14.7 DMA MODE

The SPI DMA mode is to use DMA engine to move data in/out SPI. Before the DMA transfer start, DMA engine must be set up first. In SPI DMA TX Mode, DMA gets data from other peripherals or memories to SPI as TX data. In SPI DMA RX Mode, DMA receive data from SPI and send to other peripherals or memories.



TX mode

DMA channel register			SPIn register				
SRC_WIDTH	DST_WIDTH	TOT_SIZE	NEW_TH_EN	FIFO depth	FIFO unit	DL[3:0]	DMA_SIZE
32 bits	16 bits	N/2	0	8	16 bits	15~2	N
16 bits	16 bits	N	0	8	16 bits	15~2	N
8 bits	16 bits	N*2	0	8	16 bits	15~2	N
32 bits	8 bits	N/4	1	16	8 bits	7~2	N
16 bits	8 bits	N/2	1	16	8 bits	7~2	N
8 bits	8 bits	N	1	16	8 bits	7~2	N
32 bits	8 bits	N/4	0	8	16 bits	7~2	N
16 bits	8 bits	N/2	0	8	16 bits	7~2	N
8 bits	8 bits	N	0	8	16 bits	7~2	N

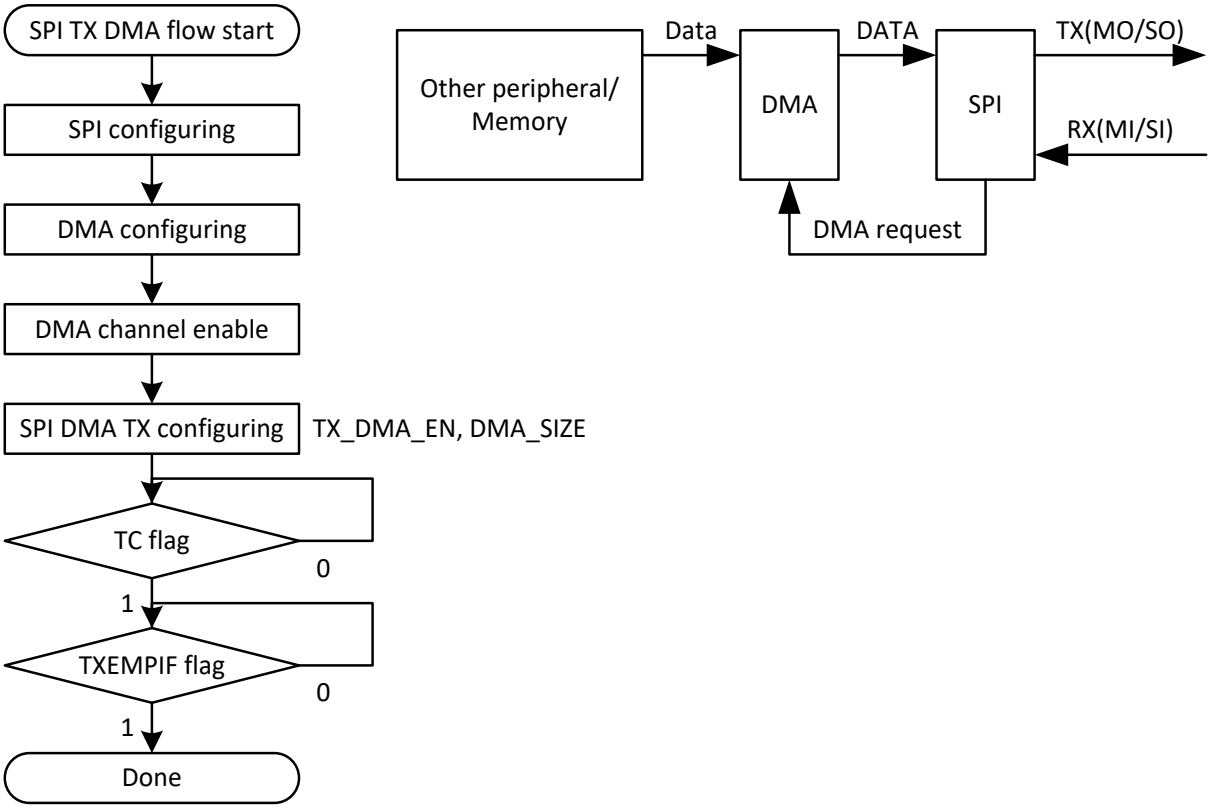
RX mode

DMA channel register			SPIn register				
SRC_WIDTH	DST_WIDTH	TOT_SIZE	NEW_TH_EN	FIFO depth	FIFO unit	DL[3:0]	DMA_SIZE
16 bits	32 bits	N	0	8	16 bits	15~2	N
16 bits	16 bits	N	0	8	16 bits	15~2	N
16 bits	8 bits	N	0	8	16 bits	15~2	N
8 bits	32 bits	N	1	16	8 bits	7~2	N
8 bits	16 bits	N	1	16	8 bits	7~2	N
8 bits	8 bits	N	1	16	8 bits	7~2	N
8 bits	32 bits	N	0	8	16 bits	7~2	N
8 bits	16 bits	N	0	8	16 bits	7~2	N
8 bits	8 bits	N	0	8	16 bits	7~2	N

✸ ***Note:***
1. ***The burst data cannot be larger than the FIFO size of SPI.***
2. ***TX FIFO threshold level = FIFO size – Burst size.***
3. ***RX FIFO threshold level = Burst size - 1.***

14.7.1 SPI DMA TX Mode

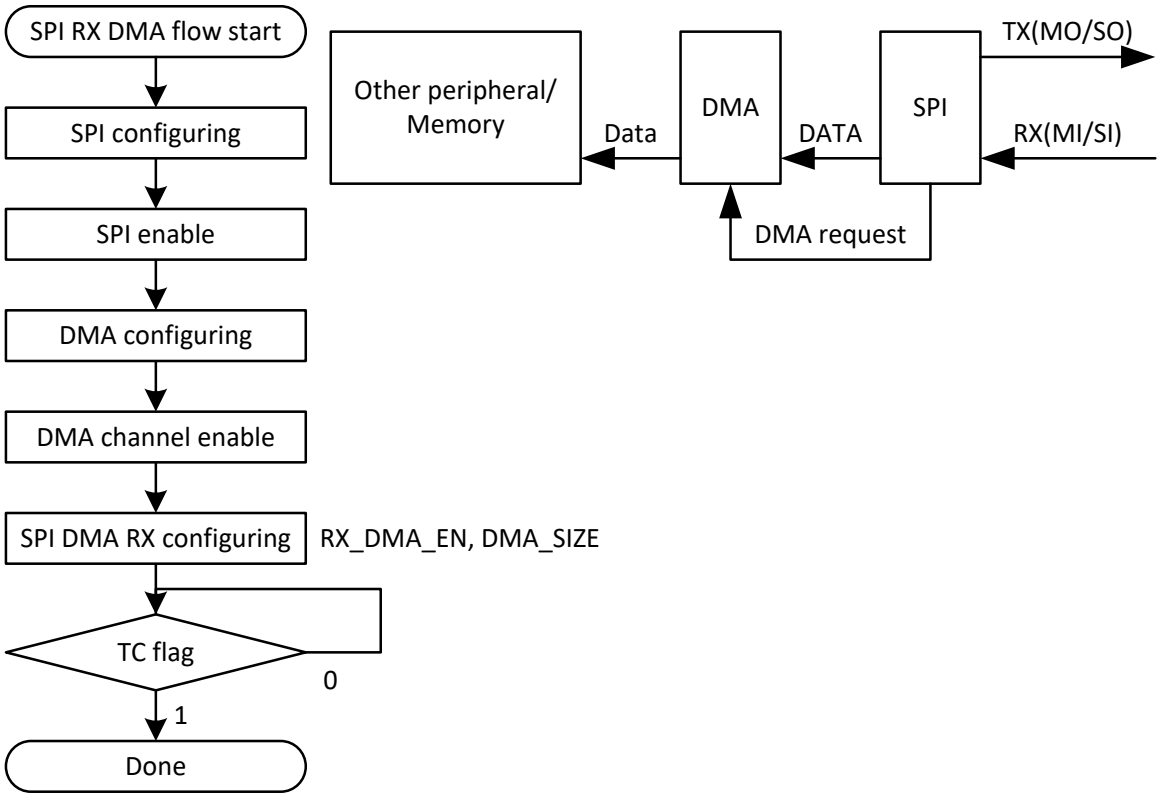
The following register programming flow is used for SPI DMA TX mode. In this flow FW check DMA TX channel transfer finish then flow can go to the end. This example will turn on DMA Channel to move data from other peripherals or memory to SPI data register. When TX FIFO is less than the threshold, DMA request is issued.



> ✸ ***Note:***
> 1. ***TX FIFO threshold = FIFO size – Burst size. Ensure that the remaining space of the FIFO can accommodate burst data.***
> 2. ***In master mode, after the transmission starts, the CS state will be keep 0 until all transmissions are completed.***

14.7.2 SPI DMA RX Mode

The following register programming flow chart used for SPI DMA RX mode. This example will turn on DMA Channel to move data from SPI data register to other peripherals or SRAM. When RX FIFO is greater than the threshold, DMA request is issued. If the SPI is master mode, when the SPI enable and RX enable are set, the SPI will start to transmit. TX will send dummy data to continuously receive RX.

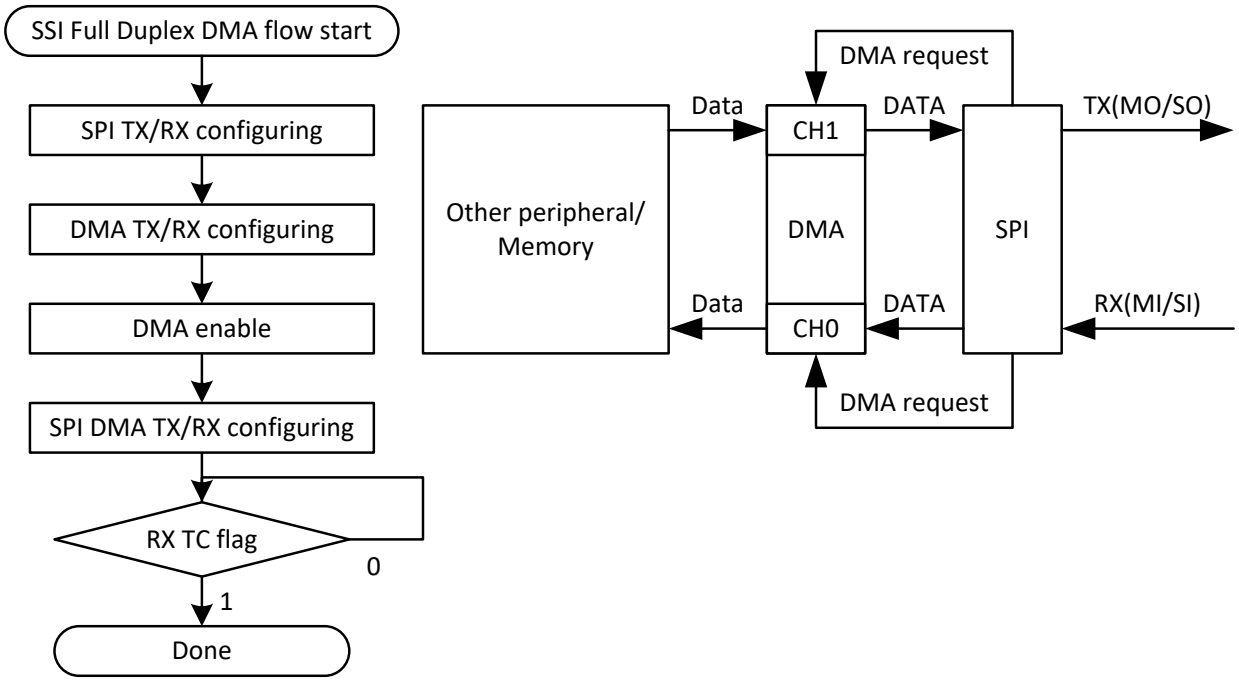


✸ ***Note:***

1. ***RX FIFO threshold level = Burst size -1.***
2. ***In master mode, after the transmission starts, the CS state will be keep 0 until all transmissions are completed.***

14.7.3 SPI DMA Full-Duplex Mode

The following programming flow is used for SPI DMA full-duplex mode. This example DMA use 2 channel to achieve SPI full-duplex function. When RX FIFO is greater than the threshold, DMA request is issued for channel 0. When TX FIFO is less than the threshold, DMA request is issued for channel 1. If the SPI is master mode, SPI transmission must wait for the TX buffer to be written.



✸ ***Note:***

1. ***TX FIFO threshold = FIFO size – Burst size. RX FIFO threshold level = Burst size -1. Because FIFO is shared by TX and RX, neither burst sizes can over half of the FIFO.***
2. ***In full-duplex mode, DMA_SIZE changes are controlled by RX DMA.***
3. ***In master mode, after the transmission starts, the CS state will be keep 0 until all transmissions are completed.***
4. ***The DMA transfer size of DMA TX and RX should be set the same, and the DMA_SIZE should be set the same as TOT_SIZE.***
5. ***TX_DMA_EN and RX_DMA_EN should be set at the same time.***

14.7.4 SPI DMA Configuration Recommendations

If data in TX FIFO ≦ TX FIFO threshold, the TX FIFO request will be issued.

If data in RX FIFO > RX FIFO threshold, the RX FIFO request will be issued.

14.7.4.1 SPI DMA TX Only

Configuration	Register
DSTADDR = SPI_DATA	DMA_Cn_DSTADDR
DST_RS = SPIn_TX, DST_HE=1	DMA_Cn_CFG
DSTAD_CTL=10 (Fixed) MODE = 1 (Peripheral)	DMA_Cn_CSR
TX_DMA_EN = 1	SPI_DMA

DMAC setting for SPI TX (DL=16bit)				Recommend setting for SPI register (Assume SPI FIFO depth == 8 half-word) NEW_TH_EN = 0 , use TXFIFO_TH	
Cn_CSR SRC_WIDTH	Cn_CSR SRC_SIZE	Cn_SIZE TOT_SIZE	Cn_CSR DST_WIDTH	TXFIFO_TH	DMA_SIZE (Half word unit)
0x2 (word)	0x0 (burst=1)	N / 2 (word)	0x1 (half word)	0x6	N
0x1(half-word)	0x0 (burst=1)	N (half-word)	0x1 (half word)	0x7	N
0x1(half-word)	0x1 (burst=4)	N (half-word)	0x1 (half word)	0x4	N
0x0(byte)	0x1 (bust=4)	N * 2 (byte)	0x1 (half word)	0x6	N
0x0(byte)	0x2 (bust=8)	N * 2 (byte)	0x1 (half word)	0x4	N

DMAC setting for SPI TX (DL=8bit)				Recommend setting for SPI register (Assume SPI FIFO depth == 16 byte) NEW_TH_EN = 1 , use NEW_TXFIFO_TH	
Cn_CSR SRC_WIDTH	Cn_CSR SRC_SIZE	Cn_SIZE TOT_SIZE	Cn_CSR DST_WIDTH	NEW_TXFIFO_TH	DMA_SIZE (Byte unit)
0x2 (word)	0x0 (burst=1)	N / 4 (word)	0x0 (byte)	0xC	N
0x1 (half-word)	0x0 (burst=1)	N / 2 (half-word)	0x0 (byte)	0xE	N
0x1 (half-word)	0x1 (burst=4)	N / 2 (half-word)	0x0 (byte)	0x8	N
0x0 (byte)	0x0 (burst=1)	N (byte)	0x0 (byte)	0xF	N
0x0 (byte)	0x1 (burst=4)	N (byte)	0x0 (byte)	0xC	N
0x0 (byte)	0x2 (burst=8)	N (byte)	0x0 (byte)	0x8	N

14.7.4.2 SPI DMA RX Only

Configuration	Register
SRCADDR = SPI_DATA	DMA_Cn_SRCADDR
SRC_RS = SPIn_RX SRC_HE=1	DMA_Cn_CFG
SRCAD_CTL=10 (Fixed) MODE = 1 (Peripheral)	DMA_Cn_CSR
RX_DMA_EN = 1	SPI_DMA

DMAC setting for SPI RX (DL=16bit)				Recommend setting for SPI register (Assume SPI FIFO depth == 8 half-word) NEW_TH_EN = 0 , use RXFIFO_TH	
Cn_CSR SRC_WIDTH	Cn_CSR SRC_SIZE	Cn_SIZE TOT_SIZE	Cn_CSR DST_WIDTH	RXFIFO_TH	DMA_SIZE (half word unit)
0x1 (half word)	0x1 (burst=4)	N (half word)	0x2 (word) 0x1 (half word) 0x0 (byte)	0x3	N
0x1 (half word)	0x0 (burst=1)	N (half word)	0x1 (half word) 0x0 (byte)	0x0	N

DMAC setting for SPI RX (DL=8bit)				Recommend setting for SPI register (Assume SPI FIFO depth == 16 byte) NEW_TH_EN = 1 , use NEW_RXFIFO_TH	
Cn_CSR SRC_WIDTH	Cn_CSR SRC_SIZE	Cn_SIZE TOT_SIZE	Cn_CSR DST_WIDTH	NEW_RXFIFO_TH	DMA_SIZE (half word unit)
0x0 (byte)	0x0 (burst=1)	N (byte)	0x0 (byte)	0x0	N
0x0 (byte)	0x1 (burst=4)	N (byte)	0x2 (word) 0x1 (half word) 0x0 (byte)	0x3	N
0x0 (byte)	0x2 (burst=8)	N (byte)	0x2 (word) 0x1 (half word) 0x0 (byte)	0x7	N

14.7.4.3 SPI Full-duplex DMA Mode (DL = 16 bit)

DMA Channel n (For SPI TX)		DMA Channel m (For SPI RX)	
Configuration	**Register**	**Configuration**	**Register**
DSTADDR = SPIx_DATA	DMA_Cn_DSTADDR	SRCADDR = SPIx_DATA	DMA_Cm_SRCADDR
DST_RS = SPIx_TX DST_HE=1	DMA_Cn_CFG	SRC_RS = SPIx_RX SRC_HE=1	DMA_Cm_CFG
DSTAD_CTL=10 (Fixed) MODE = 1 (Peripheral)	DMA_Cn_CSR	SRCAD_CTL=10 (Fixed) MODE = 1 (Peripheral)	DMA_Cm_CSR
SPI register configuration (SPI_DMA)			
DMA_SIZE = N TX_DMA_EN & RX_DMA_EN = 1 **(Set at the same time)**			

DMAC setting for SPI TX (DL = 16 bit)				Recommend setting for SPI register (Assume SPI FIFO depth == 8 half-word) NEW_TH_EN = 0 , use TXFIFO_TH	
Cn_CSR SRC_WIDTH	**Cn_CSR SRC_SIZE**	**Cn_SIZE TOT_SIZE**	**Cn_CSR DST_WIDTH**	**TXFIFO_TH**	**DMA_SIZE (half word unit)**
0x2 (word)	0x0 (burst=1)	N / 2 (word)	0x1 (half word)	0x2	N
0x1 (half-word)	0x0 (burst=1)	N (half-word)	0x1 (half word)	0x3	N
0x0 (byte)	0x1 (bust=4)	N * 2 (byte)	0x1 (half word)	0x2	N

DMAC setting for SPI RX (DL = 16 bit)				Recommend setting for SPI register (Assume SPI FIFO depth == 8 half-word) NEW_TH_EN = 0 , use RXFIFO_TH	
Cn_CSR SRC_WIDTH	**Cn_CSR SRC_SIZE**	**Cn_SIZE TOT_SIZE**	**Cn_CSR DST_WIDTH**	**RXFIFO_TH**	**DMA_SIZE (half word unit)**
0x1 (half word)	0x1 (burst=4)	N (half-word)	0x2 (word) 0x1 (half word) 0x0 (byte)	0x3	N
0x1 (half word)	0x0 (burst=1)	N (half-word)	0x1 (half word) 0x0 (byte)	0x0	N

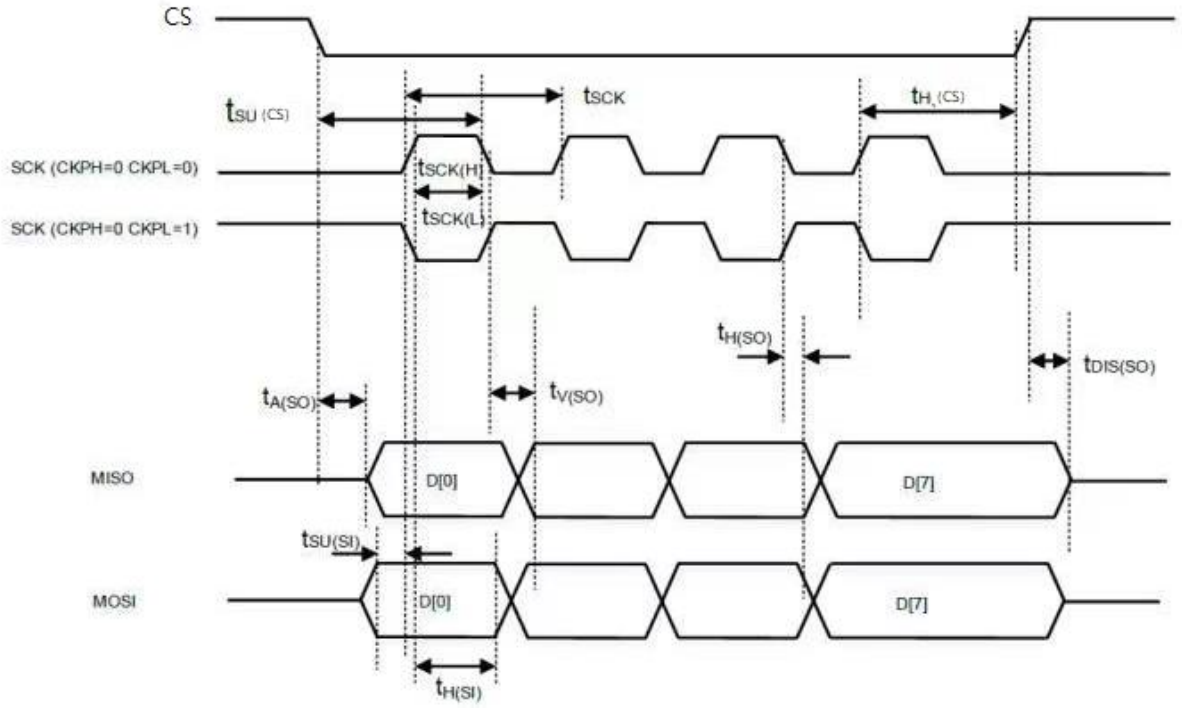
14.7.4.4 SPI Full-duplex DMA Mode (DL = 8 bit)

DMA Channel n (For SPI TX)		DMA Channel m (For SPI RX)	
Configuration	**Register**	**Configuration**	**Register**
DSTADDR = SPIx_DATA	DMA_Cn_DSTADDR	SRCADDR = SPIx_DATA	DMA_Cm_SRCADDR
DST_RS = SPIx_TX DST_HE=1	DMA_Cn_CFG	SRC_RS = SPIx_RX SRC_HE=1	DMA_Cm_CFG
DSTAD_CTL=10 (Fixed) MODE = 1 (Peripheral)	DMA_Cn_CSR	SRCAD_CTL=10 (Fixed) MODE = 1 (Peripheral)	DMA_Cm_CSR
SPI register configuration (SPI_DMA)			
DMA_SIZE = N TX_DMA_EN & RX_DMA_EN = 1 **(Set at the same time)**			

DMC setting for SPI TX (DL = 8 bit)				Recommend setting for SPI register (Assume SPI FIFO depth == 16 byte) NEW_TH_EN = 1 , use NEW_TXFIFO_TH	
Cn_CSR SRC_WIDTH	**Cn_CSR SRC_SIZE**	**Cn_SIZE TOT_SIZE**	**Cn_CSR DST_WIDTH**	**NEW_TXFIFO_TH**	**DMA_SIZE (half word unit)**
0x2 (word)	0x0 (burst=1)	N / 4 (word)	0x0 (byte)	0x4	N
0x1 (half-word)	0x0 (burst=1)	N / 2 (half-word)	0x0 (byte)	0x6	N
0x0 (byte)	0x0 (burst=1)	N (byte)	0x0 (byte)	0x7	N
0x0 (byte)	0x1 (burst=4)	N (byte)	0x0 (byte)	0x4	N

DMAC setting for SPI RX (DL = 8 bit)				Recommend setting for SPI register (Assume SPI FIFO depth == 16 byte) NEW_TH_EN = 1 , use NEW_RXFIFO_TH	
Cn_CSR SRC_WIDTH	**Cn_CSR SRC_SIZE**	**Cn_SIZE TOT_SIZE**	**Cn_CSR DST_WIDTH**	**NEW_RXFIFO_TH**	**DMA_SIZE (half word unit)**
0x0 (byte)	0x1 (burst=4)	N (byte)	0x2 (word) 0x1 (half word) 0x0 (byte)	0x3	N
0x0 (byte)	0x0 (burst=1)	N (byte)	0x0 (byte)	0x0	N

14.8 TIMING CHARACTERISTICS



14.8.1 MASTER MODE

Symbol	Parameter	Min	Typ	Max	Unit
t_{sck}	SCK out period	33			ns
t_{sck}(H)	SCK out high pulse	t_{sck}*0.4			ns
t_{sck}(L)	SCK out low pulse	t_{sck}*0.4			ns
t_{SU}(CS)	CS out setup time	t_{sck}*2			ns
t_{H}(CS)	CS out hold time	t_{sck}*1.5			ns
t_{SU}(MISO)	Data in setup time	t_{sck}*0.4			ns
t_{H}(MISO)	Data in hold time	t_{sck}*0.4			ns
t_{SU}(MOSI)	Data out setup time	t_{sck}*0.4			ns
t_{H}(MOSI)	Data out hold time	t_{sck}*0.4			ns

14.8.2 SLAVE MODE

Symbol	Parameter	Min	Typ	Max	Unit
t_{sck}	SCK in period	33			ns
t_{sck}(H)	SCK in high pulse	t_{sck}*0.4			ns
t_{sck}(L)	SCK in low pulse	t_{sck}*0.4			ns
t_{SU}(CS)	CS in setup time	t_{sck}*1.5			ns
t_{H}(CS)	CS in hold time	t_{sck}			ns
t_{SU}(MISO)	Data out setup time	t_{sck}*0.4			ns
t_{H}(MISO)	Data out hold time	t_{sck}*0.4			ns
t_{SU}(MOSI)	Data in setup time	t_{sck}*0.4			ns
t_{H}(MOSI)	Data in hold time	t_{sck}*0.4			ns

14.9 SPI REGISTERS

Base Address: 0x4001 C000 (SPI0)

14.9.1 SPI n Control register 0 (SPIn_CTRL0) (n=0)

Address Offset:0x00

✷ ***Note:***
1. Must reset SPI FSM with FRESET[1:0] after changing any configuration of SPI when SPIEN = 1.
2. HW will switch I/O configurations refer to FORMAT bit directly when SPIEN = 1.
3. Must set SELDIS = 0 when it acts as slave.

Bit	Name	Description	Attribute	Reset
31:19	Reserved		R	0
18	SELDIS	Auto-SEL disable bit. 0: Enable Auto-SEL flow control. 1: Disable Auto-SEL flow control.	R/W	1
17:15	RXFIFOTH[2:0]	RX FIFO Threshold level 000: RX FIFO threshold level = 0 001: RX FIFO threshold level = 1 … … 111: RX FIFO threshold level = 7	R/W	000b
14:12	TXFIFOTH[2:0]	TX FIFO Threshold level 000: TX FIFO threshold level = 0 001: TX FIFO threshold level = 1 … … 111: TX FIFO threshold level = 7	R/W	000b
11:8	DL[3:0]	Data length = DL[3:0] + 1 0000~0001: Reversed 0010: data length = 3 … … 1110: data length = 15 1111: data length = 16	R/W	1111b
7:6	FRESET[1:0]	SPI FSM and FIFO Reset bit 00: No effect 01: Reserved 10: Reserved 11: Reset finite state machine and FIFO. (BUF_BUSY = 0, data in shift BUF is cleared, TX_EMPTY = 1, TX_FULL = 0, RX_EMPTY = 1, RX_FULL = 0, and data in FIFO is cleared). This bit will be cleared by HW automatically.	W	0
5	Reserved		R	0
4	FORMAT	Interface format. 0: SPI 1: Reserved	R/W	0
3	MS	Master/Slave selection bit 0: Act as Master. 1: Act as Slave.	R/W	0
2	SDODIS	Slave data output disable bit (ONLY used in slave mode) 0: Enable slave data output. 1: Disable slave data output. (MISO=0)	R/W	0
1	LOOPBACK	Loop back mode enable 0: Disable 1: Data input from data output	R/W	0
0	SPIEN	SPI enable bit 0: Disable 1: Enable.	R/W	0

14.9.2 SPI n Control register 1 (SPIn_CTRL1) (n=0)

Address Offset: 0x04

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	CPHA	Clock phase for edge sampling. 0: Data changes at clock falling edge, latches at clock rising edge when CPOL = 0; Data changes at clock rising edge, latches at clock falling edge when CPOL = 1. 1: Data changes at clock rising edge, latches at clock falling edge when CPOL = 0; Data changes at clock falling edge, latches at clock rising edge when CPOL = 1.	R/W	0
1	CPOL	Clock polarity selection bit 0: SCK idles at Low level. 1: SCK idles at High level.	R/W	0
0	MLSB	MSB/LSB selection bit 0: MSB transmit first. 1: LSB transmit first.	R/W	0

14.9.3 SPI n Clock Divider register (SPIn_CLKDIV) (n=0)

Address Offset: 0x08

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DIV[7:0]	SPIn clock divider 0: SCK = SPIn_PCLK / 2 1: SCK = SPIn_PCLK / 4 2: SCK = SPIn_PCLK / 6 X: SCK = SPIn_PCLK / (2X+2)	R/W	0

14.9.4 SPI n Status register (SPIn_STAT) (n=0)

Address Offset: 0x0C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6	RXFIFOTHF	RX FIFO threshold flag 0: Data in RX FIFO ≤ RXFIFOTH 1: Data in RX FIFO > RXFIFOTH	R	0
5	TXFIFOTHF	TX FIFO threshold flag 0: Data in TX FIFO > TXFIFOTH 1: Data in TX FIFO ≤ TXFIFOTH	R	1
4	BUSY	Busy flag. 0: SPI controller is idle. 1: SPI controller is transferring.	R	0
3	RX_FULL	RX FIFO full flag. 0: RX FIFO is NOT full. 1: RX FIFO is full.	R	0
2	RX_EMPTY	RX FIFO empty flag 0: RX FIFO is NOT empty. 1: RX FIFO is empty.	R	1
1	TX_FULL	TX FIFO full flag. 0: TX FIFO is NOT full. 1: TX FIFO is full.	R	0
0	TX_EMPTY	TX FIFO empty flag 0: TX FIFO is NOT empty. In Master mode, the transmitter will begin to transmit automatically. 1: TX FIFO is empty.	R	1

14.9.5 SPI n Interrupt Enable register (SPIn_IE) (n=0)

Address Offset: 0x10

This register controls whether each of the four possible interrupt conditions in the SPI controller is enabled.

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
5	TXEMPIE	TX all empty interrupt enable (TX all empty means TX FIFO empty and SPI is not busy) 0: Disable 1: Enable	R/W	0
4	TXUDFIE	TX underflow interrupt enable (TX underflow means TX FIFO is empty and data is read from TX FIFO) 0: Disable 1: Enable	R/W	0
3	TXFIFOTHIE	TX FIFO threshold interrupt enable 0: Disable 1: Enable	R/W	0
2	RXFIFOTHIE	RX FIFO threshold interrupt enable 0: Disable 1: Enable	R/W	0
1	RXTOIE	RX time-out interrupt enable 0: Disable 1: Enable	R/W	0
0	RXOVFIE	RX Overflow interrupt enable 0: Disable 1: Enable	R/W	0

14.9.6 SPI n Raw Interrupt Status register (SPIn_RIS) (n=0)

Address Offset: 0x14

This register contains the status for each interrupt condition, regardless of whether or not the interrupt is enabled in SPIn_IE register.

This register indicates the status for SPI control raw interrupts. An SPI interrupt is sent to the interrupt controller if the corresponding bit in the SPIn_IE register is set.

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
5	TXEMPIF	TX all empty interrupt flag 0: No TX all empty interrupt 1: TX all empty triggered.	R	0
4	TXUDFIF	TX underflow interrupt flag 0: No TX underflow interrupt 1: TX underflow triggered.	R	0
3	TXFIFOTHIF	TX FIFO threshold interrupt flag 0: No TX FIFO threshold interrupt 1: TX FIFO threshold triggered.	R	0
2	RXFIFOTHIF	RX FIFO threshold interrupt flag 0: No RX FIFO threshold interrupt 1: RX FIFO threshold triggered.	R	0
1	RXTOIF	RX time-out interrupt flag RXTO occurs when the RX FIFO is not empty, and has not been read for a time-out period (32*SPIn_PCLK). The time-out period is the same for master and slave modes. 0: RXTO doesn't occur. 1: RXTO occurs.	R	0
0	RXOVFIF	RX Overflow interrupt flag RXOVF occurs when the RX FIFO is full and another frame is completely received. The ARM spec implies that the preceding frame data is overwritten by the new frame data when this occurs. 0: RXOVF doesn't occur. 1: RXOVF occurs.	R	0

14.9.7 SPI n Interrupt Clear register (SPIn_IC) (n=0)

Address Offset: 0x18

Bit	Name	Description	Attribute	Reset
31:6	Reserved		R	0
5	TXEMPIC	0: No effect 1: Clear TXEMPIF bit	W	0
4	TXUDFIC	0: No effect 1: Clear TXUDFIF bit	W	0
3	TXFIFOTHIC	0: No effect 1: Clear TXFIFOTHIF bit	W	0
2	RXFIFOTHIC	0: No effect 1: Clear RXFIFOTHIF bit	W	0
1	RXTOIC	0: No effect 1: Clear RXTOIF bit.	W	0
0	RXOVFIC	0: No effect 1: Clear RXOVFIF bit.	W	0

14.9.8 SPI n Data register (SPIn_DATA) (n=0)

Address Offset: 0x1C

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	DATA[15:0]	Write SW can write data to be sent in a future frame to this register when TX_FULL = 0 in SPIn_STAT register (TX FIFO is not full). If the TX FIFO was previously empty and the SPI controller is not busy on the bus, transmission of the data will begin immediately. Otherwise the data written to this register will be sent as soon as all previous data has been sent (and received). Read SW can read data from this register when RX_EMPTY=0 in SPIn_STAT register (Rx FIFO is not empty). When SW reads this register, the SPI controller returns data from the least recent frame in the RX FIFO. If the data length is less than 16 bit, the data is right-justified in this field with higher order bits filled with 0s.	R/W	0

14.9.9 SPI n Data Fetch register (SPIn_DF) (n=0)

Address Offset: 0x20

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	DF	SPI data fetch control bit 0: Disable 1: Enable when SCKn frequency > 6MHz	R/W	0

14.9.10 SPI n FIFO threshold register (SPIn_FIFO_TH) (n=0)

Address Offset: 0x24

Bit	Name	Description	Attribute	Reset
31	NEW_TH_EN	New TX/RX threshold setting enable 0: TX/RX threshold setting is from TX/RXFIFOTH[2:0] in SPIn_CTRL0. FIFO size is 8x16 bits FIFO. 1: TX/RX threshold setting is from NEW_TX/RXFIFO_TH[3:0] in SPIn_FIFO_TH. FIFO size is 16x8 bits FIFO.	R/W	0
30:20	Reserved		R	0

19:16	NEW_RXFIFO_TH[3:0]	RX FIFO Threshold level 0000: RX FIFO threshold level = 0 0001: RX FIFO threshold level = 1 … … 1111: RX FIFO threshold level = 15	R/W	0000b
15:4	Reserved		R	0
3:0	NEW_TXFIFO_TH[3:0]	TX FIFO Threshold level 0000: TX FIFO threshold level = 0 0001: TX FIFO threshold level = 1 … … 1111: TX FIFO threshold level = 15	R/W	0000b

14.9.11 SPI n DMA Mode register (SPIn_DMA) (n=0)

Address Offset: 0x50

Bit	Name	Description	Attribute	Reset
31	TX_DMA_EN	TX DMA enable 0: Disable 1: Enable	R/W	0
30	RX_DMA_EN	RX DMA enable 0: Disable 1: Enable	R/W	0
29:22	Reserved		R	0
21:0	DMA_SIZE	Total DMA transfer size	R/W	0

14.9.12 SPI n ARGB CTRL register (SPIn_ARGBCTRL) (n=0)

Address Offset: 0x58

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	ARGBM	SPIn ARGB mode 0: Disable 1: Enable. SCKn, SELn and MISOn share pins are GPIO.	R/W	0

15 I2C

15.1 OVERVIEW

The I2C bus is bidirectional for inter-IC control using only two wires: Serial Clock Line (SCL) and Serial Data line (SDA). Each device is recognized by a unique address and can operate as either a receiver-only device (e.g., an LCD driver) or a transmitter with the capability to both receive and send information (such as memory). Transmitters and/or receivers can operate in either master or slave mode, depending on whether the chip has to initiate a data transfer or is only addressed. The I2C is a multi-master bus and can be controlled by more than one bus master connected to it. It is also SMBus 2.0 compatible.

The I2C interface is byte oriented and has four operating modes:

- Master transmitter mode
- Master receiver mode
- Slave transmitter mode
- Slave receiver mode

15.2 FEATURES

The I2C interface complies with the entire I2C specification, supporting the ability to turn power off to the ARM Cortex-M0 without interfering with other devices on the same I2C-bus.

- Standard I2C-compliant bus interfaces may be configured as Master or Slave.
- I2C Master features:
  - Clock generation
  - Start and Stop generation
- I2C Slave features:
  - Programmable I2C Address detection
  - Optional recognition of up to four distinct slave addresses
  - Stop bit detection
- Supports different communication speeds:
  - Standard Speed (up to 100KHz)
  - Fast Speed (up to 400 KHz)
- Arbitration is handled between simultaneously transmitting masters without corruption of serial data on the bus.
- Programmable clock allows adjustment of I2C transfer rates.
- Data transfer is bidirectional between masters and slaves.
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus.
- Serial clock synchronization is used as a handshake mechanism to suspend and resume serial transfer.
- I2C-bus can be used for test and diagnostic purposes.
- Generation and detection of 7-bit/10-bit addressing and General Call.
- Support DMA transfer

15.3 PIN DESCRIPTION

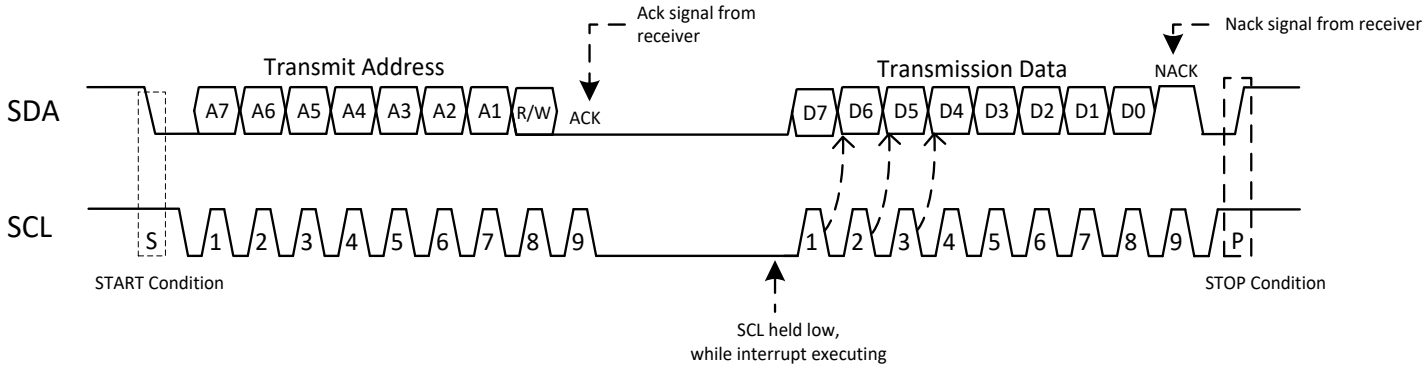
Pin Name	Type	Description	GPIO Configuration
SCLn	I/O	I2C Serial clock	Output with Open-drain Input depends on GPIOn_CFG
SDAn	I/O	I2C Serial data	Output with Open-drain Input depends on GPIOn_CFG

15.4 I2C PROTOCOL

I2C transmission structure includes a START(S) condition, 8-bit address byte, one or more data byte and a STOP (P) condition. START condition is generated by master to initial any transmission.

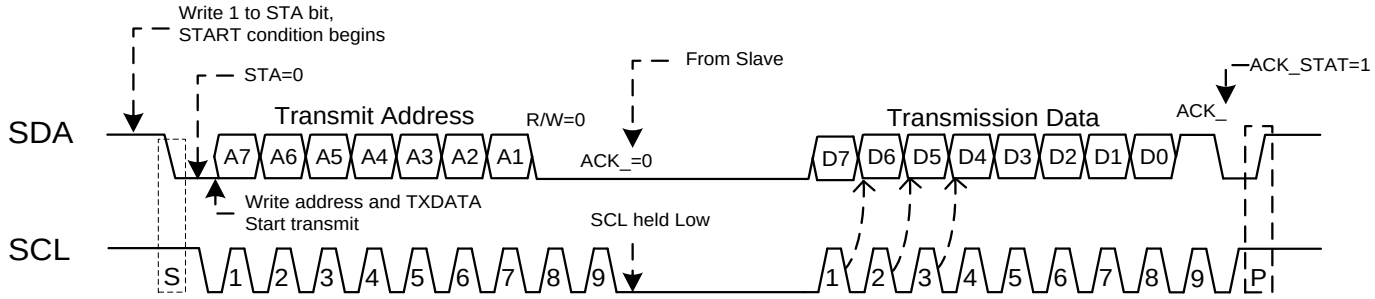
Data is transmitted with the Most Significant Bit (MSB) first. In address byte, the higher 7-bit is address bit and the lowest bit is data direction (R/W) bit. When R/W=0, it assigns a “WRITE” operation. When R/W=1, it assigns a “READ” operation.

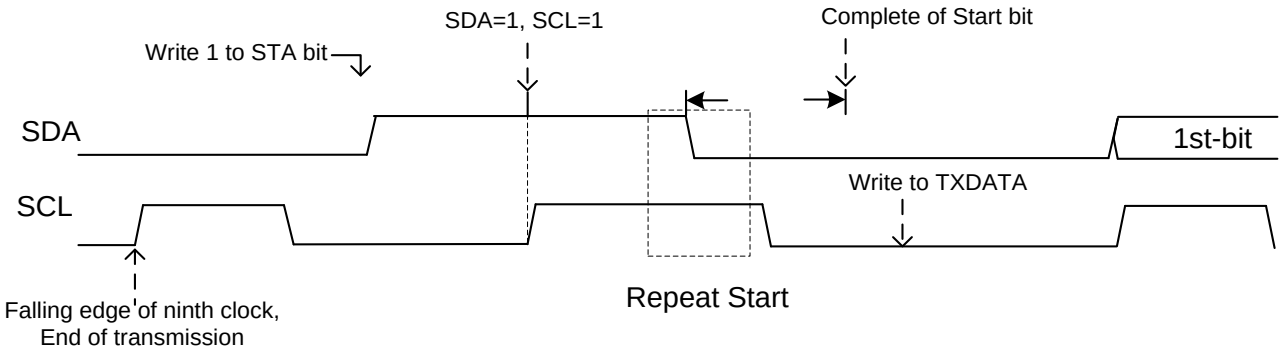
After each byte is received, the receiver (a master or a slave) must send an acknowledge (ACK) bit. If transmitter can’t receive an ACK, it will recognize a not acknowledge (NACK). In WRITE operation, the master will transmit data to the slave and then waits for ACK from slave. In READ operation, the slave will transmit data to the master and then waits for ACK from master. In the end, the master will generate a STOP condition to finish transmission.



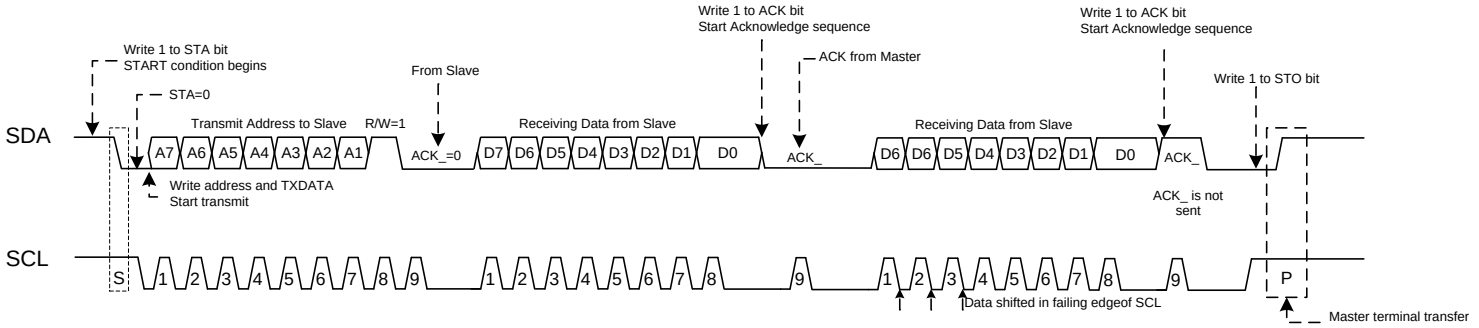
15.4.1 7-BIT ADDRESSING MODES

15.4.1.1 MASTER TRANSMITTER MODE

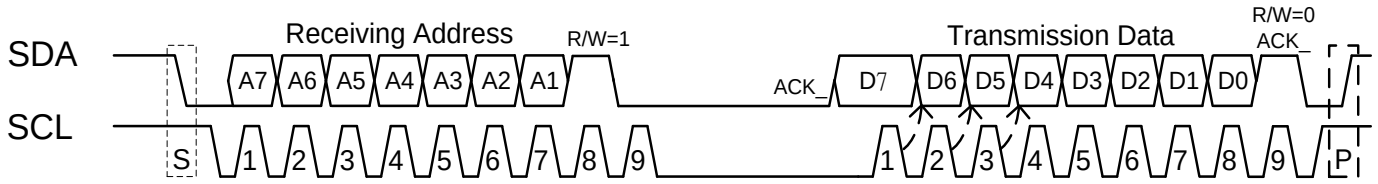




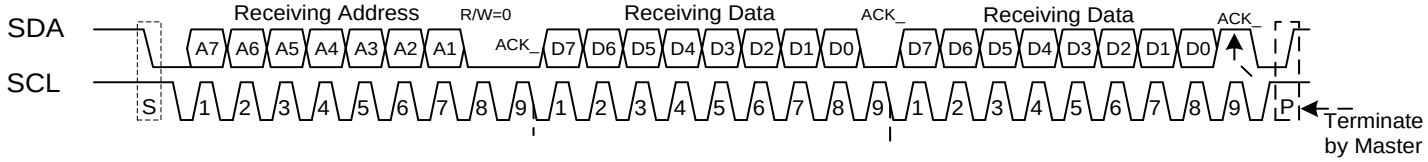
15.4.1.2 MASTER RECEIVER MODE



15.4.1.3 SLAVE TRANSMITTER MODE

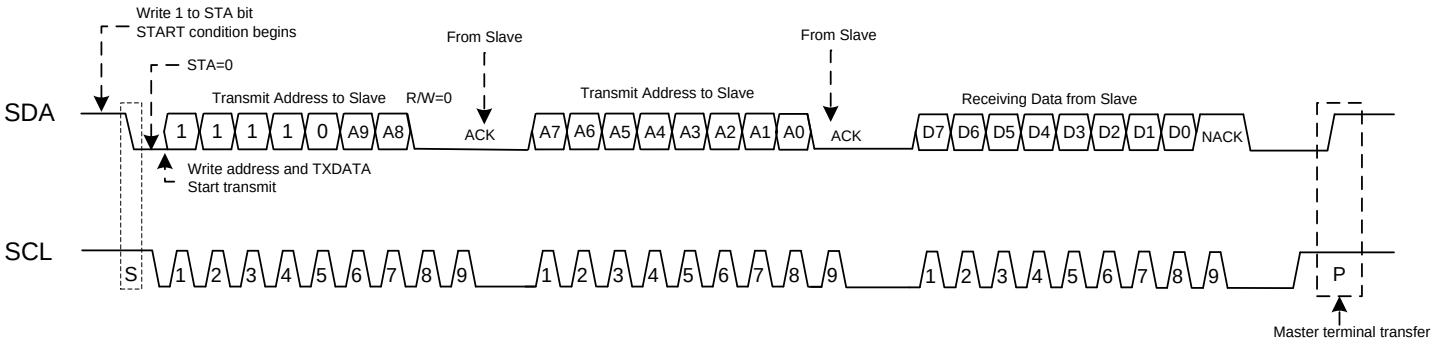


15.4.1.4 SLAVE RECEIVER MODE

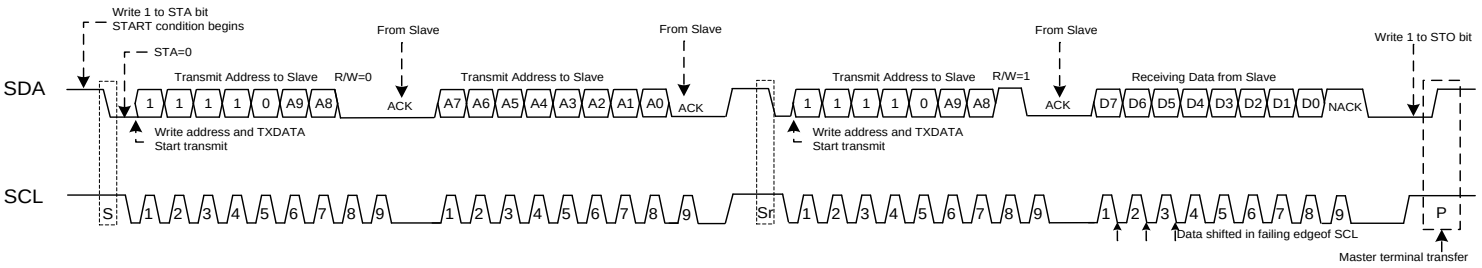


15.4.2 10-BIT ADDRESSING MODES

15.4.2.1 MASTER TRANSMITTER MODE



15.4.2.2 MASTER RECEIVER MODE



15.5 ARBITRATION

In multi-master condition, more than one master may transmit on bus in the same time. It must be decided which master has the control of bus and complete its transmission. Clock synchronization and arbitration are used to configure multi-master transmission.

Clock synchronization is executed by synchronizing the SCL signal with anther devices. When two masters want to transmit data in the same time, the clock synchronization will start by the High to Low transition on the SCL. If master 1 pulls the SCL line LOW first, it holds the SCL in LOW status until the SCL line is released to HIGH status. However, if anther master still pulls the SCL line LOW, the SCL Low to High transition of master 1 may not change SCL status (SCL line is still LOW). The SCL will transit from LOW to HIGH when the all masters release the SCL line. In the duration, the master1 will wait for SCL transition from LOW to HIGH, and then continue its transmission.

After clock synchronization, the clock of all devices is synchronized with the SCL clock. Arbitration is used to decide which master can complete its transmission by SDA signal. Two masters may send out a START condition and transmit data on bus in the same time, and may be influenced by each other. Arbitration will force one master to lose the control on bus. Data transmission will keep until two masters output different data signal. If one master transmits HIGH status and anther master transmits LOW status, the SDA will be pulled low. The master which pulls the SDA line High will detect the different with SDA and loses the control on bus. The mater which pulls the SDA line LOW status wins the bus control and continues its transmission. There is no data miss during arbitration.

Arbitration may also be lost in the master receiver mode. Loss of arbitration in this mode can only occur while the I2C block is returning a “not acknowledge” to the bus. Arbitration is lost when another device on the bus pulls this signal low. Since this can occur only at the end of a serial byte, the I2C block generates no further clock pulses.

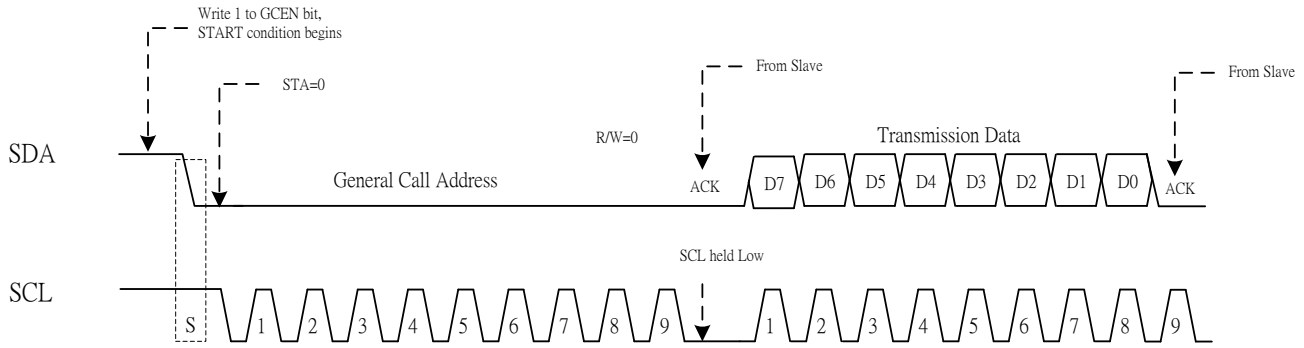
15.6 CLOCK STRETCHING

Clock stretching pauses a transaction by pulling the SCL line LOW. The transaction cannot continue until the line is released HIGH again. Clock stretching is optional.

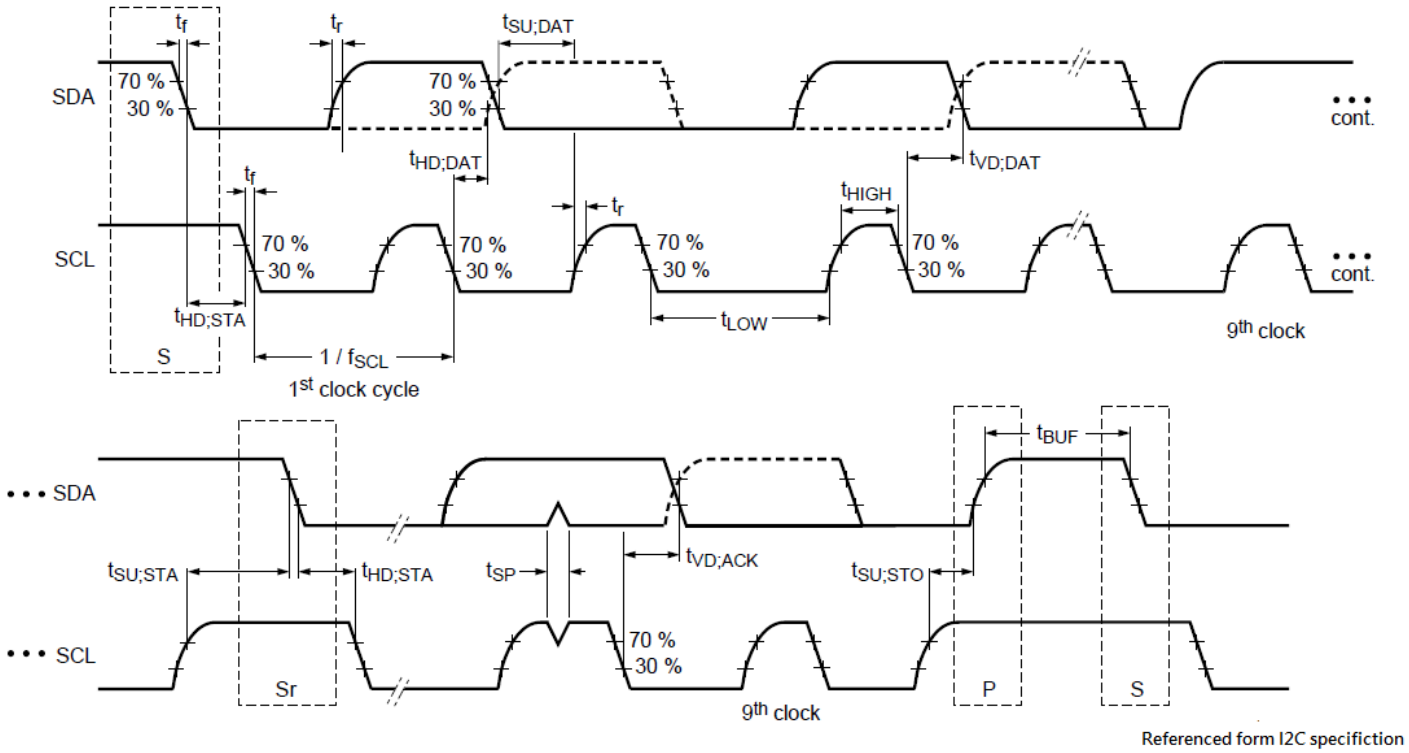
On the byte level, a device may be able to receive bytes of data at a fast rate, but needs more time to store a received byte or prepare another byte to be transmitted. Slaves can then pulls the SCL line LOW after reception and acknowledgment of a byte to force the master into a wait state until the slave is ready for the next byte transfer in a type of handshake procedure.

15.7 GENERAL CALL ADDRESS

The general call address is a special address which is reserved as all “0” of 7-bit address and is for addressing every device connected to the I2C-bus at the same time. However, if a device does not need any of the data supplied within the general call structure, it can ignore this address by not issuing an acknowledgment (ACK). If a device does require data from a general call address, it acknowledges this address and behaves as a slave-receiver. The master does not actually know how many devices acknowledged if one or more devices respond. The second and following bytes are acknowledged by every slave-receiver capable of handling this data. A slave who cannot process one of these bytes must ignore it by not-acknowledging. If one or more slaves acknowledge, the not-acknowledge will not be seen by the master. The meaning of the general call address is specified in the second byte



15.8 TIMING CHARACTERISTICS



Referenced form I2C specifiction

15.8.1 MASTER TRANSMITTER MODE

t_{HIGH}: (SCLHT + 1) * I2C_PCLK cycle

t_{LOW}: (SCLLT + 1) * I2C_PCLK cycle

$t_{HD;STA}$: (SCLLT + 4) * I2C_PCLK cycle

$t_{HD;DAT}$: 2 * I2C_PCLK cycle ~ 3 * I2C_PCLK cycle

$t_{SU;STA}$: (SCLL + 3) * I2C_PCLK cycle

$t_{SU;STO}$: (SCLLT + 2) * I2C_PCLK cycle ~ (SCLLT + 3) * I2C_PCLK cycle

15.8.2 SLAVE TRANSMITTER MODE

t_{HIGH}: controlled by Master

t_{LOW}: controlled by Master

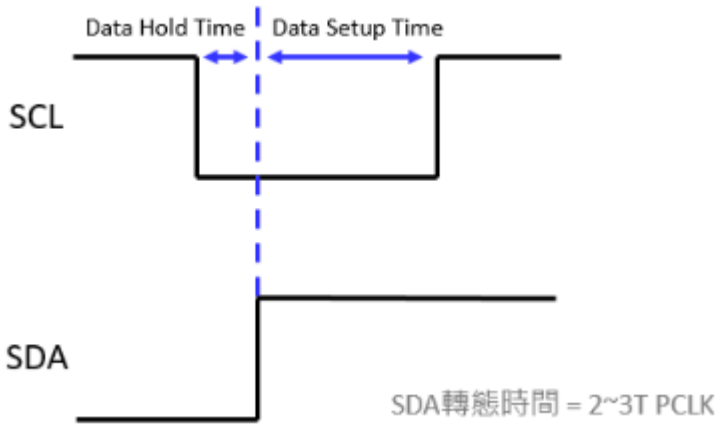
$t_{HD;DAT}$: 2 * I2C_PCLK cycle ~ 3 * I2C_PCLK cycle

$t_{SU;DAT}$: t_{LOW} - 2 * I2C_PCLK cycle ~ t_{LOW} - 3 * I2C_PCLK cycle

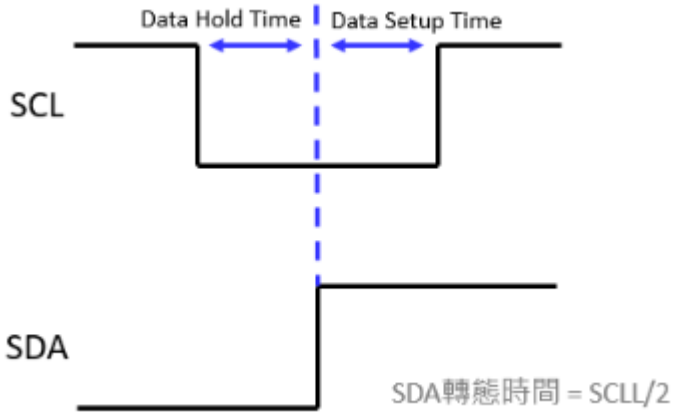
15.8.3 I2C TX_BIT Changing Timing

I2C TX_BIT chaging timing selected by SCLL_HF_SEL bit.

- SCLL_HF_SEL = 0 : Change at SCL negative edge.

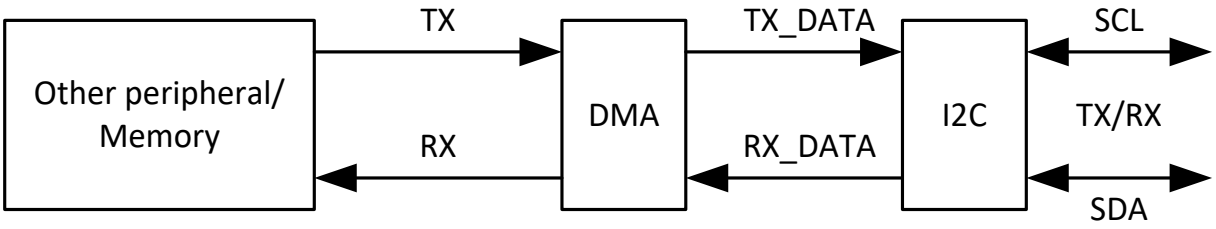


- SCLL_HF_SEL = 1 : Change at the half of SCL low edge.



15.9 DMA Mode

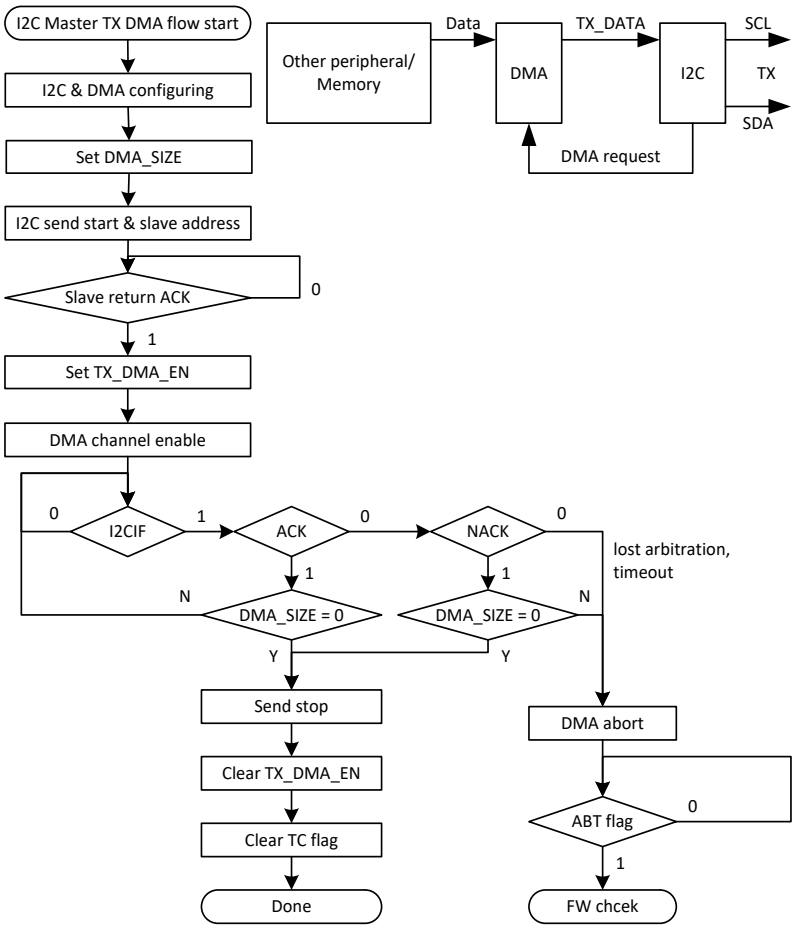
The I2C DMA mode is to use DMA engine to move data in/out I2C. Before the DMA transfer start, DMA engine must be set up first. In I2C DMA TX Mode, DMA gets data from other peripherals or memories to I2C as TX data. In I2C DMA RX Mode, DMA receive data from I2C and send to other peripherals or memories.

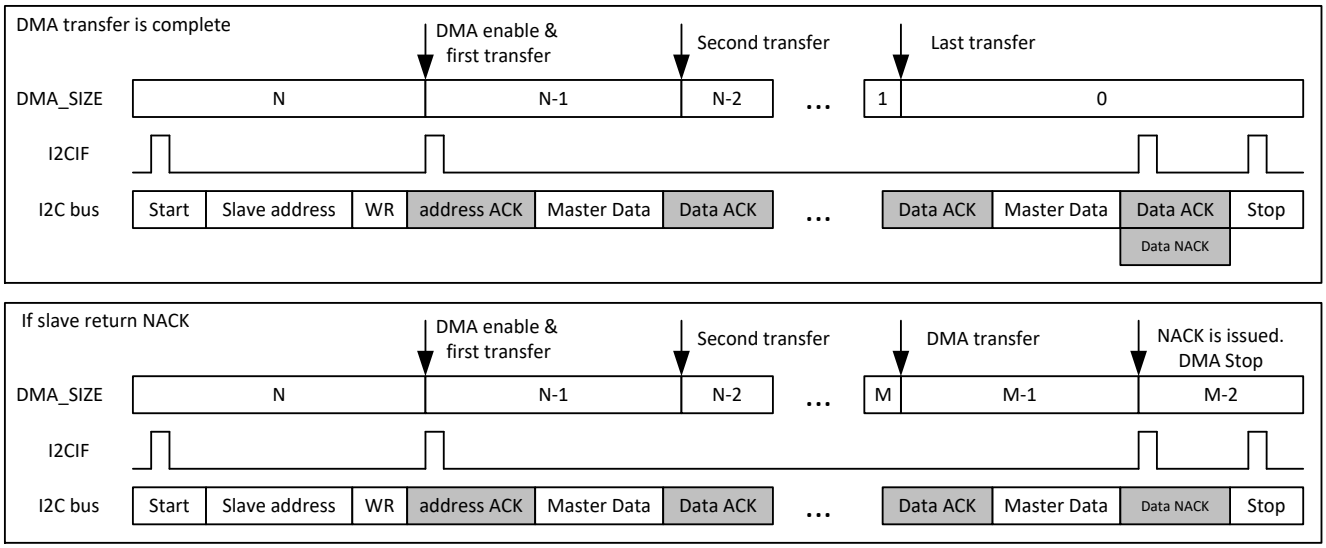


✸ ***Note:***
1. ***The burst size must be set to 1, because I2C hasn't FIFO.***
2. ***If I2C is TX mode, the DMA channel source width must be set to 0x0 (byte).***

15.9.1 I2C DMA Master TX Mode

The following register programming flow is used for I2C DMA master TX mode. This example will turn on DMA Channel to move data from some peripherals or memory to I2C TX data register. When TX FIFO is less than the threshold, DMA request is issued. If the RX terminal responds with NACK, I2CIF will be issued. When NACK is received, firmware must be abort the DMA operation. It is recommended to use I2C interrupt to avoid unexpected events.



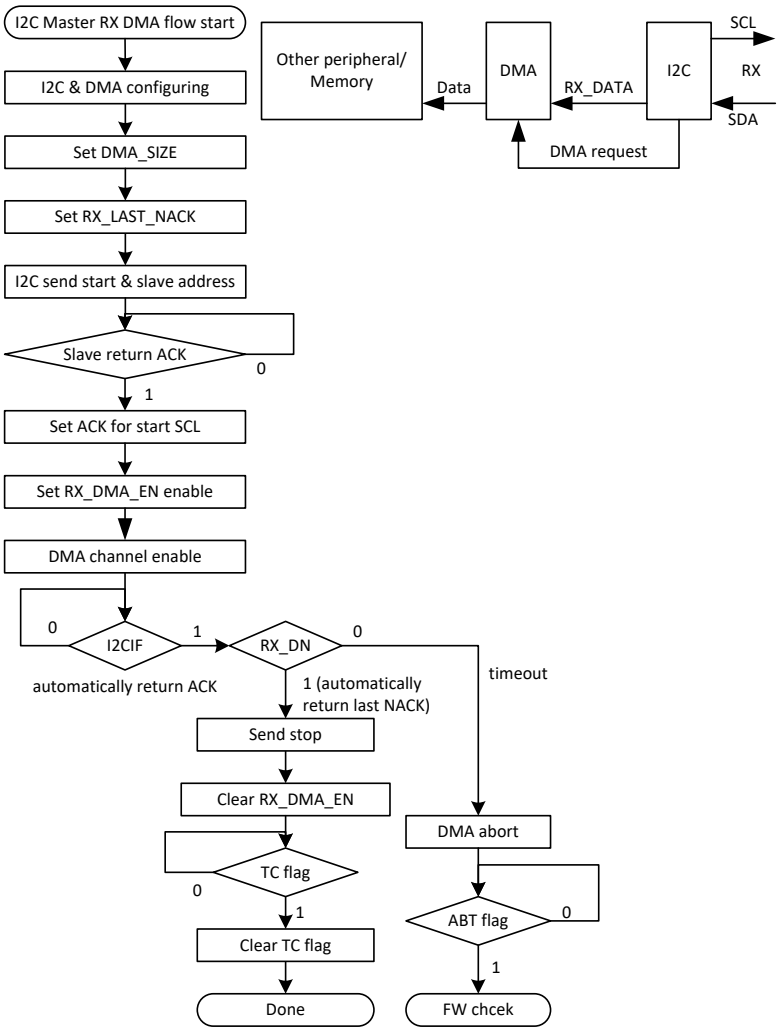


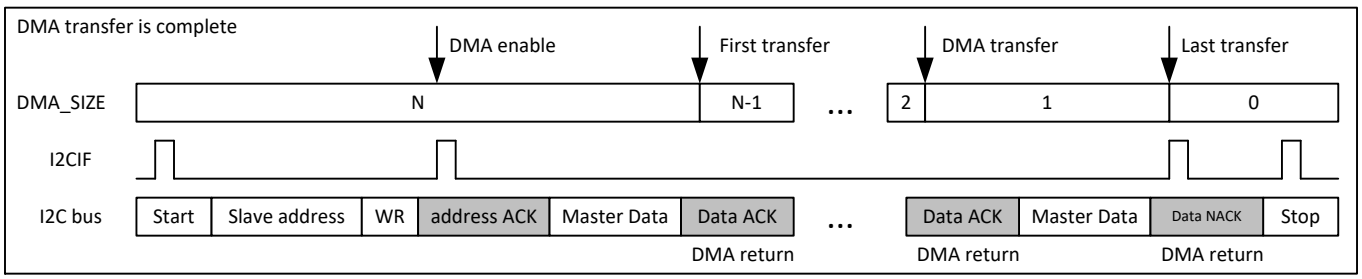
✸ ***Note:***

1. ***I2C interrupts will not be issued during DMA transfers until the last transfer is completed or a NACK is received.***
2. ***When an abnormal event occurs, an I2C interrupt will be issued, and the DMA operation needs to be aborted at this time.***

15.9.2 I2C DMA Master RX Mode

The following register programming flow chart used for I2C DMA master RX mode. This example will turn on DMA Channel to move data from I2C RX data register to other peripherals SRAM. When RX FIFO is greater than the threshold, DMA request is issued. During DMA transfer, I2C will automatically return ACK to the TX terminal until the transfer is completed.



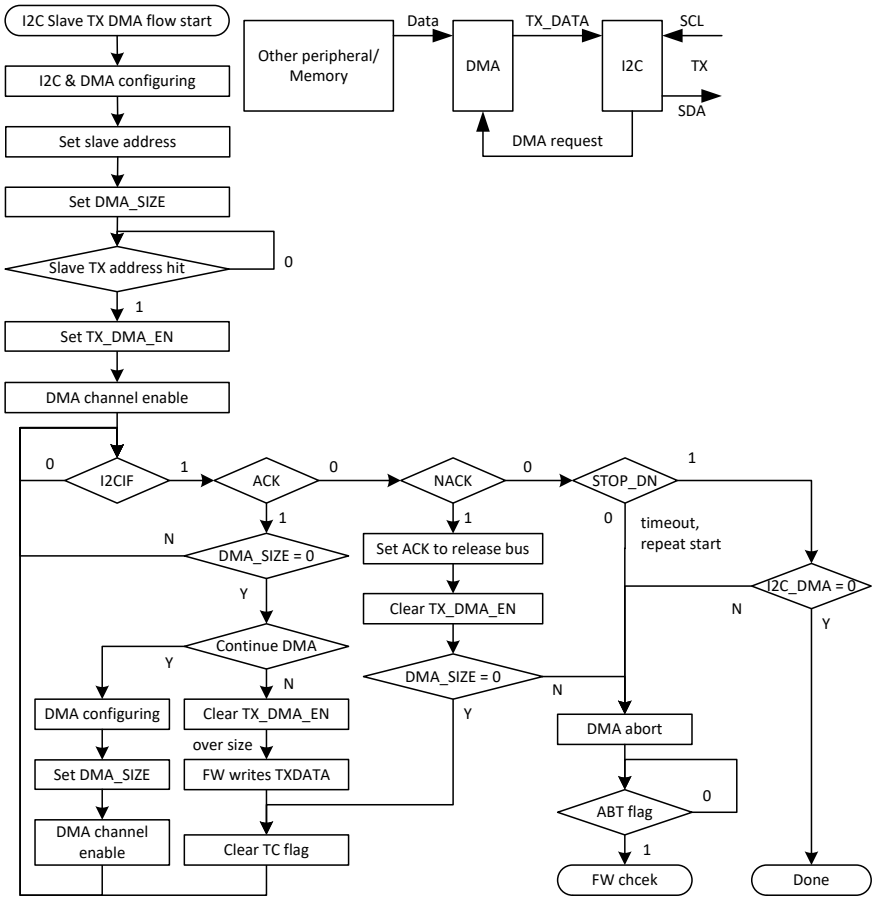


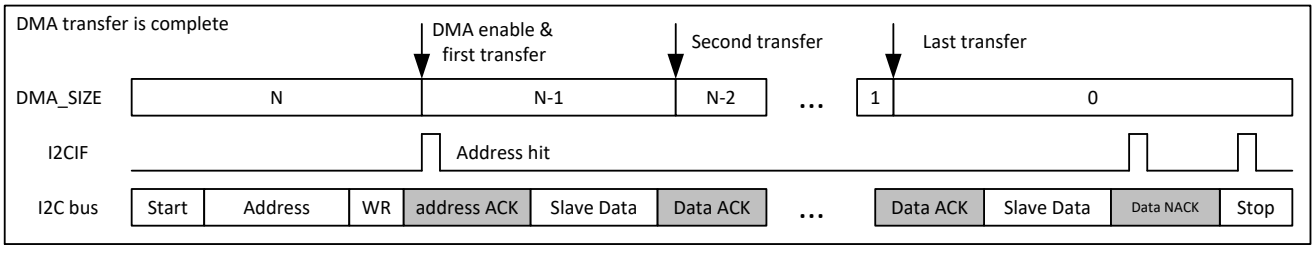
✸ ***Note:***

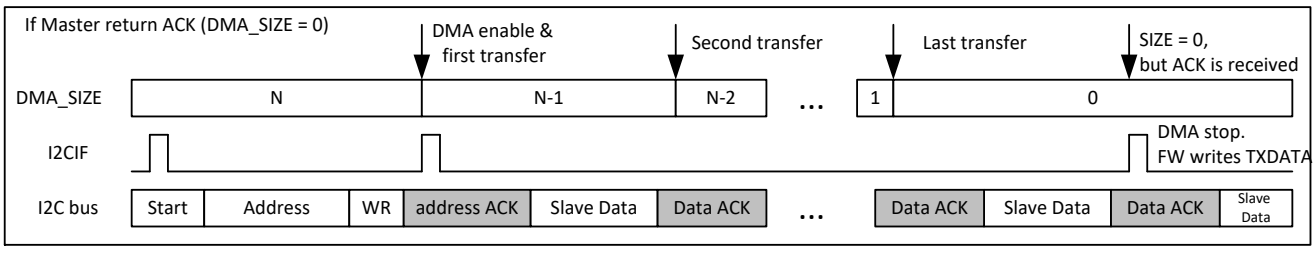
1. ***I2C interrupts will not be issued during DMA transfers until the last transfer is completed.***
2. ***When an abnormal event occurs, an I2C interrupt will be issued, and the DMA operation needs to be aborted at this time.***

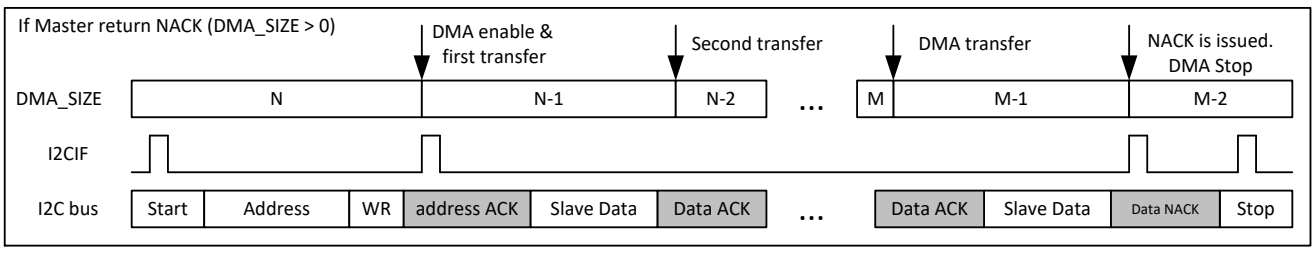
15.9.3 I2C DMA Slave TX Mode

The following register programming flow is used for I2C DMA slave TX mode. This example will turn on DMA Channel to move data from some peripherals or memory to I2C TX data register. When TX FIFO is less than the threshold, DMA request is issued. If the RX terminal responds with NACK, I2CIF will be issued. When NACK is received, firmware must be abort the DMA operation. It is recommended to use I2C interrupt to avoid unexpected events.





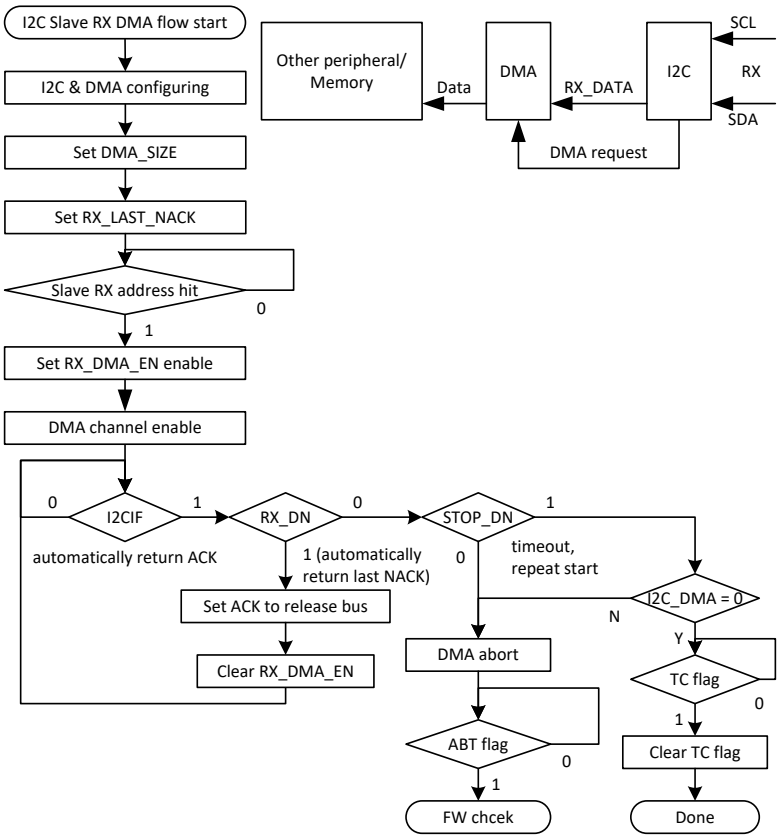


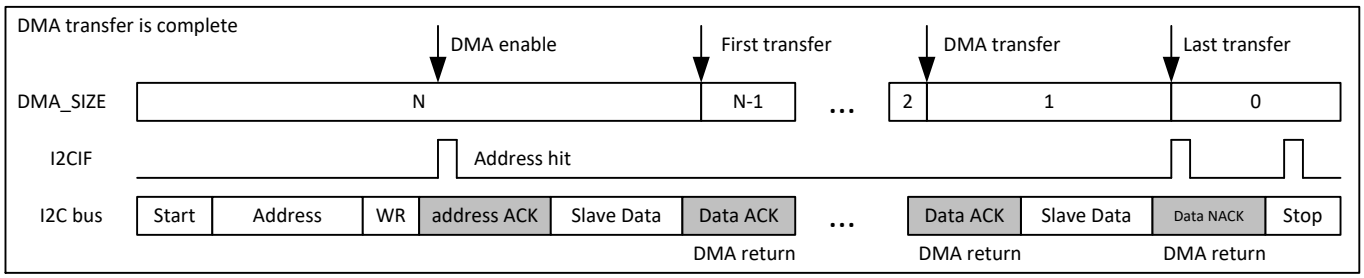


✸ ***Note:***
1. ***I2C interrupts will not be issued during DMA transfers until the last transfer is completed or a NACK is received.***
2. ***When an abnormal event occurs, an I2C interrupt will be issued, and the DMA operation needs to be aborted at this time.***

15.9.4 I2C DMA Slave RX Mode

The following register programming flow chart used for I2C DMA slave RX mode. This example will turn on DMA Channel to move data from I2C RX data register to other peripherals SRAM. When RX FIFO is greater than the threshold, DMA request is issued. During DMA transfer, I2C will automatically return ACK to the TX terminal until the transfer is completed. It is recommended to use I2C interrupt to avoid unexpected events.





✸ ***Note:***
1. ***I2C interrupts will not be issued during DMA transfers until the last transfer is completed.***
2. ***When an abnormal event occurs, an I2C interrupt will be issued, and the DMA operation needs to be aborted at this time.***

15.9.5 I2C DMA Configuration Recommendations

15.9.5.1 I2C TX DMA

Configuration	Register
DSTADDR = I2C_TXDATA	DMA_Cn_DSTADDR
DST_RS = I2Cn DST_HE=1	DMA_Cn_CFG
DSTAD_CTL=10 (Fixed) MODE = 1 (Peripheral)	DMA_Cn_CSR
TX_DMA_EN = 1	I2C_DMA

DMAC setting for I2C TX				Recommend setting for I2C DMA register
Cn_CSR **SRC_WIDTH**	Cn_CSR **SRC_SIZE**	Cn_SIZE **TOT_SIZE**	Cn_CSR **DST_WIDTH**	DMA_SIZE(byte unit)
0x0 (byte)	0x0 (burst=1)	N (byte)	0x0 (byte)	N

15.9.5.2 I2C RX DMA

Configuration	Register
SRCADDR = I2C_RXDATA	DMA_Cn_SRCADDR
SRC_RS = I2Cn SRC_HE=1	DMA_Cn_CFG
SRCAD_CTL=10 (Fixed) MODE = 1 (Peripheral)	DMA_Cn_CSR
RX_DMA_EN = 1 RX_DMA_LAST_NACK = 0 or 1	I2C_DMA

DMAC setting for I2C RX				Recommend setting for I2C DMA register
Cn_CSR **SRC_WIDTH**	Cn_CSR **SRC_SIZE**	Cn_SIZE **TOT_SIZE**	Cn_CSR **DST_WIDTH**	DMA_SIZE(byte unit)
0x0 (byte)	0x0 (burst=1)	N (byte)	0x0 (byte)	N

15.10 I2C REGISTERS

Base Address: 0x4001 8000 (I2C0)

15.10.1 I2C n Control register (I2Cn_CTRL) (n=0)

Address Offset: 0x00

Setting of the bits in this register controls operation of the I2C interface.

When STA =1 and the I2C interface is not already in master mode, it enters master mode, checks the bus and generates a START condition if the bus is free. If the bus is not free, it waits for a STOP condition (which will free the bus) and generates a START condition after a delay of a half clock period of the internal clock generator. If the I2C interface is already in master mode and data has been transmitted or received, it transmits a Repeated START condition. STA may be set at any time, including when the I2C interface is in an addressed slave mode.

When STO = 1 in master mode, a STOP condition is transmitted on the I2C bus. When the bus detects the STOP condition, STO is cleared automatically. In slave mode, setting STO bit can recover from an error condition. In this case, no STOP condition is transmitted to the bus. The HW behaves as if a STOP condition has been received and it switches to “not addressed” slave receiver mode.

If STA and STO are both set, then a STOP condition is transmitted on the I2C bus if it the interface is in master mode, and transmits a START condition thereafter. If the I2C interface is in slave mode, an internal STOP condition is generated, but is not transmitted on the bus.

✷ ***Note:***

1. ***I2CEN shall be set at last.***
2. ***HW will assign SCL0 and SDA0 pins as output pins with open-drain function instead of GPIO automatically.***
3. ***ACK and NACK bits can’t both be “1” when receiving data.***
4. ***User has to write 1 to ACK or NACK bit in Master mode to continue next RX process.***

Bit	Name	Description	Attribute	Reset
31:22	Reserved		R	0
21	ARBITRATION	Master Arbitration enable bit (NEW) 0: Disable 1: Enable	R/W	1
20	SCLL_HF_SEL	I2C TX_BIT changing timing bit (NEW) 0: Change at SCL negative edge 1: Change at the half of SCL low edge	R/W	1
19:9	Reserved		R	0
8	I2CEN	I2C Interface enable bit 0: Disable. The STO bit is forced to “0”. 1: Enable. I2EN shall not be used to temporarily release the I2C bus since the bus status is lost when I2CEN resets. The ACK flag should be used instead.	R/W	0
7:6	Reserved		R	0
5	STA	START bit. 0: No START condition or Repeated START condition will be generated. 1: Cause the I2C interface to enter master mode and transmit a START or a Repeated START condition. Automatically cleared by HW.	R/W	0
4	STO	STOP flag 0: Stop condition idle. 1: Cause the I2C interface to transmit a STOP condition in master mode, or recover from an error condition in slave mode. Automatically cleared by HW.	R/W	0
3	Reserved		R	0
2	ACK	Assert ACK flag. 0: Master mode→ No function Slave mode→Return a NACK after receiving address or data.	R/W	0

		1: An ACK will be returned during the acknowledge clock pulse on SCLn when ➢ The address in the Slave Address register has been received. ➢ The General Call address has been received while the General Call enable bit (GCEN) in SLVADDR0 register is set. ➢ A data byte has been received while the I2C is in the master receiver mode. ➢ A data byte has been received while the I2C is in the addressed slave receiver mode. HW will clear after issuing ACK automatically.		
1	NACK	Assert NACK flag. 0: No function 1: An NACK will be returned during the acknowledge clock pulse on SCLn when ➢ A data byte has been received while the I2C is in the master receiver mode. HW will clear after issuing NACK automatically.	R/W	0
0	Reserved		R	0

15.10.2 I2C n Status register (I2Cn_STAT) (n=0)

Address Offset: 0x04

Check this register when I2C interrupt occurs, and all status will be cleared automatically by writing I2Cn_CTRL or I2Cn_TXDATA register.

Following events will trigger I2C interrupt if I2C interrupt is enabled in NVIC interrupt controller.
- START/Repeat START condition
- STOP condition
- Timeout
- Data byte transmitted or received
- ACK Transmit or received
- NACK Transmit or received

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15	I2CIF	I2C Interrupt flag. 0: I2C status doesn't change. 1: Read→I2C status changes. Write→Clear this flag.	R/W	0
14:10	Reserved		R	0
9	TIMEOUT	Time-out status 0: No Timeout 1: Timeout	R	0
8	LOST_ARB	Lost arbitration 0: Not lost arbitration 1: Lost arbitration	R	0
7	SLV_TX_HIT	0: No matched slave address. 1: Slave address hit, and is called for TX in slave mode.	R	0
6	SLV_RX_HIT	0: No matched slave address. 1: Slave address hit, and is called for RX in slave mode.	R	0
5	MST	Master/Slave status 0: I2C is in Slave state. 1: I2C is in Master state.	R	0
4	START_DN	Start done status 0: No START bit. 1: MASTER mode→ a START bit was issued. SLAVE mode→a START bit was received.	R	0
3	STOP_DN	Stop done status 0: No STOP bit. 1: MASTER mode→a STOP condition was issued. SLAVE mode→a STOP condition was received.	R	0
2	NACK_STAT	NACK done status 0 : Not received a NACK 1 : Received a NACK	R	0
1	ACK_STAT	ACK done status 0 : Not received an ACK	R	0

		1 : Received an ACK		
0	RX_DN	RX done status 0: No RX with ACK/NACK transfer. 1: 8-bit RX with ACK/NACK transfer is done.	R	0

15.10.3 I2C n TX Data register (I2Cn_TXDATA) (n=0)

Address Offset: 0x08

This register contains the data to be transmitted.

In Master TX mode, CPU writes this register will trigger a TX function. In Slave TX mode, CPU has to write this register before next TX procedure.

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DATA[7:0]	Data to be transmitted.	R/W	0x00

15.10.4 I2C n RX Data register (I2Cn_RXDATA) (n=0)

Address Offset: 0x0C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DATA[7:0]	Contains the data received. Read this register when RX_DN = 1.	R	0x00

15.10.5 I2C n Slave Address 0 register (I2Cn_SLVADDR0) (n=0)

Address Offset: 0x10

Only used in slave mode. In master mode, this register has no effect.
If this register contains 0x00, the I2C will not acknowledge any address on the bus. Register ADR0 to ADR3 will be cleared to this disabled state on reset.

Bit	Name	Description	Attribute	Reset
31	ADD_MODE	Slave address mode. 0 : 7-bit address mode 1: 10-bit address mode	RW	0
30	GCEN	General call address enable bit. 0: Disable 1: Enable general call address (0x0)	RW	0
29:10	Reserved		R	0
9:0	ADDR[9:0]	The I2C slave address. ADD[9:0] is valid when ADD_MODE = 1 ADD[7:1] is valid when ADD_MODE = 0	R/W	0

15.10.6 I2C n Slave Address 1~3 register (I2Cn_SLVADDR1~3) (n=0)

Address Offset: 0x14, 0x18, 0x1C

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9:0	ADDR[9:0]	The I2C slave address. ADD[9:0] is valid when ADD_MODE = 1 ADD[7:1] is valid when ADD_MODE = 0	R/W	0

15.10.7 I2C n SCL High Time register (I2Cn_SCLHT) (n=0)

Address Offset: 0x20

✸ ***Note: I2C Bit Frequency = I2Cn_PCLK / (I2Cn_SCLHT+I2Cn_SCLLT)***

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	SCLH[7:0]	Count for SCL High Period time , shall more than 5 SCL High Period Time = (SCLH+1) * I2C0_PCLK cycle	R/W	0x04

15.10.8 I2C n SCL Low Time register (I2Cn_SCLLT) (n=0)

Address Offset: 0x24

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	SCLL[7:0]	Count for SCL Low Period time, shall more than 5 SCL Low Period Time = (SCLL+1) * I2C0_PCLK cycle	R/W	0x04

15.10.9 I2C n Timeout Control register (I2Cn_TOCTRL) (n=0)

Address Offset: 0x2C

Timeout happens when Master/Slave SCL remained LOW for:

TO * 32 * I2C0_PCLK cycle.

When I2C timeout occurs, the I2C transfer will return to “IDLE” state, and issue a TO interrupt to inform user. That means SCL/SDA will be released by HW after timeout. User can issue a STOP after timeout interrupt occurred in Master mode.

Time-out status will be cleared automatically by writing I2Cn_CTRL or I2Cn_TXDATA register.

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	TO[15:0]	Count for checking Timeout. 0: Disable Timeout checking N: Timeout period time = N*32*I2Cn_PCLK cycle	R/W	0x0

15.10.1 I2C n DMA Mode register (I2Cn_DMA) (n=0)

Address Offset: 0x50

Bit	Name	Description	Attribute	Reset
31	TX_DMA_EN	TX DMA mode enable 0: Disable 1: Enable	R/W	0
30	RX_DMA_EN	RX DMA mode enable 0: Disable 1: Enable	R/W	0
29	Reserved	Reserved	R	0
28	RX_DMA_LAST_NACK	I2C issues NACK/ACK for last data in RX DMA mode 0: I2C issue ACK for last data in RX DMA mode 1: I2C issue NACK for last data in RX DMA mode	R/W	0

27:22	Reserved	Reserved	R	0
21:0	DMA_SIZE	Total DMA transfer size (in byte)	R/W	0

16 UNIVERSAL ASYNCHRONOUS RECEIVER AND TRANSMITTER (UART)

16.1 OVERVIEW

The UART offers a flexible means of full-duplex data exchange with external equipment requiring an industry standard NRZ asynchronous serial data format. The serial interface is applied to low speed data transfer and communicate with low speed peripheral devices.

The UART offers a very wide range of baud rates using a fractional baud rate generator.

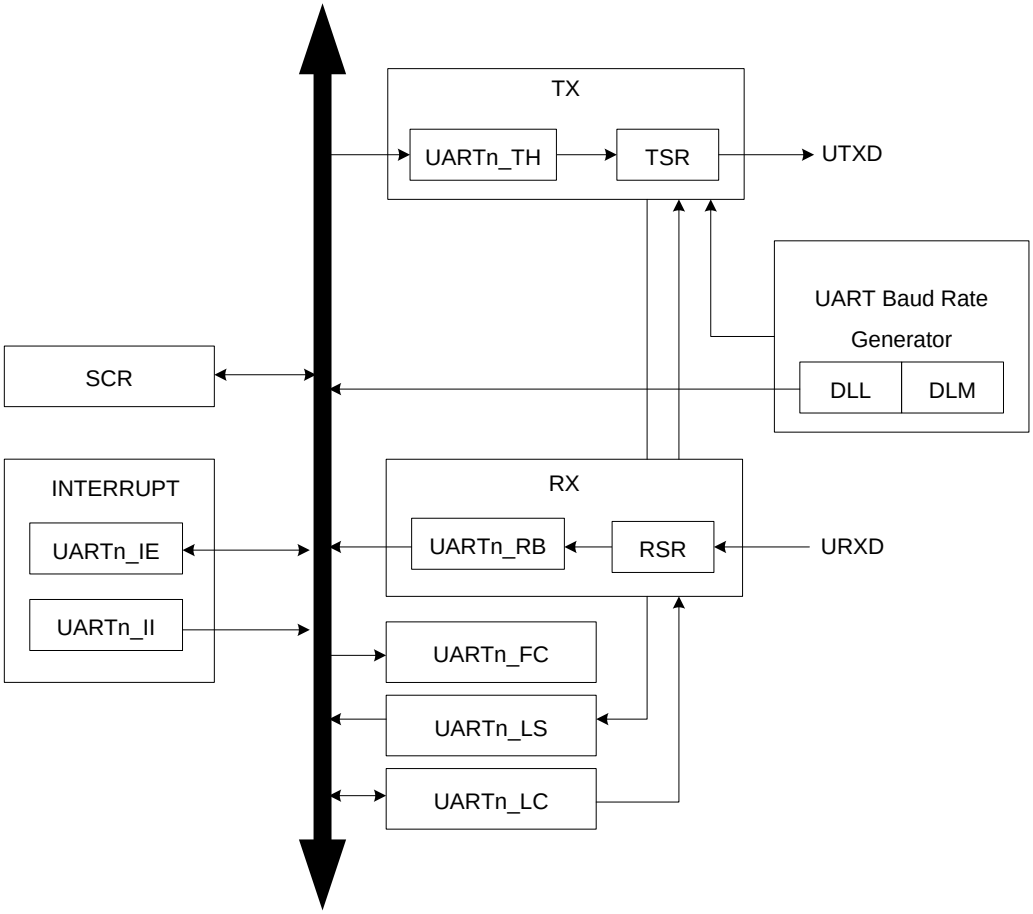
16.2 FEATURES

- Full-duplex, 2-wire asynchronous data transfer.
- Single-wire half-duplex communication
- Register locations conform to 16550 industry standard.
- Receiver FIFO trigger points at 1 byte.
- Built-in baud rate generator.
- Software or hardware flow control.
- Support DMA transfer.

16.3 PIN DESCRIPTION

Pin Name	Type	Description	GPIO Configuration
UTXDn	O	Serial Transmit data.	
URXDn	I	Serial Receive data.	Depends on GPIOn_CFG

16.4 BLOCK DIAGRAM



16.5 BAUD RATE CALCULATION

The UART baud rate is calculated as:

$$UART_{BAUDRATE} = \frac{UARTn\_PCLK}{Oversampling \times (256 \times DLM + DLL) \times (1 + DIVADDVAL / MUL)}$$

Where UARTn_PCLK is the peripheral clock, UARTn_DLM and UARTn_DLL are the standard UART baud rate divider registers, and DIVADDVAL and MULVAL are UART fractional baud rate generator specific parameters in UARTn_FD register.

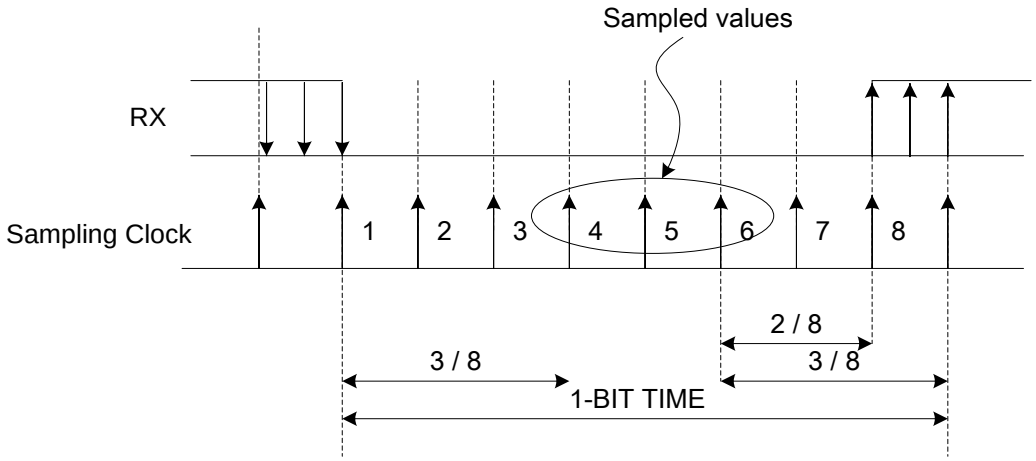
The value of MUL and DIVADDVAL should comply with the following conditions:

1. 1 ≤ MUL ≤ 15
2. 0 ≤ DIVADDVAL ≤ 14
3. DIVADDVAL< MUL
4. MUL – DIVADDVAL≠2
5. Oversampling is 8 or 16

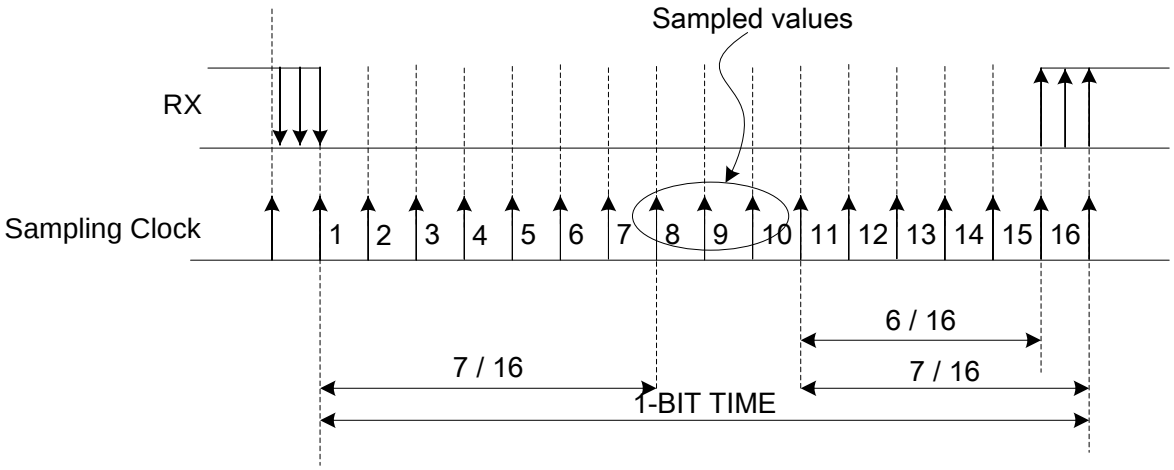
The value of the UARTn_FD register should not be modified while transmitting/receiving data or data may be lost or corrupted.

The oversampling method can be selected by programming the OVER8 bit in UARTn_FD register and can be either 16 or 8 times the baud rate clock.

● OVER8=1: Oversampling by 8 to achieve higher speed (up to UARTn_PCLK/8). In this case the maximum receiver tolerance to clock deviation is reduced.



● OVER8=0: Oversampling by 16 to increase the tolerance of the receiver to clock deviations. In this case, the maximum speed is limited to maximum UARTn_PCLK/16



If the UARTn_FD register value does not comply with these two requests, then the fractional divider output is undefined. If DIVADDVAL is zero then the fractional divider is disabled, and the clock will not be divided.

UART can operate with or without using the Fractional Divider. The desired baud rate can be achieved using several different Fractional Divider settings. The following algorithm illustrates one way of finding a set of DLM, DLL, MUL, and DIVADDVAL values. Such set of parameters yields a baud rate with a relative error of less than 1.1% from the desired one.

The following example illustrates selecting the DIVADDVAL, MUL, DLM, and DLL to generate BR = 115200 when UARTn_PCLK = 12 MHz, and Oversampling = 16.

$$UART_{BAUDRATE} = \frac{UARTn\_PCLK}{Oversampling \times (256 \times DLM + DLL) \times (1 + DIVADDVAL / MUL)}$$

$$115200 = \frac{12000000}{16 \times (256 \times DLM + DLL) \times (1 + DIVADDVAL / MUL)}$$

$$(256 \times DLM + DLL) \times (1 + DIVADDVAL / MUL) = 6.51$$

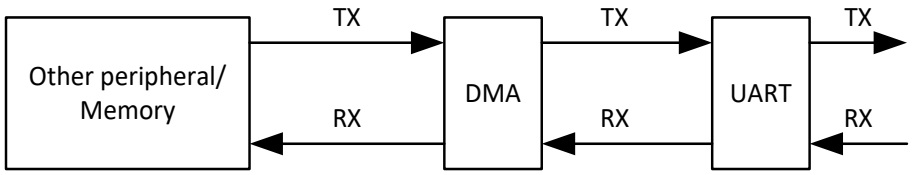
Since the value of MULVAL and DIVADDVAL should comply with the following conditions:

1. $1 \le MUL \le 15$
2. $0 \le DIVADDVAL \le 14$
3. $DIVADDVAL < MUL$
4. $MULVAL - DIVADDVAL \neq 2$

Thus, the suggested UART settings would be: DLM = 0, DLL = 4, DIVADDVAL = 5, and MUL = 8 (fill in 7 in the MULVAL bits). The baud rate generated is 115384, and has a relative error of 0.16% from the originally specified 115200.

16.6 DMA MODE

The UART DMA mode is to use DMA engine to move data in/out UART. Before the DMA transfer start, DMA engine must be set up first. In UART DMA TX Mode, DMA gets data from other peripherals or memories to UART as TX data. In UART DMA RX Mode, DMA receive data from UART and send to other peripherals or memories.



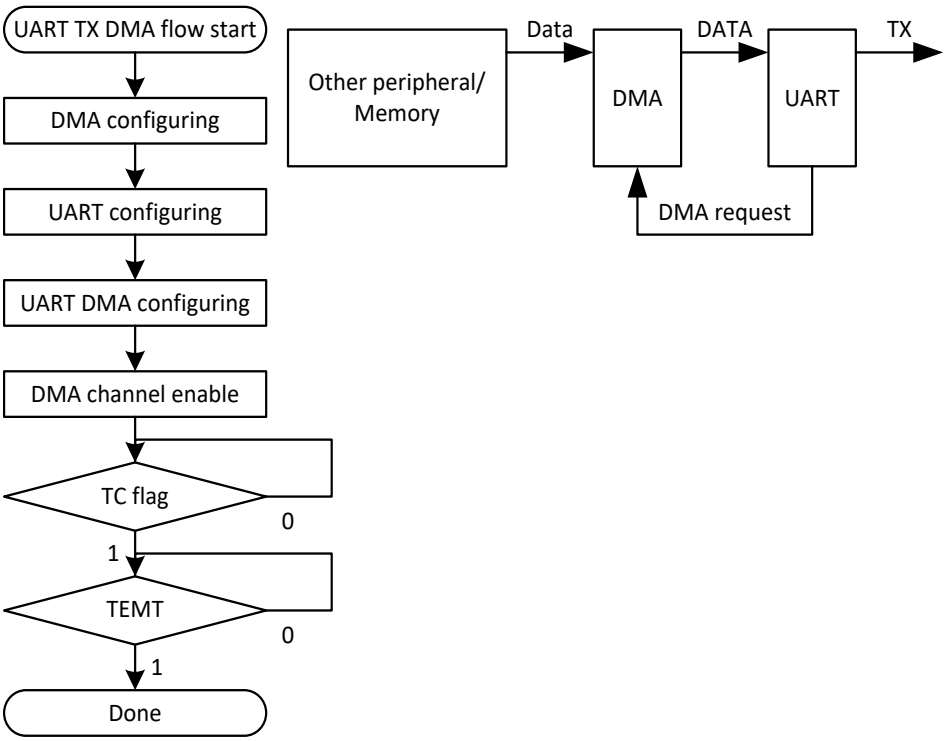
✷ ***Note:***

1. ***The burst size must be set to 1, because UART hasn't FIFO.***
2. ***When DMA destination is UART TX mode, the DMA channel source width must be set to 0x0 (byte).***

16.6.1 UART DMA TX Mode

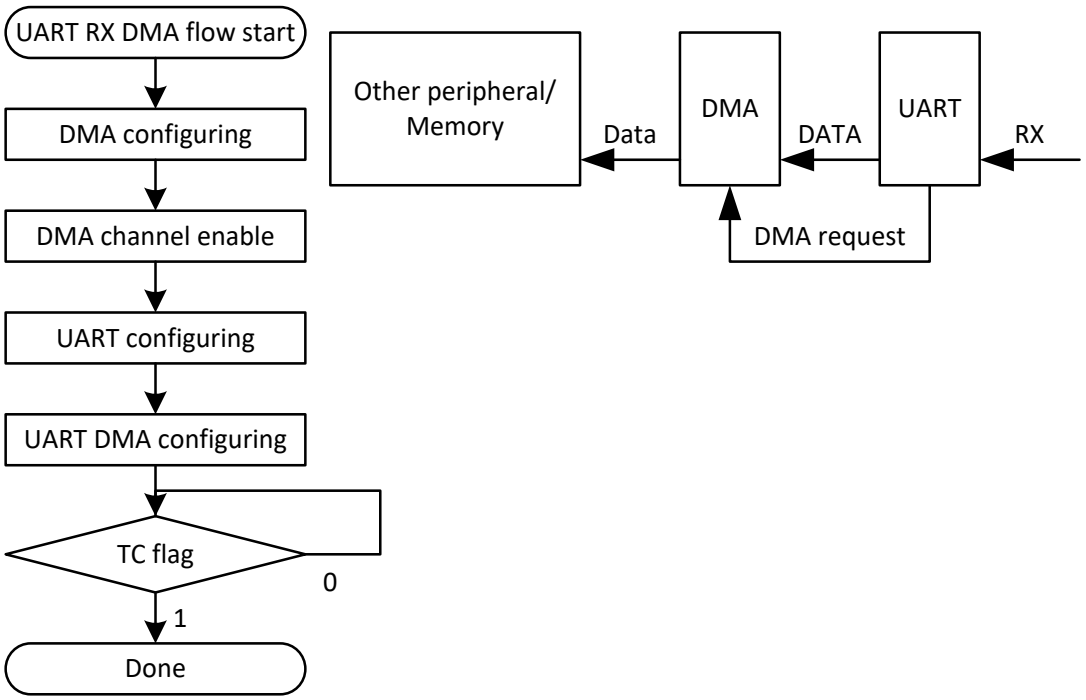
The following register programming flow is used for UART continue DMA TX mode. In this flow FW check DMA TX channel transfer finish then flow can go to the end. This example will turn on FTDMA Channel to move data from other peripherals or memory to UART transmitter holding register. When TX data buffer is empty, DMA request is issued.

✷ ***Note: In TX mode, DMA_SIZE will be ignored.***



16.6.2 UART DMA RX Mode

The following register programming flow chart used for UART DMA RX mode. This example will turn on DMA Channel to move data from UART receive buffer register to other peripherals or SRAM. When RX data buffer has data, DMA request is issued.

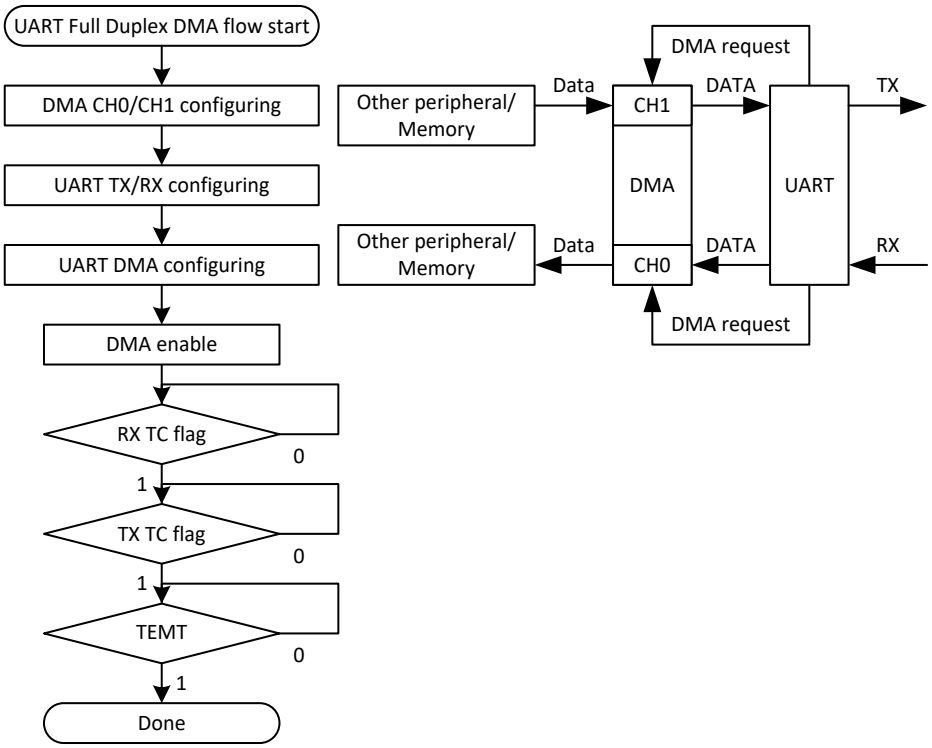


✷ ***Note: The UART DMA_SIZE shall be the same as DMA TOT_SIZE.***

16.6.3 UART DMA Full-duplex Mode

The following programming flow is used for UART DMA full-duplex mode. This example DMA use 2 channels to achieve UART full-duplex function. When RX data buffer has data, DMA request is issued for channel 0. When TX buffer is empty, DMA request is issued for channel 1.

✷ ***Note: In full-duplex mode, DMA_SIZE changes are controlled by RX DMA.***



16.6.4 UART DMA Configuration Recommendations

- UART TX DMA

DMAC setting for UART TX				Recommend setting for UART DMA register
Cn_CSR **SRC_WIDTH**	Cn_CSR **SRC_SIZE**	Cn_SIZE **TOT_SIZE**	Cn_CSR **DST_WIDTH**	DMA_SIZE(word unit)
0x0 (byte)	0x0 (burst=1)	N (byte)	0x0 (byte)	N

- UART RX DMA

DMAC setting for UART RX				Recommend setting for UART DMA register
Cn_CSR **SRC_WIDTH**	Cn_CSR **SRC_SIZE**	Cn_SIZE **TOT_SIZE**	Cn_CSR **DST_WIDTH**	DMA_SIZE(word unit)
0x0 (byte)	0x0 (burst=1)	N (byte)	0x0 (byte)	N

16.7 UART REGISTERS

Base Address: 0x4001 6000 (UART0)
0x4005 6000 (UART1)

16.7.1 UART n Receiver Buffer register (UARTn_RB) (n=0,1)

Address Offset: 0x00

This register is the byte of the UART RX FIFO, and contains the character received and can be read via the bus interface. The LSB (bit 0) contains the first-received data bit. If the character received is less than 8 bits, the unused MSBs are padded with zeros.

The Divisor Latch Access Bit (DLAB) in the UARTn_LC register must be zero in order to access this register.

Since PE, FE and BI bits correspond to the byte on the top of the UART RX FIFO (i.e. the one that will be read in the next read from this register), the right approach for fetching the valid pair of received byte and its status bits is first to read the content of the UARTn_LS register, and then to read a byte from this register.

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	RB[7:0]	Contains the received byte in the UART RX FIFO.	R	0

16.7.2 UART n Transmitter Holding register (UARTn_TH) (n=0,1)

Address Offset: 0x00

This register is the byte of the UART TX FIFO. The byte is the character in the TX FIFO and can be written via the bus interface. The LSB represents the first bit to transmit.

The Divisor Latch Access Bit (DLAB) in UARTn_LC register must be zero in order to access this register.

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	TH[7:0]	The byte will be sent when it is the byte in TX FIFO and the transmitter is available.	W	0

16.7.3 UART n Divisor Latch LSB registers (UARTn_DLL) (n =0,1)

Address Offset: 0x00

The UART Divisor Latch is part of the UART Baud Rate Generator and holds the value used (optionally with the Fractional Divider) to divide the UARTn_PCLK clock in order to produce the baud rate clock, which must be the multiple of the desired baud rate that is specified by the Oversampling Register (typically 16X).

The UARTn_DLL and UARTn_DLM registers together form a 16-bit divisor, and DLAB bit in UARTn_LC register must be one in order to access these registers.

DLL contains the lower 8 bits of the divisor and DLM contains the higher 8 bits. A zero value is treated like 0x0001.

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DLL[7:0]	The UART Divisor Latch LSB Register, along with the DLM register, determines the baud rate of the UART.	R/W	0

16.7.4 UART n Divisor Latch MSB register (UARTn_DLM) (n=0,1)

Address Offset: 0x04

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DLM[7:0]	The UART Divisor Latch MSB Register, along with the DLL register, determines the baud rate of the UART.	R/W	0

16.7.5 UART n Interrupt Enable register (UARTn_IE) (n=0,1)

Address Offset: 0x04

The DLAB bit in the UARTn_LC register must be zero in order to access this register.

Bit	Name	Description	Attribute	Reset
31:5	Reserved		R	0
4	TEMTIE	TEMT interrupt enable bit. The status of this interrupt can be read from TEMT bit in UARTn_LS register. 0: Disable 1: Enable	R/W	0
3	Reserved		R	0
2	RLSIE	Receive Line Status (RLS) interrupt enable bit. The status of this interrupt can be read from UARTn_LS[4:1]. 0: Disable 1: Enable	R/W	0
1	THREIE	THRE interrupt enable bit. The status of this interrupt can be read from THRE bit in UARTn_LS register. 0: Disable 1: Enable	R/W	0
0	RDAIE	RDA interrupt enable bit. Enables the Receive Data Available interrupt. It also controls the Character Receive Time-out interrupt. 0: Disable 1: Enable	R/W	0

16.7.6 UART n Interrupt Identification register (UARTn_II) (n=0,1)

Address Offset: 0x08

This register provides a status code that denotes the priority and source of a pending interrupt.

The interrupts are frozen during a UARTn_II register access. If an interrupt occurs during a UARTn_II register access, the interrupt is recorded for the next UARTn_II register access.

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	100b
3:1	INTID[2:0]	Interrupt identification which identifies an interrupt corresponding to the UARTn RX FIFO. 0x3: 1 - Receive Line Status (RLS). 0x2: 2a - Receive Data Available (RDA). 0x1: 3a - THRE Interrupt. 0x7: 3b – TEMT Interrupt Other: Reserved	R	0
0	INTSTATUS	Interrupt status. The pending interrupt can be determined by evaluating UARTn_II[3:1]. 0: At least one interrupt is pending. 1: No interrupt is pending.	R	1

Given the status of UARTn_II[3:0], an interrupt handler routine can determine the cause of the interrupt and how to clear the active interrupt. The UARTn_II register must be read in order to clear the interrupt prior to exiting the Interrupt service routine.

Interrupt	UARTn_II [3:0]	Priority	Interrupt Source	Interrupt Reset
RLS	0110	Highest	Overrun error (OE), Parity error (PE), Framing error (FE) or Break interrupt (BI)	Read UARTn_LS register
RDA	0100	2nd	RX data in FIFO reached trigger level (FCR0=1)	Read UARTn_RB register or UART FIFO drops below trigger level
THRE	0010	3rd	THRE	Read UARTn_II register (if source of interrupt) or Write THR register
TEMT	1110	3rd	TEMT	Read UARTn_II register (if source of interrupt) or Write THR register

16.7.7 UART n FIFO Control register (UARTn_FIFOCTRL) (n=0,1)

Address Offset: 0x08

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:6	RXTL[1:0]	RX Trigger Level. These two bits determine how many receiver UART FIFO characters must be written before an interrupt is activated. 00: Trigger level 0 (1 character) Other: Reserved	W	0
5:1	Reserved		R	0
0	FIFOEN	FIFO enable 0: No effect 1: Enable for both UART Rx and TX FIFOs and UARTn_FIFOCTRL[7:1] access. This bit must be set for proper UART operation.	W	1

16.7.8 UART n Line Control register (UARTn_LC) (n=0,1)

Address Offset: 0x0C

This register determines the format of the data character that is to be transmitted or received.

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	DLAB	Divisor Latch Access bit 0: Disable access to Divisor Latches. 1: Enable access to Divisor Latches.	R/W	0
6	BC	Break Control bit 0: Disable break transmission. 1: Enable break transmission. Output pin UART TXD is forced to logic 0.	R/W	0
5:4	PS[1:0]	Parity Select bits 00: Odd parity. Number of 1s in the transmitted character and the attached parity bit will be odd. 01: Even Parity. Number of 1s in the transmitted character and the attached parity bit will be even. 10: Forced 1 stick parity. 11: Forced 0 stick parity.	R/W	0
3	PE	Parity Enable bit 0: Disable parity generation and checking. 1: Enable parity generation and checking.	R/W	0
2	SBS	Stop Bit Select bit 0: 1 stop bit 1: 2 stop bits (1.5 if WLS bits=00)	R/W	0
1:0	WLS[1:0]	Word Length Select bits 00: 5-bit character length 01: 6-bit character length 10: 7-bit character length 11: 8-bit character length	R/W	0

16.7.9 UART n Line Status register (UARTn_LS) (n=0,1)

Address Offset: 0x14

✸ ***Note:***

1. ***The break interrupt (BI) is associated with the character in the UARTn_RB FIFO.***
2. ***The framing error (FE) is associated with the character in the UARTn_RB FIFO.***
3. ***The parity error (PE) is associated with the character in UARTn_RB FIFO.***

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	RXFE	Error in RX FIFO flag. RXFE =1 when a character with a RX error such as framing error, parity error, or break interrupt, is loaded into the UARTn_RB register. This bit is cleared when the UARTn_LS register is read and there are no subsequent errors in the UART FIFO. 0: UARTn_RB register contains no UART RX errors or FIFOEN=0 1: UARTn_RB register contains at least one UART RX error.	R	0
6	TEMT	Transmitter Empty flag TEMT=1 when both THR and TSR are empty; TEMT is cleared when either the TSR or the THR contain valid data. 0: THR and/or TSR contains valid data. 1: THR and TSR are empty.	R	1
5	THRE	Transmitter Holding Register Empty flag THRE indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART to issue THRE interrupt to if THREIE=1. THRE=1 when a character is transferred from the THR into the TSR. The bit is reset to logic 0 concurrently with the loading of the Transmitter Holding Register by the CPU. 0: THR contains valid data. 1: THR (TX FIFO) is empty.	R	1
4	BI	Break Interrupt flag. When RXD1 is held in the spacing state (all zeros) for one full character transmission (start, data, parity, stop), a break interrupt occurs. Once the break condition has been detected, the receiver goes idle until RXD1 goes to marking state (all ones). A UARTn_LS register read clears BI bit. The time of break detection is dependent on FIFOEN bit in UARTn_FIFOCTRL register. 0: Break interrupt status is inactive. 1: Break interrupt status is active.	R	0
3	FE	Framing Error flag. When the stop bit of a received character is a logic 0, a framing error occurs. A UARTn_LS register read clears FE bit. The time of the framing error detection is dependent on FIFOEN bit in UARTn_FIFOCTRL register. Upon detection of a framing error, the RX will attempt to re-synchronize to the data and assume that the bad stop bit is actually an early start bit. However, it cannot be assumed that the next received byte will be correct even if there is no Framing Error. 0: Framing error status is inactive. 1: Framing error status is active.	R	0
2	PE	Parity Error flag. When the parity bit of a received character is in the wrong state, a parity error occurs. A UARTn_LS register read clears PE bit. Time of parity error detection is dependent on FIFOEN bit in UARTn_FIFOCTRL register. 0: Parity error status is inactive. 1: Parity error status is active.	R	0
1	OE	Overrun Error flag. The overrun error condition is set as soon as it occurs. A UARTn_LS register read clears OE bit. OE=1 when UART RSR has a new character assembled and the UARTn_RB FIFO is full. In this case, the UARTn_RB FIFO will not be overwritten and the character in the UARTn_RS register will be lost. 0: Overrun error status is inactive. 1: Overrun error status is active.	R	0
0	RDR	Receiver Data Ready flag RDR=1 when the UARTn_RB FIFO holds an unread character and is cleared when the UARTn_RB FIFO is empty. 0: UARTn_RB FIFO is empty. 1: UARTn_RB FIFO contains valid data.	R	0

16.7.10 UART n Scratch Pad register (UARTn_SP) (n=0,1)

Address Offset: 0x1C

This register has no effect on the UART operation. This register can be written and/or read at user's discretion. There is no provision in the interrupt interface that would indicate to the host that a read or write of this register has occurred.

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PAD[7:0]	A readable, writable byte.	R/W	0

16.7.11 UART n Fractional Divider register (UARTn_FD) (n=0,1)

Address Offset: 0x28

This register controls the clock prescaler for the baud rate generation and can be read and written at the user's discretion. This prescaler takes the APB clock and generates an output clock according to the specified fractional requirements.

In most applications, the UART samples received data 16 times in each nominal bit time, and sends bits that are 16 input clocks wide. OVER8 bit allows software to control the ratio between the input clock and bit clock. This is required for smart card mode, and provides an alternative to fractional division for other modes.

✸ ***Note:***

1. If the fractional divider is active (DIVADDVAL>0) and UARTn_DLM=0, the value of the UARTn_DLL register must ≥ 3.

2. MUL – DIVADDVAL ≠ 2, so MULVAL[3:0] - DIVADDVAL[3:0] ≠ 1.

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0
8	OVER8	Oversampling value 0: Oversampling by 16 1: Oversampling by 8	R/W	0
7:4	MULVAL[3:0]	Baud rate pre-scaler multiplier value MUL = MULVAL[3:0] +1 0000: Baud rate pre-scaler multiplier value is 1 for HW 0001: Baud rate pre-scaler multiplier value is 2 for HW … … 1111: Baud rate pre-scaler multiplier value is 16 for HW.	R/W	0
3:0	DIVADDVAL[3:0]	Baud rate generation pre-scaler divisor value. If this field is 0, fractional baud rate generator will not impact the UART baud rate	R/W	0

16.7.12 UART n Control register (UARTn_CTRL) (n=0,1)

Address Offset: 0x30

In addition to HW flow control (Auto-CTS and Auto-RTS mechanisms), this register enables implementation of SW flow control.

When TXEN = 1, the UART transmitter will keep sending data as long as they are available. As soon as TXEN bit becomes 0, UART transmission will stop.

It is strongly suggested to let the UART HW implemented auto flow control features take care of limit the scope of TXEN to SW flow control.

✸ ***Note: It is advised that TXEN and RXEN are set in the same instruction if needed in order to minimize the setup and the hold time of the receiver.***

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	TXEN	0: Disable TX function. The transfer of characters from the UARTn_TH register or TX FIFO into the transmit shift register is blocked. While a character is being sent, the transmission of that character is completed, but no further characters are sent until this bit is set again. 1: Data written to the UARTn_TH register is output on the TXD pin as soon as any preceding data has been sent.	R/W	1
6	RXEN	0: Disable RX related function 1: Enable RX	R/W	1
5:4	Reserved		R	0
3:1	MODE[2:0]	UARTn Mode 000: UART mode. HW will switch GPIO to UTXDn and URXDn. Other: Reserved	R/W	0
0	UARTEN	UART enable 0: Disable . All UART shared pins act as GPIO. 1: Enable. HW switches GPIO to UART pin according to MODE bits automatically.	R/W	0

16.7.13UART n Half-duplex Enable register (UARTn_HDEN) (n=0,1)

Address Offset: 0x34

After reset the UART will be in full-duplex mode, meaning that both TX and RX work independently. After setting the HDEN bit, the UART will be in half-duplex mode. In this mode, the UART ensures that the receiver is locked when idle, or will enter a locked state after having received a complete ongoing character reception. Line conflicts must be handled in SW.

The behavior of the UART is unpredictable when data is presented for reception while data is being transmitted. For this reason, the value of the HDEN register should not be modified while sending or receiving data, or data may be lost or corrupted.

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	HDEN	Half-duplex mode enable bit 0: Disable 1: Enable	R/W	0

16.7.1 UART n DMA Mode register (UARTn_DMA) (n=0,1)

Address Offset: 0x50

Bit	Name	Description	Attribute	Reset
31:23	Reserved		R	0
22	DMA_EN	UART DMA mode enable 0: Disable 1: Enable	R/W	0
21:0	DMA_SIZE	Total DMA transfer size (in byte)	R/W	0

17 FOC

17.1 OVERVIEW

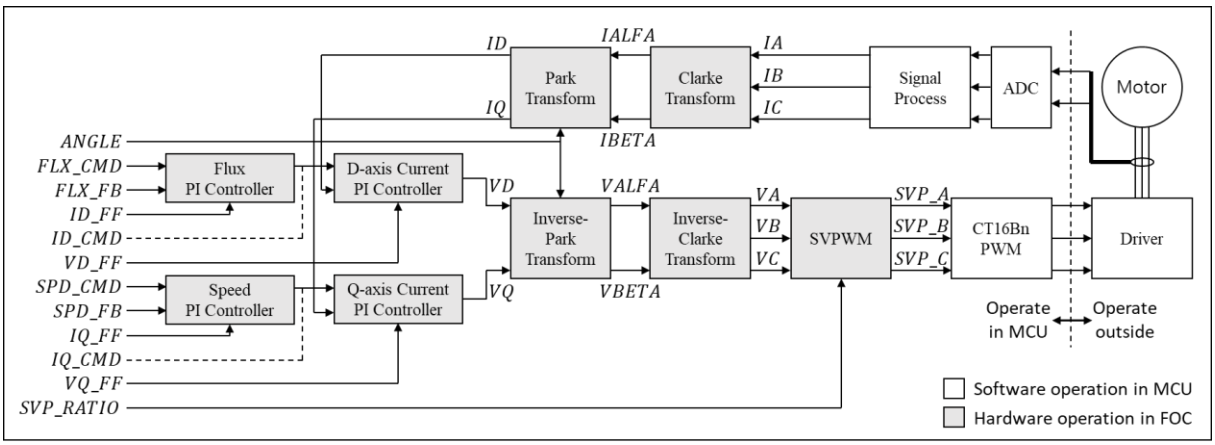
Field oriented control (FOC) block is designed for various motor control applications including induction motor (IM), permanent magnet synchronous motor (PMSM), and brushless DC motor (BLDC). The control block contains 1 Clarke transformer, 1 inverse-Clarke transformer, 1 Park transformer, 1 inverse-Park transformer, 4 PI-controllers for flux, speed, d-axis current, q-axis current individually, and 1 SVPWM calculator. After setting current feedbacks and angle (obtained from hall sensor or sensor-less algorithm), the output duty can be calculated from the FOC block.

17.2 FEATURES

- 1 Clarke and inverse-Clarke transformer.
- 1 Park and inverse-Park transformer.
- 4 PI-controllers for flux, speed, d-axis current, and q-axis current control.
- 1 SVPWM calculator with third harmonic injection function
- Build in sin/cos table for Park and inverse-Park transformations.
- Each sub-block can be turn on/off through CTRL register.
- Calculate under Q15 format.

17.3 FUNCTION DESCRIPTIONS

The block diagram is shown in the following figure.



The FOC block first transforms current signals from A/B/C domain into α/β domain and d/q domain by the Clarke transformer and Park transformer. Then, the d/q domain currents are controlled by flux, speed, d-axis current, q-axis current PI controllers. The d/q domain output voltages can be calculated from PI controllers. In the next step, d/q domain voltage are converted into α/β domain and a/b/c domain through the inverse-Park transformer and inverse-Clarke transformer, and then the a/b/c domain voltage are boosted by SVPWM block using third harmonic injection algorithm. Finally, the results (SVP_A, SVP_B, SVP_C) can be fetched and processed as CT16Bn PWM duty by MCU.

17.3.1 Clarke and Inverse-Clarke Transformers

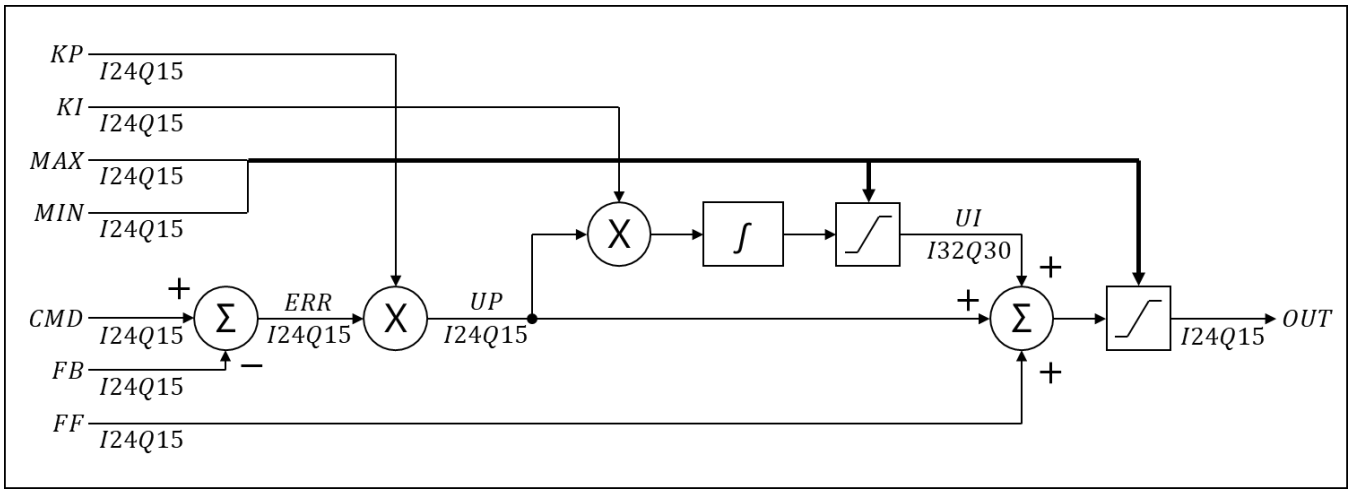
The Clarke transformer converts IA, IB, IC registers from a/b/c domain to α/β domain and sets corresponding value to IALFA, IBETA registers. On the other hand, the inverse-Clarke transformer converts VALFA, VBETA registers from α/β domain to a/b/c domain and sets corresponding value to VA, VB, VC. All registers are in I24Q15 format (1 sign bit, 8 integer bits and 15 fraction bits).

17.3.2 Park and Inverse-Park Transformers

The Park transformer converts α/β domain current into d/q domain current through rotating IALFA, IBETA register values clockwise by ANGLE register value. The results are stored in ID, IQ registers. On the other hand, the inverse-Park transformer converts d/q domain voltage into α/β domain voltage through rotating VD, VQ registers values counter-clockwise by ANGLE register value. The results are stored in VALFA, VBETA. ANGLE register is in I16Q15 format, and others are in I24Q15 format.

17.3.3 PI Controllers

PI controller block diagram is shown in the following figure.



The PI controller contains 4 setting registers (KP, KI, MAX, MIN), 3 input registers (CMD, FB, FF), 1 output register (OUT), and 3 status registers (ERR, UP, UI). KP is the proportional gain, and KI is the integral gain. Both registers determine the control response. MAX and MIN are upper limit and lower limit of output. UI and OUT are limited by these value. CMD, FB, and FF registers represents the command, feedback, and feedforward. OUT is the PI output. If flux PI controller is enabled, CMD of d-axis current PI controller is automatically overwritten with OUT value from flux PI controller, otherwise CMD of d-axis current PI controller remains its value. Similarly, CMD of q-axis current PI controller is changed if speed PI controller is enabled. ERR, UP, and UI registers are calculation status inside the controller. Calculation results can be derived and inspected from these values. UI register is in I32Q30 format, and others are in I24Q15 format.

17.3.4 SVPWM Calculator

SVPWM calculator converts VA, VB, VC into SVP_A, SVP_B, SVP_C by applying third harmonic injection, and the output amplitude can be boosted up to 1.155 times larger than original signals. Additionally, SVP_RATIO represents the output amplification ratio. With proper setting value, DC bus voltage compensation can be performed by this register. All registers are in I24Q15 format.

17.4 FOC REGISTERS

Base Address: 0x4007 4000

17.4.1 FOC Control register (FOC_CTRL)

Address Offset: 0x00

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15	FOC_INT	FOC interrupt 0: No FOC interrupt 1: FOC interrupt is triggered	R/W	0
14	IE	FOC interrupt enable bit 0: Disable 1: Enable	R/W	0
13:10	Reserved		R	0
9	IQ_PI_EN	Current Q PI function enable bit 0: Disable 1: Enable	R/W	0
8	ID_PI_EN	Current D PI function enable bit 0: Disable 1: Enable	R/W	0
7	SPD_PI_EN	Speed PI function enable bit 0: Disable 1: Enable	R/W	0
6	FLX_PI_EN	Flux PI function enable bit 0: Disable 1: Enable	R/W	0
5	SVP_EN	SVPWM function enable bit 0: Disable 1: Enable	R/W	0
4	ICLARKE_EN	IClarke function enable bit 0: Disable 1: Enable	R/W	0
3	IPARK_EN	IPark function enable bit 0: Disable 1: Enable	R/W	0
2	PARK_EN	Park function enable bit 0: Disable 1: Enable	R/W	0
1	CLARKE_EN	Clarke function enable bit 0: Disable 1: Enable	R/W	0
0	FOC_EN	FOC enable bit 0: Disable 1: Enable	R/W	0

17.4.2 FOC Control register 1 (FOC_CTRL1)

Address Offset: 0x04

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1	RDY	FOC ready status 0: FOC is operating 1: FOC is not operating	R	0
0	START_PLUSE	FOC operation trigger bit 0: No action 1: Trigger 1 pulse to start FOC operating	R/W	0

17.4.3 FOC Motor-A Phase Current register (FOC_IA)

Address Offset: 0x08

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IA	Motor A-phase current in I24Q15 format	R/W	0

17.4.4 FOC Motor-B Phase Current register (FOC_IB)

Address Offset: 0x0C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IB	Motor B-phase current in I24Q15 format	R/W	0

17.4.5 FOC Motor-C Phase Current register (FOC_IC)

Address Offset: 0x10

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IC	Motor C-phase current in I24Q15 format	R/W	0

17.4.6 FOC Motor Stator Output Angle register (FOC_ANGLE)

Address Offset: 0x14

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	ANGLE	Motor stator output electrical angle in I24Q15 format	R/W	0

17.4.7 FOC Stator Flux Command register (FOC_FLX_CMD)

Address Offset: 0x18

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_CMD	Motor stator flux command in I24Q15 format	R/W	0

17.4.8 FOC Motor Stator Flux Feedback register (FOC_FLX_FB)

Address Offset: 0x1C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_FB	Motor stator flux feedback in I24Q15 format	R/W	0

17.4.9 FOC Current D Feedforward register (FOC_ID_FF)

Address Offset: 0x20

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_FF	Field current D feedforward in I24Q15 format	R/W	0

17.4.10 FOC Rotor Speed Command register (FOC_SPD_CMD)

Address Offset: 0x24

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_CMD	Motor rotor speed command in I24Q15 format	R/W	0

17.4.11 FOC Motor Rotor Speed Feedback register (FOC_SPD_FB)

Address Offset: 0x28

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_FB	Motor rotor speed feedback in I24Q15 format	R/W	0

17.4.12 FOC Q-axis Current Feedforward register (FOC_IQ_FF)

Address Offset: 0x2C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_FF	Torque current feedforward in I24Q15 format	R/W	0

17.4.13 FOC D-axis Current Command register (FOC_ID_CMD)

Address Offset: 0x30

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_CMD	D-axis current command in I24Q15 format	R/W	0

17.4.14 FOC D-axis Voltage Feedforward register (FOC_VD_FF)

Address Offset: 0x34

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	VD_FF	D-axis voltage feeedforward in I24Q15 format	R/W	0

17.4.15 FOC Q-axis Current Command register (FOC_IQ_CMD)

Address Offset: 0x38

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_CMD	Q-axis current command in I24Q15 format	R/W	0

17.4.16 FOC Q-axis Voltage Feedforward register (FOC_VQ_FF)

Address Offset: 0x3C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	VQ_FF	Q-axis voltage feeedforward in I24Q15 format	R/W	0

17.4.17 FOC SVPWM Output Ratio register (FOC_SVP_RATIO)

Address Offset: 0x40

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SVP_RATIO	SVPWM output ratio in I24Q15 format	R/W	0x8000

17.4.18 FOC Motor Alpha-axis Current register (FOC_IALFA)

Address Offset: 0x44

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IALFA	Motor alpha-axis current in I24Q15 format	R/W	0

17.4.19 FOC Motor Beta-axis Current register (FOC_IBETA)

Address Offset: 0x48

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IBETA	Motor beta-axis current in I24Q15 format	R/W	0

17.4.20 FOC Motor D-axis Current register (FOC_ID)

Address Offset: 0x4C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID	Motor D-axis current in I24Q15 format	R/W	0

17.4.21 FOC Motor Q-axis Current register (FOC_IQ)

Address Offset: 0x50

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ	Motor Q-axis current in I24Q15 format	R/W	0

17.4.22 FOC Driver D-axis Output Voltage register (FOC_VD)

Address Offset: 0x54

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	VD	Driver D-axis output voltage in I24Q15 format	R/W	0

17.4.23 FOC Driver Q-axis Output Voltage register (FOC_VQ)

Address Offset: 0x58

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	VQ	Driver Q-axis output voltage in I24Q15 format	R/W	0

17.4.24 FOC Driver Alpha-axis Output Voltage register (FOC_VALFA)

Address Offset: 0x5C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	VALFA	Driver alpha-axis output voltage in I24Q15 format	R/W	0

17.4.25 FOC Driver Beta-axis Output Voltage register (FOC_VBETA)

Address Offset: 0x60

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	VBETA	Motor beta-axis current in I24Q15 format	R/W	0

17.4.26 FOC Driver A-phase Output Voltage register (FOC_VA)

Address Offset: 0x64

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	VA	Driver A-phase output voltage in I24Q15 format	R/W	0

17.4.27 FOC Driver B-phase Output Voltage register (FOC_VB)

Address Offset: 0x68

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	VB	Driver B-phase output voltage in I24Q15 format	R/W	0

17.4.28 FOC Driver C-phase Output Voltage register (FOC_VC)

Address Offset: 0x6C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	VC	Driver C-phase output voltage in I24Q15 format	R/W	0

17.4.29 FOC SVPWM Maximum Value register (FOC_SVP_MAX)

Address Offset: 0x70

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SVP_MAX	Maximum value between A, B, C-phase output voltage in I24Q15 format	R/W	0

17.4.30 FOC SVPWM Minimum Value register (FOC_SVP_MIN)

Address Offset: 0x74

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SVP_MIN	Minimum value between A, B, C-phase output voltage in I24Q15 format	R/W	0

17.4.31 FOC SVPWM Delta Value register (FOC_SVP_DLTA)

Address Offset: 0x78

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SVP_DLTA	Difference between minimum and maximum value of SVPWM calculation in I24Q15 format	R/W	0

17.4.32 FOC SVPWM A-phase Output Voltage register (FOC_SVP_A)

Address Offset: 0x7C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SVP_A	SVPWM A-phase output voltage in I24Q15 format	R/W	0

17.4.33 FOC SVPWM B-phase Output Voltage register (FOC_SVP_B)

Address Offset: 0x80

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SVP_B	SVPWM B-phase output voltage in I24Q15 format	R/W	0

17.4.34 FOC SVPWM C-phase Output Voltage register (FOC_SVP_C)

Address Offset: 0x84

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SVP_C	SVPWM C-phase output voltage in I24Q15 format	R/W	0

17.4.35 FOC Proportional Gain for Flux PI Controller register (FOC_FLX_PI_KP)

Address Offset: 0x88

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_PI_KP	Proportional gain for flux PI controller in I24Q15 format	R/W	0

17.4.36 FOC Integral Gain for Flux PI Controller register (FOC_FLX_PI_KI)

Address Offset: 0x8C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_PI_KI	Integral gain for flux PI controller in I24Q15 format	R/W	0

17.4.37 FOC Maximum Output Limit for Flux PI Controller register (FOC_FLX_PI_MAX)

Address Offset: 0x90

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_PI_MAX	Maximum output limit for flux PI controller in I24Q15 format	R/W	0

17.4.38 FOC Minimum Output Limit for Flux PI Controller register (FOC_FLX_PI_MIN)

Address Offset: 0x94

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_PI_MIN	Minimum output limit for flux PI controller in I24Q15 format	R/W	0

17.4.39 FOC Command for Flux PI Controller register (FOC_FLX_PI_CMD)

Address Offset: 0x98

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_PI_CMD	Command for flux PI controller in I24Q15 format	R/W	0

17.4.40 FOC Feedback for Flux PI Controller register (FOC_FLX_PI_FB)

Address Offset: 0x9C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_PI_FB	Feedback for flux PI controller in I24Q15 format	R/W	0

17.4.41 FOC Feedforward for Flux PI Controller register (FOC_FLX_PI_FF)

Address Offset: 0xA0

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_PI_FF	Feedforward for flux PI controller in I24Q15 format	R/W	0

17.4.42 FOC Output of Flux PI Controller register (FOC_FLX_PI_OUT)

Address Offset: 0xA4

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_PI_OUT	Output of flux PI controller in I24Q15 format	R/W	0

17.4.43 FOC Error of Flux PI Controller register (FOC_FLX_PI_ERR)

Address Offset: 0xA8

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_PI_ERR	Error of flux PI controller in I24Q15 format	R/W	0

17.4.44 FOC Proportional Output for Flux PI Controller register (FOC_FLX_PI_UP)

Address Offset: 0xAC

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_PI_UP	Proportional output for flux PI controller in I24Q15 format	R/W	0

17.4.45 FOC Integral Output for Flux PI Controller register (FOC_FLX_PI_UI)

Address Offset: 0xB0

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	FLX_PI_UI	Integral output for flux PI controller in I24Q15 format	R/W	0

17.4.46 FOC Proportional Gain for Speed PI Controller register (FOC_SPD_PI_KP)

Address Offset: 0xB4

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_PI_KP	Proportional gain for speed PI controller in I24Q15 format	R/W	0

17.4.47 FOC Integral Output for Speed PI Controller register (FOC_SPD_PI_KI)

Address Offset: 0xB8

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_PI_KI	Integral gain for speed PI controller in I24Q15 format	R/W	0

17.4.48 FOC Maximum Output Limit for Speed PI Controller register (FOC_SPD_PI_MAX)

Address Offset: 0xBC

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_PI_MAX	Maximum output limit for speed PI controller in I24Q15 format	R/W	0

17.4.49 FOC Minimum Output Limit for Speed PI Controller register (FOC_SPD_PI_MIN)

Address Offset: 0xC0

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_PI_MIN	Minimum output limit for speed PI controller in I24Q15 format	R/W	0

17.4.50 FOC Command for Speed PI Controller register (FOC_SPD_PI_CMD)

Address Offset: 0xC4

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_PI_CMD	Command for speed PI controller in I24Q15 format	R/W	0

17.4.51 FOC Feedback for Speed PI Controller register (FOC_SPD_PI_FB)

Address Offset: 0xC8

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_PI_FB	Feedback for speed PI controller in I24Q15 format	R/W	0

17.4.52 FOC Feedforward for Speed PI Controller register (FOC_SPD_PI_FF)

Address Offset: 0xCC

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_PI_FF	Feedforward for speed PI controller in I24Q15 format	R/W	0

17.4.53 FOC Output of Speed PI Controller register (FOC_SPD_PI_OUT)

Address Offset: 0xD0

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_PI_OUT	Output of speed PI controller in I24Q15 format	R/W	0

17.4.54 FOC Error of Speed PI Controller register (FOC_SPD_PI_ERR)

Address Offset: 0xD4

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_PI_ERR	Error of speed PI controller in I24Q15 format	R/W	0

17.4.55 FOC Proportional Output for Speed PI Controller register (FOC_SPD_PI_UP)

Address Offset: 0xD8

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_PI_UP	Proportional output for speed PI controller in I24Q15 format	R/W	0

17.4.56 FOC Integral Output for Speed PI Controller register (FOC_SPD_PI_UI)

Address Offset: 0xDC

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	SPD_PI_UI	Integral output for speed PI controller in I24Q15 format	R/W	0

17.4.57 FOC Proportional Gain for D-axis Current PI Controller register (FOC_ID_PI_KP)

Address Offset: 0xE0

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_PI_KP	Proportional gain for D-axis current PI controller in I24Q15 format	R/W	0

17.4.58 FOC Integral Gain for D-axis Current PI Controller register (FOC_ID_PI_KI)

Address Offset: 0xE4

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_PI_KI	Integral gain for D-axis current PI controller in I24Q15 format	R/W	0

17.4.59 FOC Maximum Output Limit for D-axis Current PI Controller register (FOC_ID_PI_MAX)

Address Offset: 0xE8

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_PI_MAX	Maximum output limit for D-axis current PI controller in I24Q15 format	R/W	0

17.4.60 FOC Minimum Output Limit for D-axis Current PI Controller register (FOC_ID_PI_MIN)

Address Offset: 0xEC

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_PI_MIN	Minimum output limit for D-axis current PI controller in I24Q15 format	R/W	0

17.4.61 FOC Command for D-axis Current PI Controller register (FOC_ID_PI_CMD)

Address Offset: 0xF0

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_PI_CMD	Command for D-axis current PI controller in I24Q15 format	R/W	0

17.4.62 FOC Feedback for D-axis Current PI Controller register (FOC_ID_PI_FB)

Address Offset: 0xF4

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_PI_FB	Feedback for D-axis current PI controller in I24Q15 format	R/W	0

17.4.63 FOC Feedforward for D-axis Current PI Controller register (FOC_ID_PI_FF)

Address Offset: 0xF8

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_PI_FF	Feedforward for D-axis current PI controller in I24Q15 format	R/W	0

17.4.64 FOC Output of D-axis Current PI Controller register (FOC_ID_PI_OUT)

Address Offset: 0xFC

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_PI_OUT	Output of D-axis current PI controller in I24Q15 format	R/W	0

17.4.65 FOC Error of D-axis Current PI Controller register (FOC_ID_PI_ERR)

Address Offset: 0x100

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_PI_ERR	Error of D-axis current PI controller in I24Q15 format	R/W	0

17.4.66 FOC Proportional Output for D-axis Current PI Controller register (FOC_ID_PI_UP)

Address Offset: 0x104

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_PI_UP	Proportional output for D-axis current PI controller in I24Q15 format	R/W	0

17.4.67 FOC Integral Output for D-axis Current PI Controller register (FOC_ID_PI_UI)

Address Offset: 0x108

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	ID_PI_UI	Integral output for D-axis current PI controller in I24Q15 format	R/W	0

17.4.68 FOC Proportional Gain for Q-axis Current PI Controller register (FOC_IQ_PI_KP)

Address Offset: 0x10C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_PI_KP	Proportional gain for Q-axis current PI controller in I24Q15 format	R/W	0

17.4.69 FOC Integral Gain for Q-axis Current PI Controller register (FOC_IQ_PI_KI)

Address Offset: 0x110

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_PI_KI	Integral gain for Q-axis current PI controller in I24Q15 format	R/W	0

17.4.70 FOC Maximum Output Limit for Q-axis Current PI Controller register (FOC_IQ_PI_MAX)

Address Offset: 0x114

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_PI_MAX	Maximum output limit for Q-axis current PI controller in I24Q15 format	R/W	0

17.4.71 FOC Minimum Output Limit for Q-axis Current PI Controller register (FOC_IQ_PI_MIN)

Address Offset: 0x118

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_PI_MIN	Minimum output limit for Q-axis current PI controller in I24Q15 format	R/W	0

17.4.72 FOC Command for Q-axis Current PI Controller register (FOC_IQ_PI_CMD)

Address Offset: 0x11C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_PI_CMD	Command for Q-axis current PI controller in I24Q15 format	R/W	0

17.4.73 FOC Feedback for Q-axis Current PI Controller register (FOC_IQ_PI_FB)

Address Offset: 0x120

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_PI_FB	Feedback for Q-axis current PI controller in I24Q15 format	R/W	0

17.4.74 FOC Feedforward for Q-axis Current PI Controller register (FOC_IQ_PI_FF)

Address Offset: 0x124

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_PI_FF	Feedforward for Q-axis current PI controller in I24Q15 format	R/W	0

17.4.75 FOC Output of Q-axis Current PI Controller register (FOC_IQ_PI_OUT)

Address Offset: 0x128

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_PI_OUT	Output of Q-axis current PI controller in I24Q15 format	R/W	0

17.4.76 FOC Error of Q-axis Current PI Controller register (FOC_IQ_PI_ERR)

Address Offset: 0x12C

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_PI_ERR	Error of Q-axis current PI controller in I24Q15 format	R/W	0

17.4.77 FOC Proportional Output for Q-axis Current PI Controller register (FOC_IQ_PI_UP)

Address Offset: 0x130

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_PI_UP	Proportional output for Q-axis current PI controller in I24Q15 format	R/W	0

17.4.78 FOC Integral Output for Q-axis Current PI Controller register (FOC_IQ_PI_UI)

Address Offset: 0x134

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	IQ_PI_UI	Integral output for Q-axis current PI controller in I24Q15 format	R/W	0

18 ACC

18.1 OVERVIEW

Hardware calculation accelerator (ACC) includes 1 arctangent calculator (ATAN), 1 divider (DIV), and 1 square root calculator (SQRT). All calculation block can be enabled/disabled by ACC_EN, and all start signals can be set individually (ATAN_START_PULSE, DIV_START_PULSE, SQRT_START_PULSE). ACC uses 3 idle signals (ATAN_IDLE, DIV_IDLE, SQRT_IDLE) to show the calculation status. Besides, ACC supports 3 interrupts (ATAN_INT, SQRT_INT, ATANT_INT).

18.2 FEATURES

- 1 arctangent calculator
- 1 divider
- 1 square root calculator
- 1 common enable signal (ACC_EN)
- 3 independent start signals (ATAN_START_PULSE, DIV_START_PULSE, SQRT_START_PULSE)
- 3 independent status signals (ATAN_IDLE, PULSE_IDLE, SQRT_IDLE)
- 1 common interrupt enable bit (IE)

18.3 FUNCTION DESCRIPTIONS

18.3.1 Arctangent Calculator

The arctangent calculator accepts 2 input (ATAN_X, ATAN_Y) and calculates the angle of the vector (ATAN_X, ATAN_Y) as ATAN_OUT. ATAN_X and ATAN_Y are in I16Q15 format. ATAN_OUT is in I18Q15 format ranging from $-\pi$ to π.

18.3.2 Divider

The divider accepts 1 divident input (DIV_DVD) and 1 divisor input (DIV_DVS), and then calculates 1 quotient output (DIV_QUO) and 1 remainder output (DIV_REM). DIV_DVD and DIV_QUO are in I32 format. DIV_DVS and DIV_REM are in I16 format.

18.3.3 Square Root Calculator

The square root calculator accepts 1 input (SQRT_IN) and calculates 1 output (SQRT_OUT). SQRT_IN is in U32 format. SQRT_OUT is in U16 format.

18.4 ACC REGISTERS

Base Address: 0x4007 6000

18.4.1 ACC Hardware Calculation Accelerator Control register (ACC_CTRL)

Address Offset: 0x00

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10	SQRT_INT	SQRT interrupt signal 0: No SQRT interrupt 1: SQRT interrupt is triggered	R/C	0
9	SQRT_RDY	SQRT ready status 0: SQRT is operating 1: SQRT is not operating	R	0
8	SQRT_START_PULSE	SQRT operation trigger bit 0: No action 1: Trigger 1 pulse to start SRQT operating	R/W	0
7	DIV_INT	DIV interrupt signal 0: No DIV interrupt 1: DIV interrupt is triggered	R/C	0
6	DIV_RDY	DIV ready status 0: DIV is operating 1: DIV is not operating	R	0
5	DIV_START_PULSE	DIV operation trigger bit 0: No action 1: Trigger 1 pulse to start DIV operating	R/W	0
4	ATAN_INT	ATAN interrupt signal 0: No ATAN interrupt 1: ATAN interrupt is triggered	R/C	0
3	ATAN_RDY	ATAN ready status 0: ATAN is operating 1: ATAN is not operating	R	0
2	ATAN_START_PULSE	ATAN operation trigger bit 0: No action 1: Trigger 1 pulse to start ATAN operating	R/W	0
1	IE	ACC interrupt enable bit 0: Disable 1: Enable	R/W	0
0	ACC_EN	Hardware accelerator enable bit 0: Disable 1: Enable	R/W	0

18.4.2 ACC Arctangent Input register (ACC_ATAN_IN)

Address Offset: 0x04

Bit	Name	Description	Attribute	Reset
31:16	ATAN_Y	Arctangent y-axis input value in I16Q15 format	R/W	0
15:0	ATAN_X	Arctangent x-axis input value in I16Q15 format	R/W	0

18.4.3 ACC Arctangent Output register (ACC_ATAN_OUTPUT)

Address Offset: 0x08

Bit	Name	Description	Attribute	Reset

31:18	Reserved		R	0
17:0	ATAN_OUT	Dividend input for division in signed 32-bit format	R/W	0

18.4.4 ACC Dividend Input for Division register (ACC_DIV_DVD)

Address Offset: 0x0C

Bit	Name	Description	Attribute	Reset
31:0	DIV_DVD	Dividend input for division in signed 32-bit format	R/W	0

18.4.5 ACC Divisor Input for Division register (ACC_DIV_DVS)

Address Offset: 0x10

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	DIV_DVS	Divisor input for division in signed 16-bit format	R/W	0

18.4.6 ACC Quotient Output for Division register (ACC_DIV_QUO)

Address Offset: 0x14

Bit	Name	Description	Attribute	Reset
31:0	DIV_QUO	Quotient output of division in signed 32-bit format	R/W	0

18.5 ACC Remainder Output register (ACC_DIV_REM)

Address Offset: 0x18

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	DIV_REM	Remainder output of division in signed 16-bit format	R/W	0

18.6 ACC SQRT Input register (ACC_SQRT_IN)

Address Offset: 0x1C

Bit	Name	Description	Attribute	Reset
31:0	SQRT_IN	Square root input value in unsigned 32-bit format	R/W	0

18.7 ACC SQRT Output register (ACC_SQRT_OUT)

Address Offset: 0x20

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	SQRT_OUT	Square root output value in unsigned 16-bit format	R/W	0

19 CYCLIC REDUNDANCY CHECK (CRC)

19.1 OVERVIEW

The CRC (cyclic redundancy check) calculation unit is used to get a CRC code from 16- or 32-bit data word and a generator polynomial. Among other applications, CRC-based techniques are used to verify data transmission or storage integrity. The CRC calculation circuit helps compute a signature of the software during runtime, to be compared with a reference signature generated at link time and stored at a given memory location.

19.2 FEATURES

1. Support

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

   CRC-16 polynomial: $X^{16}+X^{15}+X^2+1$

   CRC-16-CCITT polynomial: $X^{16}+X^{12}+X^5+1$

	CRC-16-CCITT $X^{16}+X^{12}+X^5+1$	**CRC-16** $X^{16}+X^{15}+X^2+1$	**CRC-32(-IEEE802.3)** $X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^{7+}X^5+X^4+X^2+X+1$
Poly	0x1021	0x8005	0x04C11DB7
Seed(Init)	0xFFFF	0x0000	0xFFFFFFFF
XOROut	0x0000	0x0000	0xFFFFFFFF
RefIn	No	Yes	Yes
RefOut	No	Yes	Yes

2. Handles 16-, 32-bit data size
3. Single input/output 32-bit data register
4. Input buffer to avoid bus stall during calculation
5. CRC computation done in 4T IHRC clock cycles for the 32-bit data size
6. Polynomial representations of cyclic redundancy checks

19.3 CRC REGISTERS

Base Address: 0x4003 8000

19.3.1 CRC Control register (CRC_CTRL)

Address offset: 0x00

Bit	Name	Description	Attribute	Reset
31:6	Reserved	Reserved	R/W	0
4	BUSY	CRC calculation busy flag. 0: CRC calculation Idle/Finished. 1: CRC calculation is in process.	R	0
3	URCRCEN	Enable bit of the CRC calculation for the User ROM, except the last page. 1: Start the CRC operation for the User ROM, except the last page. This bit is set only by SW and reset by HW. 0: Stop/Finish operation.	R/W	0
2	RESET	Reset bit 0: No effect. 1: Reset the CRC calculation circuit. (Reset the initial seed value and BUSY bit to 0). Clear this bit when the reset operation had finished by HW.	R/W	0
1:0	CRC[1:0]	CRC Polynomial 00: CRC-16-CCITT 01: CRC-16 10: CRC-32 11: Reserved	R/W	0

19.3.2 CRC Data register (CRC_DATA)

Address offset: 0x04

✸ ***Note: Support 8-bit (Byte) Write ONLY!***

Bit	Name	Description	Attribute	Reset
31:0	DATA[31:0]	Data to be input or read. Write: Input 8-bit data to the CRC calculator and start to calculation process. Read: Output the previous CRC calculation result depends on the CRC Polynomial.	R/W	0

20 FLASH

20.1 OVERVIEW

SONiX 32-bit MCU integrated device feature in-system programmable (ISP) FLASH memory for convenient, upgradeable code storage. The FLASH memory may be programmed via the SONiX 32-bit MCU programming interface or by application code for maximum flexibility. SONiX 32-bit MCU provides security options at the disposal of the designer to prevent unauthorized access to information stored in FLASH memory.

- The MCU is stalled during Flash program and erase operations, although peripherals (Timers, WDT, I/O, PWM, etc.) remain active.
- Watchdog timer should be cleared if enabled before the Flash write or erase operation.
- The erase operation sets all the bits in the Flash page to logic 1.
- HW will hold system clock and automatically move out data from RAM and do programming, after programming finished, HW will release system clock and let MCU execute the next instruction.

20.2 EMBEDDED FLASH MEMORY

The Flash memory is organized as 32-bit wide memory cells that can be used for storing both code and data constants, and is located at a specific base address in the memory map of chip.

The high-performance Flash memory module in chip has the following key features:

- Memory organization: the Flash memory is organized as a User ROM, Boot ROM.

User ROM	8K × 32 bits divided into 64 pages of 512 Bytes
Boot ROM	1K × 32 bits divided into 8 pages of 512 Bytes

The Flash interface implements instruction access and data access based on the AHB protocol. It implements the logic necessary to carry out Flash memory operations (Program/Erase). Program/Erase operations can be performed over the whole product voltage range.

20.3 FEATURES

- Read interface (32-bit)
- Flash Program / Erase operation
- Code Option includes Code Security (CS)

Write operations to the main memory block and the code options are managed by an embedded Flash Memory Controller (FMC). The high voltage needed for Program/Erase operations is internally generated. The main Flash memory can be read/write protected against different levels of Code Security (CS).

During a write operation to the Flash memory, any attempt to read the Flash memory will stall the bus. The read operation will proceed correctly once the write operation has completed. This means that code or data fetches cannot be made while a write/erase operation is ongoing.

For write and erase operations on the Flash memory, the IHRC will be turn ON by FMC. The Flash memory can be programmed and erased using ICP and ISP.

✸ ***Note: When using ISP of FMC in debug mode, the system clock must be higher than 12MHz.***

20.4 ORGANIZATION

Block	Name	Base Address	Size (Byte)
User ROM	Page 0	0x00000000 ~ 0x000001FF	512
	Page 1	0x00000200 ~ 0x000003FF	512
	. . .	. . .	. . .
	Page 63	0x00007E00 ~ 0x00007FFF	512
Boot Loader	Page 0	0x1FFF0000 ~ 0x1FFF01FF	512
	Page 1	0x1FFF0200 ~ 0x1FFF03FF	512
	. . .	. . .	. . .
	Page 7	0x1FFF0E00 ~ 0x1FFF0FFF	512

20.5 READ

The embedded Flash module can be addressed directly, as a common memory space. Any data read operation accesses the content of the Flash module through dedicated read senses and provides the requested data.

The read interface consists of a read controller on one side to access the Flash memory, and an AHB interface on the other side to interface with the CPU. The main task of the read interface is to generate the control signals to read from the Flash memory as required by the CPU.

20.6 PROGRAM/ERASE

The Flash memory erase operation can be performed at page level.

To ensure that there is no over-programming, the Flash programming and erase controller blocks are clocked by IHRC.

20.7 EMBEDDED BOOT LOADER

The embedded boot loader is used to reprogram the Flash memory using the UART0 serial interface. This program is located in the Boot ROM and is programmed by SONiX during production.

20.8 FLASH MEMORY CONTROLLER (FMC)

The FMC handles the program and erase operations of the Flash memory.

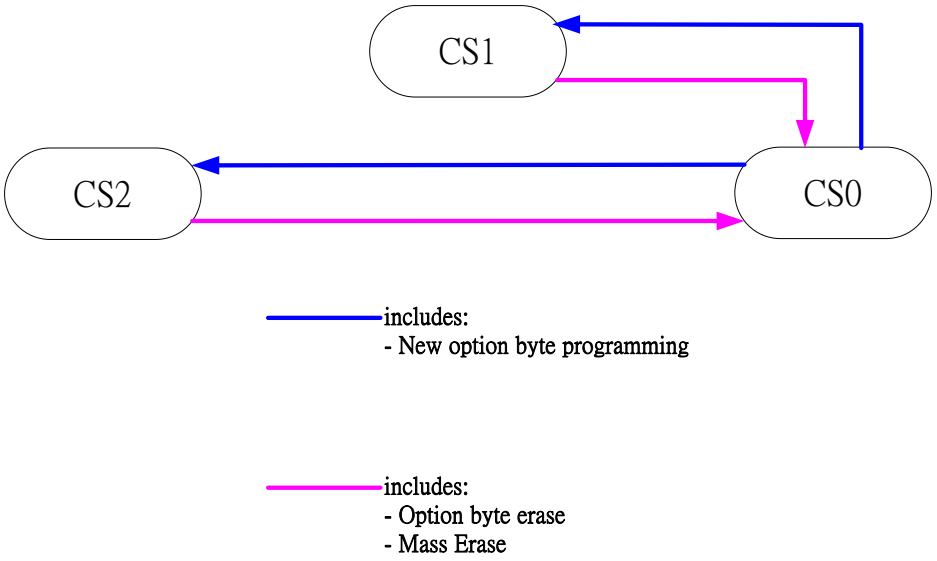
20.8.1 Code Security (CS)

Code Security is a mechanism that allows the user to enable different levels of security in the system so that access to the on-chip Flash and use of the ISP can be restricted.

✸ ***Note: Any Code Security change becomes effective only after the MCU has been Reboot.***

User ROM		CS0	CS1	CS2	Description
WRITER	Read	O	X	X	
	Erase	O	O(*)	O(*)	(*) WRITER will change the CS level to CS0.
	Program	O	O	O	
FW (EEPROM emulation)	Read	O	O	O	
	Erase	O	O	O	
	Program	O	O	O	
SWD	Read	O	X	X	
	Erase	O	X	X	
	Program	O	X	X	

✸ ***Note: User may try to change security level from CS2 to CS0, or from CS1 to CS0. HW shall:***
1. ***Mass erase the User ROM first. User shall NOT execute this operation in debug mode, since the SWD communication may fail during the mass erase procedure.***
2. ***Update security level.***



20.8.2 Program FLASH Memory

The Flash memory can be programmed 32 bits (4 bytes) at a time. CPU can program the main Flash memory by performing standard page write operations. The PG bit in the FLASH_CTRL register must be set. When the data is filled in the FLASH _DATA register, FMC preliminarily increases the data address, and checks the address to be programmed. If the following errors happen, the program operation is skipped and a warning is issued by the PGERR bit in FLASH_STATUS register.

- Start to Erase/Program and find that the address is over page boundary
- Start to Erase/Program and find that the address is illegal (>ROM size)
- Fill in Data and the address is already over Page Boundary
- The address to be programmed contains a value different from 0xFFFFFFFF before programming.

The main Flash memory programming sequence in standard mode is as follows:

1. Set the PG bit in the FLASH_CTRL register.
2. Fill in the target address in the FLASH_ADDR register.
3. Wait for the BUSY bit to be reset.
4. Perform the continuous data write until all of the data had been filled in the FLASH_DATA register.
5. Wait for the BUSY bit to be reset.
6. Set the START bit to start programming.
7. Wait for the BUSY bit to be reset.
8. (Optional) Read the programmed value and verify.

20.8.3 Erase

The Flash memory can be erased page by page.

20.8.3.1 Page Erase

A page of the Flash memory can be erased using the Page Erase feature of the FMC. To erase a page, the procedure below should be followed:

1. Set the PER bit in the FLASH_CTRL register
2. Program the FLASH_ADDR register to select a page to be erased
3. Set the START bit in the FLASH_CTRL register
4. Wait for the BUSY bit to be reset
5. (Optional) Read the erased page and verify

20.8.3.2 Mass Erase

When the Flash memory read protection is changed from protected to unprotected, a Mass Erase of the User ROM is performed by HW before reprogramming the read protection option.

20.9 READ PROTECTION

The read protection is activated by setting the Code Security bytes in Code option.

When the Flash memory read protection is changed from protected to unprotected, a Mass Erase of the User ROM is performed by HW before reprogramming the read protection option.

20.10 HW CHECKSUM

HW checksum is the checksum of User ROM. If the read protection is enabled, the users can still readout the HW checksum through Writer or ISP AP.

20.11 FMC REGISTERS

Base Address: 0x4006 2000

20.11.1 Flash Low Power Control register (FLASH_LPCTRL)

Address offset: 0x00

Reset value: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:16	FMCKEY	FMC verify key. Read as 0. When writing to the register you must write 0x5AFA to FMCKEY, otherwise behavior of writing to the register is ignored.	W	0
15:6	Reserved		R	0
5:0	LPMODE[5:0]	Flash Low Power mode selection bit 000000b: HCLK≦24MHz 011001b: 24MHz<HCLK≦48MHz 101001b: 48MHz<HCLK Other: Reserved, and may cause unexpected error to force MCU enter Hardfault handler.	R/W	000b

20.11.2 Flash Status register (FLASH_STATUS)

Address offset: 0x04

Reset value: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	ERR	Error flag 0: Read→No error. Write→Clear this flag. 1: Set by HW when - Start to Erase/Program and find that the address is over page boundary - Start to Erase/Program and find that the address is illegal. (>ROM size) - Fill in Data and the address is already over Page Boundary - The address to be programmed contains a value different from 0xFFFFFFFF before programming.	R/W	0
1	Reserved		R	0
0	BUSY	Busy flag 0: Flash operation is not busy. 1: Flash operation is in progress. This is set on the beginning of a Flash operation and reset when the operation finishes or when an error occurs by HW.	R	0

20.11.3 Flash Control register (FLASH_CTRL)

Address offset: 0x08

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	CHK	Checksum calculation chosen This bit is set only by SW and reset when the BUSY bit resets.	R/W	0
6	START	Start Erase/Programming operation 1: Triggers an Erase/Programming operation. This bit is set only by SW and resets when the BUSY bit resets. 0: Stop/Finish operation.	R/W	0
5:3	Reserved		R	0
2	MER	Mass erase chosen mode bit Erase of all user pages chosen.	R/W	0

		This bit is set only by SW and reset when the BUSY bit resets.		
1	PER	Page Erase mode chosen. This bit is set only by SW and reset when the BUSY bit resets.	R/W	0
0	PG	Flash Programming mode chosen. This bit is set only by SW and reset when the BUSY bit resets.	R/W	0

20.11.4 Flash Data register (FLASH_DATA)

Address offset: 0x0C

For Page Program operations, this should be updated by SW to indicate the data to be programmed.

Bit	Name	Description	Attribute	Reset
31:0	DATA[31:0]	Data to be programmed.	R/W	0

20.11.5 Flash Address register (FLASH_ADDR)

Address offset: 0x10

The Flash address to be erased or programmed should be updated by SW, and the PG bit or PER bit shall be set before filling in the Flash address.

✸ ***Note: Write access to this register is blocked when the BUSY bit in the FLASH_STATUS register is set.***

Bit	Name	Description	Attribute	Reset
31:0	FAR[31:0]	Flash Address Choose the Flash start address to erase when Page Erase is selected, or to program when Page Program is selected.	R/W	0

20.11.6 Flash Checksum register (FLASH_CHKSUM)

Address offset: 0x14

Bit	Name	Description	Attribute	Reset
31:16	BRCHKSUM[15:0]	Checksum of Boot ROM.	R	0
15:0	URCHKSUM[15:0]	Checksum of User ROM.	R	0

20.11.7 Flash Write Protect register (FLASH_WP)

Address offset: 0x20

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	WP[15:0]	Flash Write protect key. 0x5AFA: The User ROM, except the last page, is Write-protected (Not able to be erased and programmed by FW). Other: The whole User ROM can be accessed.	R/W	0

21 SERIAL-WIRE DEBUG (SWD)

21.1 OVERVIEW

SWD functions are integrated into the ARM Cortex-M0. The ARM Cortex-M0 is configured to support up to four breakpoints and two watch points.

21.2 FEATURES

- Supports ARM Serial Wire Debug (SWD) mode.
- Direct debug access to all memories, registers, and peripherals.
- No target resources are required for the debugging session.
- Up to four breakpoints.
- Up to two data watch points that can also be used as triggers.

21.3 PIN DESCRIPTION

Pin Name	Type	Description	GPIO Configuration
SWCLK	I	Serial Wire Clock pin in SWD mode.	
SWDIO	I/O	Serial Wire Data Input/Output pin in SWD mode.	

21.4 DEBUG NOTE

21.4.1 LIMITATIONS

Debug mode changes the way in which reduced power modes work internal to the ARM Cortex-M0 CPU, and this ripples through the entire system. These differences mean that power measurements should not be made while debugging, the results will be higher than during normal operation in an application.

During a debugging session, the SysTick Timer is automatically stopped whenever the CPU is stopped. Other peripherals are not affected.

21.4.2 DEBUG RECOVERY

User code may disable SWD function in order to use P3.5 and P3.6 as GPIO, and may not debug by SWD function to debug or download FW any more.

SONiX provide Boot loader to check the status of P1.3 (BOOT pin) during boot procedure. If P1.3 is Low during Boot procedure, MCU will execute code in Boot loader instead of User code, so SWD function is not disabled.

Exit Boot loader, user code can still configure P1.3 as other functions such as GPIO.

✷ ***Note: We strongly recommanded NOT using BOOT pin as output pin to drive the LED, otherwise, the BOOT pin status may be low during boot procedure.***

21.4.3 INTERNAL PULL-UP/DOWN RESISTORS on SWD PINS

To avoid any uncontrolled IO levels, the device embeds internal pull-up and pull-down resistor on the SWD input pins:

- SWDIO/JTMS: Internal pull-up
- SWCLK/JTCK: Internal pull-down

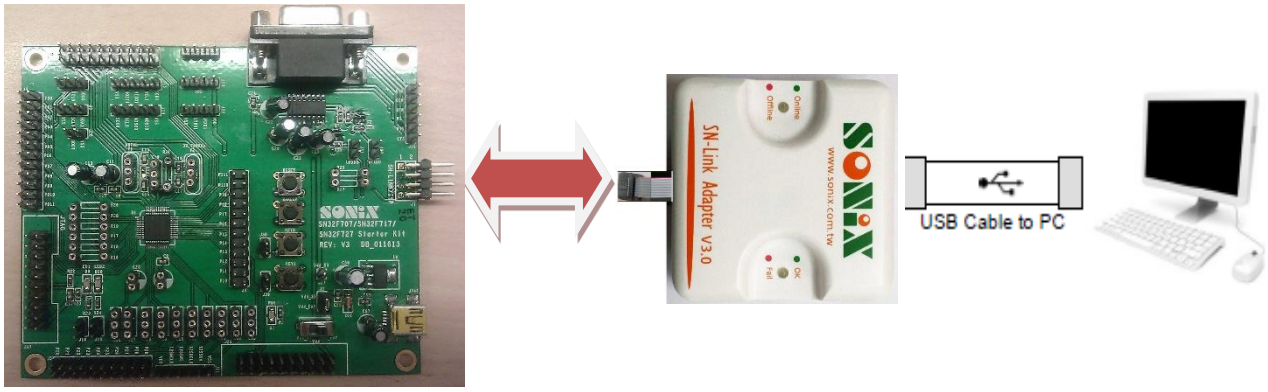
Once a SWD function is disabled by SW, the GPIO controller takes control again.

22 DEVELOPMENT TOOL

SONiX provides an Embedded ICE emulator system to offer 32-bit series MCU firmware development.

SONiX 32-bit series Embedded ICE Emulator System includes:

- SONiX 32-bit MCU Starter-Kit.
- SN-LINK-V3
- USB cable to provide communications between the SN-LINK-V3 and PC.
- IDE Tools (KEIL RVMDK)



SONiX 32-bit MCU Starter-Kit. SN-LINK-V3 IDE Tools

SONiX 32-bit series Embedded ICE Emulator Feature:

- Target's Operating Voltage: 1.8V~5.5V.
- Up to 4 hardware break points.
- System clock rate up to 60MHz.
- Oscillator supports IHRC, ILRC, EHS/ELS X'tal.

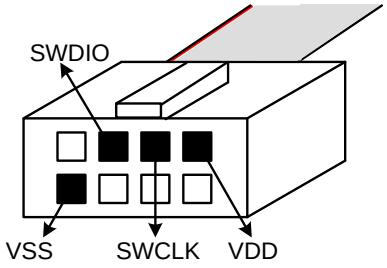
SONiX 32-bit series Embedded ICE Emulator Limitation:

- SWCLK and SWDIO pins are shared with GPIO pins. In embedded ICE mode, the shared GPIO function can't work.

22.1 SN-LINK-V3

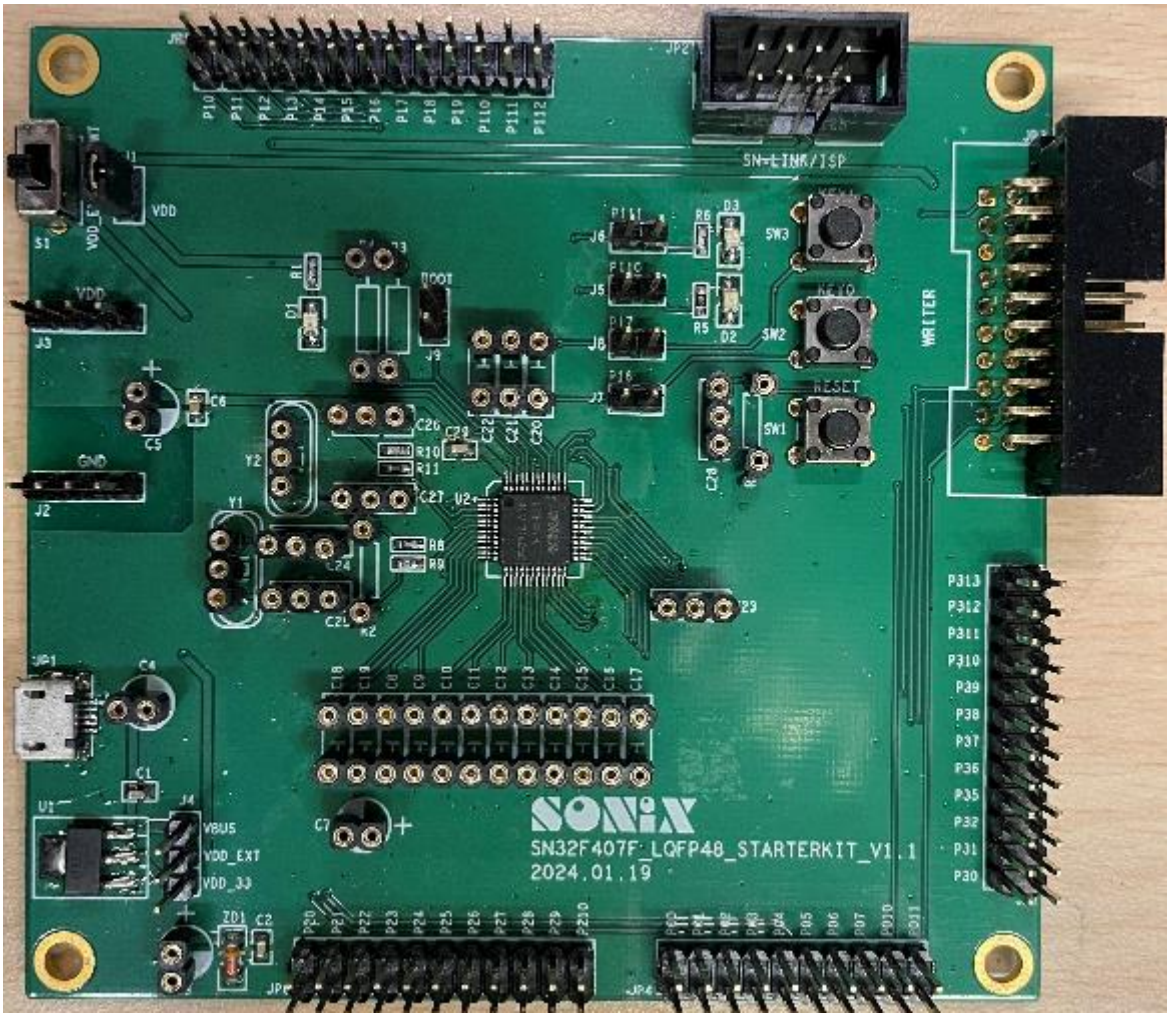
SN-LINK-V3 is a high speed emulator for SONiX 32-bit MCU. It debugs and programs based on SWD protocol. In addition to debugger functions, the SN-LINK-V3 also may be used as a programmer to load firmware from PC to MCU for engineering production, even mass production.

SN-LINK-V3 communicates with SONiX 32-bit MCU through SWD interface. The pin definition of the Modular cable is as following:



22.2 SN32F400 STARTER-KIT

SONiX 32-bit MCU Starter-kit is an easy-development platform. It includes real chip and I/O connectors to input signal or drive extra device of user's application. It is a simple platform to develop application as target board not ready. The starter-kit can be replaced by target board because of integrated SWD debugger circuitry.



- JP1: Micro USB connector.
- S1: VDD power source is 3.3V/5.0V from board, Writer, or external power.
- U2: SN32F407 real chip.
- D1: Power LED.
- C8~C19: 12-ch ADC capacitors.
- RESET button: External reset trigger source.
- Y1: External high-speed X'tal
- Y2: External low-speed 32.768KHz X'tal
- JP2: SN-LINK connector
- J9: Short to force MCU stay in Boot loader.

23 ELECTRICAL CHARACTERISTIC

23.1 ABSOLUTE MAXIMUM RATING

Supply voltage (Vdd)..... - 0.3V ~ 5.5V
Input in voltage (Vin)..... Vss – 0.2V ~ Vdd + 0.2V
Operating ambient temperature (Topr)..... -40℃ ~ + 105℃
Storage ambient temperature (Tstor) ..... –40℃ ~ + 125℃

23.2 ELECTRICAL CHARACTERISTIC

All of voltages refer to Vss, Typical Vdd = 3.3V, Fosc = 12MHz, ambient temperature is 25℃ unless otherwise note.

PARAMETER	SYM.	DESCRIPTION		MIN.	TYP.	MAX.	UNIT
Operating Voltage	Vdd	Supply voltage for core and external rail		1.8	3.3	5.5	V
VDD rise rate	V_{POR}	VDD rise rate to ensure internal power-on reset		0.05	-	-	V/ms
Power Consumption							
Supply Current	Idd1	Normal mode	System clock = 12MHz [1][2][3]	-	2	-	mA
			System clock = 24MHz [1][3][4]	-	4	-	mA
			System clock = 36MHz [1][3][4]	-	4.5	-	mA
			System clock = 48MHz [1][3][4]	-	6	-	mA
			System clock = 60MHz [1][3][4]	-	7	-	mA
	Idd2	Sleep Mode	System clock = 12MHz [1][2][3][5]	-	800	-	uA
	Idd3		System clock = 32KHz [1][3][5]		100		uA
	Idd4	Deep-sleep Mode	Vdd=3.3V [1][3][5]	-	1	5	uA
Port Pins, RESET pin							
High-level input voltage	V_{IH}			0.7Vdd	-	Vdd	V
Low-level input voltage	V_{IL}			Vss	-	0.3Vdd	V
Input voltage	V_i			0	-	Vdd	V
Output voltage	V_o			0	-	Vdd	V
I/O port pull-up resistor	R_{PU}	Vin = Vss , Vdd = 5.0V Vin = Vss , Vdd = 3.3V		40 50	60 75	80 100	KΩ
I/O port pull-down resistor	R_{PD}	Vin = 5.0V Vin = 3.3V		25 20	35 25	50 35	KΩ
I/O High-level output source current	I_{OH1}	Standard port and RESET pins, except P3.10 and P3.11	V_{OP} = Vdd– 0.5V	23	30	50	mA
	I_{OH2}	P3.10, P3.11		10	12	50	mA
I/O Low-level output sink current	I_{OL1}	Standard port and RESET pins, except P3.10 and P3.11	V_{OP} = Vss + 0.5V	25	30	50	mA
	I_{OL2}	P3.10, P3.11			12		mA

ADC						
ADC Operating Voltage	V_{ADC}		2.0	-	5.5	V
AIN0 ~ AIN11 input voltage	V_{ani}		0	-	Avrefh	V
ADC Resolution	RES		10		12	Bit
ADC reference Voltage	V_{ref}		2.0	-	V_{ADC}	V
*ADC enable time	T_{ast}	Ready to start convert after set ADENB = "1"	100	-	-	us
*ADC current consumption	I_{ADC}	Vdd=3.3V, ADS=0	-	500	-	uA
ADC Clock Frequency	F_{ADCLK}		-	-	16	MHz
ADC Conversion Cycle Time	F_{ADCYL}		16	-	-	$1/F_{ADCLK}$
ADC Sampling Rate	F_{ADSMP}		-	-	1000	KHz
Differential Nonlinearity	DNL	Vdd= AVREFH=5.5V	-1	-	+1	LSB
Integral Nonlinearity	INL	Vdd= AVREFH=5.5V	-2	+/-1	+2	LSB
No Missing Code	NMC	Vdd= AVREFH=5.5V	10 [6]	-	12	Bit
ADC offset Voltage	$V_{ADCoffset}$		-5	-	+5	mV
ADC Internal voltage reference	$V_{ADCIREF}$	Internal VDD reference voltage, Vdd=2.0V~5.5V		VDD		V
		Internal 4.5V reference voltage, Vdd=5.0V~5.5V	4.47	4.5	4.53	V
		Internal 3V reference voltage, Vdd=3.5V~5.5V	2.98	3	3.02	V
		Internal 2V reference voltage, Vdd=2.5V~5.5V	1.986	2	2.013	V
		Internal 1.5V reference voltage, Vdd=2.0V~5.5V	1.49	1.5	1.51	V
COMPARATOR						
CMP Operating Voltage	V_{CMP}		1.8		5.5	V
Supply Current	I_{COMP}	Vdd=5.5V		75	100	uA
Input Offset Voltage	Vos	Vcm=Vdd/2	-2		+2	mV
Response Time	Trs	Positive input voltage = 1/2*Vdd. Negative input voltage transitions from Vss to Vdd.		50*	100	ns
Output Slew Rate Time	Tsr	Vo=rising Vss~Vdd or falling Vdd~Vss. Vdd=5V.	-	20	-	ns
Common Mode Input Voltage Range	Vcmr	Vdd=5.5V	Vss+0.5		Vdd-0.5	V
CMP Internal voltage reference	V_{DAC}	Internal DAC output voltage, N = DATA[7:0]		N*V_{CMP} /256		V
OPA						
Power Supply Range	V_{OPA}		1.8		5.5	V
Supply Current	I_{OPA}	Vdd=5V. Unique gain buffer (Vin=Vss).			1.5	mA
Common Mode Input Voltage Range	V_{CM}	Vdd=5.0V	Vss-0.3		Vdd+0.3	V
Input Offset Voltage	V_{OFFSET}	V_{CM}=Vss			±15	mV
Power Supply Rejection Ratio	PSRR	V_{CM}=Vss	70	85	95	dB
Common Mode Rejection Ratio	CMRR	V_{CM}=-0.3V~2.5V. Vdd=5V.	70	85	-	dB
		V_{CM}=-0.3V~5.3V. Vdd=5V.	65	80	-	dB
Phase Margin	Φm	C_L=100pF		60		deg
Gain Margin	Gm			10		dB
Unity Gain Bandwidth	B1	C_L=100pF	6	10		MHz
Output Slew Rate	T_{OSR}	Output voltage transitions from Vss to Vdd.			0.5	us
		Output voltage transitions from Vdd to Vss.			0.5	us
Open-Loop Gain (Large Signal)	A_{OL}	Vout=0.2V~Vdd-0.2V. V_{CM} =Vss.	90	110	-	dB
Maximum Output Voltage Swing	Vol, Voh		Vss+100		Vdd-100	mV
Output Short Current	I_{SC}			±25		mA
FLASH						
Supply Voltage	Vdd1		1.35	1.50	1.65	V
Endurance time	T_{EN}	Erase + Program	20K	*100K	-	Cycle
Page Erase current	I_{PER}		-	1	2	mA
Program current	I_{PG}		-	-	7	mA
Page erase time	T_{PE}	1-Page (512 bytes)	4	-	5	ms
Mass erase time	T_{MER}		20	-	40	ms
2-Word Programming time	T_{PG}	2-Word (64 bits)	48		60	us
MISC						
Low Voltage Detector	LVD	Interrupt/Reset Level 1	2.00	2.10	2.20	V
		Level 2	2.40	2.50	2.60	V

			Level 3	2.80	2.90	3.00	V
			Level 4	3.20	3.30	3.40	V
			Level 5	3.60	3.70	3.80	V
			Level 6	4.00	4.10	4.20	V
			Level 7	4.40	4.50	4.60	V
ESD_HBM	V_{ESD_HBM}	ESD human body mode		8000	-	-	V
ESD_MM	V_{ESD_MM}	ESD machine mode		200	-	-	V
ESD_CDM	V_{ESD_CDM}	ESD charged device model		750	-	-	V
IHRC Freq.	F_{IHRC}	*T=25*℃, Vdd=1.8V~ 5.5V		47.76	48	48.24	MHz
		T=-40℃*~105*℃, Vdd=1.8V~5.5V		46.80	48	49.20	MHz

*** These parameters are for design reference, not tested.**

[1] IDD measurements were performed with all pins configured as GPIO outputs driven LOW and pull-up resistors disabled, code while(1); executed, and VDD = 3.3V

[2] IHRC and ILRC are enabled, external X'tal are disabled, and PLL is disabled.

[3] LVD and all peripherals are disabled.

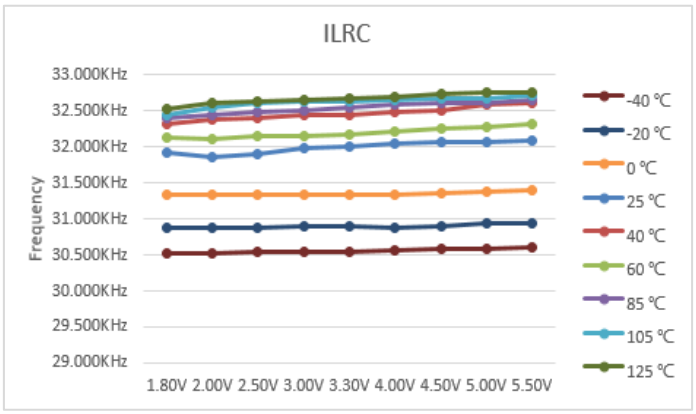
[4] IHRC is disabled, external high X'tal is enabled, and PLL is enabled.

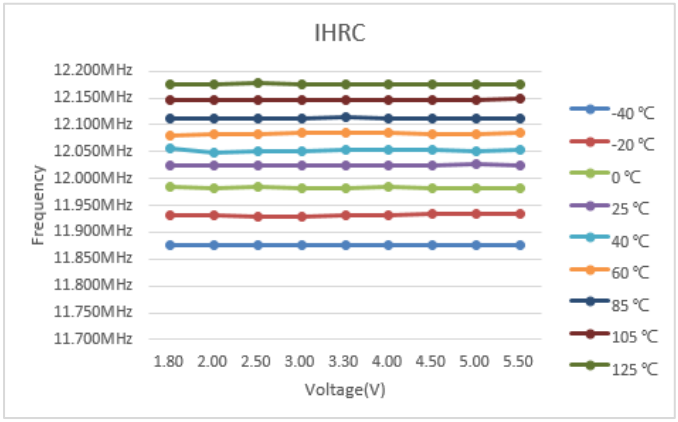
[5] All oscillators and analog blocks are turned off.

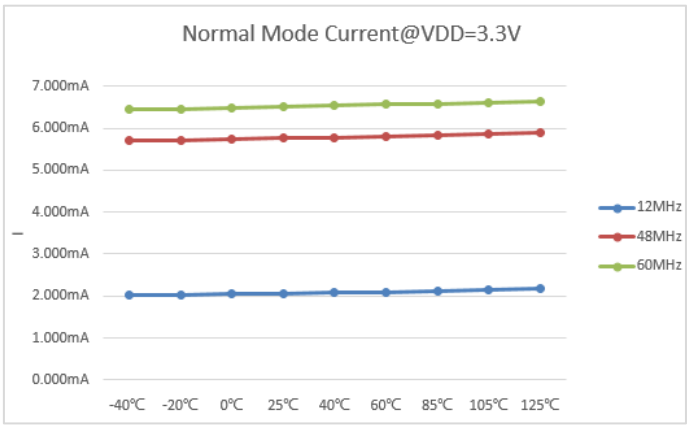
[6] The resolution of ADC would be only 8-bit when HCLK = EHS.

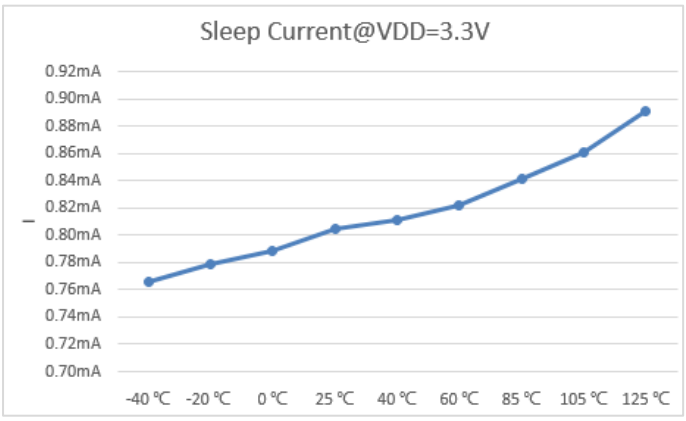
23.3 CHARACTERISTIC GRAPHS

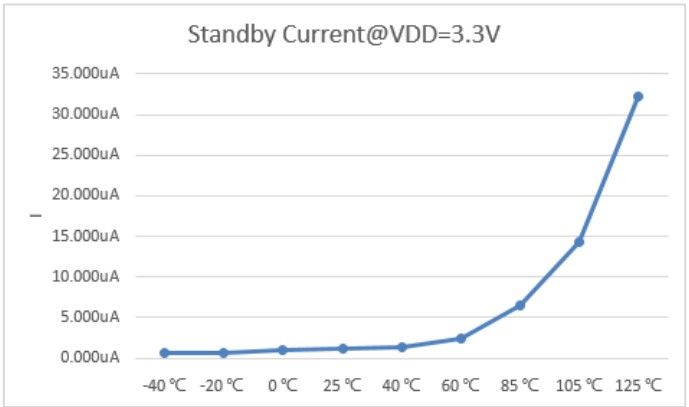
The Graphs in this section are for design guidance, not tested or guaranteed. In some graphs, the data presented are outside specified operating range. This is for information only and devices are guaranteed to operate properly only within the specified range.

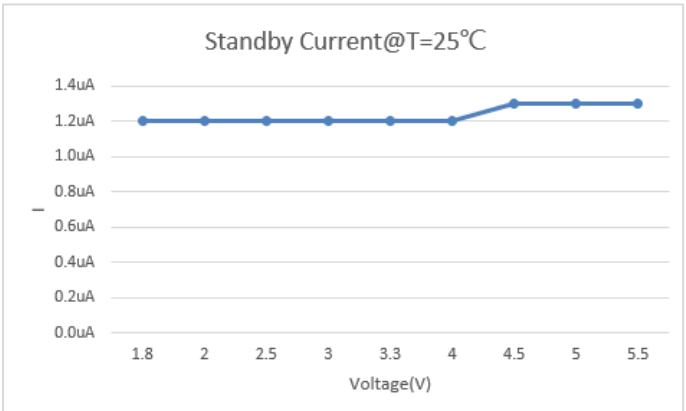


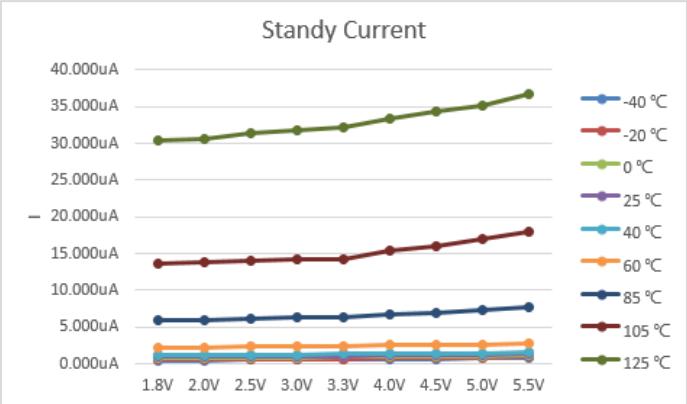












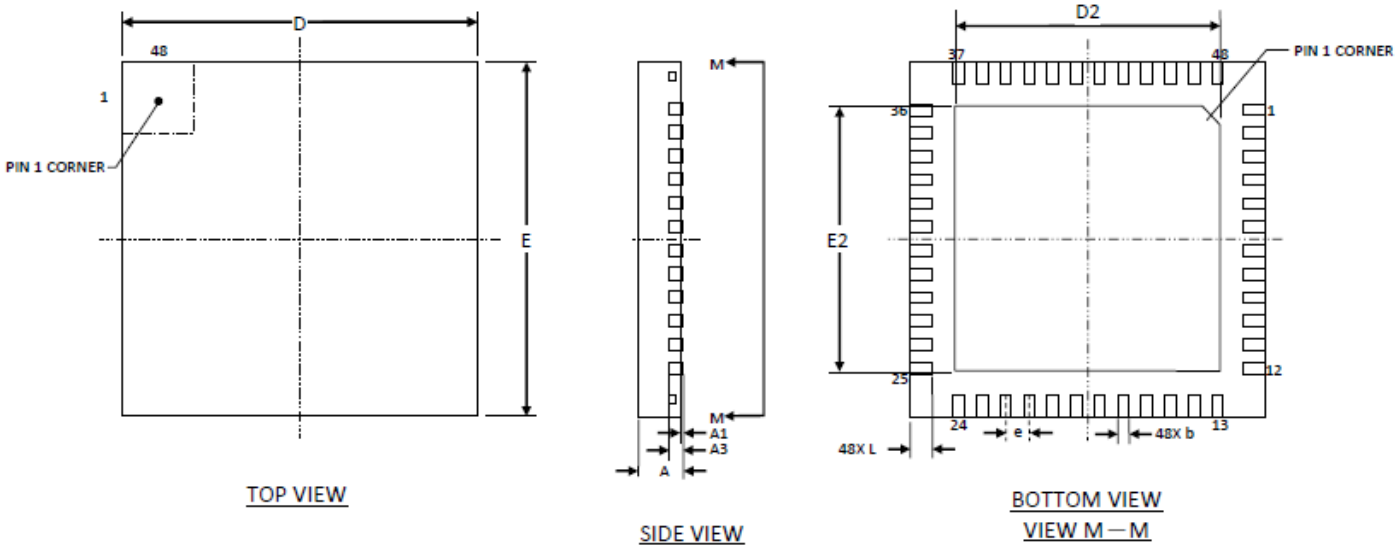
*Supply Current V.S. Operating Temperature (Operating Conditions: All pins configured as GPIO outputs driven Low and pull-up resistors disabled and VDD = 3.3V)

24 FLASH ROM PROGRAMMING PIN

Programming Information of SN32F400 Series											
Chip Name		SN32F407		SN32F405		SN32F403					
Writer Connector JP5											
Number	Name	Number	Pin	Number	Pin	Number	Pin	Number	Pin	Number	Pin
1	VDD	48	VDD	1	VDD	1	VDD				
2	GND	47	VSS	32	VSS	25	VSS				
3	CLK	43	P0.11	30	P0.11	23	P0.11				
4	CE										
5	PGM	33	P3.6	25	P3.6	18	P3.6				
6	OE	32	P3.5	24	P3.5	17	P3.5				
7	D1										
8	D0										
9	D3										
10	D2										
11	D5										
12	D4										
13	D7										
14	D6										
15	VDD										
16	-										
17	HLS										
18	RST										
19	-										
20	ALSB/PDB	42	P0.10	29	P0.10	22	P0.10				

25 PACKAGE INFORMATION

25.1 QFN 48 PIN 6x6

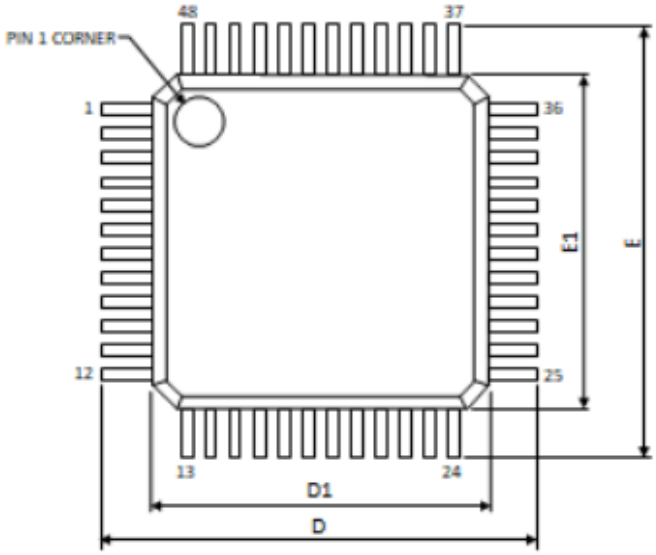


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.80	0.90	0.028	0.031	0.035
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3		0.203 REF			0.008 REF	
b	0.15	0.20	0.25	0.006	0.008	0.010
D		6.00 BSC			0.236 BSC	
E		6.00 BSC			0.236 BSC	
e		0.40 BSC			0.016 BSC	
D2	3.70	4.20	4.70	0.146	0.165	0.185
E2	3.70	4.20	4.70	0.146	0.165	0.185
L	0.30	0.40	0.50	0.012	0.016	0.020

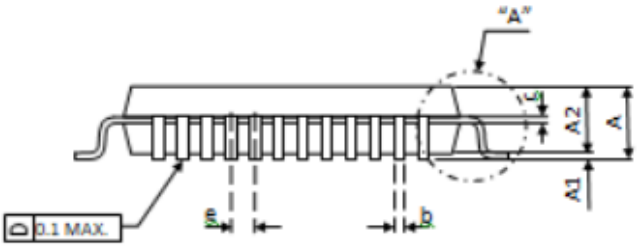
Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)

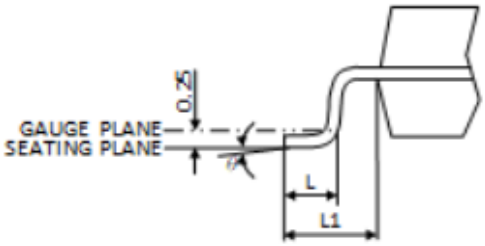
25.2 LQFP 48 PIN



TOP VIEW



SIDE VIEW



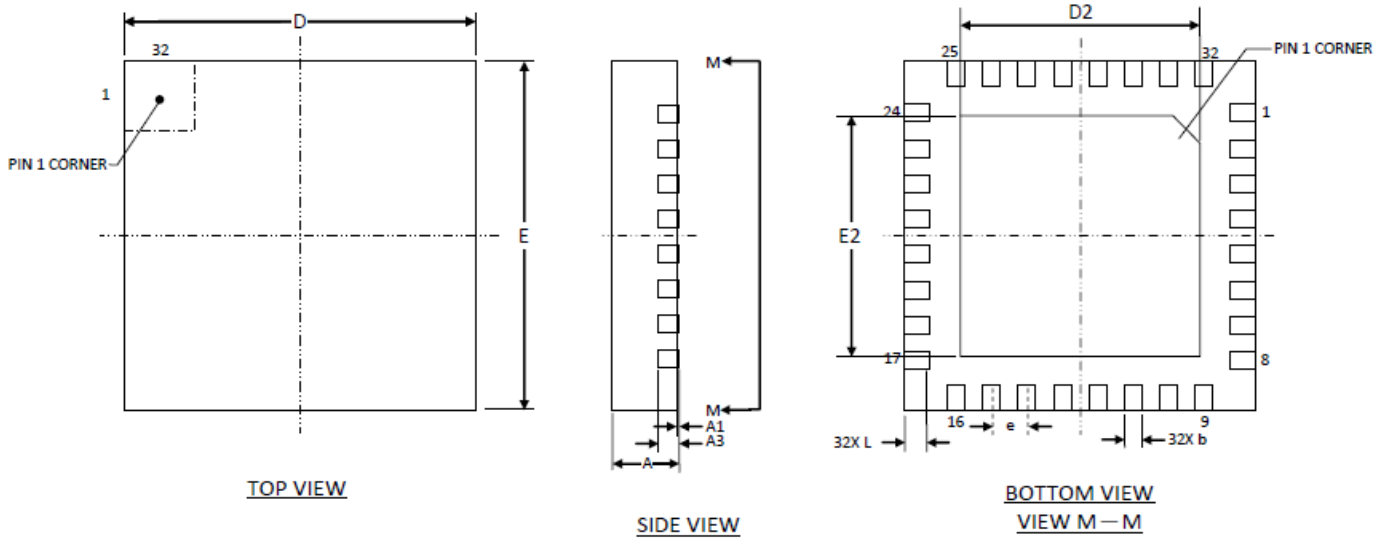
DETAIL "A"

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.60	--	--	0.063
A1	0.05	--	0.15	0.002	--	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.17	0.22	0.27	0.007	0.009	0.011
c	0.09	--	0.20	0.004	--	0.008
D	9.00 BSC			0.354 BSC		
D1	7.00 BSC			0.276 BSC		
E	9.00 BSC			0.354 BSC		
E1	7.00 BSC			0.276 BSC		
e	0.50 BSC			0.020 BSC		
L	0.40	0.60	0.80	0.016	0.024	0.031
L1	1.00 REF			0.039 REF		
θ	0°	3.5°	7°	0°	3.5°	7°

Notes :

1. CONTROLLING DIMENSION : MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

25.3 QFN 32 PIN 4x4

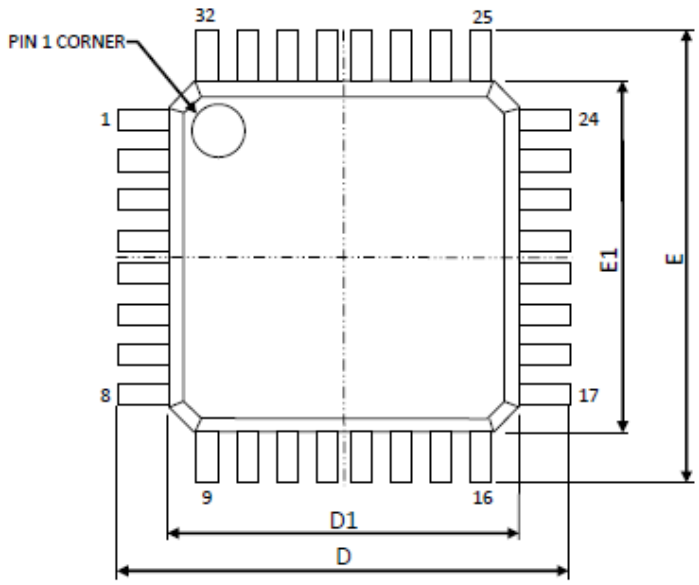


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.80	0.90	0.028	0.031	0.035
A1	0.00	0.02	0.05	0.000	0.000	0.002
A3		0.20 REF			0.008 REF	
b	0.15	0.20	0.25	0.006	0.008	0.010
D		4.00 BSC			0.157 BSC	
E		4.00 BSC			0.157 BSC	
e		0.40 BSC			0.016 BSC	
D2	2.00	2.45	2.9	0.080	0.096	0.114
E2	2.00	2.45	2.9	0.080	0.096	0.114
L	0.25	0.35	0.45	0.010	0.013	0.017

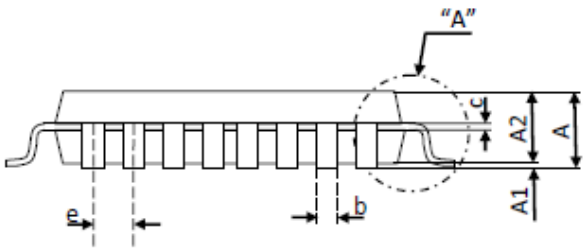
Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)

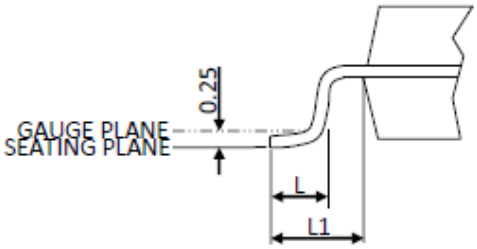
25.4 LQFP 32 PIN



TOP VIEW



SIDE VIEW



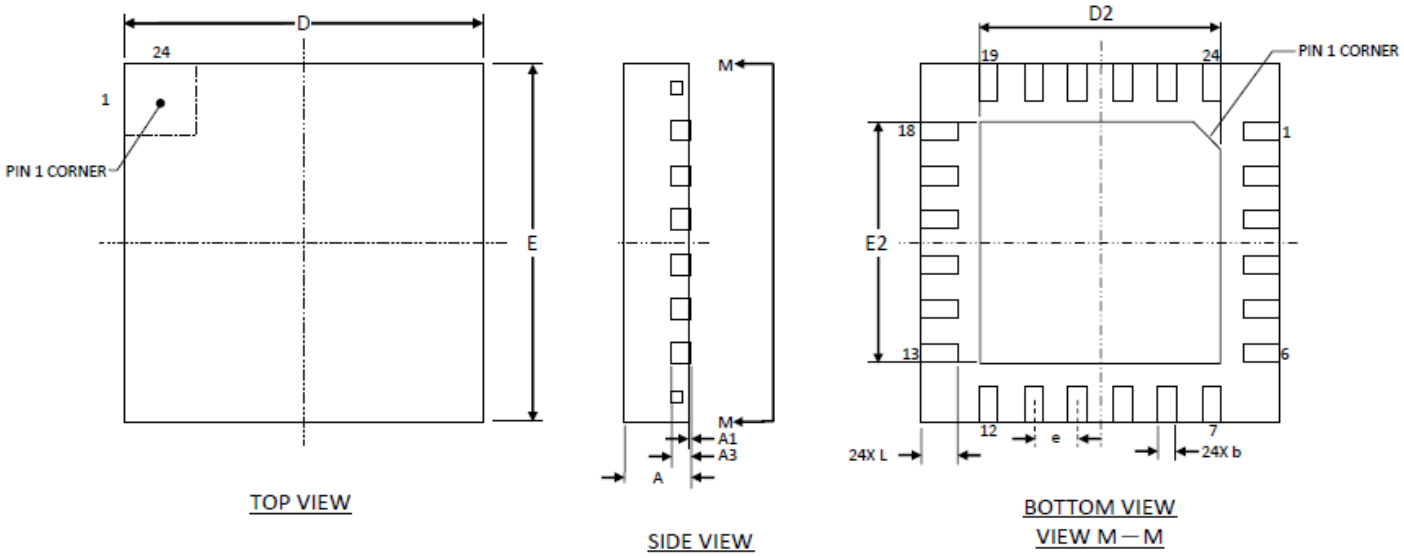
DETAIL "A"

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.60	--	--	0.063
A1	0.05	--	0.25	0.002	--	0.01
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.30	--	0.45	0.012	--	0.018
c	0.09	--	0.20	0.004	--	0.008
D	9.00 BSC			0.354 BSC		
D1	7.00 BSC			0.276 BSC		
E	9.00 BSC			0.354 BSC		
E1	7.00 BSC			0.276 BSC		
e	0.80 BSC			0.031 BSC		
L	0.40	0.60	0.80	0.016	0.024	0.031
L1	1.00 REF			0.039 REF		

Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

25.5 QFN 24 PIN 4x4



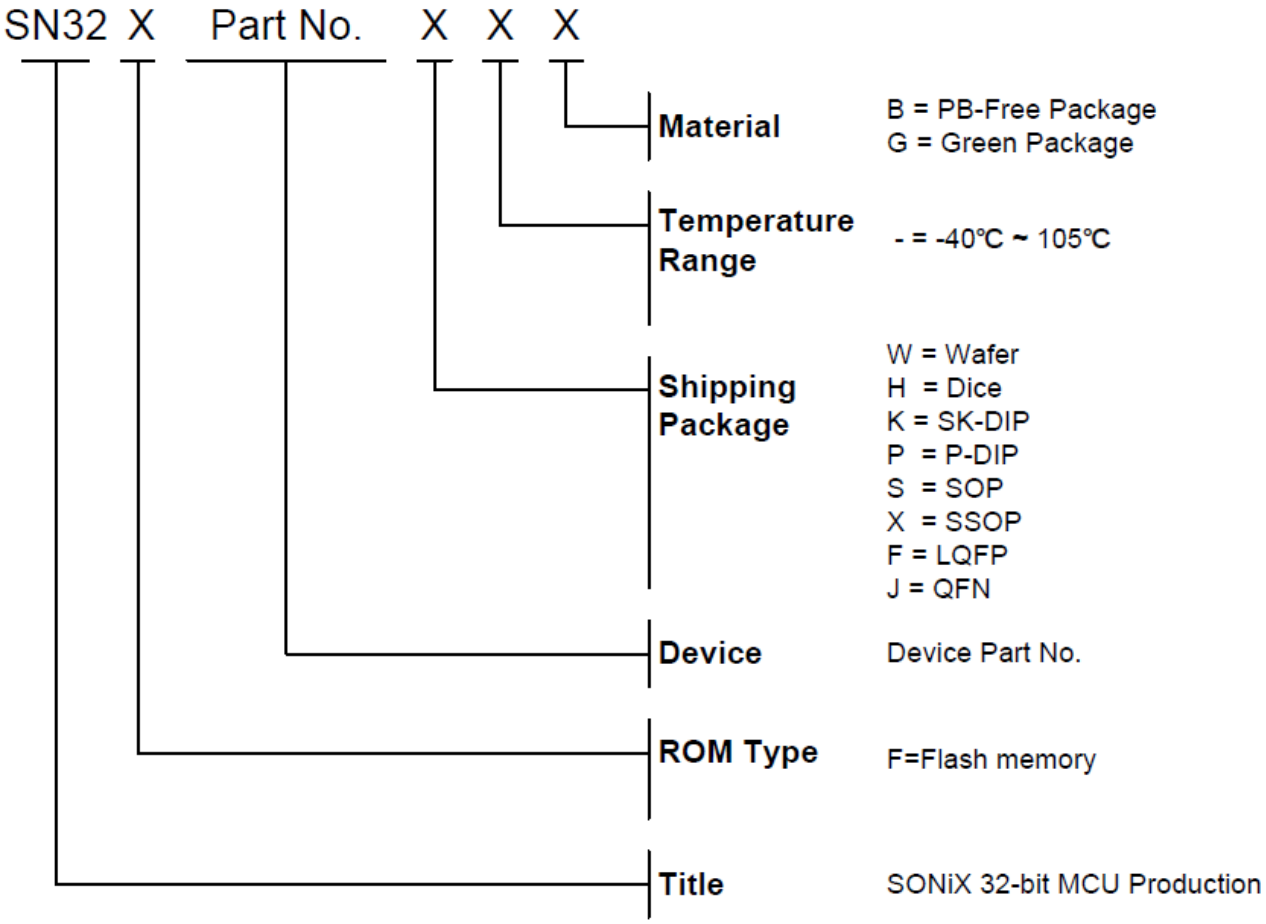
SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.80	0.90	0.028	0.031	0.035
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3	0.203 REF			0.008 REF		
b	0.15	0.25	0.30	0.007	0.010	0.012
D	4.00 BSC			0.157 BSC		
E	4.00 BSC			0.157 BSC		
e	0.50 BSC			0.020 BSC		
D2	1.90	2.35	2.80	0.075	0.093	0.110
E2	1.90	2.35	2.80	0.075	0.093	0.110
L	0.30	0.40	0.50	0.012	0.016	0.020

Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)

26 MARKING DEFINITION

26.1 MARKING INDETIFICATION SYSTEM



26.2 INTRODUCTION

There are many different types in SONiX 32-bit MCU production line.

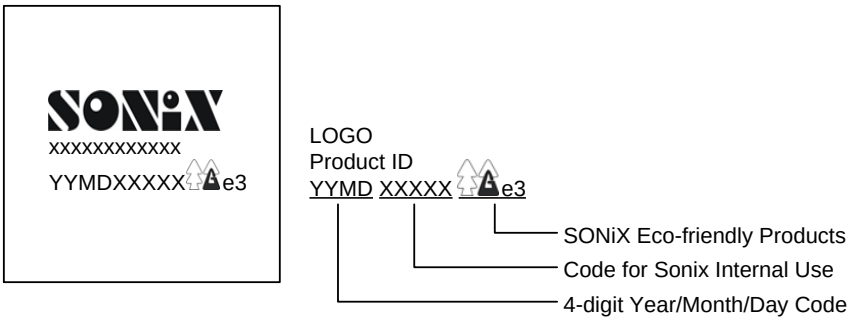
This note lists the marking definitions of all 32-bit MCU for order or obtaining information.

26.3 MARKING EXAMPLE

Name	ROM Type	Device	Package	Temperature	Material
SN32F407JG	Flash memory	400	QFN	-40℃~105℃	Green Package
SN32F407FG	Flash memory	400	LQFP	-40℃~105℃	Green Package
SN32F405JG	Flash memory	400	QFN	-40℃~105℃	Green Package
SN32F405FG	Flash memory	400	LQFP	-40℃~105℃	Green Package
SN32F403JG	Flash memory	400	QFN	-40℃~105℃	Green Package
SN32F400W	Flash memory	400	Wafer	-40℃~105℃	-
SN32F400H	Flash memory	400	Dice	-40℃~105℃	-

26.4 DATECODE SYSTEM

The figure below is an example of the marking. Contents such as the product ID or symbol may vary according to different packages.



Main Office:
Address: 10F-1, NO. 36, Taiyuan Street., Chupei City, Hsinchu, Taiwan R.O.C.
Tel: 886-3-5600 888
Fax: 886-3-5600 889

Taipei Office:
Address: 15F-2, NO. 171, Song Ted Road, Taipei, Taiwan R.O.C.
Tel: 886-2-2759 1980
Fax: 886-2-2759 8180

Hong Kong Office:
Unit No.705,Level 7 Tower 1,Grand Central Plaza 138 Shatin Rural Committee Road, Shatin, New Territories, Hong Kong.
Tel: 852-2723-8086
Fax: 852-2723-9179

Technical Support by Email:
Sn8fae@sonix.com.tw