SN34F780 Series

USER'S MANUAL

SN34F788
SN34F787
SN34F785

SONiX 32-Bit Cortex-M4F Micro-Controller

AMENDENT HISTORY

Version	Date	Description
0.1	2023/10/03	First version released.
1.0	2023/11/24	1. Fix typing error 2. Update Features 3. Update Pin Descriptions 4. Add FPU MVFR0 and MVFR1 registers 5. Update reset range of the BTUP_STS register after reset event 6. Update Wactchdog Reset notes 7. Update Reboot Reset notes 8. Add PLL Frequency Setting table 9. Update EHS warm-up time 10. Add EHS SMD Crystal Limitation 11. Update SCU PWRMODE register description 12. Update SCU CHIPID and VERSION registers description 13. Add PWRMODEMISC register 14. Update WWDTCLKSEL of CLKSEL register default value 15. Update SCU PnSTR registers description 16. Add ALWAYSON LVDCTRL register note 17. Update ALWAYSON OSCCTL register 18. Add ALWAYSON OSCMISC register 19. Update sleep, deep sleep and deep power-down mode notes 20. Update sleep and deep sleep mode wakeup time 21. Update Tigger Detection of GPIO description 22. Add ADC Channel table 23. Update Inverse PWM output with Dead-band period notes 24. Update WDT clock source and WDT_CTRL register 25. Update WWDT_CTRL register. 26. Add RTC Calibration flow and Example. 27. Update SPI Master maximum speed. 28. Add Flash Physical Read Address and Shadow Memory read structure. 29. Update Flash checksum calculate description 30. Update Development Tool description
1.1	2024/02/15	1. Fix typing error. 2. Add RTC_TIME1, RTC_TIME2, BCD_RTC_TIME1 and BCD_RTC_TIME2 registers note. 3. Update minimum ratio of SSPCLK/SCLK. 4. Update AHB/APB CLOCK table. 5. Update FLASH ROM PROGRAMMING PIN. 6. Update ALWAYSON_OSCMISC register description. 7. Update ADC Conversion Time description. 8. Add electrical characteristics of VDDIO1 and related pins. 9. Add RTC Configuration Procedure. 10. Add WWDT Programming Sequence note. 11. Update deep sleep mode note.
1.2	2024/03/06	1. Add 3.4.3 LVD control register note. 2. Update 2.7 System Control Register bit4 SEVONPEND description. 3. Update 3.2 System Clock description. 4. Update 8.9.2.3 Write CT16B0/1/2/5 MR0~3_ALIAS buffers description. 5. Update 9.1 WDT Overview description. 6. Update 10.1 WWDT Overview description. 7. Update 13.2 SSP Feature description. 8. Update 24.2 Electrical Characteristic.
1.3	2024/03/29	1. Update 8.1 Overview. 2. Update 13.7.1 SPI n Control register 0 Note's description. 3. Update 24.2 Electrical Characteristic. 4. Update 26 PACKAGE INFORMATION diagram. 5. Update 27.3 MARKING EXAMPLE

1.4	2024/07/11	1. Update 12.11.1 Request Selection DMA1 SRC_RS/DST_RS Table. 2. Update 12.13.14 DMA 1 Channel m Configuration register description. 3. Update 13.2 FEATURES description. 4. Update 13.7.5 SPI n Interrupt Control register description. 5. Update 15.11.8 UART n FIFO Control register description.
1.5	2024/11/21	1. Update 3.2.1.1 Internal High-speed RC Oscillator (IHRC) description. 2. Update 7.9.2 Auto Power down Mode description. 3. Update 8.2 Features description. 4. Update 8.10.18 CT16Bn Capture Control register description. 5. Update 24.2 ELECTRICAL CHARACTERISTIC
1.6	2025/04/16	1. Update 8.6.1 PWM Mode 1 and 8.6.2 PWM Mode 2 description. 2. Update 8.10 CT16Bn REGISTERS description. 3. Update 8.10.1 CT16Bn Timer Control register description. 4. Update 8.10.2 CT16Bn Timer Control register description. 5. Update 8.10.3 CT16Bn Timer Counter register description. 6. Update 8.10.4 CT16Bn Prescale register description. 7. Update 8.10.6 CT16Bn Count Control register description. 8. Update 16.9.4 TTCAN Trigger Types description. 9. Update 16.9.4.5 Transmit Stop Trigger description. 10. Update 16.9.5 TTCAN Watch Trigger description. 11. Update 24.2 ELECTRICAL CHARACTERISTIC 12. Update 25 FLASH ROM PROGRAMMING PIN
1.7	205/05/05	1. Remove Message for MM/SM write protect description. 2. Update 21.3 FEATURES description. 3. Update 21.4.2 SM Layout description and table. 4. Update 21.8.2 Message Description table. 5. Update 21.8.3 Message Function table. 6. Update 21.8.3.5 Message for MM/SM protect status description. 7. Update 21.8.5 CODE SECURITY description and table. 8. Update 21.9.1 FLASH Message 0 register description. 9. Update 21.9.4 FLASH Message 3 register description.

Table of Content

1 PRODUCT OVERVIEW

1.1 FEATURES

- **Memory configuration**
  512KB on-chip Flash programming memory (include 8KB Boot ROM).
  160KB SRAM.

- **Operation Frequency up to 192MHz**

- **Interrupt sources**
  ARM Cortex-M4 built-in Nested Vectored Interrupt Controller (NVIC) and FPU.

- **I/O pin configuration**
  Up to 55 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 17/19/21/23mA driving/sinking current

- **2 Programmable Watchdog Timers (WDT, WWDT)**
  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/2.4V/2.6V/2.8V/3.0V for VDD
  Interrupt: 2.4V/2.6V/2.8V/3.0V 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, Deep Power-down

- **GPIO Alternate function (AFIO)**
  Flexible assignment of peripheral functions to desired pins.

- **CRC**
  CRC-16, CRC-16-CCCITT, CRC-32

- **Two 8-Channels DMA Controller**

- **Timer**
  4 16-bit timer support up-counting, down-counting, and center-aligned mode.
  3 16-bit timers support up-counting mode.
  32 sets PWM
  8 sets inverse PWM with programmable dead-band
  PCLK up to 192MHz
  2 16-bit basic timers support up-counting mode

- **Working voltage 2.4V ~ 3.6V**

- **16+2 CH 12-bit SAR ADC**
  16 external ADC input
  1 internal battery measurement
  4-level internal reference voltage source (VDD, 2.5V, 2V, 1.5V)
  4 sets 16-frame DMA FIFO
  Single/Scan/Continuous mode

- **Interface**
  -Three I2C controllers supporting I2C-bus specification with multiple address recognition.
  -Six UART controllers with fractional baud rate generation.
  -Three SSP controllers supporting SPI and I2S
  -Two CAN 2.0A/B interfaces
  -LCD-TFT (8080) interface
  -SDIO 2.0 Controller
  -Ethernet MAC Controller 10/100Mbps

- **System clocks**
  -External high clock: Crystal type 10MHz~25MHz
  -External low clock: Crystal type 32.768 KHz
  -Internal high clock: RC type 12 MHz
  -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.

- **Serial Wire Debug (SWD)**

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

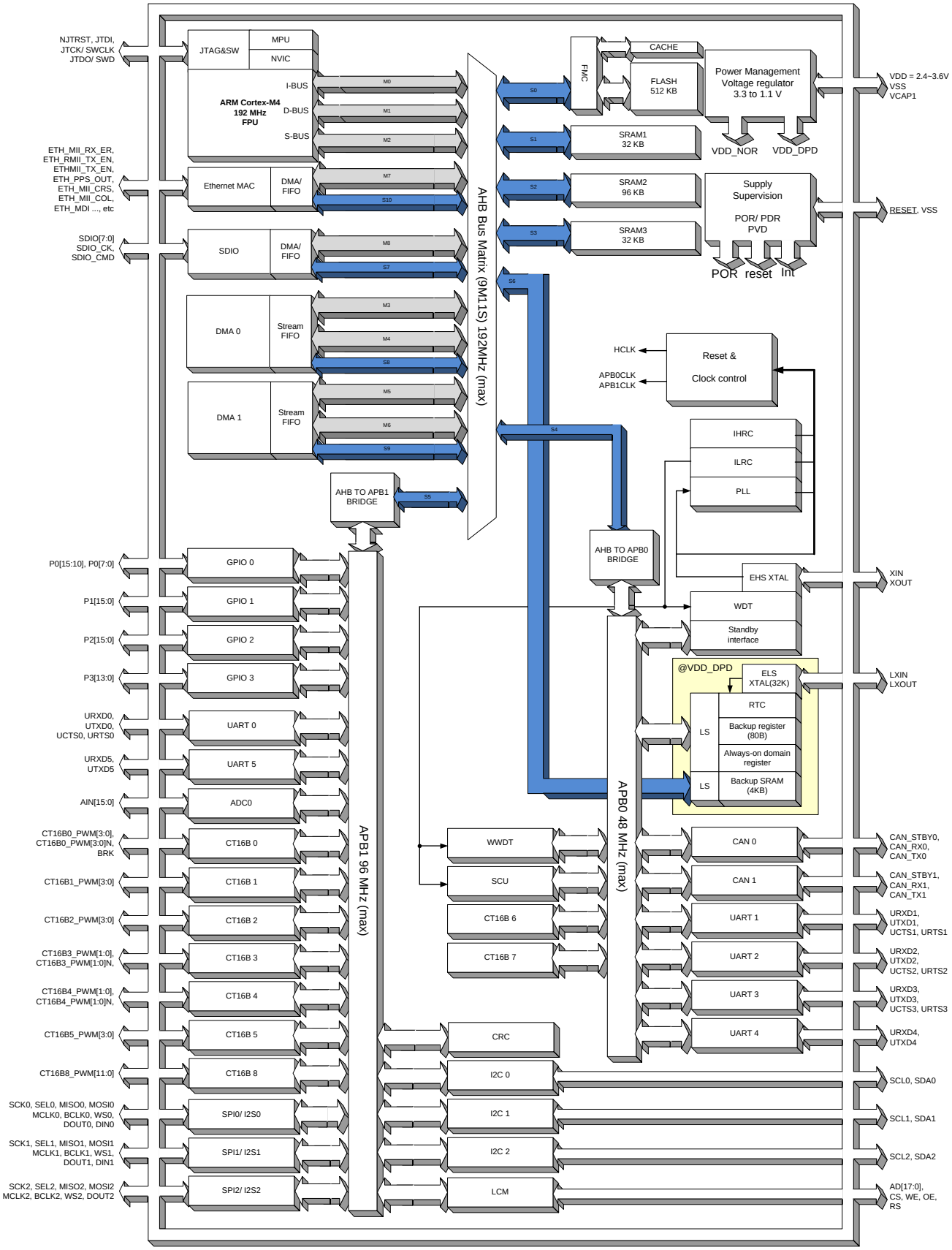
- **Package (Chip form support)**
  LQFP64/48/32 pin
  QFN32 pin

☞ **Features Selection Table**

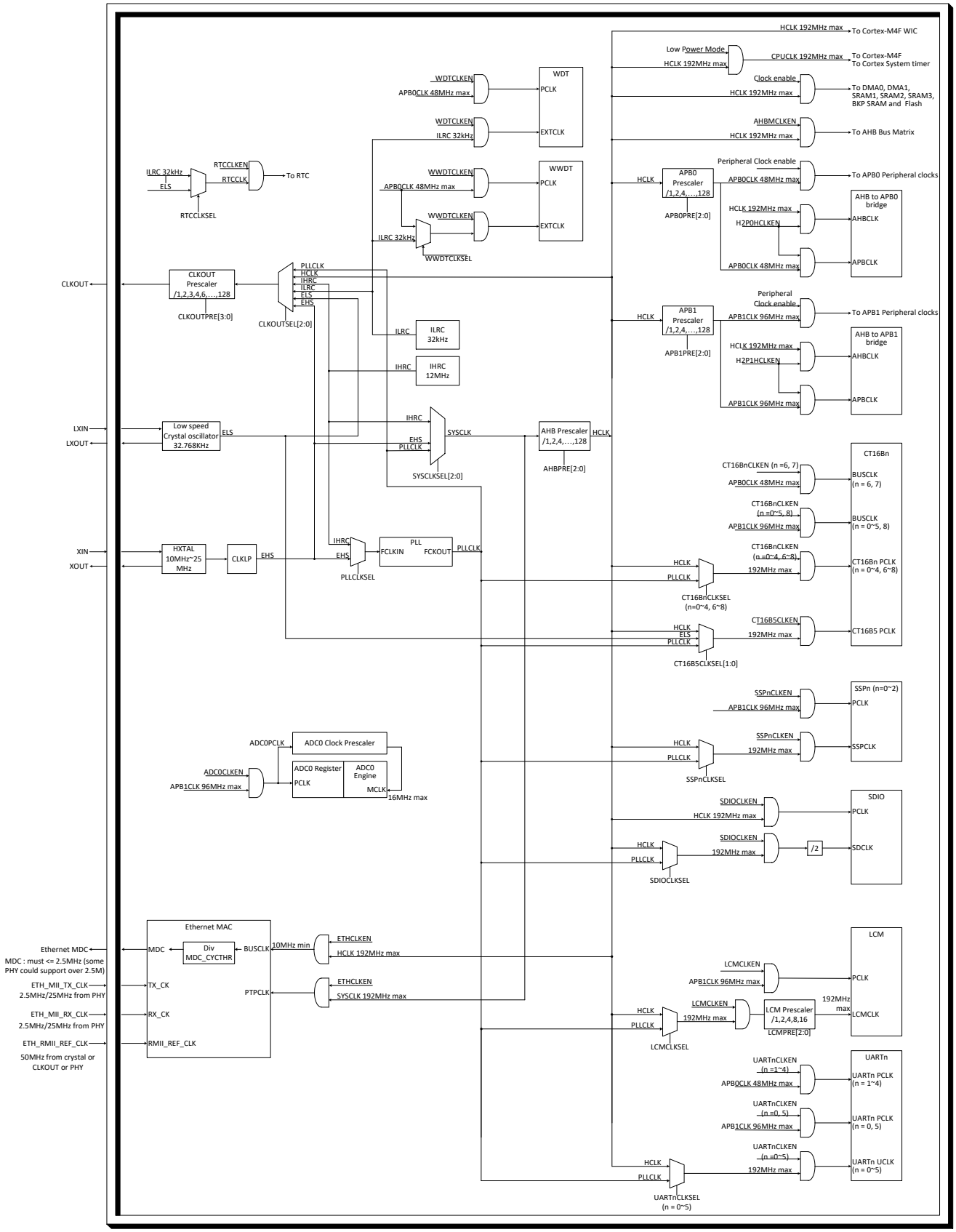
Chip	ROM (KB)	RAM (KB)	F_{CPU} (Max MHz)	UART	SPI/ I2S	I2C	CAN	Timer	PWM	ADC	SDIO	LCM	ETHMAC	GPIO	PKG
SN34F788F	512*	160	192	6	3	3	2	9	32+8	16	V	V	V	55	LQFP64
SN34F787F	512*	160	192	6	3	3	2	9	32+8	10	V	V		41	LQFP48
SN34F785F/J	512*	160	192	5	3	3	2	9	27+8	9				27	LQFP32 QFN32

* Include 8KB Boot ROM

1.2 SYSTEM BLOCK DIAGRAM

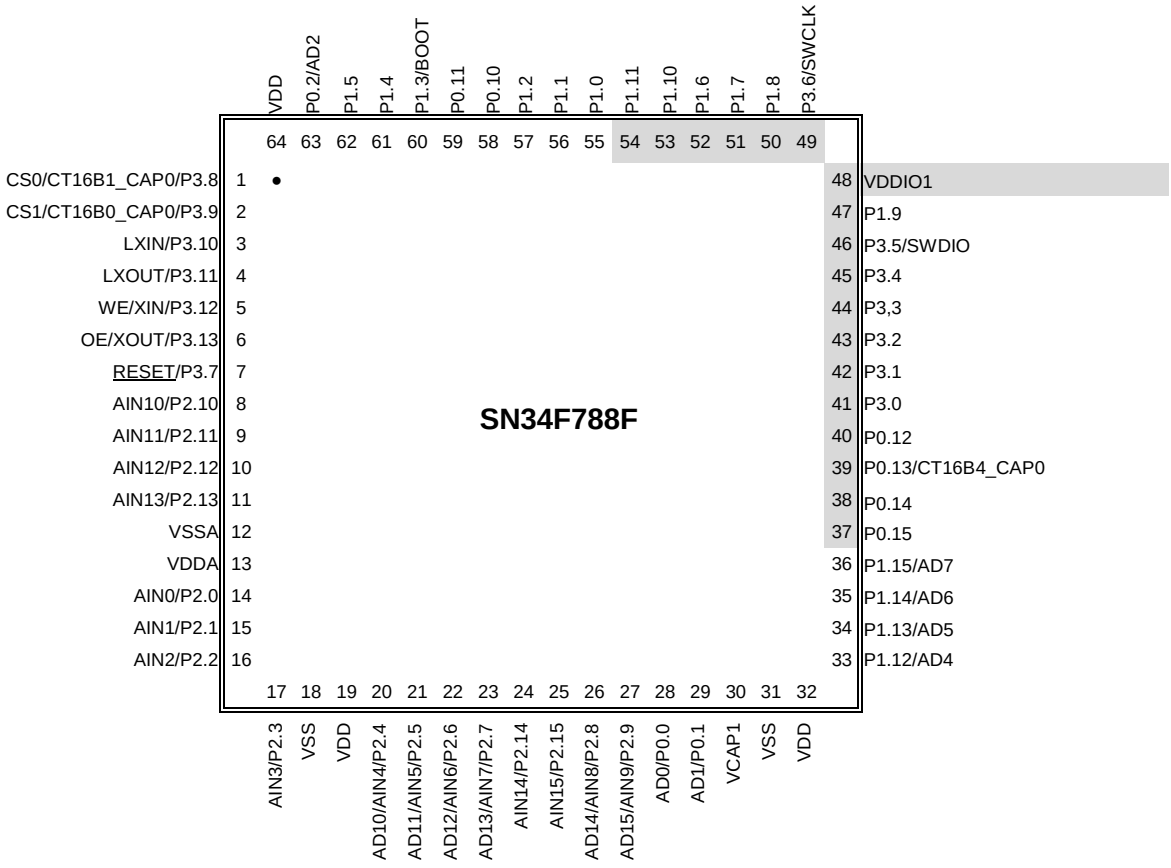


1.3 CLOCK GENERATION BLOCK DIAGRAM



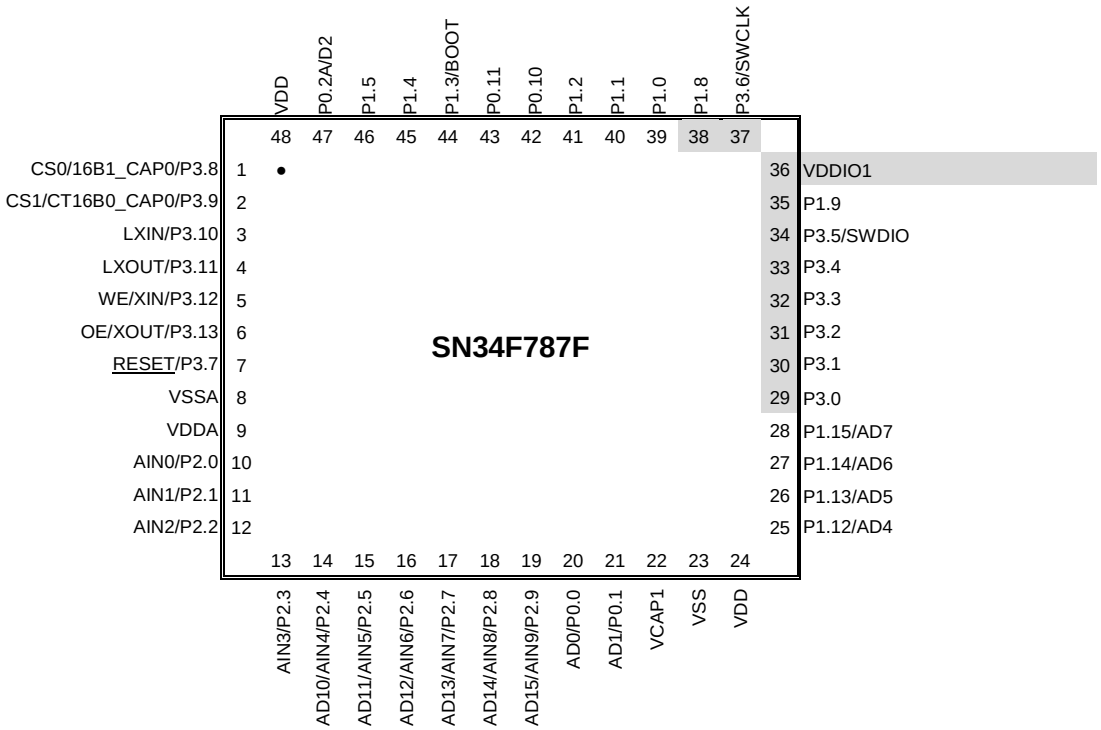
1.4 PIN ASSIGNMENT

1.4.1 SN34F788FG (LQFP 64 pins)



✷ ***Note: 1. 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.***

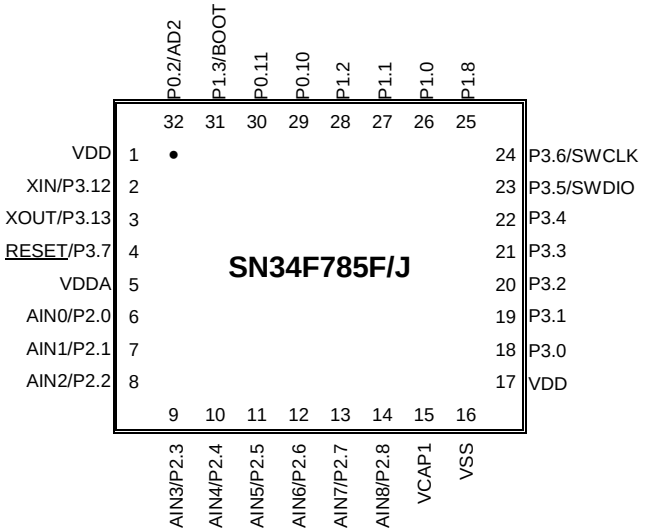
1.4.2 SN34F787FG (LQFP 48 pins)



✸ ***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.

1.4.3 SN34F785F/JG (LQFP 32 pins/QFN 32 pins 4x4)



✷ ***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. This package does not have a VDDIO1 pin, so the voltage source of P0.12~P0.15, P3.0~P3.6 and P1.6~P1.11 pins are VDD.

1.5 PIN ALLOCATION TABLE

PIN	SPI	I2C	I2S	UART	CT16	CAN	Eth	LCM	SDIO	ADC	OTHER
P0.0	SCK0 MO2 SI2		DOUT2		B0_PWM2 B0_PWM3N B0_PWM1N B2_PWM3 B0_PWM0 B2_PWM0 B8_PWM0	CAN_RX0		AD0	SDIO_D0		
P0.1	SEL0 SCK1	SCL1	BCLK1	UTXD2 URXD2 IRDA_TXD2 IRDA_RXDL2	B0_PWM1 B0_PWM2N B0_PWM3 B2_PWM2 B8_PWM1	CAN_TX0	ETH_MII_RX_ER	AD1	SDIO_D1		
P0.2	MI0 SO0	SDA1		URXD2 IRDA_RXDL2	B0_PWM0 B5_PWM3 B0_PWM3N B2_PWM3 B8_PWM2	CAN_RX1	ETH_RMII_TX_EN ETH_MII_TX_EN	AD2	SDIO_D2		
P0.10		SCL0		UTXD0 UTXD4 IRDA_TXD0 IRDA_TXD4	B0_BRK B1_PWM2 B2_PWM0 B3_PWM0N B4_PWM0 B4_PWM1N B8_PWM5	CAN_TX1		CS			CLKOUT3
P0.11		SDA0		URXD0 UCTS3 IRDA_RXDL0	B2_CAP0 B1_PWM3 B2_PWM1 B3_PWM1N B4_PWM1 B4_PWM0N B8_PWM6						
P0.12	SCK1	SDA2			B0_PWM3 B0_PWM0N B5_PWM3 B8_PWM8				SDIO_D1		CLKOUT2
P0.13	SEL1				B4_CAP0 B0_PWM2 B5_PWM2 B8_PWM7				SDIO_D0		
P0.14	SCK1	SDA0	BCLK1	URXD5 IRDA_RXDL5	B0_PWM1 B5_PWM1 B8_PWM6				SDIO_D7		
P0.15		SCL0		UTXD5 IRDA_TXD5	B0_PWM0 B5_PWM0 B8_PWM5				SDIO_D6		
P0.20		SDA1			B8_PWM1			LCM_CS			

PIN	SPI	I2C	I2S	UART	CT16	CAN	Eth	LCM	SDIO	ADC	OTHER
P1.0	SCK0 SCK2	SDA1	BCLK0 BCLK2	UTXD4 URXD0 IRDA_TXD4 IRDA_RXDL0	B2_PWM1 B5_PWM2 B8_PWM3	CAN_RX0					SWO
P1.1	MI0 MI2 SO0 SO2	SCL0 SDA2	DIN1	URXD4 IRDA_RXDL4	B1_PWM0 B5_PWM0 B8_PWM4	CAN_TX0		CS			
P1.2	MO0 MO2 SI0 SI2	SDA0	DOUT0 DOUT1 DOUT2	URXD4 IRDA_RXDL4	B1_PWM1 B4_PWM0N B5_PWM1	CAN_STBY0 CAN_RX1					
P1.3			DIN0	URXD2 IRDA_RXDL2	B3_PWM0N B4_PWM1N B8_PWM7	CAN_STBY1					
P1.4		SCL0		UTXD2 URXD4 IRDA_TXD2 IRDA_RXDL4	B2_PWM0 B2_PWM2 B3_PWM0 B4_PWM1 B8_PWM8	CAN_RX1	ETH_MII_TXD3		SDIO_D4		
P1.5	SEL1	SDA0 SDA1	WS1	UTXD4 IRDA_TXD4	B2_PWM1 B2_PWM3 B3_PWM1 B4_PWM0 B8_PWM9	CAN_TX1			SDIO_D5		
P1.6	MI2 SO2		BCLK1	URXD2 URXD3 IRDA_RXDL2 IRDA_RXDL3	B2_PWM2 B5_PWM2 B8_PWM0				SDIO_D3		
P1.7	SEL0 SCK2		BCLK2	UTXD2 UTXD3 IRDA_TXD2 IRDA_TXD3	B2_PWM1 B5_PWM1 B8_PWM11				SDIO_D2		
P1.8	SEL0 SCK1 SEL2	SCL1	WS0 WS2	URXD1 URTS3 UTXD0 IRDA_TXD0 IRDA_RXDL1	B2_PWM0 B8_PWM10			AD3			
P1.9	SEL1	SDA1 SCL1	WS1	UTXD1 URXD3 IRDA_TXD1 IRDA_RXDL3	B5_PWM0 B8_PWM9	CAN_STBY0					
P1.10	MI1 MO2 SO1 SI2	SCL1 SDA1	DIN1 DOUT2	UTXD4 IRDA_TXD4	B2_PWM3 B8_PWM1				SDIO_CK		
P1.11	MO1 SI1	SDA1		URXD4 URTS2 IRDA_RXDL4	B8_PWM2			A0 (RS)	SDIO_CMD		
P1.12	SEL1 SCK2		WS1 BCLK2		B0_BRK B0_PWM0 B1_PWM0 B3_PWM1 B4_PWM1 B5_PWM0	CAN_RX0 CAN_RX1	ETH_RMII_TXD0 ETH_MII_TXD0	AD4	SDIO_CK		
P1.13	SCK1	SCL1 SCL2	BCLK1	UCTS2	B0_PWM0N B1_PWM1 B3_PWM1N B4_PWM1N	CAN_TX0 CAN_TX1	ETH_RMII_TXD1 ETH_MII_TXD1	AD5	SDIO_CMD		
P1.14	MI1 SO1	SDA1 SDA2		URTS2	B0_PWM1N B1_PWM2 B3_PWM0 B4_PWM0 B8_PWM3	CAN_STBY0		AD6			
P1.15	MO1 SI1	SCL2	DOUT1		B5_CAP0 B3_PWM1 B0_PWM2N B1_PWM3 B3_PWM0N B4_PWM1 B8_PWM4			AD7			

PIN	SPI	I2C	I2S	UART	CT16	CAN	Eth	LCM	SDIO	ADC	OTHER
P2.0		SCL1		UTXD3 URXD0 UCTS1 IRDA_TXD3 IRDA_RXDL0	B2_PWM0 B1_PWM0 B5_PWM0	CAN_STBY1	ETH_MII_CRS		SDIO_CK	ADC0_AIN0 AVREFH	
P2.1		SDA1		URXD3 UTXD0 UTXD3 URTS1 IRDA_TXD0 IRDA_TXD3 IRDA_RXDL3	B2_PWM1 B3_PWM0N B5_PWM1	CAN_RX1	ETH_RMII_REF_CLK ETH_MII_RX_CLK		SDIO_CMD	ADC0_AIN1	
P2.2				UTXD1 IRDA_TXD1	B2_PWM2 B3_PWM0 B5_PWM2	CAN_TX1	ETH_MDIO	CS		ADC0_AIN2	
P2.3				URXD1 IRDA_RXDL1	B2_PWM3 B3_PWM1 B5_PWM3		ETH_MII_COL			ADC0_AIN3	
P2.4	SEL0 SEL2		WS0 WS2	UTXD5 IRDA_TXD5	B0_PWM1N B1_PWM1 B3_PWM1N B4_PWM1N			AD10		ADC0_AIN4	
P2.5	SCK0		BCLK0	URXD5 IRDA_RXDL5	B0_PWM1 B0_PWM0N B2_PWM0 B4_PWM1			AD11	SDIO_D3	ADC0_AIN5	
P2.6	MI0 SO0			UCTS2	B8_PWM2 B0_PWM0 B0_PWM2N B3_PWM0 B4_PWM0N B4_PWM1 B5_PWM0 B0_BRK			AD12	SDIO_D4	ADC0_AIN6	
P2.7	MO0 SI0		DOUT0		B0_PWM0N B0_PWM2 B1_PWM2 B4_PWM0 B5_PWM1		ETH_MII_RX_DV ETH_RMII_CRS_DV	AD13	SDIO_D5	ADC0_AIN7	
P2.8	MO2 SI2		DIN0 DOUT2	URXD1 IRDA_RXDL1	B0_PWM1N B0_PWM2 B5_PWM2	CAN_STBY1	ETH_MII_RXD2	AD14	SDIO_D6	ADC0_AIN8	
P2.9	SCK1		BCLK1	URTS2	B1_PWM3 B0_PWM1 B0_PWM2N B5_PWM3	CAN_STBY0	ETH_MII_RXD3	AD15	SDIO_D7	ADC0_AIN9	
P2.10		SCL2		UTXD5 IRDA_TXD5				AD16		ADC0_AIN10	
P2.11	MO2 MO1 SI2 SI1	SDA2	DOUT2 DOUT1	URXD5 IRDA_RXDL5			ETH_MDC	AD17		ADC0_AIN11	
P2.12	MI1 SO1				B0_PWM1		ETH_MII_TXD2			ADC0_AIN12	
P2.13	MO1 SI1		DOUT1				ETH_MII_TX_CLK			ADC0_AIN13	
P2.14				UTXD2 IRDA_TXD2	B0_PWM3 B3_PWM0		ETH_RMII_RXD0 ETH_MII_RXD0			ADC0_AIN14	
P2.15				URXD2 IRDA_RXDL2	B0_PWM3N B3_PWM1		ETH_RMII_RXD1 ETH_MII_RXD1			ADC0_AIN15	

PIN	SPI	I2C	I2S	UART	CT16	CAN	Eth	LCM	SDIO	ADC	OTHER
P3.0	SCK0	SCL2		UTXD1 IRDA_TXD1	B0_PWM0	CAN_STBY1		A0			CLKOUT1
P3.1	MI0 SCK1 SO0	SCL0	BCLK0 BCLK1	UTXD0 IRDA_TXD0	B0_PWM1	CAN_RX1			SDIO_D0		
P3.2	MO0 MO1 SI0 SI1	SDA0	WS0 DOUT1	URXD0 IRDA_RXDL0	B0_PWM2	CAN_TX1			SDIO_D1		
P3.3	SEL1	SCL1	DOUT0 WS1	UCTS0 UTXD5 IRDA_TXD5	B0_PWM3	CAN_RX0	ETH_RMII_RX_ER		SDIO_D2		
P3.4	MI1 SO1	SDA1	DIN0	URXD1 URXD5 URTS0 IRDA_RXDL1 IRDA_RXDL5		CAN_TX0	ETH_TXER		SDIO_D3		
P3.5	MI2 SO2										SWDIO
P3.6	MO2 SI2		DOUT2	UTXD1 IRDA_TXD1							SWCLK
P3.7											RESET
P3.8					B1_CAP0 B8_PWM10		ETH_PHY_LINKSTS	AD8			
P3.9					B0_CAP0 B8_PWM11		ETH_PHY_PDN	AD9			
P3.10								WE			LXIN
P3.11								OE			LXOUT
P3.12		SDA0 SDA1		UTXD3 IRDA_TXD3							XIN
P3.13		SCL0 SCL1		URXD3 IRDA_RXDL3							XOUT

1.6 PIN DESCRIPTIONS

Pad NAME	TYPE	DESCRIPTION
VDD, VSS	P	Power supply input pins for digital circuit.
VDDA, VSSA	P	Power supply input pins for analog circuit.
VDDIO1	P	I/O driver power input pins for P0.12~P0.15, P3.0~P3.6, P1.6~P1.11. This power input shall not be floating.
VCAP1	P	The Core power regulator voltage output pin.
P0.0	I/O	**P0.0 —** General purpose digital input/output pin.
P0.1	I/O	**P0.1 —** General purpose digital input/output pin.
P0.2	I/O	**P0.2 —** General purpose digital input/output pin.
P0.10/CLKOUT3	I/O	**P0.10 —** General purpose digital input/output pin.
	O	**CLKOUT3 —** Clockout pin
P0.11/CT16B2_CAP0	I/O	**P0.11 —** General purpose digital input/output pin.
	I	**CT16B2_CAP0 —** Capture input 0 for CT16B2.
P0.12/CLKOUT2	I/O	**P0.12 —** General purpose digital input/output pin.
	O	**CLKOUT2 —** Clockout pin
P0.13/CT16B4_CAP0	I/O	**P0.13 —** General purpose digital input/output pin.
	I	**CT16B4_CAP0 —** Capture input 0 for CT16B4.
P0.14	I/O	**P0.14 —** General purpose digital input/output pin.
P0.15	I/O	**P0.15 —** General purpose digital input/output pin.
P0.20	I/O	**P0.20 —** General purpose digital input/output pin.
P1.0	I/O	**P1.0 —** General purpose digital input/output pin.
P1.1	I/O	**P1.1 —** General purpose digital input/output pin.
P1.2	I/O	**P1.2 —** General purpose digital input/output pin.
P1.3	I/O	**P1.3 —** General purpose digital input/output pin.
P1.4	I/O	**P1.4 —** General purpose digital input/output pin.
P1.5	I/O	**P1.5 —** General purpose digital input/output pin.
P1.6	I/O	**P1.6 —** General purpose digital input/output pin.
P1.7	I/O	**P1.7 —** General purpose digital input/output pin.
P1.8	I/O	**P1.8 —** General purpose digital input/output pin.

P1.9	I/O	**P1.9 —** General purpose digital input/output pin.
P1.10	I/O	**P1.10 —** General purpose digital input/output pin.
P1.11	I/O	**P1.11 —** General purpose digital input/output pin.
P1.12	I/O	**P1.12 —** General purpose digital input/output pin.
P1.13	I/O	**P1.13 —** General purpose digital input/output pin.
P1.14	I/O	**P1.14 —** General purpose digital input/output pin.
P1.15/CT16B5_CAP0	I/O	**P1.15 —** General purpose digital input/output pin.
	I	**CT16B5_CAP0 —** Capture input 0 for CT16B5.
P2.0/AIN0/AVREFH	I/O	**P2.0 —** General purpose digital input/output pin.
	I	**AIN0 —** ADC channel input 0.
	P	**AVREFH —** ADC external or internal reference input pin.
P2.1/AIN1	I/O	**P2.1 —** General purpose digital input/output pin.
	I	**AIN1 —** ADC channel input 1.
P2.2/AIN2	I/O	**P2.2 —** General purpose digital input/output pin.
	I	**AIN2 —** ADC channel input 2.
P2.3/AIN3	I/O	**P2.3 —** General purpose digital input/output pin.
	I	**AIN3 —** ADC channel input 3.
P2.4/AIN4	I/O	**P2.4 —** General purpose digital input/output pin.
	I	**AIN4 —** ADC channel input 4.
P2.5/AIN5	I/O	**P2.5 —** General purpose digital input/output pin.
	I	**AIN5 —** ADC channel input 5.
P2.6/AIN6	I/O	**P2.6 —** General purpose digital input/output pin.
	I	**AIN6 —** ADC channel input 6.
P2.7/AIN7	I/O	**P2.7 —** General purpose digital input/output pin.
	I	**AIN7 —** ADC channel input 7.
P2.8/AIN8	I/O	**P2.8 —** General purpose digital input/output pin.
	I	**AIN8 —** ADC channel input 8.
P2.9/AIN9	I/O	**P2.9 —** General purpose digital input/output pin.
	I	**AIN9 —** ADC channel input 9.
P2.10/AIN10	I/O	**P2.10 —** General purpose digital input/output pin.
	I	**AIN10 —** ADC channel input 10.

P2.11/AIN11	I/O	**P2.11 —** General purpose digital input/output pin.
	I	**AIN11 —** ADC channel input 11.
P2.12/AIN12	I/O	**P2.12 —** General purpose digital input/output pin.
	I	**AIN12 —** ADC channel input 12.
P2.13/AIN13	I/O	**P2.13 —** General purpose digital input/output pin.
	I	**AIN13 —** ADC channel input 13.
P2.14/AIN14	I/O	**P2.14 —** General purpose digital input/output pin.
	I	**AIN14 —** ADC channel input 14.
P2.15/AIN15	I/O	**P2.15 —** General purpose digital input/output pin.
	I	**AIN15 —** ADC channel input 15.
P3.0/CLKOUT1	I/O	**P3.0 —** General purpose digital input/output pin.
	O	**CLKOUT1 —** 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.
P3.3	I/O	**P3.3 —** General purpose digital input/output pin.
P3.4	I/O	**P3.4 —** 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/CT16B1_CAP0	I/O	**P3.8 —** General purpose digital input/output pin.
	I	**CT16B1_CAP0 —** Capture input 0 for CT16B1.
P3.9/CT16B0_CAP0	I/O	**P3.9 —** General purpose digital input/output pin.
	I	**CT16B0_CAP0 —** Capture input 0 for CT16B0.
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.

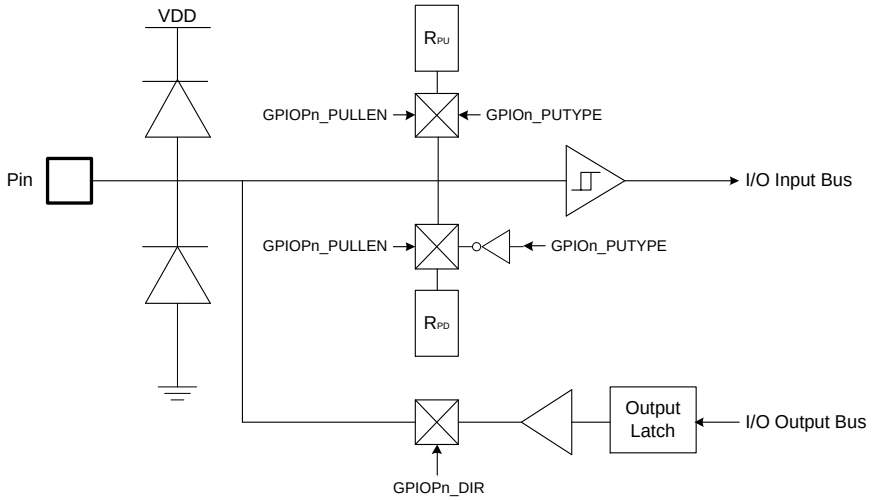
	I	**XIN —** External high-speed X'tal input pin.
P3.13/XOUT	I/O	**P3.13 —** General purpose digital input/output pin.
	O	**XOUT —** External high-speed X'tal output pin.

Please refer to GPIO Alternate Function Control registers for setting of each GPIOs.

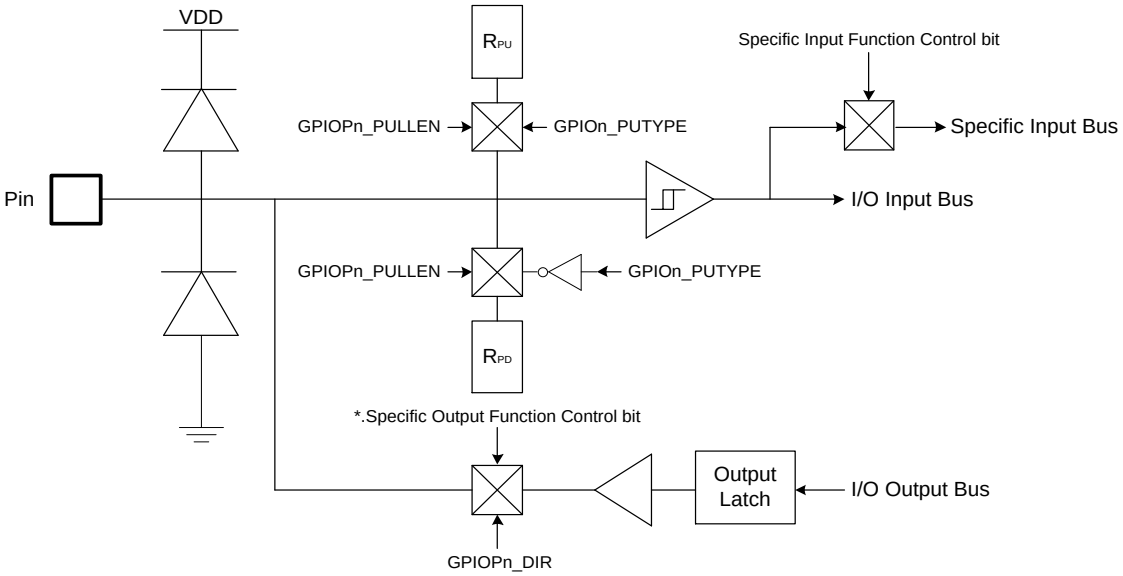
✸ ***Note: VDDIO1 is the I/O driver power input pin for P0.12~P0.15, P3.0~P3.6 and P1.6~P1.11. This power input shall not be floating.***

1.7 PIN CIRCUIT DIAGRAMS

- **Normal Bi-direction I/O Pin with Pull-up/Pull-down resistor.**

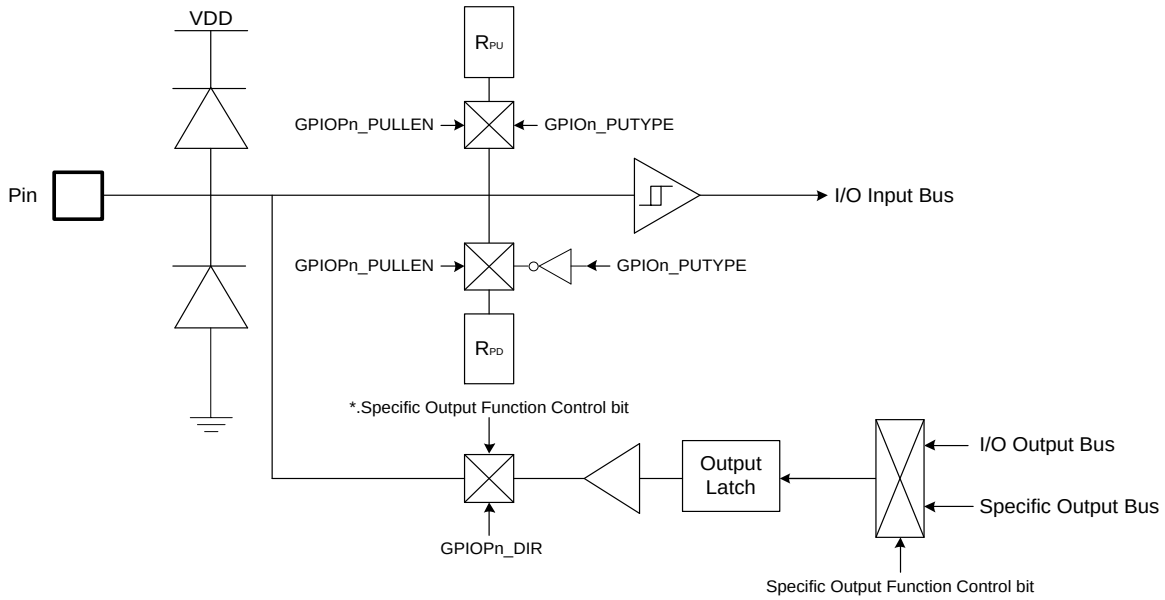


- **Bi-direction I/O Pin Shared with Specific Digital Input Function, e.g. Event counter, SPI, I2C…**



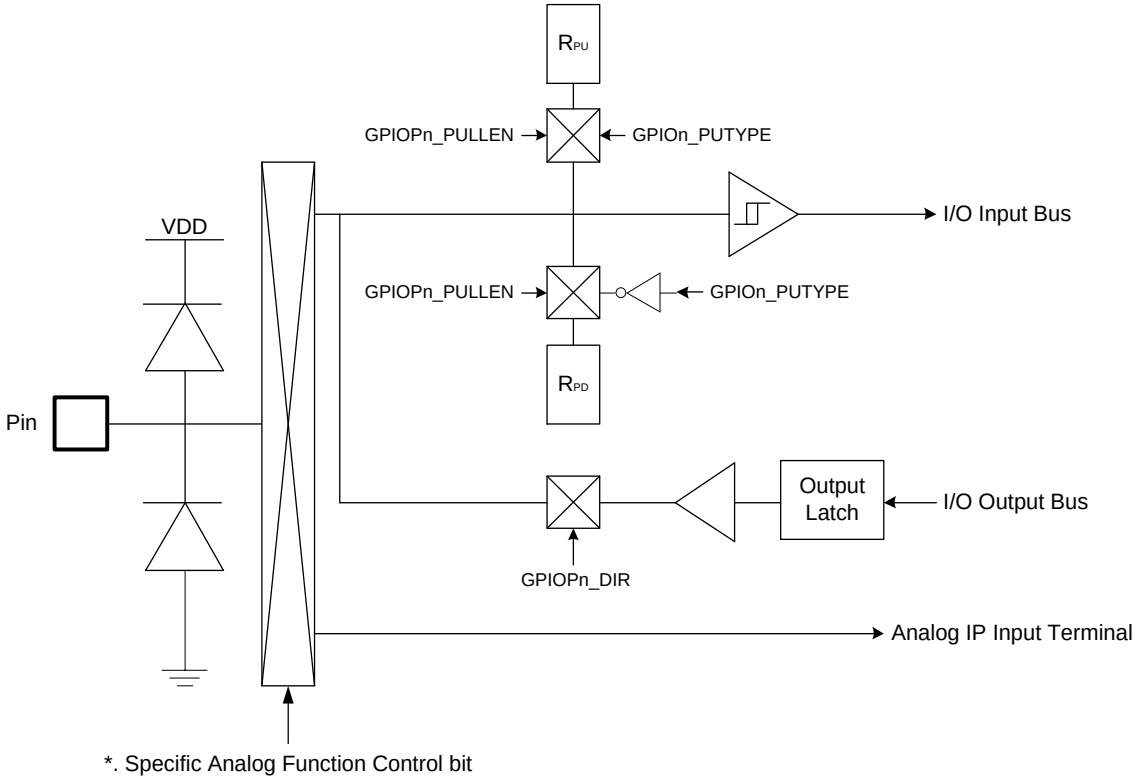
***. Some specific functions switch I/O direction directly, not through GPIOn_DIR register.**

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



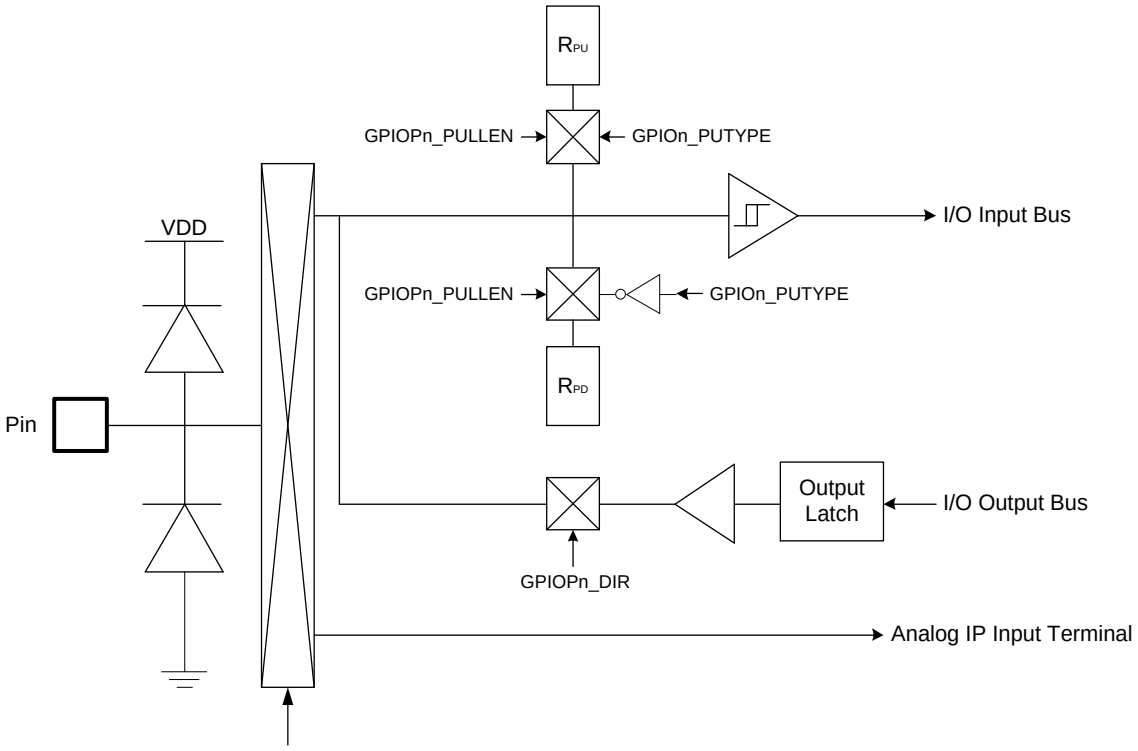
***. Some specific functions switch I/O direction directly, not through GPIOn_DIR register.**

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



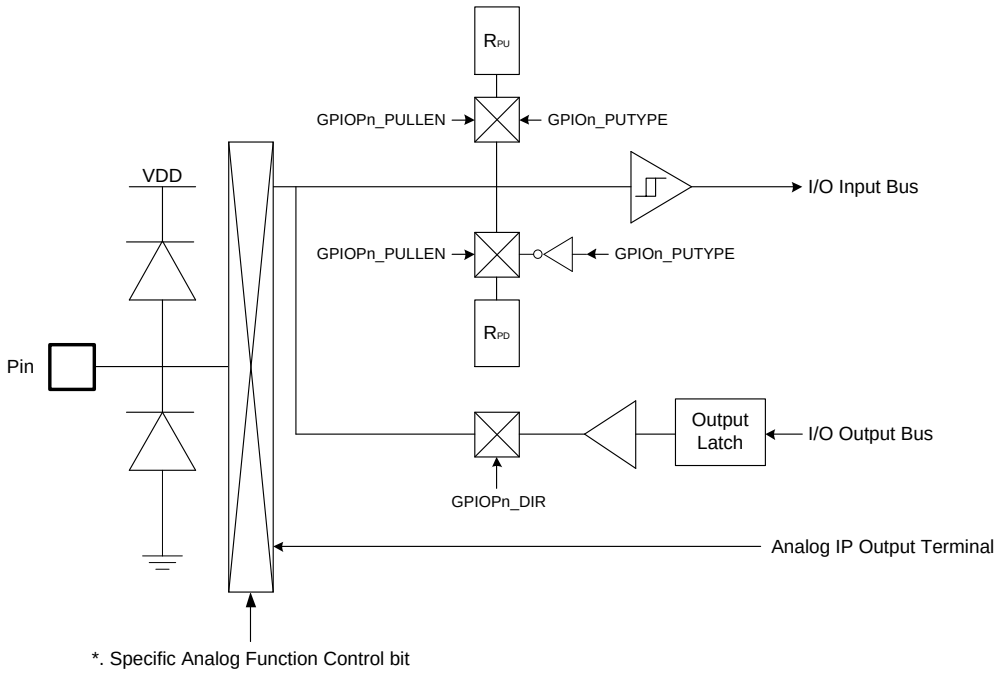
***. Some specific functions switch I/O direction directly, not through GPIOn_DIR register.**

◆ Bi-direction I/O Pin Shared with Xtal Input Function



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

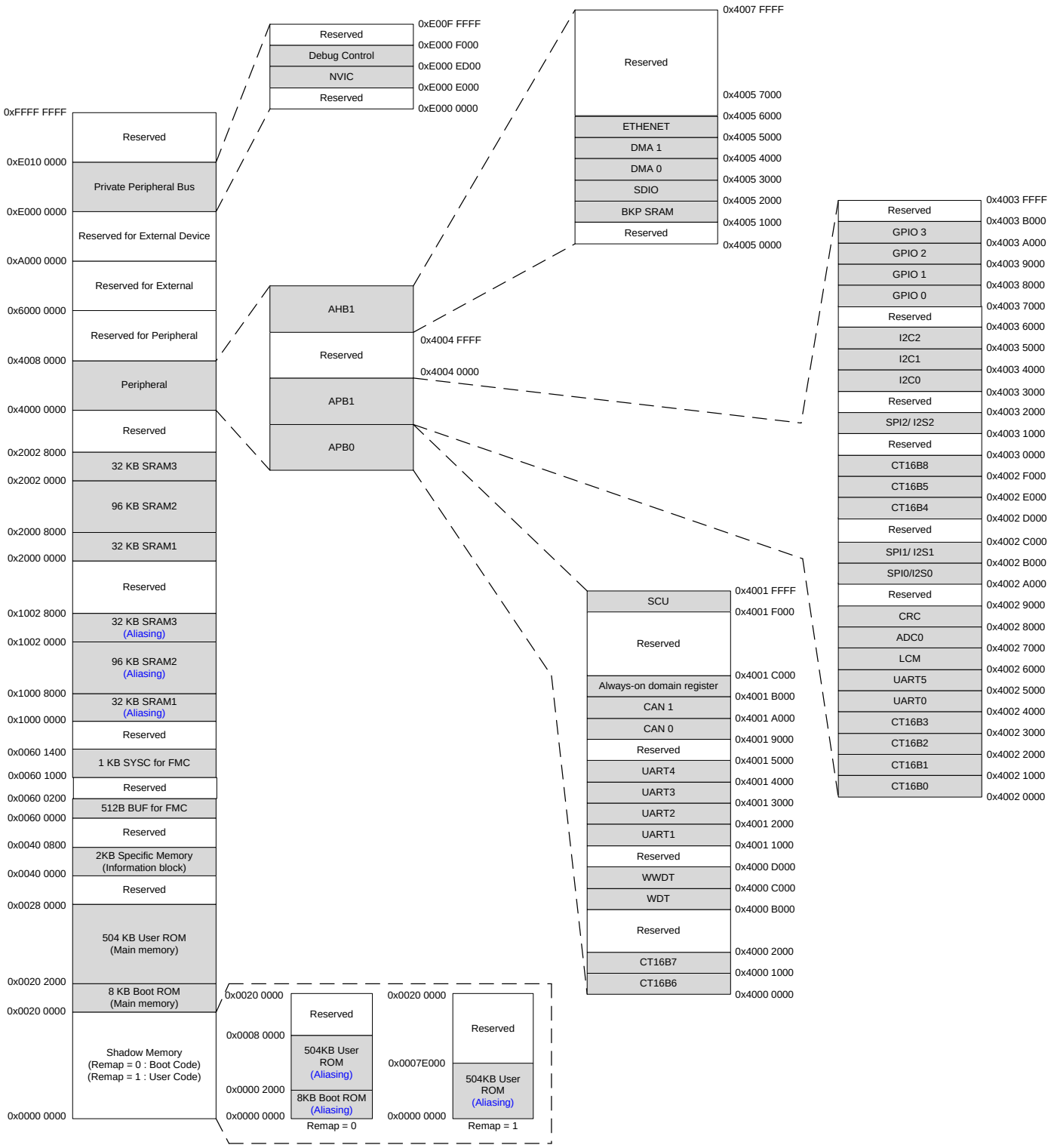
◆ Bi-direction I/O Pin Shared with Xtal Output Function



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

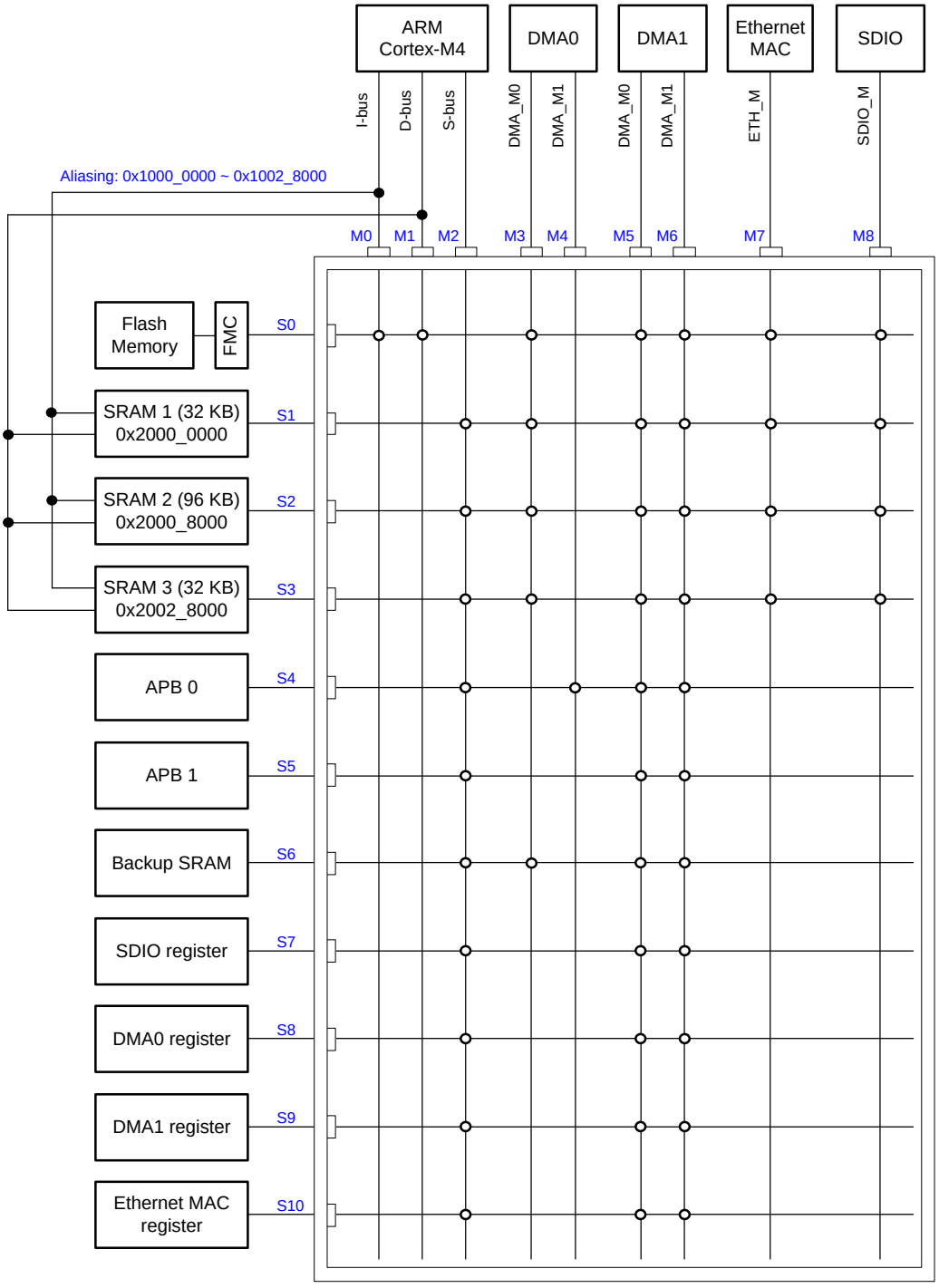
2 CENTRAL PROCESSOR UNIT (CPU)

2.1 MEMORY MAP



2.2 AHB BUS MATRIX

The AHB bus matrix of SN34F780 is shown in the figure below. The circled node in the figure indicates that the master can access the slave. When multiple masters access different slaves, they can operate simultaneously.



2.3 SYSTEM TICK TIMER

The SysTick timer is an integral part of the Cortex-M4F. 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-M4F, it facilitates porting of software by providing a standard timer that is available on Cortex-M4F based devices.

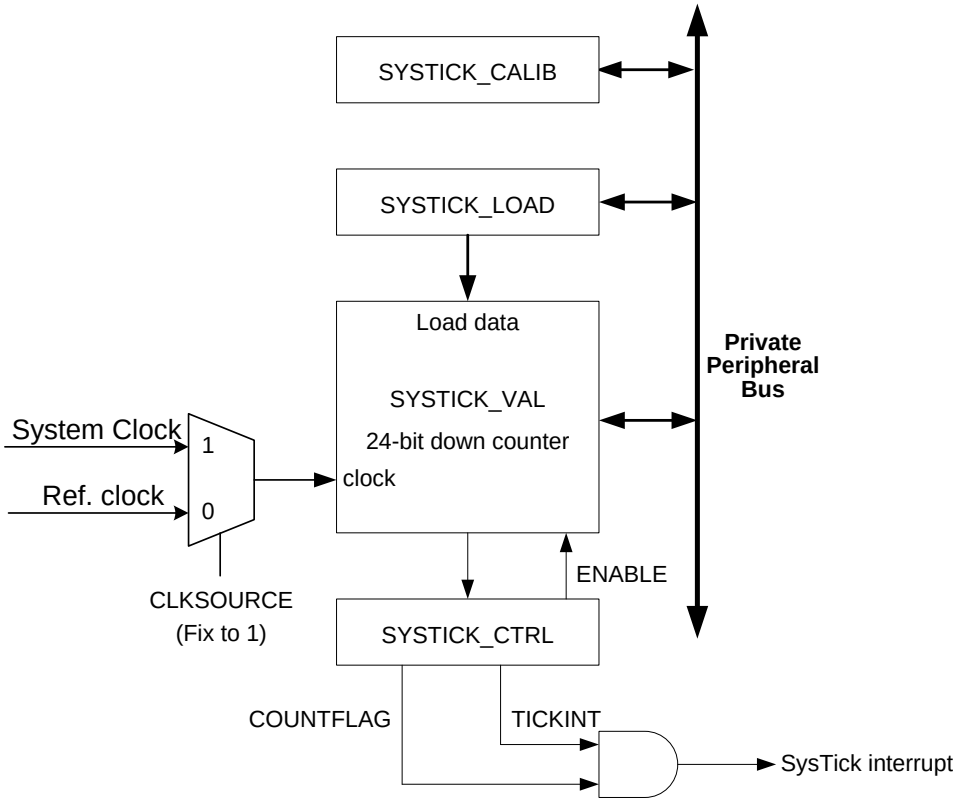
Refer to the *Cortex-M4 User Guide* for details.

2.3.1 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.***

2.3.2 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.

2.3.3 SYSTICK REGISTERS

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

2.3.3.1 System Tick Timer Control and Status register (SYSTICK_CTRL)

Address: 0xE000 E010 (Refer to Cortex-M4 Spec)

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

[1] CPUCLK source is from AHB clock. CPUCLK is stopped in sleep/deep sleep/deep power-down mode.

✷ ***Note: The CPUCLK stops in low power modes. Therefore, the SysTick cannot be used to wake up the system.***

2.3.3.2 System Tick Timer Reload value register (SYSTICK_LOAD)

Address: 0xE000 E014 (Refer to Cortex-M4 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 192 MHz.

RELOAD = (system tick clock frequency × 10 ms) −1 = (192 MHz × 10 ms) −1 = 0x001D4BFF.

Bit	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23:0	RELOAD	R/W	0	Value to load into the SYSTICK_VAL when the counter is enabled and when it reaches 0.

2.3.3.3 System Tick Timer Current Value register (SYSTICK_VAL)

Address: 0xE000 E018 (Refer to Cortex-M4 Spec)

Bit	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23:0	CURRENT	R/W	0	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.

2.3.3.4 System Tick Timer Calibration Value register (SYSTICK_CALIB)

Address: 0xE000 E01C (Refer to Cortex-M4 Spec)

Bit	Field	Access	Initial	Description
31	NOREF	R	1	Indicates the reference clock to M4 is provided or not. 1: No reference clock provided.
30	SKEW	R	0	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.
29:24	–	–	0	Reserved
23:0	TENMS	R/W	0x1D4BFF	Reload value for 10ms timing, subject to system clock skew errors. If the value reads as zero, the calibration value is not known.

2.4 NESTED VECTORED INTERRUPT CONTROLLER (NVIC)

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

- The NVIC supports 240 vectored interrupts.
- 16 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.

2.4.1 INTERRUPT AND EXCEPTION VECTORS

Priority	Type of Priority	Function	Description	Address Offset
-	-	-	Reserved	0x0000 0000
-3	Fixed	Reset	Reset	0x0000 0004
-2	Fixed	NMI_Handler	Non maskable interrupt.	0x0000 0008
-1	Fixed	HardFault	All class of fault	0x0000 000C
0	Settable	MemManage	Memory management	0x0000 0010
1	Settable	BusFault	Pre-fetch fault, memory access fault	0x0000 0014
2	Settable	UsageFault	Undefined instruction or illegal state	0x0000 0018
-	-	Reserved	Reserved	0x0000 001C~0x0000 002B
3	Settable	SVCCalll	System service call via SWI instruction	0x0000 002C
4	Settable	Debug Monitor	Debug Monitor	0x0000 0030
-	-	Reserved	Reserved	0x0000 0034
5	Settable	PendSV	Pendable request for system service	0x0000 0038
6	Settable	SysTick	System Tick timer	0x0000 003C
7	Settable	IRQ0	WWDT	0x0000 0040
8	Settable	IRQ1	LVD	0x0000 0044
9	Settable	IRQ2	WDT	0x0000 0048
10	Settable	IRQ3	SCU	0x0000 004C
11	Settable	IRQ4	FLASH	0x0000 0050
12	Settable	IRQ5		0x0000 0054
13	Settable	IRQ6		0x0000 0058
14	Settable	IRQ7		0x0000 005C
15	Settable	IRQ8		0x0000 0060
16	Settable	IRQ9		0x0000 0064
17	Settable	IRQ10		0x0000 0068

18	Settable	IRQ11	DMA0_CH0	0x0000 006C
19	Settable	IRQ12	DMA0_CH1	0x0000 0070
20	Settable	IRQ13	DMA0_CH2	0x0000 0074
21	Settable	IRQ14	DMA0_CH3	0x0000 0078
22	Settable	IRQ15	DMA0_CH4	0x0000 007C
23	Settable	IRQ16	DMA0_CH5	0x0000 0080
24	Settable	IRQ17	DMA0_CH6	0x0000 0084
25	Settable	IRQ18	ADC0	0x0000 0088
26	Settable	IRQ19	CAN0_TX	0x0000 008C
27	Settable	IRQ20	CAN0_RX	0x0000 0090
28	Settable	IRQ21	CAN0_TT	0x0000 0094
29	Settable	IRQ22	CAN0_EW	0x0000 0098
30	Settable	IRQ23		0x0000 009C
31	Settable	IRQ24	CT16B0	0x0000 00A0
32	Settable	IRQ25	CT16B8	0x0000 00A4
33	Settable	IRQ26		0x0000 00A8
34	Settable	IRQ27	CT16B3	0x0000 00AC
35	Settable	IRQ28	CT16B2	0x0000 00B0
36	Settable	IRQ29	CT16B5	0x0000 00B4
37	Settable	IRQ30		0x0000 00B8
38	Settable	IRQ31	I2C0	0x0000 00BC
39	Settable	IRQ32		0x0000 00C0
40	Settable	IRQ33	I2C1	0x0000 00C4
41	Settable	IRQ34		0x0000 00C8
42	Settable	IRQ35	SPI0	0x0000 00CC
43	Settable	IRQ36	SPI1	0x0000 00D0
44	Settable	IRQ37	UART0	0x0000 00D4
45	Settable	IRQ38	UART1	0x0000 00D8
46	Settable	IRQ39	UART2	0x0000 00DC
47	Settable	IRQ40		0x0000 00E0
48	Settable	IRQ41		0x0000 00E4
49	Settable	IRQ42		0x0000 00E8
50	Settable	IRQ43	CT16B4	0x0000 00EC

51	Settable	IRQ44		0x0000 00F0
52	Settable	IRQ45	CT16B1	0x0000 00F4
53	Settable	IRQ46		0x0000 00F8
54	Settable	IRQ47	DMA0_CH7	0x0000 00FC
55	Settable	IRQ48		0x0000 0100
56	Settable	IRQ49	SDIO	0x0000 0104
57	Settable	IRQ50		0x0000 0108
58	Settable	IRQ51	SPI2	0x0000 010C
59	Settable	IRQ52	UART3	0x0000 0110
60	Settable	IRQ53	UART4	0x0000 0114
61	Settable	IRQ54	CT16B6	0x0000 0118
62	Settable	IRQ55	CT16B7	0x0000 011C
63	Settable	IRQ56	DMA1_CH0	0x0000 0120
64	Settable	IRQ57	DMA1_CH1	0x0000 0124
65	Settable	IRQ58	DMA1_CH2	0x0000 0128
66	Settable	IRQ59	DMA1_CH3	0x0000 012C
67	Settable	IRQ60	DMA1_CH4	0x0000 0130
68	Settable	IRQ61	ETH	0x0000 0134
69	Settable	IRQ62		0x0000 0138
70	Settable	IRQ63	CAN1_TX	0x0000 013C
71	Settable	IRQ64	CAN1_RX	0x0000 0140
72	Settable	IRQ65	CAN1_TT	0x0000 0144
73	Settable	IRQ66	CAN1_EW	0x0000 0148
74	Settable	IRQ67		0x0000 014C
75	Settable	IRQ68	DMA1_CH5	0x0000 0150
76	Settable	IRQ69	DMA1_CH6	0x0000 0154
77	Settable	IRQ70	DMA1_CH7	0x0000 0158
78	Settable	IRQ71	UART5	0x0000 015C
79	Settable	IRQ72	I2C2	0x0000 0160
80	Settable	IRQ73		0x0000 0164
81	Settable	IRQ74		0x0000 0168
82	Settable	IRQ75		0x0000 016C
83	Settable	IRQ76		0x0000 0170

84	Settable	IRQ77		0x0000 0174
85	Settable	IRQ78		0x0000 0178
86	Settable	IRQ79		0x0000 017C
87	Settable	IRQ80		0x0000 0180
88	Settable	IRQ81	FPU	0x0000 0184
89	Settable	IRQ82		0x0000 0188
90	Settable	IRQ83		0x0000 018C
91	Settable	IRQ84		0x0000 0190
92	Settable	IRQ85		0x0000 0194
93	Settable	IRQ86		0x0000 0198
94	Settable	IRQ87		0x0000 019C
95	Settable	IRQ88		0x0000 01A0
96	Settable	IRQ89		0x0000 01A4
97	Settable	IRQ90	LCM	0x0000 01A8
98	Settable	IRQ91	ETH_WOL	0x0000 01AC
99	Settable	IRQ92	GPIO3	0x0000 01B0
100	Settable	IRQ93	GPIO2	0x0000 01B4
101	Settable	IRQ94	GPIO1	0x0000 01B8
102	Settable	IRQ95	GPIO0	0x0000 01BC

2.4.2 NVIC REGISTERS

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

2.4.2.1 IRQ0~239 Interrupt Set-Enable Register (NVIC_ISERn) (n=0~7)

Address: 0xE000 E100 + 0x4 * n (Refer to Cortex-M4 Spec.)

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

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

2.4.2.2 IRQ0~239 Interrupt Clear-Enable Register (NVIC_ICERn) (n=0~7)

Address: 0xE000 E180 + 0x4 * n (Refer to Cortex-M4 Spec.)

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

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

2.4.2.3 IRQ0~239 Interrupt Set-Pending Register (NVIC_ISPRn) (n=0~7)

Address: 0xE000 E200 + 0x4 * n (Refer to Cortex-M4 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	Field	Access	Initial	Description
31:0	SETPEND[31:0]	R/W	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

2.4.2.4 IRQ0~239 Interrupt Clear-Pending Register (NVIC_ICPRn)

Address: 0xE000 E280 + 0x4 * n (Refer to Cortex-M4 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	Field	Access	Initial	Description
31:0	CLRPEND[31:0]	R/W	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

2.4.2.5 IRQ0~239 Interrupt Active Bit Register (NVIC_IABRn)

Address: 0xE000 E300 + 0x4 * n (Refer to Cortex-M4 Spec.)

A bit reads as one if the status of the corresponding interrupt is active or active and pending.

Bit	Field	Access	Initial	Description
31:0	ACTIVE [31:0]	R	0	Interrupt active flags. 0: interrupt not active. 1: interrupt active.

2.4.2.6 IRQ0~239 Interrupt Priority Register (NVIC_IPRn) (n=0~59)

Address: 0xE000 E400 + 0x4 * n (Refer to Cortex-M4 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	Field	Access	Initial	Description
31:24	PRI_(4*n+3)	R/W	0	Each priority field holds a priority value, 0-240. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[31:28] of each field, bits [27:24] read as zero and ignore writes. This means writing 255 to a priority register saves value 240 to the register.
23:16	PRI_(4*n+2)	R/W	0	Each priority field holds a priority value, 0-240. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[23:20] of each field, bits [19:16] read as zero and ignore writes. This means writing 255 to a priority register saves value 240 to the register.
15:8	PRI_(4*n+1)	R/W	0	Each priority field holds a priority value, 0-240. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[15:12] of each field, bits [11:8] read as zero and ignore writes. This means writing 255 to a priority register saves value 240 to the register.
7:0	PRI_4*n	R/W	0	Each priority field holds a priority value, 0-240. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[7:4] of each field, bits [3:0] read as zero and ignore writes. This means writing 255 to a priority register saves value 240 to the register.

2.5 VECTOR TABLE OFFSET REGISTER (VTOR)

Address: 0xE000 ED08 (Refer to Cortex-M4 Spec)

The VTOR indicates the offset of the vector table base address from memory address 0x00000000. Bit 29 determines whether the vector table is in the code or SRAM memory region.

✸ ***Note:***

1. ***When setting TBLOFF, you must align the offset to the number of exception entries in the vector table. The minimum alignment is 32 words, enough for up to 16 interrupts. For more interrupts, adjust the alignment by rounding up to the next power of two.***
2. ***The start address of the new vector table must be a multiple of 0x80 because bits 0~6 of VTOR cannot be written.***

Bit	Field	Access	Initial	Description
31:7	TBLOFF	R/W	0	Vector table base offset field. It contains bits[29:7] of the offset of the table base from the bottom of the memory map. The lower bit in the TBLOFF field is implementation defined.
6:0	–	–	0	Reserved

2.6 APPLICATION INTERRUPT AND RESET CONTROL REGISTER (AIRCR)

Address: 0xE000 ED0C (Refer to Cortex-M4 Spec)

The entire MCU, including the core, can be reset by SW by setting the SYSRESREQ bit in the AIRC register in Cortex-M4 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	Field	Access	Initial	Description
31:16	VECTKEY	R/W	0	Register key. Read as unknown. Write 0x05FA to VECTKEY, otherwise the write is ignored.
15	ENDIANESS	R	0	Data endianness implemented 0: Little-endian 1: Big-endian
14:3	–	–	0	Reserved
10:8	PRIGROUP	R/W	0	Interrupt priority grouping field.
7:3	–	–	0	Reserved
2	SYSRESETREQ	W	0	System reset request. This bit read as 0. 0: No effect 1: Requests a system level reset.
1	VECTCLRACTIVE	W	0	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.
0	VECTRESET	W	0	Reserved for Debug use. Writing 1 to this bit causes a local system reset This bit self-clears. The effect of writing a 1 to this bit if the processor is not halted in Debug state is Unpredictable. When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is UNPREDICTABLE.

2.7 SYSTEM CONTROL REGISTER (SCR)

Address: 0xE000 ED10 (Refer to Cortex-M4 Spec)

The SCR controls features of entry to and exit from low power state.

> ➢ ***Note: SLEEPDEEP is set 1 to enable WIC function for wake-up system sleep mode.***

Bit	Field	Access	Initial	Description
31:5	–	–	0	Reserved
4	SEVONPEND	R/W	0	Send Event on Pending bit: 0 = only enabled interrupts or events can wakeup the processor, disabled interrupts are excluded 1 = enabled events and all interrupts, including disabled interrupts, can wakeup the processor. When an event or interrupt enters pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event is registered and affects the next WFE. The processor also wakes up on execution of an SEV instruction or an external event.

Bit	Field	Access	Initial	Description
3	–	–	0	Reserved
2	SLEEPDEEP	R/W	0	Controls whether the processor uses sleep or deep sleep as its low power mode: 0 = sleep (CPU) 1 = deep sleep (CPU)
1	SLEEPONEXIT	R/W	0	Indicates sleep-on-exit when returning from Handler mode to Thread mode: 0 = do not sleep when returning to Thread mode 1 = enter sleep, or deep sleep, on return from an ISR. Setting this bit to 1 enables an interrupt driven application to avoid returning to an empty main application.
0	–	–	0	Reserved

2.8 FLOATING POINT UNIT (FPU)

The Cortex-M4 FPU is an implementation of the single precision variant of the ARMv7-M Floating-Point Extension (FPv4-SP). It provides floating-point computation functionality that is compliant with the ANSI/IEEE Std 754-2008, IEEE Standard for Binary Floating-Point Arithmetic, referred to as the IEEE 754 standard. The FPU supports all single-precision data-processing instructions and data types described in the ARM Architecture Reference Manual.

The FPU fully supports single-precision add, subtract, multiply, divide, multiply and accumulate, and square root operations. It also provides conversions between fixed-point and floating-point data formats, and floating-point constant instructions

The FPU contains 32 single-precision extension registers, which you can also access as 16 double word registers for load, store, and move operations.

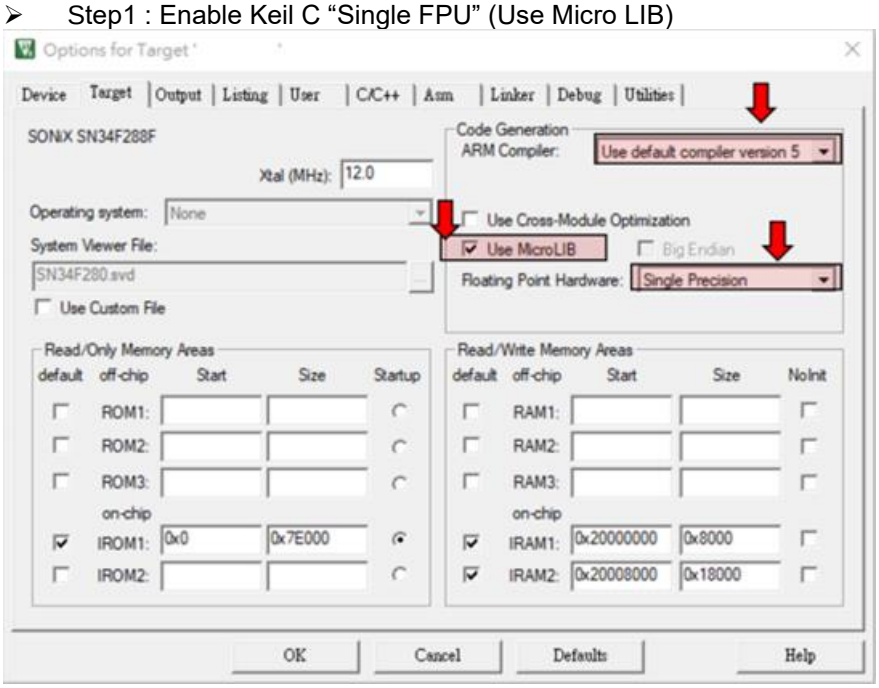
Refer to the *Cortex-M4 User Guide* for details.

2.8.1 ENABLING THE FPU

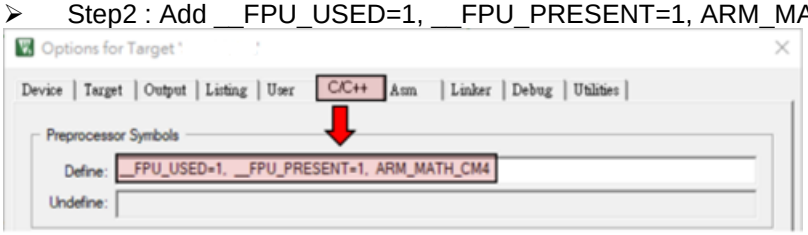
The FPU is disabled from reset. You must enable it before you can use any floating-point instructions.

2.8.1.1 Initialize method

- Step1 : Enable Keil C “Single FPU” (Use Micro LIB)

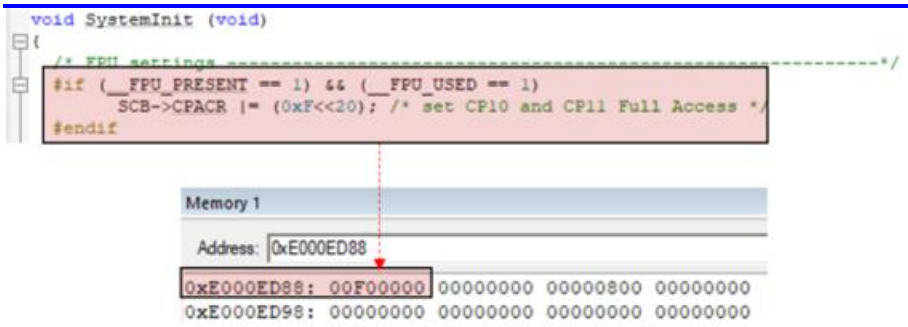


- Step2 : Add __FPU_USED=1, __FPU_PRESENT=1, ARM_MATH_CM4



- Step3 : main.c #include “arm_math.h”

- Step4 : Call SystemInit() of system_SN34F780.c for enable FPU



2.8.2 FPU REGISTERS

2.8.2.1 Coprocessor Access Control Register (SCB_CPACR)

Address: 0xE000 ED88 (Refer to Cortex-M4 Spec.)

The CPACR register specifies the access privileges for coprocessors.

Bit	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23:22	CP11	R/W	0	Access privileges for coprocessor 1. 00 = Access denied. Any attempted access generates a NOCP UsageFault. 01 = Privileged access only. An unprivileged access generates a NOCP fault. 10 = Reserved. The result of any access is Unpredictable. 11 = Full access.
21:20	CP10	R/W	0	Access privileges for coprocessor 0. 00 = Access denied. Any attempted access generates a NOCP UsageFault. 01 = Privileged access only. An unprivileged access generates a NOCP fault. 10 = Reserved. The result of any access is Unpredictable. 11 = Full access.
19:0	–	–	0	Reserved

2.8.2.2 Floating-point Context Control Register (FPU_FPCCR)

Address: 0xE000 EF34 (Refer to Cortex-M4 Spec.)

The FPCCR register sets or returns FPU control data.

Bit	Field	Access	Initial	Description
31	ASPEN	R/W	1	Enables CONTROL<2> setting on execution of a floating-point instruction. This results in automatic hardware state preservation and restoration, for floating-point context, on exception entry and exit. 0 = Disable CONTROL<2> setting on execution of a floating-point instruction. 1 = Enable CONTROL<2> setting on execution of a floating-point instruction.
30	LSPEN	R/W	1	0 = Disable automatic lazy state preservation for floating-point context. 1 = Enable automatic lazy state preservation for floating-point context.
29:9	–	–	0	Reserved
8	MONRDY	R/W	0	0 = DebugMonitor is disabled or priority did not permit setting MON_PEND when the floating-point stack frame was allocated. 1 = DebugMonitor is enabled and priority permits setting MON_PEND when the floating-point stack frame was allocated.
7	–	–	0	Reserved
6	BFRDY	R/W	0	0 = BusFault is disabled or priority did not permit setting the BusFault handler to the pending state when the floating-point stack frame was allocated. 1 = BusFault is enabled and priority permitted setting the BusFault handler

Bit	Field	Access	Initial	Description
				to the pending state when the floating-point stack frame was allocated.
5	MMRDY	R/W	0	0 = MemManage is disabled or priority did not permit setting the MemManage handler to the pending state when the floating-point stack frame was allocated. 1 = MemManage is enabled and priority permitted setting the MemManage handler to the pending state when the floating-point stack frame was allocated.
4	HFRDY	R/W	0	0 = Priority did not permit setting the HardFault handler to the pending state when the floating-point stack frame was allocated. 1 = Priority permitted setting the HardFault handler to the pending state when the floating-point stack frame was allocated.
3	THREAD	R/W	0	0 = Mode was not Thread Mode when the floating-point stack frame was allocated. 1 = Mode was Thread Mode when the floating-point stack frame was allocated.
2	–	–	0	Reserved
1	USER	R/W	0	0 = Privilege level was not user when the floating-point stack frame was allocated. 1 = Privilege level was user when the floating-point stack frame was allocated.
0	LSPACT	R/W	0	0 = Lazy state preservation is not active. 1 = Lazy state preservation is active. floating-point stack frame has been allocated but saving state to it has been deferred.

2.8.2.3 Floating-point Context Address Register (FPU_FPCAR)

Address: 0xE000 EF38 (Refer to Cortex-M4 Spec.)

The FPCAR register holds the location of the unpopulated floating-point register space allocated on an exception stack frame.

Bit	Field	Access	Initial	Description
31:3	ADDRESS	R/W	0	The location of the unpopulated floating-point register space allocated on an exception stack frame.
2:0	–	–	0	Reserved

2.8.2.4 Floating-point Status Control Register (FPU_FPSCR)

Address: Not mapped (Refer to Cortex-M4 Spec.)

The FPSCR register provides all necessary User level control of the floating-point system. Accessible only when software has enabled access to CP10 and CP11. You read or write the FPSCR, or transfer the FPSCR flags to the corresponding APSR flags, using the VMRS and VMSR instructions.

Bit	Field	Access	Initial	Description
31	N	R/W	0	Condition code flags. Floating-point comparison operations update these flags. N = Negative condition code flag. Z = Zero condition code flag. C = Carry condition code flag. V = Overflow condition code flag.
30	Z	R/W	0	
29	C	R/W	0	
28	V	R/W	0	
27	–	–	0	Reserved
26	AHP	R/W	0	Alternative half-precision control bit: 0 = IEEE half-precision format selected. 1 = Alternative half-precision format selected.
25	DN	R/W	0	Default NaN mode control bit: 0 = NaN operands propagate through to the output of a floating-point operation. 1 = Any operation involving one or more NaNs returns the Default NaN.
24	FZ	R/W	0	Flush-to-zero mode control bit: 0 = Flush-to-zero mode disabled. Behavior of the floating-point system is fully compliant with the IEEE 754 standard. 1 = Flush-to-zero mode enabled.

Bit	Field	Access	Initial	Description
23:22	RMODE[1:0]	R/W	0	Rounding Mode control field. The encoding of this field is: 00 = Round to Nearest (RN) mode 01 = Round towards Plus Infinity (RP) mode 10 = Round towards Minus Infinity (RM) mode 11 = Round towards Zero (RZ) mode. The specified rounding mode is used by almost all floating-point instructions.
21:8	–	–	0	Reserved
7	IDC	R/W	0	Input Denormal cumulative exception bit, see bits [4:0].
6:5	–	–	0	Reserved
4	IXC	R/W	0	Cumulative exception bits for floating-point exceptions, see also bit [7]. Each of these bits is set to 1 to indicate that the corresponding exception has occurred since 0 was last written to it. IDC, bit[7] = Input Denormal cumulative exception bit. IXC = Inexact cumulative exception bit. UFC = Underflow cumulative exception bit. OFC = Overflow cumulative exception bit. DZC = Division by Zero cumulative exception bit. IOC = Invalid Operation cumulative exception bit.
3	UFC	R/W	0	
2	OFC	R/W	0	
1	DZC	R/W	0	
0	IOC	R/W	0	

2.8.2.5 Floating-point Default Status Control Register (FPU_FPDSCR)

Address: 0xE000 EF3C (Refer to Cortex-M4 Spec.)

The FPDSCR register holds the default values for the floating-point status control data (FPSCR).

Bit	Field	Access	Initial	Description
31:27	–	–	0	Reserved
26	AHP	R/W	0	Default value for AHP bit of FPSCR
25	DN	R/W	0	Default value for DN bit of FPSCR
24	FZ	R/W	0	Default value for FZ bit of FPSCR
23:22	RMode[1:0]	R/W	0	Default value for RMode bits of FPSCR
21:0	–	–	0	Reserved

2.8.2.6 Media and VFP Feature Register 0 (FPU_MVFR0)

Address: 0xE000 EF40 (Refer to Cortex-M4 Spec.)

The VFP Feature Registers, MVFR0 and MVFR1, are read-only registers which describe the features supported by the FPU. These registers are accessible in Privileged modes only.

Bit	Field	Access	Initial	Description
31:28	RM	R	1	All VFP rounding modes supported
27:24	SV	R	0	VFP short vector unsupported
23:20	SR	R	1	VFP hardware square root supported
19:16	D	R	1	VFP hardware divide supported
15:12	TE	R	0	Only untrapped exception handling can be selected
11:8	DP	R	0	Double precision supported in VFPv3
7:4	SP	R	2	Single precision supported in VFPv3
3:0	RB	R	1	16x64-bit media register bank supported

2.8.2.7 Media and VFP Feature Register 1 (FPU_MVFR1)

Address: 0xE000 EF44 (Refer to Cortex-M4 Spec.)

The VFP Feature Registers, MVFR0 and MVFR1, are read-only registers which describe the features supported by the FPU. These registers are accessible in Privileged modes only.

Bit	Field	Access	Initial	Description
31:20	–	–	0x110	Reserved
19:16	SP	R	0	Single-precision floating-point operations supported for VFP

Bit	Field	Access	Initial	Description
				0b0000 = not supported
15:12	I	R	0	Integer operations supported for VFP 0b0000 = not supported
11:8	LS	R	0	Load and store instructions supported for VFP 0b0000 = not supported
7:4	DN	R	1	Propagation of NaN values supported for VFP
3:0	FZ	R	1	Full denormal arithmetic supported for VFP

2.9 CODE OPTION TABLE

Address: 0x0040 0500

Bit	Field	Access	Initial	Description
31:3	–	–	0	Reserved
2	BOOTPINEN	R/W	1	Boot Pin enable 0: Disable 1: Enable (default)
1:0	–	–	0	Reserved

2.10 UNIQUE NUMBER

The unique number is a 8-byte unique device serial number of each IC. In other words, the unique number is different and discontinuous for each IC. Users can use the unique number to pair in RF application, or use as USB string serial number.

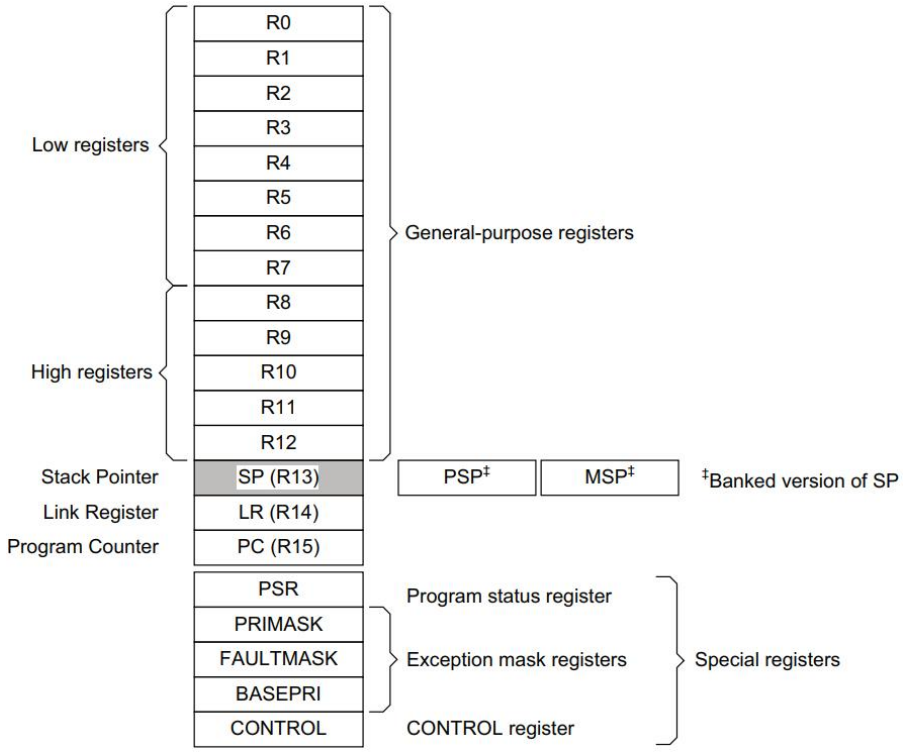
Address: 0x0040 0180

Bit	Field	Access	Initial	Description
31:0	L4BYTE[31:0]	R	By Die	Lower 4 bytes of Unique number

Address: 0x0040 0190

Bit	Field	Access	Initial	Description
31:0	H4BYTE[31:0]	R	By Die	High 4 bytes of Unique number

2.11 CORE REGISTER OVERVIEW



Register	Description (Refer to Cortex-M4 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. 31 30 29 28 27 26 25 24 23 … 16 15 … 10 9 8 … 0 APSR: N Z C V Q | Reserved IPSR: Reserved | ISR_NUMBER EPSR: Reserved | ICI/IT | T | Reserved | ICI/IT | Reserved
PRIMASK	The PRIMASK register prevents activation of all exceptions with configurable priority.
FAULTMASK	The FAULTMASK register prevents activation of all exceptions except for Non-Maskable Interrupt (NMI).
BASEPRI	The BASEPRI register defines the minimum priority for exception processing. When BASEPRI is set to a nonzero value, it prevents the activation of all exceptions with the same or lower priority level as the BASEPRI value.
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. Power-on reset (POR)
2. Watchdog Timer reset (WDT)
3. Programmable low voltage detection reset (LVD)
4. A low level on the RST pin (external reset, EXT).
5. Cortex-M4 Software reset (SW)
6. Deep power-down wake-up (DPDWK)
7. Reboot command (REBOOT)
8. Multiple peripherals reset (Peripheral reset command, PERRST)
9. Individual peripheral module reset (IPRST)

Reset range after reset event is issued

Function & Register	Address	POR	DPDWK	EXT	LVD	WDT	SW	PERRST	IPRST	REBOOT
Initialization	N/A	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
SCU register (offset 0x000)	0x4001 F000	Yes	No[1]	No[1]	No[1]	No[1]	No[1]	No	No	No
SCU register (offset 0x004)	0x4001 F004	Yes	No	Yes	Yes	Yes	Yes	No	No	No
SCU register (offset 0x010~0x0C4)	0x4001 F010 ~ 0x4001 F0C4	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
SCU RTC register	0x4001 F200 ~ 0x4001 F22C	Yes	No	No	No	No	No	No	No	No
SCU register (offset 0x800~0xBFC)	0x4001 F800 ~ 0x4001 FBFC	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
Always-on domain register (offset 0x000~0x240)	0x4001 B000 ~ 0x4001 B240	Yes	No	Yes	Yes	Yes	Yes	No	No	No
Always-on domain register - POR Miscellaneous Control register	0x4001 B110	Yes	No	No	No	No	No	No	No	No
GPIO0/1/2/3	0x4002 0000 ~ 0x4002 3FFF	Yes	No	Yes	Yes	Yes	Yes	Maskable	Maskable	Yes
CPU	CM4 registers	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes
Other peripherals	Others	Yes	Yes	Yes	Yes	Yes	Yes	Maskable	Maskable	Yes
Flash Memory Control	0x0060 1000 ~ 0x0060 1FFC	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
	Initialization: 1. Set clock mux/pre-scaler to default value 2. Set System clock source to IHRC, PLL disable 3. Remap bit to default value(from flash) 4. Flash Initialization flow									

Yes : Reset after event issue. No : Not reset after event issue.
[1] Reference the reset range of the BTUP_STS register

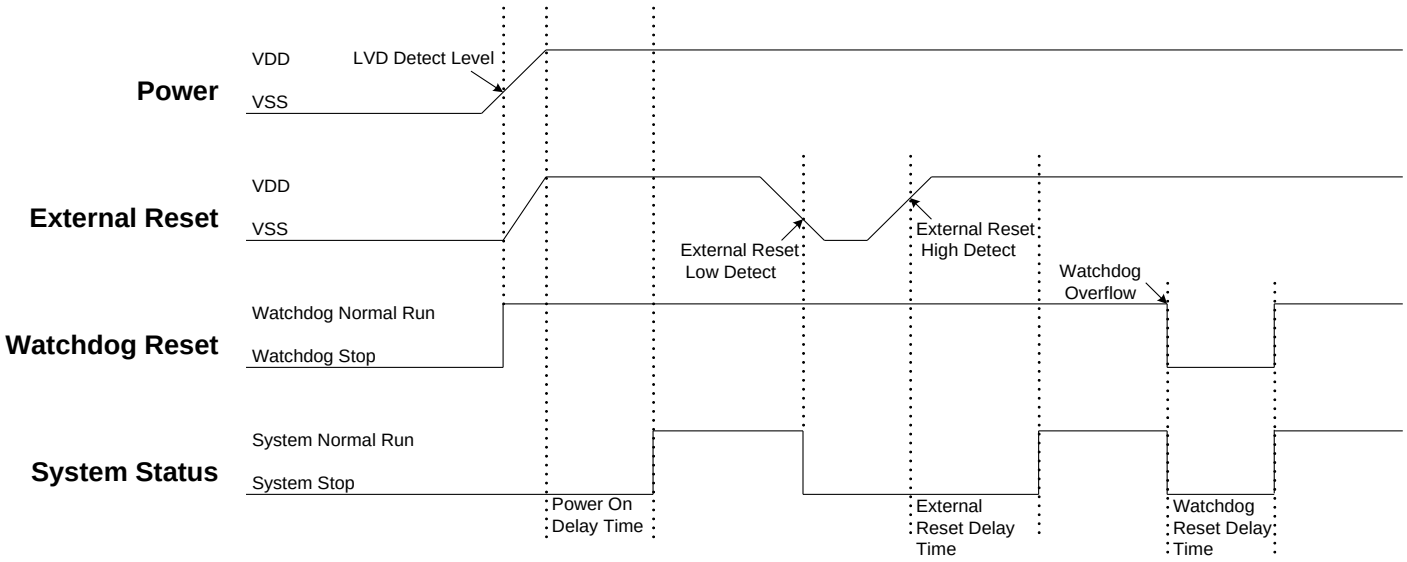
Reset range of the BTUP_STS register after reset event is issued

Boot-up Status Register	POR	Deep Power Down Wake-up	EXT Reset	LVD Reset	WDT Reset	CM4 SW Reset
[24:17] BTUP_STS	Yes	Keep current wakeup event Other is cleared	Yes	Yes	Yes	Yes
[15] SWRSTF	Yes	Yes	No	No	No	No
[12] LVDRSTF	Yes	Yes	No	No	No	No
[10] DPDWKF	Yes	No	Yes	Yes	Yes	Yes
[9] WDTRSTF	Yes	Yes	No	No	No	No
[8] EXTRSTF	Yes	Yes	No	No	No	No
[2] GPIO_HOLD	Yes	No	Yes	Yes	Yes	Yes

Yes : Reset after event issue. No : Not reset after event issue.

The reset source can be identified by checking the reset flags in Boot-up Status register (SCU_BTUP_STS) and Power Miscellaneous Control register (ALWAYS_POR_MISC).
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.
- **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. Program executes from User code if Boot byte is 0x5A, otherwise 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 underflow. Watchdog timer underflow 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 underflow status. If watchdog timer underflow 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. Program executes from User code if Boot byte is 0x5A, otherwise from Boot loader.

Watchdog timer application note is as following.

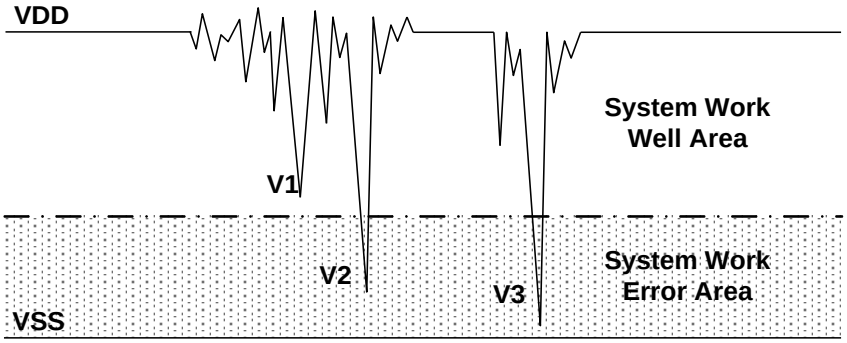
- Before clearing watchdog timer, check I/O status and check RAM contents can improve system error.
- Don't clear watchdog timer in interrupt vector and interrupt service routine. That can improve main routine fail.
- Clearing watchdog timer program is only at one part of the program. This way is the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

✸ ***Note:***
1. ***Please refer to the "WATCHDOG TIMER" and the "WINDOW WATCHDOG TIMER" about watchdog timer detail information.***
2. ***In debug mode, when the system is paused, the WDT/WWDT counter clock will be stopped to ensure that WDT reset is not issued.***

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.

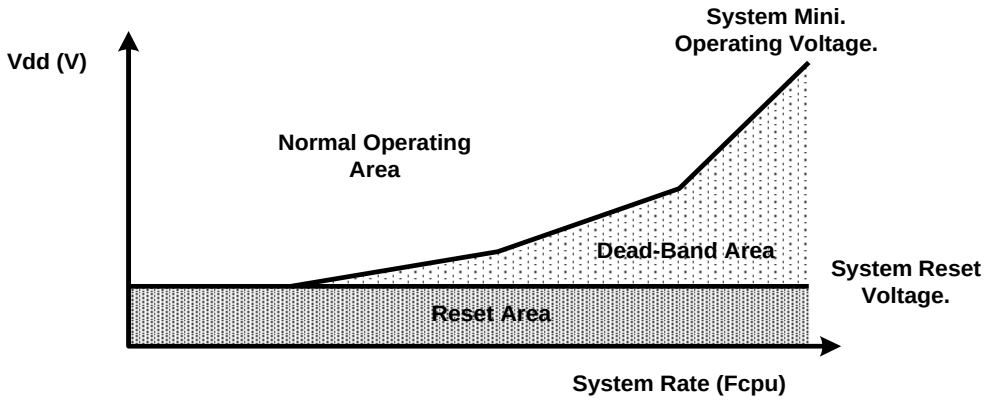
DC application:
The power source of DC application is usually using battery. When low battery condition and MCU drive any loading, the power drops and keeps in dead-band. Under the situation, the power won't drop deeper and not touch the system reset voltage. That makes the system under dead-band.

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.

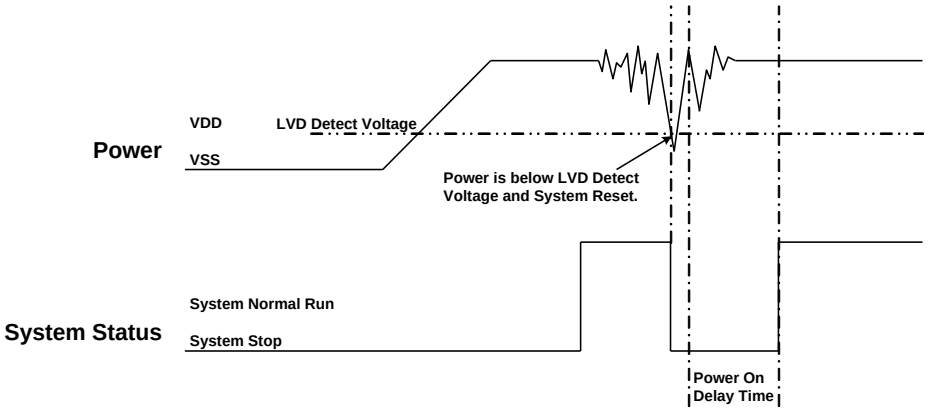
3.1.3.3 BROWN-OUT RESET IMPROVEMENT

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

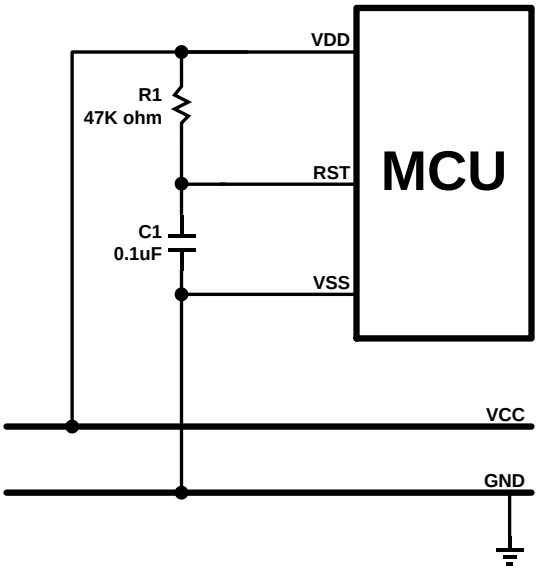
External reset function is controlled by External RESET pin control (alternate function of P3.7) bit. External reset pin is Schmitt Trigger structure and low level active. The system is running when reset pin is high level voltage input. The reset pin receives the low voltage and the system is reset. The external reset operation actives in power down mode and normal running mode. 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. Program executes from User code if Boot byte is 0x5A,

otherwise from Boot loader.

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

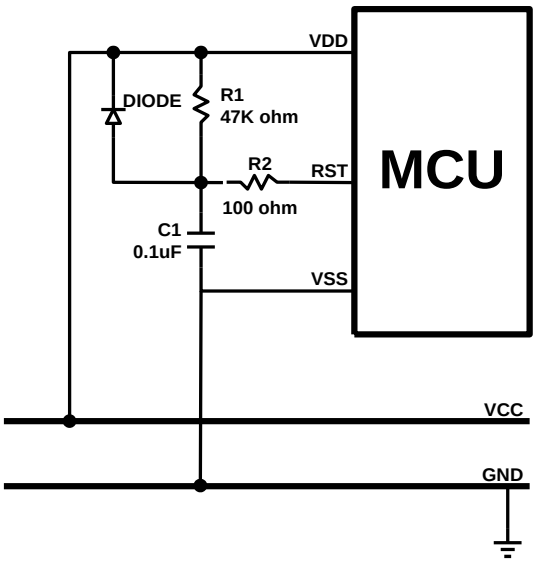
3.1.4.1 SIMPLY RC RESET CIRCUIT



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

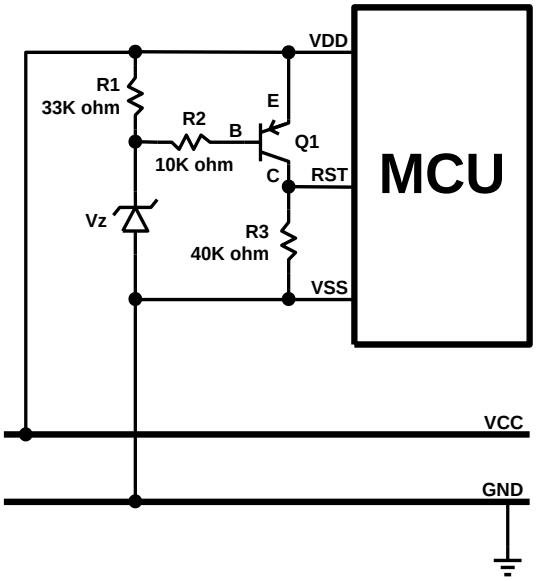


This is the better reset circuit. The R1 and C1 circuit operation is like the simply reset circuit to make a power on signal.

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.

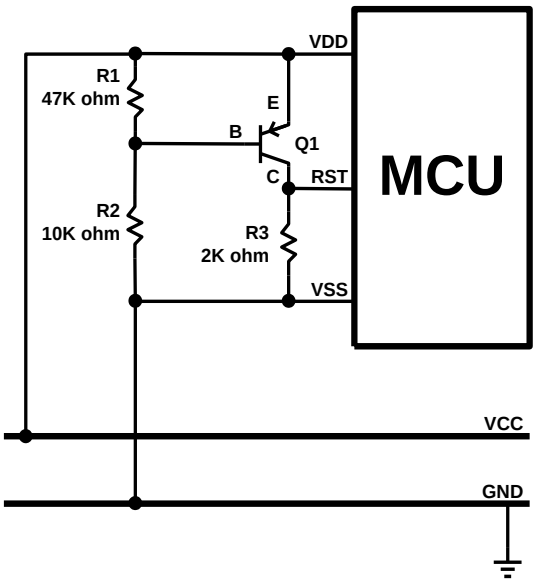
✷ ***Note: The R2 100 ohm resistor of "Simply reset circuit" and "Diode & RC reset circuit" is necessary to limit any current flowing into reset pin from external capacitor C in the event of reset pin breakdown due to Electrostatic Discharge (ESD) or Electrical Over-stress (EOS).***

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

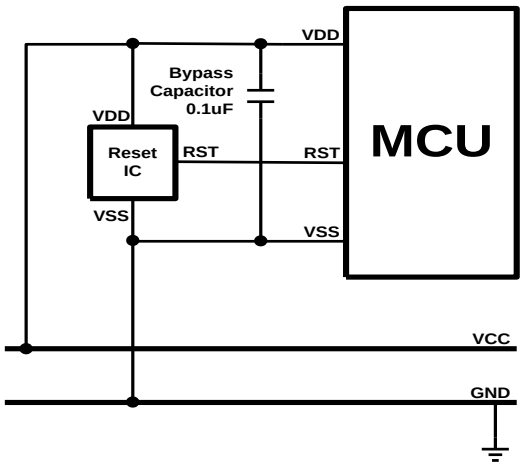


The voltage bias reset circuit is a low cost voltage detector and can **improve brown out reset condition completely**. The operating voltage is not accurate as Zener diode reset circuit. Use R1, R2 bias voltage to be the active level. When VDD voltage level is above or equal to "0.7V x (R1 + R2) / R1", 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

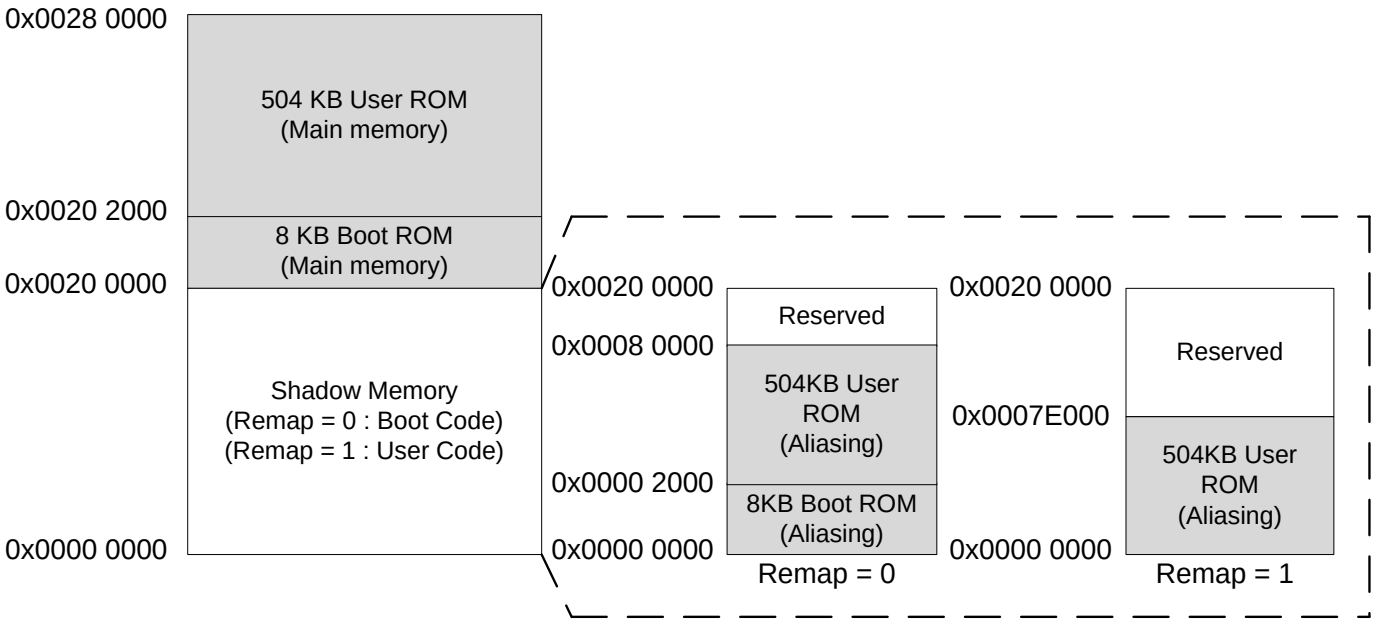
The entire MCU, including the core, can be reset by software by setting the SYSRESREQ bit in the AIRCR (Application Interrupt and Reset Control Register) in Cortex-M4 spec.

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.
4. Program executes from User code if Boot byte is 0x5A, otherwise from Boot loader.

3.1.6 REBOOT RESET

The REBOOT bit of SCU_PWRMODE register is set to issue a reset to initiate the CPU and bus state. Reboot command can use to change memory remapping. Program REMAP bit before REBOOT is set. The default value of REMAP bit is latched from Flash's boot byte.

The Operating sequence of remap is as follows:
1. Write the new value to REMAP bit of SCU_PWRMODE register.
2. Set REBOOT bit of SCU_PWRMODE register.
3. CPU, data bus and all peripherals will be reset.
4. After reset is complete, CPU reboots from address 0x0 and the shadow memory domain is new code domain (boot loader or user code by REMAP bit).



✷ ***Note:***
1. ***After system resets, the shadow memory domain is selected by boot byte setting. The REMAP will reload the boot byte value.***
2. ***In debug mode, LVD, SW, WDT and EXT resets do not affect the REMAP bit. After reset, the shadow memory will remain in its original state.***

3.1.7 PERIPHERALS RESET

The PERRST bit of SCU_PWRMODE register is set to issue a reset to initiate multiple peripherals. Program reset mask registers[1] before PERRST is set. The mask bits will auto clear after PRERST done.

The Individual RST bit[2] is set to issue a reset to initiate specified peripheral. The bits will auto clear after reset done.

[1] Reset mask registers : SCU_AHBRSTMSK, APB0RSTMSK and APB1RSTMSK register
[2] The Individual RST bits : AHBRST, APB0RST, APB1RST registers

Peripheral reset method	Multiple peripheral reset (PERRST)	Individual peripheral reset (IPRST)
Description	Reset multiple peripherals simultaneously.	Reset a single peripheral.
Reset range	No mask peripherals	Specified peripheral
Operating sequence	1. Set reset mask bits for the specified peripherals don't be reset 2. Set the PERRST bit to start the reset operation 3. After the reset is completed, the PERRST bit and the reset mask bits are automatically cleared to 0	1. Set specified peripheral RST bit to start the reset operation 2. After the reset is completed, the specified peripheral RST bit is automatically cleared to 0
Peripheral Reset Register	PERRST (The bit 10 of SCU_PWRMODE register)	AHBRST, APB0RST, APB1RST registers
Reset Mask Register	SCU_AHBRSTMSK, APB0RSTMSK and APB1RSTMSK register	N/A
(Option) Reset configuration	Delay and active cycle (SCU PRERSTCTL register)	N/A

3.2 SYSTEM CLOCK

Different clock sources can be used to drive the system clock (SYSCLK):
- 12 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 32 KHz).

3.2.1 INTERNAL RC CLOCK SOURCE

3.2.1.1 Internal High-speed RC Oscillator (IHRC)

- **Structure:** 12 MHz RC type.
- **Main Purpose:** System high clock source and PLL clock source.
- **Warm-up Time: 256***F_{IHRC}

The internal high-speed oscillator is 12MHz RC type. The accuracy is ±2% under commercial condition.
The IHRC can be switched on and off using the IHRCEN bit in IHRC Oscillator Control register (SCU_IHRCCTRL).

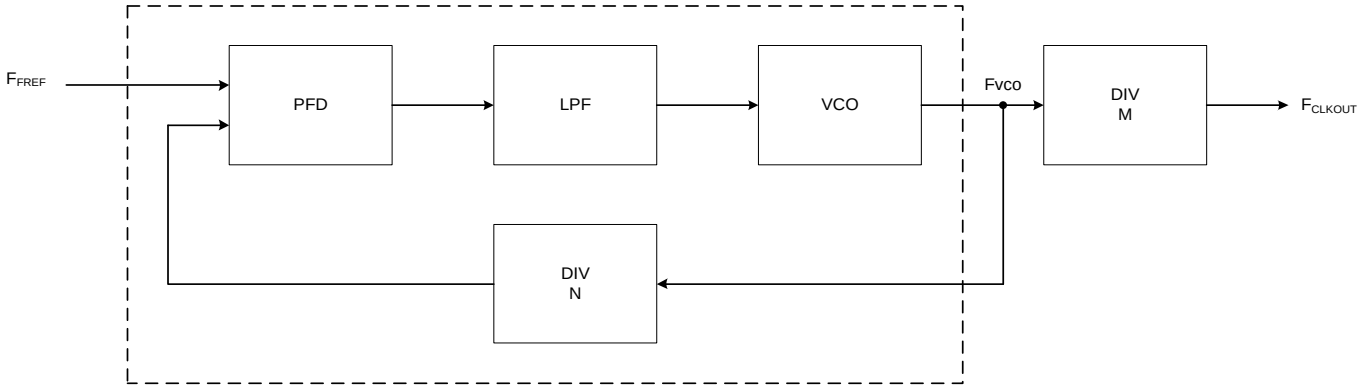
3.2.1.2 Internal Low-speed RC Oscillator (ILRC)

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 PLL

SONiX 32-bit MCU uses the PLL to create the clocks for the core and peripherals. The input frequency range is 10MHz to 50MHz. 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 320MHz to 800MHz. These clocks are divided by M by the programmable post divider to create the output clock(s). The VCO output clock is then divided by N 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.



3.2.2.1 PLL Frequency selection

The PLL frequency equations:

$$F_{VCO} = F_{FREF} * N$$

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

The PLL frequency is determined by the following parameters:

- F_{FREF}: Frequency from the PLLCLKSEL multiplexer.
- F_{VCO}: Frequency of the Voltage Controlled Oscillator (VCO); 320 to 800 MHz.
- F_{CLKOUT}: Frequency of PLL output.
- M: System PLL post divider ratio, controlled by FS[1:0] bits in PLL control register (SCU_PLLCTRL).
- N: System PLL feedback divider ratio, controlled by NS[6:0] bits in PLL control register (SCU_PLLCTRL).

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

1. 10MHz ≤ F_{FREF} ≤ 50MHz
2. 320MHz ≤ F_{VCO} ≤ 800MHz
3. N = 6 ~ 80
4. M = 4, 8, 16 or 32 (duty 50% +/- 4%)
5. F_{CLKOUT} = 10MHz ~ 200MHz with jitter < ±500 ps
6. The maximum speed of HCLK is 192MHz. When F_{CLKOUT} exceeds 192MHz, it must be ensured that HCLK does not exceed 192MHz.

3.2.2.2 PLL Configuration

Follow the steps below to configure PLL:

1. Configuration PLL parameter
   - Select PLL input reference clock. (PLLCLKSEL of SCU_PLLCTRL)
   - Set loop divider. (NS[6:0] of SCU_PLLCTRL)
   - Set output divider. (FS[1:0] of SCU_PLLCTRL)
   - Enable PLL. (PLLEN of SCU_PLLCTRL)
2. Execute frequency change sequence
   - Set FCS bit for enter frequency change sequence. (FCS of SCU_PWRMODE)
   - Set FCS_EINT bit of SCU_IE and NVIC for wake-up system.
   - Set WFI to enter idle mode.
3. Frequency change sequence in idle mode (CM4 WFI)
   - The frequency change sequence with system clock gating in idle mode.
   - The system clock is released after the frequency change sequence is completed.
   - SCU interrupt is issued to wake-up idle mode.
4. Wake-up after frequency change sequence
   - CPU wake-up and enters SCU interrupt subroutine (ISR)
   - Clear INT_FCS flag for next FCS operation. (INT_FCS of SCU_RIS)
   - Check PLL ready flag. (PLLSTABLE of SCU_PLLCTRL)
   - Check PLL configuration status (SCU_PLLSTS register)

When FCS_PLLRSTOFF = 0, the timing of frequency conversion sequence is 4T ~ 5T*ILRC.

Disable PLL steps:

1. Clear PLLEN
2. Execute frequency change sequence
   - Set FCS bit for enter frequency change sequence. (FCS of SCU_PWRMODE)
   - Set FCS_EINT bit of SCU_IE and NVIC for wake-up system.
   - Set WFI to enter idle mode.
3. Frequency change sequence in idle mode (CM4 WFI)
   - The frequency change sequence with system clock gating in idle mode.
   - The system clock is released after the frequency change sequence is completed.
   - SCU interrupt is issued to wake-up idle mode.
4. Wake-up after frequency change sequence
   - CPU wake-up and enters SCU interrupt subroutine (ISR)
   - Clear INT_FCS flag for next FCS operation. (INT_FCS of SCU_RIS)
   - Check PLL ready flag. (PLLSTABLE of SCU_PLLCTRL)
   - Check PLL configuration status (SCU_PLLSTS register)

✸ ***Note:***

1. ***If FW changes configuration while the PLL is enabled, the PLL needs to reset for lock frequency. If the FCS_PLLRSTOFF bit of SCU_PWRMODE is cleared, PLL will be reset while the frequency change sequence.***
2. ***Before changing the system clock frequency, make sure the new AHB clock is less than or equal to the flash operating frequency. The flash operating frequency can be adjusted up before changing to avoid program code read errors.***

3.2.2.3 Frequency Setting

$F_{VCO} = F_{FREF} * N$

$F_{CKOUT} = F_{VCO} / M$

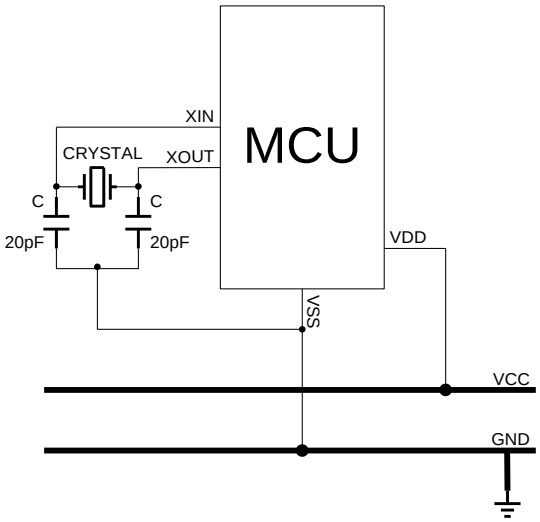
F_{FREF}(MHz)	NS[6:0]	N	F_{VCO}(MHz)= F_{FREF} *N 320 ~ 800 MHz	FS[1:0]	M	F_{CKOUT}(MHz)= F_{VCO}/M System Clock Max. 192Mz
10	010 0000b	**32**	**320**	11	**4**	80
10	010 1000b	**40**	**400**	11	**4**	100
10	011 1100b	**60**	**600**	11	**4**	150
10	100 1100b	**76**	**760**	11	**4**	190
12	010 0000b	**32**	**384**	11	**4**	96
12	010 1000b	**40**	**480**	11	**4**	120
12	011 1100b	**60**	**720**	11	**4**	180
12	100 0000b	**64**	**768**	11	**4**	192
25	001 0000b	**16**	**400**	11	**4**	100
25	001 0100b	**20**	**500**	11	**4**	125
25	001 1000b	**24**	**600**	11	**4**	150
25	001 1100b	**28**	**700**	11	**4**	175

3.2.3 EXTERNAL CLOCK SOURCE

3.2.3.1 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.

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:** 10MHz~25MHz EHS external crystal/ceramic resonator
- **Main Purpose:** System high clock source and PLL clock source.
- **Warm-up Time:** 4096*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.

EHS crystal circuit	Control bit	Register
Crystal on/off	EHSEN	Oscillator control register (ALWAYSON_OSCCTL)
Internal resistor on/off	EHSINROFF	
Frequency range 0x1, L1: 10MHz ~ 12MHz 0x2, L2: 12MHz ~ 16MHz 0x3, L3: 16MHz ~ 25MHz	EHSFREQ[1:0]	Oscillator control register (ALWAYSON_OSCMISC)
Ready flag	EHSRDY	Oscillator ready register (ALWAYSON _OSCRDY)

- **SMD Crystal Limitation**

Crystal	12MHz		16MHz	25MHz
VDD	**2.4V~3.6V**		**2.4V~3.6V**	**2.44V~3.6V**
EHSFREQ[1:0]	0x1, L1	0x2, L2	0x2, L2	0x3, L3
Max. C0	3pF	4pF	3pF	3pF
Max. ESR	120ohm	150ohm	90ohm	60ohm
1st cycle warm-up time	Typ. 0.71ms	Typ. 1.59ms	Typ. 0.49ms	Typ. 0.13ms
current consumption	Typ. 1.72mA	Typ. 2.85mA	Typ. 2.46mA	Typ. 7.14mA

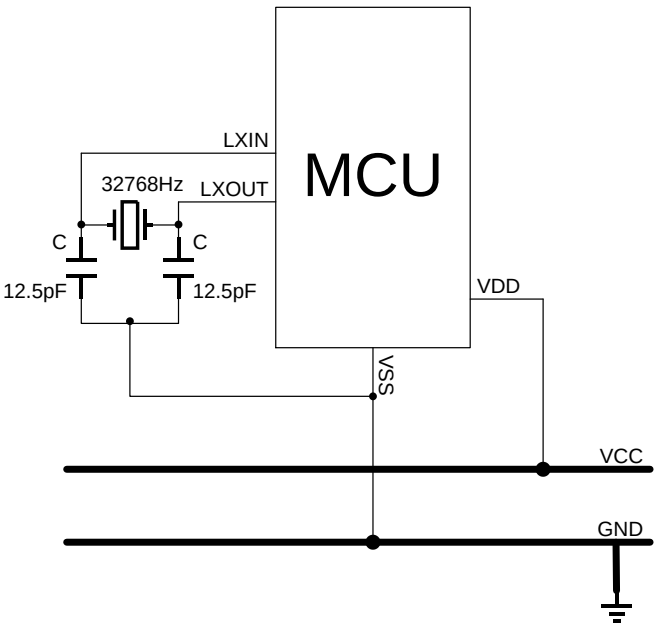
DIP is easier to start-up than SMD so there are no limitations.

3.2.3.2 External Low-speed (ELS) Clock

- **Structure:** 32.768KHz external crystal/ ceramic resonator
- **Main Purpose:** RTC and CT16B5 clock source.
- **Warm-up Time:** 32768*F_{ELS}. 32K crystal builds in speed-up warm-up time function.

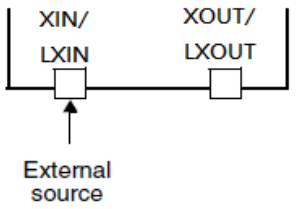
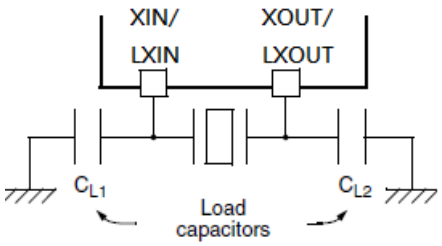
The low-speed oscillator can use 32768 crystal oscillator circuit.
Crystal devices are driven by LXIN, LXOUT pins. The 32768 crystal and 10pF capacitor must be as near as possible to MCU.

ELS crystal circuit	Control bit	Register
Crystal on/off	ELSEN	POR Miscellaneous Control register (ALWAYSON_POR_MISC)
Clock filter on/off	ELSFLOFF	
Ready flag	ELSRDY	



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

3.2.3.3 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 Oscillator control register (ALWAYSON_OSCCTL). ELS X'TAL must have a frequency of 32.768 KHz. You select this mode by setting ELSEN bit in POR Miscellaneous Control register (ALWAYSON_POR_MISC).
External X'TAL (EHS/ELS X'TAL)		The 10 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 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.

Make sure the target clock source is ready before switching. Unexpected conditions may occur if a clock source that is not ready is selected. Ready bits in ALWAYS_OSCRDY, SCU_IHRCRDY, SCU_PLLCTRL registers indicate which clock(s) is (are) ready.

SYSCLKST bits in SCU_PLLSTS register indicate which clock is currently used as system clock.

3.2.4.1 System clock Configuration

Follow the steps below to configure system clock:

1. Select system clock source. (SYSCLKSEL[2:0] of SCU_PLLCTRL)
2. Execute frequency change sequence
   - Set FCS bit for enter frequency change sequence. (FCS of SCU_PWRMODE)
   - Set FCS_EINT bit of SCU_IE and NVIC for wake-up system.
   - Set WFI to enter idle mode.
3. Frequency change sequence in idle mode (CM4 WFI)
   - The frequency change sequence with system clock gating in idle mode.
   - The system clock is released after the frequency change sequence is completed.
   - SCU interrupt is issued to wake-up idle mode.
4. Wake-up after frequency change sequence
   - CPU wake-up and enters SCU interrupt subroutine (ISR)
   - Clear INT_FCS flag for next FCS operation. (INT_FCS of SCU_RIS)
   - Check system clock status. (SYSCLKSTS of SCU_PLLSTS)

✷ ***Note:***
1. Before changing the system clock frequency, make sure the new AHB clock is less than or equal to the flash operating frequency. The flash operating frequency can be adjusted up before changing to avoid program code read errors.

3.2.5 AHB/APB CLOCK

Bus clock	Reference clock	Pre-scaler control	Pre-scaler range	Max. Speed
AHB HCLK	SYSCLK	AHBPRE[2:0]	/1~ /128, (2^N)	192MHz
APB0 APB0CLK	HCLK	APB0PRE[2:0]	/1~ /128, (2^N)	48MHz
APB1 APB1CLK	HCLK	APB1PRE[2:0]	/1~ /128, (2^N)	96MHz

Follow the steps below to configure AHB/APB clock pre-scaler:
1. Program the CLKPRE register to configure the AHB/APB clock pre-scaler. When programming, bits 31~16 must be written with 0x5AFA at the same time.
2. Configuration bus clock pre-scaler
   - AHB clock pre-scaler. (AHBPRE[2:0] of SCU_CLKPRE)
   - APB0 clock pre-scaler. (APB0PRE[2:0] of SCU_CLKPRE)
   - APB1 clocks pre-scaler. (APB1PRE[2:0] of SCU_CLKPRE)

✷ ***Note:***
1. Before changing the system clock frequency, make sure the new AHB clock is less than or equal to the flash operating frequency. The flash operating frequency can be adjusted up before changing to avoid program code read errors.

3.2.6 FREQUENCY CHANGE SEQUENCE

The Frequency Change Sequence (FCS) is a way to change the system clock and PLL configuration without re-booting. If the configured values of PLL are changed, it needs to reset and execute the re-lock procedure. FCS provides the safe sequence to complete the PLL re-lock procedure. The main clock will be stopped in the interval to regress the lock process for PLL, so CPU will be in the idle mode (CM4 WFI). The new PLL configuring values should be programmed through the PLLCTRL register.

3.2.6.1 Operation Steps

1. Configuration PLL parameter.
   - Select PLL input reference clock. (PLLCLKSEL of SCU_PLLCTL)
   - Set loop divider. (NS[6:0] of SCU_PLLCTL)
   - Set output divider. (FS[1:0] of SCU_PLLCTL)
   - Enable PLL. (PLLEN of SCU_PLLCTL)
2. Select system clock source. (SYSCLKSEL[2:0] of SCU_PLLCTL)
3. Program SCU_PWRMODE register.
   - (Optional) EFC_STB_OFF as 1: flash normal run.
   - (Optional) FCS_PLLRSTOFF as 0: Reset PLL for re-locking
4. Ensure all peripherals will not disturb the changing sequence.
   - Disable all peripheral interrupt to avoid wake-up system.
5. Clear SCU_RIS register to avoid wake-up idle mode.
6. Execute frequency change sequence
   - Set FCS bit for enter frequency change sequence. (FCS of SCU_PWRMODE)
   - Set FCS_EINT bit of SCU_IE and NVIC for wake-up system.
   - Set WFI to enter idle mode.
7. Frequency change sequence in idle mode (CM4 WFI)

- The frequency change sequence with system clock gating in idle mode.
- (Option) If FCS_PLLRSTOFF = 0, it resets PLL.
- Update PLL and SYSCLK setting.
- (Option) If FCS_PLLRSTOFF = 0, it waits for PLL locking.
- The system clock is released after the frequency change sequence is completed.
- SCU interrupt is issued to wake-up idle mode.

8. Wake-up after frequency change sequence
   - CPU wake-up, enter SCU interrupt subroutine (ISR) and clear INT_FCS flag. (INT_FCS of SCU_RIS)
9. Restore all interrupt setting.
10. Check system clock and PLL status.
    - Check PLL configuration and system clock status (SCU_PLLSTS register)
    - Check PLL ready flag. (PLLSTABLE of SCU_PLLCTL)

When FCS_PLLRSTOFF = 0, the timing of frequency change sequence is 4T ~ 5T*ILRC.
When FCS_PLLRSTOFF = 1, the timing of frequency change sequence is 1T ~ 2T*ILRC.

✷ ***Note:***
1. ***If FW changes configuration while the PLL is enabled, the PLL needs to reset for lock frequency. If the FCS_PLLRSTOFF bit of SCU_PWRMODE is cleared, PLL will be reset while the frequency change sequence.***
2. ***Before changing the system clock frequency, make sure the new AHB clock is less than or equal to the flash operating frequency. The flash operating frequency can be adjusted up before changing to avoid program code read errors.***

3.2.7 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

CLKOUT circuit	Control bit	Register
Clock enable	CLKOUTEN	APB0 Clock Control register (SCU_APB0CLKG) APB0 Clock Sleep Control register (SCU_SLP_APB0CLKG)
Source select	CLKOUTSEL[2:0]	Peripheral Clock Source Select register (SCU_CLKSEL)
Clock pre-scaler	CLKOUTPRE[3:0]	System and Peripheral Clock Prescale register (SCU_CLKPRE)

Follow the steps below to configure CLKOUT:
1. Disable CLKOUTEN.
2. Select CLKOUT share pin. (P3.0, P0.12 and P0.10 alternate function bits)
3. Select CLKOUT clock source. (CLKOUTSEL[2:0] of SCU_CLKSEL)
4. Program the CLKPRE register to configure the CLKOUT clock pre-scaler. When programming, bits 31~16 must be written with 0x5AFA at the same time. (CLKOUTPRE[3:0] of SCU_CLKPRE)
5. Enable CLKOUTEN.

3.3 SYSTEM CONTROL UNIT REGISTERS

Base Address: 0x4001 F000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	SCU_BTUP_STS	SCU Boot-up Status register
0x0004	SCU_BTUP_CTRL	SCU Boot-up Control register
0x0008 – 0x000C	–	Reserved
0x0010	SCU_CHIPID	SCU RTC Chip ID register
0x0014	SCU_VERSION	SCU RTC Version register
0x0018 – 0x001C	–	Reserved
0x0020	SCU_PWRMODE	SCU Power Mode register
0x0024	SCU_RIS	SCU Raw Interrupt Status register
0x0028	SCU_IE	SCU Interrupt Enable register
0x002C	SCU_PERRSTCTL	SCU PERRST reset control register
0x0030	SCU_PLLCTRL	SCU PLL Control register
0x0034 – 0x004C	–	Reserved
0x0050	SCU_AHBCLKG	SCU AHB Clock Control register
0x0054	–	Reserved
0x0058	SCU_SLP_AHBCLKG	SCU AHB Clock Sleep Control register
0x005C	–	Reserved
0x0060	SCU_APB0CLKG	SCU APB0 Clock Control register
0x0064	SCU_APB1CLKG	SCU APB1 Clock Control register
0x0068	SCU_SLP_APB0CLKG	SCU APB0 Sleep Mode Clock Control register
0x006C	SCU_SLP_APB1CLKG	SCU APB1 Sleep Mode Clock Control register
0x0070 – 0x00BC	–	Reserved
0x00C0	SCU_SLP_WAKUPST	SCU Sleep Wakeup Events Status register
0x00C4	SCU_SLP_WAKUPEN	SCU Sleep Wakeup Events Enable register
0x00C8 –0x0124	–	Reserved
0x0128	SCU_PWRMODEMISC	SCU Power Mode Clear register
0x012C –0x01FC	–	Reserved
0x0200 – 0x022C	–	Please reference RTC function registers
0x0230 – 0x07FC	–	Reserved
0x0800	SCU_CLKSEL	SCU Peripheral Clock Source Select register
0x0804	SCU_CLKPRE	SCU System and Peripheral Clock Prescale register
0x0808	SCU_IHRCCTRL	SCU IHRC Oscillator Control register
0x080C	SCU_PLLSTS	SCU PLL and System Clock Status register
0x0810	SCU_AHBRST	SCU AHB Peripheral Reset Control register
0x0814	SCU_APB0RST	SCU APB0 Peripheral Reset Control register
0x0818	SCU_APB1RST	SCU APB1 Peripheral Reset Control register
0x081C – 0x083C	–	Reserved
0x0840	SCU_PRECTRL	SCU Peripheral Clock Prescale register
0x0844 – 0x08FC	–	Reserved
0x0900	SCU_P0STR	SCU GPIO0.0~0.15 Driving/Sinking Strength Control register
0x0904	SCU_P1STR	SCU GPIO1.0~1.15 Driving/Sinking Strength Control register
0x0908	SCU_P2STR	SCU GPIO2.0~2.15 Driving/Sinking Strength Control register
0x090C	SCU_P3STR	SCU GPIO3.0~3.13 Driving/Sinking Strength Control register
0x0910	SCU_P0STR1	SCU GPIO0.20 Driving/Sinking Strength Control register
0x0914 – 0x0A0C	–	Reserved
0x0A10	SCU_AHBRSTMSK	SCU AHB Peripheral PERRST Reset Mask register
0x0A14	SCU_APB0RSTMSK	SCU APB0 Peripheral PERRST Reset Mask register
0x0A18	SCU_APB1RSTMSK	SCU APB1 Peripheral PERRST Reset Mask register
0x0A1C – 0x0BE8	–	Reserved
0x0BEC	SCU_GMAC_SC	SCU GMAC Special Control register
0x0BF0	SCU_SRAM1CTRL	SCU SRAM1 controller sideband signal register
0x0BF4	SCU_SRAM2CTRL	SCU SRAM2 controller sideband signal register
0x0BF8	SCU_SRAM3CTRL	SCU SRAM3 controller sideband signal register

0x0BFC	SCU_BKPSRAMCTRL	SCU BKPSRAM controller sideband signal register

3.3.1 SCU Boot-up Status register (SCU_BTUP_STS)

Address Offset: 0x00

When entering deep-sleep mode or deep power-down, the system will clear bits 17~24 to 0. Therefore, after the system is woken up, only the current wakeup event is kept when returning to normal mode.

The system can be restarted by SW, LVD, WDT or EXT reset event. Bits 17~24 and the DPDWKF bit will be cleared when these reset events occur. After the system returns to normal mode, the corresponding reset flag will be set.

Bits 17~24 and bits 8~15 will be cleared by hardware before getting into deep power mode.

The GPIO_HOLD bit will be set to high when entering deep power down mode. It will be cleared to 0 in SW, LVD, WDT or EXT reset event.

Bit	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23	ETHWKF	RW1C	0	Ethernet wake-on-LAN wakeup flag (wakeup_event 7) 0: No wakeup by Ethernet wake-on-LAN 1: Wakeup by Ethernet wake-on-LAN
22	–	–	0	Reserved
21	P3WKF	RW1C	0	GPIO3 interrupt wakeup flag (wakeup_event 5) 0: No wakeup by GPIO3 interrupt 1: Wakeup by GPIO3 interrupt
20	P2WKF	RW1C	0	GPIO2 interrupt wakeup flag (wakeup_event 4) 0: No wakeup by GPIO2 interrupt 1: Wakeup by GPIO2 interrupt
19	P1WKF	RW1C	0	GPIO1 interrupt wakeup flag (wakeup_event 3) 0: No wakeup by GPIO1 interrupt 1: Wakeup by GPIO1 interrupt
18	P0WKF	RW1C	0	GPIO0 interrupt wakeup flag (wakeup_event 2) 0: No wakeup by GPIO0 interrupt 1: Wakeup by GPIO0 interrupt
17	RTCWKF	RW1C	0	The RTC alarm wakeup flag (wakeup_event 1) 0: No wakeup by RTC alarm 1: Wakeup by RTC alarm
16	–	–	0	Reserved
15	SWRSTF	RW1C	0	Software reset flag (Cortex-M4 Software reset). Write one to clear 0: No software reset occurred 1: Software reset occurred
14:13	–	–	0	Reserved
12	LVDRSTF	RW1C	0	LVD reset flag 0: No LVD reset occurred 1: LVD reset occurred
11	–	–	0	Reserved
10	DPDWKF	RW1C	0	Deep power-down wakeup flag 0: No wakeup from DPD mode 1: Wakeup from DPD mode
9	WDTRSTF	RW1C	0	Watchdog reset flag. 0: No Watchdog reset occurred 1: Watchdog reset occurred
8	EXTRSTF	RW1C	0	External reset flag 0: No External reset occurred 1: External reset occurred
7:3	–	–	0	Reserved
2	GPIO_HOLD	RW1C	0	Auto Hold the GPIO output function when deep power down mode. 0: GPIO was controlled by the pin alternative function 1: GPIO output function was held Set by hardware when enter deep power down mode, and clear by software programming.
1:0	–	–	0	Reserved

3.3.2 SCU Boot-up Control register (SCU_BTUP_CTRL)

Address Offset: 0x04

Bit	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23	ETHWKEN	RW	1	Enable the wakeup by the Ethernet Wake-on-LAN Frame (wakeup_event 7) 0: Disable 1: Enable
22	–	–	0	Reserved
21	P3WKEN	RW	1	Enable the wakeup from the GPIO3 interrupt (wakeup_event 5) 0: Disable 1: Enable
20	P2WKEN	RW	1	Enable the wakeup by the GPIO2 interrupt (wakeup_event 4) 0: Disable 1: Enable
19	P1WKEN	RW	1	Enable the wakeup by the GPIO1 interrupt (wakeup_event 3) 0: Disable 1: Enable
18	P0WKEN	RW	1	Enable the wakeup by the GPIO0 interrupt (wakeup_event 2) 0: Disable 1: Enable
17	RTCWKEN	RW	1	Enable the wakeup by the RTC alarm (wakeup_event 1) 0: Disable 1: Enable
16:0	–	–	0	Reserved

3.3.3 SCU RTC Chip ID register (SCU_CHIPID)

Address Offset: 0x10

The CHIDID register is protected to avoid unexpected programming. The unlock RTC ID is the same CHIP_ID, and writes the value to the RTC version register. It will unlock the write protected and allow one write operation to the CHIPID register, and the register will resume the lock state after the write operation. When CHIP_ID programmed, the unlock key will also be updated to new CHIP_ID. User should use new CHIP_ID to unlock the protected register in the next write procedure.

Bit	Field	Access	Initial	Description
31:0	CHIP_ID	RPW	0x4281	Chip ID

3.3.4 SCU RTC Version register (SCU_VERSION)

Address Offset: 0x14

Bit	Field	Access	Initial	Description
31:0	VERSION	R	0x30100	IP version

3.3.5 SCU Power Mode register (SCU_PWRMODE)

Address Offset: 0x20

Bit	Field	Access	Initial	Description
31	EFC_STB_OFF	RW	0	eFlash stand-by command off 0: eFlash enters standby mode in all power down mode and frequency change sequence. 1: eFlash normal run
30:29	–	–	0	Reserved
28	FCS_PLLRSTOFF	RW	0	PLL resets in the frequency change sequence 0: Reset PLL for locking 1: keep PLL active in FCS
27:25	–	–	0	Reserved
24	REMAP	RW	0	Target remap programmable register After issuing the reboot function for the changed address remapping of the BOOT_CODE/USER_CODE. 0: Shadow memory is BOOT CODE 1: Shadow memory is USER CODE
23:11	–	–	0	Reserved
10	PERRST	RWS	0	Peripheral reset command 0: No effect 1: Issue a reset to initiate multiple peripherals The bit is set to issue a reset signal that the interval can be adjusted by programming PERRSTCTL, and be cleared by hardware after the reset is done.
9:8	–	–	0	Reserved
7	REBOOT	RWS	0	Reboot command 0: No effect 1: Issue a reset to initiate the CPU and bus state
6	FCS	RWS	0	Frequency Change Sequence (FCS) 0: No effect 1: Enter the frequency change sequence cleared after finishing FCS. The command does not execute with the SLEEP, DSLEEP, or DPD command. It can be clear to 1'b0 when PWRMODECLR bit is set to 1'b0.
5:4	–	–	0	Reserved
3	SLEEP	RWS	0	Sleep-mode sequence 0: Not sleep mode 1: Enter the sleep mode This bit will be automatically cleared after waking up. The command does not execute with the FCS, DSLEEP, or DPD command. It can be clear to 1'b0 when PWRMODECLR bit is set to 1'b0.
2	DSLEEP	RWS	0	Deep sleep mode sequence 0: Not Deep sleep mode 1: Enter the Deep sleep mode This bit will be automatically cleared after waking up. The command does not execute with the FCS, SLEEP, or DPD command. It can be clear to 1'b0 when PWRMODECLR bit is set to 1'b0.
1	DPD	RWS	0	Deep power-down mode sequence 0: Not Deep power-down mode 1: Enter the Deep power-down mode This bit will be automatically cleared after waking up. The command does not execute with the FCS, SLEEP, or DSLEEP command. It can be clear to 1'b0 when PWRMODECLR bit is set to 1'b0.
0	–	–	0	Reserved

3.3.6 SCU Raw Interrupt Status register (SCU_RIS)

Address Offset: 0x24

Bit	Field	Access	Initial	Description
31:19	–	–	0	Reserved
18	INT_RTC_SEC	RW1C	0	RTC second out interrupt status 0: No RTC second out occurred 1: RTC second out occurred
17	INT_RTC_PER	RW1C	0	RTC periodic interrupt status 0: No RTC periodic occurred 1: RTC periodic occurred
16	INT_RTC_ALARM	RW1C	0	RTC alarm interrupt status 0: No RTC alarm occurred 1: RTC alarm occurred
15:12	–	–	0x0	Reserved, don't modify it
11	INT_REMAPCHG	RW1C	0	Remap is changed status after reboot command 0: Not change 1: REMAP flag is changed
10:7	–	–	0	Reserved, don't modify it
6	INT_FCS	RW1C	0	FCS command finish interrupt status 0: Not finish 1: FCS command finished
5:4	–	–	0	Reserved
3	INT_WAKEUP	RW1C	0	Sleep mode wakeup interrupt status 0: No wakeup from sleep mode 1: Wakeup from sleep mode
2	INT_DS_WAKEUP	RW1C	0	Deep sleep mode wakeup interrupt status 0: No wakeup from deep sleep mode 1: Wakeup from deep sleep mode
1:0	–	–	0	Reserved

3.3.7 SCU Interrupt Enable register (SCU_IE)

Address Offset: 0x28

Bit	Field	Access	Initial	Description
31:19	–	–	0	Reserved
18	RTC_SEC_EINT	RW	0	RTC second interrupt enable bit 0: Disable 1: Enable
17	RTC_PER_EINT	RW	0	RTC periodic interrupt enable bit 0: Disable 1: Enable
16	RTC_ALARM_EINT	RW	0	RTC alarm interrupt enable bit 0: Disable 1: Enable
15:12	–	–	0	Reserved
11	REMAPCHG_EINT	RW	0	REMAP change interrupt enable 0: Disable REMAP change interrupt 1: Enable REMAP change interrupt
10:7	–	–	0x0	Reserved
6	FCS_EINT	RW	0	FCS command finish interrupt enable bit 0: Disable 1: Enable
5:4	–	–	0x0	Reserved
3	WAKEUP_EINT	RW	0	Sleep mode Wake-up event interrupt enable bit 0: Disable 1: Enable
2	DS_WAKEUP_EINT	RW	0	Deep sleep mode Wake-up event interrupt enable bit 0: Disable

Bit	Field	Access	Initial	Description
				1: Enable
1:0	–	–	0x0	Reserved

3.3.8 SCU PERRST reset control register (SCU_PERRSTCTL)

Address Offset: 0x2C

Bit	Field	Access	Initial	Description
31:16	PERRSTDC	RW	0	PERRST reset delay cycles of APB0 bus clock
15:0	PERRSTAC	RW	0x0F	PERRST reset active cycles of APB0 bus clock

3.3.9 SCU PLL Control register (SCU_PLLCTRL)

Address Offset: 0x30

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

Bit	Field	Access	Initial	Description
31	–	–	0	Reserved
30:24	NS	RW	0x40	7-bit programmable loop divider. PLLCLK = FREF * NS/FS Valid value: 6~80
23:10	–	–	0	Reserved
9:8	FS	RW	0x3	PLL frequency range output divider 0: PLL CKOUT= PLLCLK / 32 1: PLL CKOUT= PLLCLK / 16 2: PLL CKOUT = PLLCLK / 8 3: PLL CKOUT = PLLCLK / 4 This field indicates that the output frequency range of the embedded the PLL controls the PLL frequency output. (M output divider)
7:5	SYSCLKSEL	RW	0	System clock source 0: IHRC 2: EHS 4: PLL Other: Reserved This field changes the system clock source in the FCS mode.
4	PLLCLKSEL	RW	0	PLL clock source 0: IHRC 1: EHS This field changes the system clock source in the FCS mode.
3:2	–	–	0	Reserved
1	PLLSTABLE	R	0	PLL locking and stability status 0: Not stable 1: Stable
0	PLLEN	RW	0	PLL enable control bit 0: Disable 1: Enable The PLL cannot directly be turned ON/OFF when the bit is set. The procedure should accompany a FCS command to stop or enable PLL.

3.3.10 SCU AHB Clock Control register (SCU_AHBCLKG)

Address Offset: 0x50

The SCU_AHBCLKG 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. ***User shall disable the AHB clock for individual peripheral to decrease power consumption by demand.***

Bit	Field	Access	Initial	Description
31:15	–	–	0	Reserved
14	ETHSRAMEN	RW	1	Enable clock for Ethernet MAC SRAM 0: Disable 1: Enable
13:11	–	–	0	Reserved
10	BKPSRAMEN	RW	1	Enable clock for Backup SRAM 0: Disable 1: Enable
9	–	–	0	Reserved
8	ETHCLKEN	RW	0	Enable clock for Ethernet MAC 0: Disable 1: Enable
7	–	–	0	Reserved
6	SDIOCLKEN	RW	0	Enable clock for SDIO 0: Disable 1: Enable
5:2	–	–	0	Reserved
1	DMA1CLKEN	RW	0	Enable clock for DMA1 0: Disable 1: Enable
0	DMA0CLKEN	RW	0	Enable clock for DMA0 0: Disable 1: Enable

3.3.11 SCU AHB Clock Sleep Control register (SCU_SLP_AHBCLKG)

Address Offset: 0x58

The SCU_SLP_AHBCLKG register enables the AHB clock to individual system and peripheral blocks in sleep mode.

✷ ***Note:***
1. ***User shall disable the AHB clock for individual peripheral to decrease power consumption by demand.***

Bit	Field	Access	Initial	Description
31:15	–	–	0	Reserved
14	ETHSRAMEN	RW	1	Enable clock for Ethernet MAC SRAM in sleep mode 0: Disable 1: Enable
13	H2P1HCLKEN	RW	0	Enable clock for AHB to APB1 bridge in sleep mode 0: Disable 1: Enable

Bit	Field	Access	Initial	Description
12	H2P0HCLKEN	RW	0	Enable clock for AHB to APB0 bridge in sleep mode 0: Disable 1: Enable
11	AHBMCLKEN	RW	0	Enable clock for AHB bus Matrix in sleep mode 0: Disable 1: Enable
10	BKPSRAMEN	RW	1	Enable clock for Backup SRAM in sleep mode 0: Disable 1: Enable
9	–	–	0	Reserved
8	ETHCLKEN	RW	0	Enable clock for Ethernet MAC in sleep mode 0: Disable 1: Enable
7	–	–	0	Reserved
6	SDIOCLKEN	RW	0	Enable clock for SDIO in sleep mode 0: Disable 1: Enable
5	FLASHEN	RW	0	Enable clock for FLASH and FMC in sleep mode 0: Disable 1: Enable
4	SRAM3EN	RW	0	Enable clock for SRAM3 in sleep mode 0: Disable 1: Enable
3	SRAM2EN	RW	0	Enable clock for SRAM2 in sleep mode 0: Disable 1: Enable
2	SRAM1EN	RW	0	Enable clock for SRAM1 in sleep mode 0: Disable 1: Enable
1	DMA1CLKEN	RW	0	Enable clock for DMA1 in sleep mode 0: Disable 1: Enable
0	DMA0CLKEN	RW	0	Enable clock for DMA0 in sleep mode 0: Disable 1: Enable

3.3.12 SCU APB0 Clock Control register (SCU_APB0CLKG)

Address Offset: 0x60

The SCU_APB0CLKG register enables the APB0 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. ***User shall disable the APB0 clock for individual peripheral to decrease power consumption by demand.***

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15	CLKOUTEN	RW	0	Enable clock for CLKOUT 0: Disable 1: Enable
14	CAN1SRAMCLKEN	RW	1	Enable clock for CAN1 SRAM 0: Disable 1: Enable
13	CAN0SRAMCLKEN	RW	1	Enable clock for CAN0 SRAM 0: Disable 1: Enable

Bit	Field	Access	Initial	Description
12	–	–	0	Reserved
11	–	–	0	Reserved
10	ALWAYSONREGEN	RW	1	Enable clock for Always-on domain registers. 0: Disable 1: Enable
9	CT16B7CLKEN	RW	0	Enable clock for CT16B7 0: Disable 1: Enable
8	CT16B6CLKEN	RW	0	Enable clock for CT16B6 0: Disable 1: Enable
7	UART4CLKEN	RW	0	Enable clock for UART4 0: Disable 1: Enable
6	UART3CLKEN	RW	0	Enable clock for UART3 0: Disable 1: Enable
5	UART2CLKEN	RW	0	Enable clock for UART2 0: Disable 1: Enable
4	UART1CLKEN	RW	0	Enable clock for UART1 0: Disable 1: Enable
3	CAN1CLKEN	RW	0	Enable clock for CAN1 0: Disable 1: Enable
2	CAN0CLKEN	RW	0	Enable clock for CAN0 0: Disable 1: Enable
1	WWDTCLKEN	RW	0	Enable clock for WWDT 0: Disable 1: Enable
0	WDTCLKEN	RW	0	Enable clock for WDT 0: Disable 1: Enable

3.3.13 SCU APB1 Clock Control register (SCU_APB1CLKG)

Address Offset: 0x64

The SCU_APB1CLKG register enables the APB1 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. ***User shall disable the APB1 clock for individual peripheral to decrease power consumption by demand.***

Bit	Field	Access	Initial	Description
31:22	–	–	0	Reserved
21	LCMCLKEN	RW	0	Enable clock for LCM 0: Disable 1: Enable
20	P3CLKEN	RW	1	Enable clock for GPIO3 0: Disable 1: Enable
19	P2CLKEN	RW	1	Enable clock for GPIO2 0: Disable

Bit	Field	Access	Initial	Description
				1: Enable
18	P1CLKEN	RW	1	Enable clock for GPIO1 0: Disable 1: Enable
17	P0CLKEN	RW	1	Enable clock for GPI0 0: Disable 1: Enable
16	CRCCLKEN	RW	0	Enable clock for CRC 0: Disable 1: Enable
15	CT16B8CLKEN	RW	0	Enable clock for CT16B8 0: Disable 1: Enable
14	CT16B5CLKEN	RW	0	Enable clock for CT16B5 0: Disable 1: Enable
13	CT16B4CLKEN	RW	0	Enable clock for CT16B4 0: Disable 1: Enable
12	CT16B3CLKEN	RW	0	Enable clock for CT16B3 0: Disable 1: Enable
11	CT16B2CLKEN	RW	0	Enable clock for CT16B2 0: Disable 1: Enable
10	CT16B1CLKEN	RW	0	Enable clock for CT16B1 0: Disable 1: Enable
9	CT16B0CLKEN	RW	0	Enable clock for CT16B0 0: Disable 1: Enable
8	I2C2CLKEN	RW	0	Enable clock for I2C2 0: Disable 1: Enable
7	I2C1CLKEN	RW	0	Enable clock for I2C1 0: Disable 1: Enable
6	I2C0CLKEN	RW	0	Enable clock for I2C0 0: Disable 1: Enable
5	UART5CLKEN	RW	0	Enable clock for UART5 0: Disable 1: Enable
4	UART0CLKEN	RW	0	Enable clock for UART0 0: Disable 1: Enable
3	SSP2CLKEN	RW	0	Enable clock for SSP2 0: Disable 1: Enable
2	SSP1CLKEN	RW	0	Enable clock for SSP1 0: Disable 1: Enable
1	SSP0CLKEN	RW	0	Enable clock for SSP0 0: Disable 1: Enable
0	ADC0CLKEN	RW	0	Enable clock for ADC0 0: Disable 1: Enable

3.3.14 SCU APB0 Sleep Mode Clock Control register (SCU_SLP_APB0CLKG)

Address Offset: 0x68

The SCU_SLP_APB0CLKG register enables the APB0 clock to individual system and peripheral blocks in sleep mode.

✷ ***Note:***
1. User shall disable the APB0 clock for individual peripheral to decrease power consumption by demand.

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15	CLKOUTEN	RW	0	Enable clock for CLKOUT in sleep mode 0: Disable 1: Enable
14	CAN1SRAMCLKEN	RW	1	Enable clock for CAN1 SRAM in sleep mode 0: Disable 1: Enable
13	CAN0SRAMCLKEN	RW	1	Enable clock for CAN0 SRAM in sleep mode 0: Disable 1: Enable
12	–	–	0	Reserved
11	–	–	0	Reserved
10	ALWAYSONREGEN	RW	1	Enable clock for Always-On domain registers in sleep mode 0: Disable 1: Enable
9	CT16B7CLKEN	RW	0	Enable clock for CT16B7 in sleep mode 0: Disable 1: Enable
8	CT16B6CLKEN	RW	0	Enable clock for CT16B6 in sleep mode 0: Disable 1: Enable
7	UART4CLKEN	RW	0	Enable clock for UART4 in sleep mode 0: Disable 1: Enable
6	UART3CLKEN	RW	0	Enable clock for UART3 in sleep mode 0: Disable 1: Enable
5	UART2CLKEN	RW	0	Enable clock for UART2 in sleep mode 0: Disable 1: Enable
4	UART1CLKEN	RW	0	Enable clock for UART1 in sleep mode 0: Disable 1: Enable
3	CAN1CLKEN	RW	0	Enable clock for CAN1 in sleep mode 0: Disable 1: Enable
2	CAN0CLKEN	RW	0	Enable clock for CAN0 in sleep mode 0: Disable 1: Enable
1	WWDTCLKEN	RW	0	Enable clock for WWDT in sleep mode 0: Disable 1: Enable
0	WDTCLKEN	RW	0	Enable clock for WDT in sleep mode 0: Disable 1: Enable

3.3.15 SCU APB1 Sleep Mode Clock Control register (SCU_SLP_APB1CLKG)

Address Offset: 0x6C

The SCU_SLP_APB1CLKG register enables the APB1 clock to individual system and peripheral blocks in sleep mode.

✷ ***Note:***
1. User shall disable the APB1 clock for individual peripheral to decrease power consumption by demand.

Bit	Field	Access	Initial	Description
31:22	–	–	0	Reserved
21	LCMCLKEN	RW	0	Enable clock for LCM in sleep mode 0: Disable 1: Enable
20	P3CLKEN	RW	1	Enable clock for GPIO3 in sleep mode 0: Disable 1: Enable
19	P2CLKEN	RW	1	Enable clock for GPIO2 in sleep mode 0: Disable 1: Enable
18	P1CLKEN	RW	1	Enable clock for GPIO1 in sleep mode 0: Disable 1: Enable
17	P0CLKEN	RW	1	Enable clock for GPIO0 in sleep mode 0: Disable 1: Enable
16	CRCCLKEN	RW	0	Enable clock for CRC in sleep mode 0: Disable 1: Enable
15	CT16B8CLKEN	RW	0	Enable clock for CT16B8 in sleep mode 0: Disable 1: Enable
14	CT16B5CLKEN	RW	0	Enable clock for CT16B5 in sleep mode 0: Disable 1: Enable
13	CT16B4CLKEN	RW	0	Enable clock for CT16B4 in sleep mode 0: Disable 1: Enable
12	CT16B3CLKEN	RW	0	Enable clock for CT16B3 in sleep mode 0: Disable 1: Enable
11	CT16B2CLKEN	RW	0	Enable clock for CT16B2 in sleep mode 0: Disable 1: Enable
10	CT16B1CLKEN	RW	0	Enable clock for CT16B1 in sleep mode 0: Disable 1: Enable
9	CT16B0CLKEN	RW	0	Enable clock for CT16B0 in sleep mode 0: Disable 1: Enable
8	I2C2CLKEN	RW	0	Enable clock for I2C2 in sleep mode 0: Disable 1: Enable
7	I2C1CLKEN	RW	0	Enable clock for I2C1 in sleep mode 0: Disable 1: Enable
6	I2C0CLKEN	RW	0	Enable clock for I2C0 in sleep mode 0: Disable 1: Enable
5	UART5CLKEN	RW	0	Enable clock for UART5 in sleep mode 0: Disable

Bit	Field	Access	Initial	Description
				1: Enable
4	UART0CLKEN	RW	0	Enable clock for UART0 in sleep mode 0: Disable 1: Enable
3	SSP2CLKEN	RW	0	Enable clock for SSP2 in sleep mode 0: Disable 1: Enable
2	SSP1CLKEN	RW	0	Enable clock for SSP1 in sleep mode 0: Disable 1: Enable
1	SSP0CLKEN	RW	0	Enable clock for SSP0 in sleep mode 0: Disable 1: Enable
0	ADC0CLKEN	RW	0	Enable clock for ADC0 in sleep mode 0: Disable 1: Enable

3.3.16 SCU Sleep Wakeup Events Status register (SCU_SLP_WAKUPST)

Address Offset: 0xC0

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1	SLP_WAKUP_INT	RW1C	0	Wakeup from sleep by interrupt 0: No occurred 1: Occurred
0	–	–	0	Reserved

3.3.17 SCU Sleep Wakeup Events Enable register (SCU_SLP_WAKUPEN)

Address Offset: 0xC4

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1	SLP_WAKUP_INTEN	RW	1	Sleep wakeup enable for NVIC interrupts 0: Disable 1: Enable
0	–	–	0	Reserved

3.3.18 SCU Power Mode Clear register (SCU_PWRMODEMISC)

Address Offset: 0x128

Bit	Field	Access	Initial	Description
31:13	–	–	0	Reserved
12	PWRMODECLR	W	1	Always 1'b1 when reading. Writing 0 will clear FCS/SLEEP/DSLEEP/DPD bits of PWRMODE to 1'b0.
11:0	–	–	0	Reserved

3.3.19 SCU Peripheral Clock Source Select register (SCU_CLKSEL)

Address Offset: 0x800

Bit	Field	Access	Initial	Description
31:29	CLKOUTSEL	RW	0	Clock output source 0: Disable 1: ILRC 32kHz 2: ELS 4: HCLK 5: IHRC 6: EHS 7: PLL Other: Reserved
28:23	–	–	0	Reserved
22	LCMCLKSEL	RW	0	LCM clock source 0: LCMCLK=HCLK 1: LCMCLK=PLLCLK
21	WWDTCLKSEL	RW	1	WWDT clock source 0: EXTCLK=APB0CLK 1: EXTCLK=ILRC 32kHz
20	–	–	0	Reserved
19	SDIOCLKSEL	RW	0	SDIO clock source 0: SDIOCLK=HCLK 1: SDIOCLK=PLLCLK
18	SSP2CLKSEL	RW	0	SSP2 SSPCLK clock source 0: SSP2CLK=HCLK 1: SSP2CLK=PLLCLK
17	SSP1CLKSEL	RW	0	SSP1 SSPCLK clock source 0: SSP1CLK=HCLK 1: SSP1CLK=PLLCLK
16	SSP0CLKSEL	RW	0	SSP0 SSPCLK clock source 0: SSP0CLK=HCLK 1: SSP0CLK=PLLCLK
15	UART5CLKSEL	RW	0	UART5 UCLK clock source 0: UART5UCLK=HCLK 1: UART5UCLK=PLLCLK
14	UART4CLKSEL	RW	0	UART4 UCLK clock source 0: UART4UCLK=HCLK 1: UART4UCLK=PLLCLK
13	UART3CLKSEL	RW	0	UART3 UCLK clock source 0: UART3UCLK=HCLK 1: UART3UCLK=PLLCLK
12	UART2CLKSEL	RW	0	UART2 UCLK clock source 0: UART2UCLK=HCLK 1: UART2UCLK=PLLCLK
11	UART1CLKSEL	RW	0	UART1 UCLK clock source 0: UART1UCLK=HCLK 1: UART1UCLK=PLLCLK
10	UART0CLKSEL	RW	0	UART0 UCLK clock source 0: UART0UCLK=HCLK 1: UART0UCLK=PLLCLK
9	CT16B8CLKSEL	RW	0	CT16B8 counter clock source 0: CT16B8 PCLK=HCLK 1: CT16B8 PCLK=PLLCLK
8	CT16B7CLKSEL	RW	0	CT16B7 counter clock source 0: CT16B7 PCLK=HCLK 1: CT16B7 PCLK=PLLCLK
7	CT16B6CLKSEL	RW	0	CT16B6 counter clock source 0: CT16B6 PCLK=HCLK 1: CT16B6 PCLK=PLLCLK
6:5	CT16B5CLKSEL	RW	0	CT16B5 counter clock source 0: CT16B5 PCLK=HCLK 1: CT16B5 PCLK=PLLCLK 2: CT16B5 PCLK=ELS

Bit	Field	Access	Initial	Description
				Other: Reserved
4	CT16B4CLKSEL	RW	0	CT16B4 counter clock source 0: CT16B4 PCLK=HCLK 1: CT16B4 PCLK=PLLCLK
3	CT16B3CLKSEL	RW	0	CT16B3 counter clock source 0: CT16B3 PCLK=HCLK 1: CT16B3 PCLK=PLLCLK
2	CT16B2CLKSEL	RW	0	CT16B2 counter clock source 0: CT16B2 PCLK=HCLK 1: CT16B2 PCLK=PLLCLK
1	CT16B1CLKSEL	RW	0	CT16B1 counter clock source 0: CT16B1 PCLK=HCLK 1: CT16B1 PCLK=PLLCLK
0	CT16B0CLKSEL	RW	0	CT16B0 counter clock source 0: CT16B0 PCLK=HCLK 1: CT16B0 PCLK=PLLCLK

3.3.20 SCU System and Peripheral Clock Prescale register (SCU_CLKPRE)

Address Offset: 0x804

✸ ***Note: Must reset the corresponding peripheral with peripheral reset registers after changing the pre-scaler value.***

Bit	Field	Access	Initial	Description
31:16	WRPKEY	W	0	Write Protect key Read as 0. When writing to AHBPRE/APB0PRE/APB1PRE/CLKOUTPRE bits you must write 0x5AFA to WRPKEY, otherwise behavior of writing to these bits is ignored.
15:13	–	–	0	Reserved
12:9	CLKOUTPRE	RW	0	CLKOUT pre-scaler 0: CLKOUT = clock source/1 1: CLKOUT = clock source/2 2: CLKOUT = clock source/3 3: CLKOUT = clock source/4 4: CLKOUT = clock source/6 5: CLKOUT = clock source/8 6: CLKOUT = clock source/16 7: CLKOUT = clock source/32 8: CLKOUT = clock source/64 9: CLKOUT = clock source/128 Other: Reserved It is write-protected by WRPKEY.
8:6	APB1PRE	RW	1	APB1CLK pre-scaler 0: APB1CLK = HCLK/1 1: APB1CLK = HCLK/2 2: APB1CLK = HCLK/4 3: APB1CLK = HCLK/8 4: APB1CLK = HCLK/16 5: APB1CLK = HCLK/32 6: APB1CLK = HCLK/64 7: APB1CLK = HCLK/128 It is write-protected by WRPKEY.
5:3	APB0PRE	RW	2	APB0CLK pre-scaler 0: APB0CLK = HCLK/1 1: APB0CLK = HCLK/2 2: APB0CLK = HCLK/4 3: APB0CLK = HCLK/8

Bit	Field	Access	Initial	Description
				4: APB0CLK = HCLK/16 5: APB0CLK = HCLK/32 6: APB0CLK = HCLK/64 7: APB0CLK = HCLK/128 It is write-protected by WRPKEY.
2:0	AHBPRE	RW	0	HCLK pre-scaler 0: HCLK = SYSCLK/1 1: HCLK = SYSCLK/2 2: HCLK = SYSCLK/4 3: HCLK = SYSCLK/8 4: HCLK = SYSCLK/16 5: HCLK = SYSCLK/32 6: HCLK = SYSCLK/64 7: HCLK = SYSCLK/128 It is write-protected by WRPKEY.

3.3.21 SCU IHRC Oscillator Control register (SCU_IHRCCTRL)

Address Offset: 0x808

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

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1	IHRCRDY	R	1	IHRC ready flag 0: No ready 1: Ready
0	IHRCEN	RW	1	Internal high-speed clock enable 0: Disable 1: Enable Note: This bit can NOT be cleared if the IHRC is selected as system clock or is selected to become the system clock.

3.3.22 SCU PLL and System Clock Status register (SCU_PLLSTS)

Address Offset: 0x80C

Bit	Field	Access	Initial	Description
31	–	–	0	Reserved
30:24	NSSTS	R	0x40	7-bit programmable loop divider status
23:10	–	–	0	Reserved
9:8	FSSTS	R	0x3	PLL frequency range output divider status 0: PLL CKOUT= PLLCLK / 32 1: PLL CKOUT= PLLCLK / 16 2: PLL CKOUT = PLLCLK / 8 3: PLL CKOUT = PLLCLK / 4
7:5	SYSCLKSTS	R	0	System clock status 0: IHRC 2: EHS 4: PLL Other: Reserved
4	PLLCLKSTS	R	0	PLL clock status 0: IHRC 1: EHS
3:1	–	–	0	Reserved

Bit	Field	Access	Initial	Description
0	PLLENSTS	R	0	PLL enable status 0: Disable 1: Enable

3.3.23 SCU AHB Peripheral Reset Control register (SCU_AHBRST)

Address Offset: 0x810

Bit	Field	Access	Initial	Description
31:9	–	–	0	Reserved
8	ETHRST	RWS	1	Ethernet MAC reset control 0: No effect 1: Reset Ethernet MAC By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
7	–	–	0	Reserved
6	SDIORST	RWS	1	SDIO reset control 0: No effect/Reset is done 1: Reset SDIO By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
5:2	–	–	0	Reserved
1	DMA1RST	RWS	1	DMA1 reset control 0: No effect/Reset is done 1: Reset DMA1 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
0	DMA0RST	RWS	1	DMA0 reset control 0: No effect/Reset is done 1: Reset DMA0 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.

3.3.24 SCU APB0 Peripheral Reset Control register (SCU_APB0RST)

Address Offset: 0x814

Bit	Field	Access	Initial	Description
31:10	–	–	0	Reserved
9	CT16B7RST	RWS	1	CT16B7 reset control 0: No effect/Reset is done 1: Reset CT16B7 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
8	CT16B6RST	RWS	1	CT16B6 reset control 0: No effect/Reset is done 1: Reset CT16B6 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
7	UART4RST	RWS	1	UART4 reset control 0: No effect/Reset is done 1: Reset UART4 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
6	UART3RST	RWS	1	UART3 reset control 0: No effect/Reset is done 1: Reset UART3 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
5	UART2RST	RWS	1	UART2 reset control

Bit	Field	Access	Initial	Description
				0: No effect/Reset is done 1: Reset UART2 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
4	UART1RST	RWS	1	UART1 reset control 0: No effect/Reset is done 1: Reset UART1 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
3	CAN1RST	RWS	1	CAN1 reset control 0: No effect/Reset is done 1: Reset CAN1 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
2	CAN0RST	RWS	1	CAN0 reset control 0: No effect/Reset is done 1: Reset CAN0 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
1	WWDTRST	RWS	1	WWDT reset control 0: No effect/Reset is done 1: Reset WWDT By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
0	WDTRST	RWS	1	WDT reset control 0: No effect/Reset is done 1: Reset WDT By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.

3.3.25 SCU APB1 Peripheral Reset Control register (SCU_APB1RST)

Address Offset: 0x818

Bit	Field	Access	Initial	Description
31:22	–	–	0	Reserved
21	LCMRST	RWS	1	LCM reset control 0: No effect/Reset is done 1: Reset LCM By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
20	GPIO3RST	RWS	0	GPIO3 reset control 0: No effect/Reset is done 1: Reset GPIO3
19	GPIO2RST	RWS	0	GPIO2 reset control 0: No effect/Reset is done 1: Reset GPIO2
18	GPIO1RST	RWS	0	GPIO1 reset control 0: No effect/Reset is done 1: Reset GPIO1
17	GPIO0RST	RWS	0	GPIO0 reset control 0: No effect/Reset is done 1: Reset GPIO0
16	CRCRST	RWS	1	CRC reset control 0: No effect/Reset is done 1: Reset CRC By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
15	CT16B8RST	RWS	1	CT16B8 reset control 0: No effect/Reset is done 1: Reset CT16B8 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.

Bit	Field	Access	Initial	Description
14	CT16B5RST	RWS	1	CT16B5 reset control 0: No effect/Reset is done 1: Reset CT16B5 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
13	CT16B4RST	RWS	1	CT16B4 reset control 0: No effect/Reset is done 1: Reset CT16B4 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
12	CT16B3RST	RWS	1	CT16B3 reset control 0: No effect/Reset is done 1: Reset CT16B3 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
11	CT16B2RST	RWS	1	CT16B2 reset control 0: No effect/Reset is done 1: Reset CT16B2 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
10	CT16B1RST	RWS	1	CT16B1 reset control 0: No effect/Reset is done 1: Reset CT16B1 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
9	CT16B0RST	RWS	1	CT16B0 reset control 0: No effect/Reset is done 1: Reset CT16B0 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
8	I2C2RST	RWS	1	I2C2 reset control 0: No effect/Reset is done 1: Reset I2C2 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
7	I2C1RST	RWS	1	I2C1 reset control 0: No effect/Reset is done 1: Reset I2C1 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
6	I2C0RST	RWS	1	I2C0 reset control 0: No effect/Reset is done 1: Reset I2C0
5	UART5RST	RWS	1	UART5 reset control 0: No effect/Reset is done 1: Reset UART5 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
4	UART0RST	RWS	1	UART0 reset control 0: No effect/Reset is done 1: Reset UART0 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
3	SSP2RST	RWS	1	SSP2 reset control 0: No effect/Reset is done 1: Reset SSP2 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
2	SSP1RST	RWS	1	SSP1 reset control 0: No effect/Reset is done 1: Reset SSP1 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
1	SSP0RST	RWS	1	SSP0 reset control 0: No effect/Reset is done

Bit	Field	Access	Initial	Description
				1: Reset SSP0 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.
0	ADC0RST	RWS	1	ADC0 reset control 0: No effect/Reset is done 1: Reset ADC0 By default, the peripheral IP clock is gated, then the IP will be held in reset until the IP clock is enabled.

3.3.26 SCU Peripheral Clock Pre-scaler register (SCU_PRECTRL)

Address Offset: 0x840

Bit	Field	Access	Initial	Description
31:3	–	–	0	Reserved
2:0	LCMPRE	RW	0	LCM clock pre-scaler 0: LCMCLK/1 1: LCMCLK/2 2: LCMCLK/4 3: LCMCLK/8 4: LCMCLK/16 Other: Reserved

3.3.27 SCU GPIOn.0~n.15 Driving/Sinking Strength Control register (SCU_PnSTR) (n=0,1,2,3)

Address Offset: 0x900+n*4

Bit	Field	Access	Initial	Description
31:30	STR15	RW	0	Pn.15 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
29:28	STR14	RW	0	Pn.14 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
27:26	STR13	RW	0	Pn.13 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
25:24	STR12	RW	0	Pn.12 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
23:22	STR11	RW	0	Pn.11 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
21:20	STR10	RW	0	Pn.10 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
19:18	STR9	RW	0	Pn.9 driving/sinking strength

Bit	Field	Access	Initial	Description
				0: 17mA 1: 19mA 2: 21mA 3: 23mA
17:16	STR8	RW	0	Pn.8 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
15:14	STR7	RW	0	Pn.7 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
13:12	STR6	RW	0	Pn.6 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
11:10	STR5	RW	0	Pn.5 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
9:8	STR4	RW	0	Pn.4 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
7:6	STR3	RW	0	Pn.3 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
5:4	STR2	RW	0	Pn.2 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
3:2	STR1	RW	0	Pn.1 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
1:0	STR0	RW	0	Pn.0 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA

3.3.28 SCU GPIO0.20 Driving/Sinking Strength Control register (SCU_P0STR1)

Address Offset: 0x910

Bit	Field	Access	Initial	Description
31:10	–	–	0	Reserved
9:8	STR20	RW	0	P0.20 driving/sinking strength 0: 17mA 1: 19mA 2: 21mA 3: 23mA
7:0	–	–	0	Reserved

3.3.29 SCU AHB Peripheral PERRST Reset Mask register (SCU_AHBRSTMSK)

Address Offset: 0xA10

Bit	Field	Access	Initial	Description
31:9	–	–	0	Reserved
8	ETHRSTMSK	RW	0	Ethernet MAC reset mask 0: No mask 1: Mask PERRST reset for Ethernet MAC
7	–	–	0	Reserved
6	SDIORSTMSK	RW	0	SDIO reset mask 0: No mask 1: Mask PERRST reset for SDIO
5:2	–	–	0	Reserved
1	DMA1RSTMSK	RW	0	DMA1 reset mask 0: No mask 1: Mask PERRST reset for DMA1
0	DMA0RSTMSK	RW	0	DMA0 reset mask 0: No mask 1: Mask PERRST reset for DMA0

3.3.30 SCU APB0 Peripheral PERRST Reset Mask register (SCU_APB0RSTMSK)

Address Offset: 0xA14

Bit	Field	Access	Initial	Description
31:10	–	–	0	Reserved
9	CT16B7RSTMSK	RW	0	CT16B7 reset mask 0: No mask 1: Mask PERRST reset for CT16B7
8	CT16B6RSTMSK	RW	0	CT16B6 reset mask 0: No mask 1: Mask PERRST reset for CT16B6
7	UART4RSTMSK	RW	0	UART4 reset mask 0: No mask 1: Mask PERRST reset for UART4
6	UART3RSTMSK	RW	0	UART3 reset mask 0: No mask 1: Mask PERRST reset for UART3
5	UART2RSTMSK	RW	0	UART2 reset mask 0: No mask 1: Mask PERRST reset for UART2
4	UART1RSTMSK	RW	0	UART1 reset mask 0: No mask 1: Mask PERRST reset for UART1
3	CAN1RSTMSK	RW	0	CAN1 reset mask 0: No mask

Bit	Field	Access	Initial	Description
				1: Mask PERRST reset for CAN1
2	CAN0RSTMSK	RW	0	CAN0 reset mask 0: No mask 1: Mask PERRST reset for CAN0
1	WWDTRSTMSK	RW	0	WWDT reset mask 0: No mask 1: Mask PERRST reset for WWDT
0	WDTRSTMSK	RW	0	WDT reset mask 0: No mask 1: Mask PERRST reset for WDT

3.3.31 SCU APB1 Peripheral PERRST Reset Mask register (SCU_APB1RSTMSK)

Address Offset: 0xA18

Bit	Field	Access	Initial	Description
31:22	–	–	0	Reserved
21	LCMRSTMSK	RW	0	LCM reset mask 0: No mask 1: Mask PERRST reset for LCM
20	GPIO3RSTMSK	RW	0	GPIO3 reset mask 0: No mask 1: Mask PERRST reset for GPIO3
19	GPIO2RSTMSK	RW	0	GPIO2 reset mask 0: No mask 1: Mask PERRST reset for GPIO2
18	GPIO1RSTMSK	RW	0	GPIO1 reset mask 0: No mask 1: Mask PERRST reset for GPIO1
17	GPIO0RSTMSK	RW	0	GPIO0 reset mask 0: No mask 1: Mask PERRST reset for GPIO0
16	CRCRSTMSK	RW	0	CRC reset mask 0: No mask 1: Mask PERRST reset for CRC
15	CT16B8RSTMSK	RW	0	CT16B8 reset mask 0: No mask 1: Mask PERRST reset for CT16B8
14	CT16B5RSTMSK	RW	0	CT16B5 reset mask 0: No mask 1: Mask PERRST reset for CT16B5
13	CT16B4RSTMSK	RW	0	CT16B4 reset mask 0: No mask 1: Mask PERRST reset for CT16B4
12	CT16B3RSTMSK	RW	0	CT16B3 reset mask 0: No mask 1: Mask PERRST reset for CT16B3
11	CT16B2RSTMSK	RW	0	CT16B2 reset mask 0: No mask 1: Mask PERRST reset for CT16B2
10	CT16B1RSTMSK	RW	0	CT16B1 reset mask 0: No mask 1: Mask PERRST reset for CT16B1
9	CT16B0RSTMSK	RW	0	CT16B0 reset mask 0: No mask 1: Mask PERRST reset for CT16B0
8	I2C2RSTMSK	RW	0	I2C2 reset mask 0: No mask 1: Mask PERRST reset for I2C2
7	I2C1RSTMSK	RW	0	I2C1 reset mask 0: No mask

Bit	Field	Access	Initial	Description
				1: Mask PERRST reset for I2C1
6	I2C0RSTMSK	RW	0	I2C0 reset mask 0: No mask 1: Mask PERRST reset for I2C0
5	UART5RSTMSK	RW	0	UART5 reset mask 0: No mask 1: Mask PERRST reset for UART5
4	UART0RSTMSK	RW	0	UART0 reset mask 0: No mask 1: Mask PERRST reset for UART0
3	SSP2RSTMSK	RW	0	SSP2 reset mask 0: No mask 1: Mask PERRST reset for SSP2
2	SSP1RSTMSK	RW	0	SSP1 reset mask 0: No mask 1: Mask PERRST reset for SSP1
1	SSP0RSTMSK	RW	0	SSP0 reset mask 0: No mask 1: Mask PERRST reset for SSP0
0	ADC0RSTMSK	RW	0	ADC0 reset mask 0: No mask 1: Mask PERRST reset for ADC0

3.3.32 SCU GMAC Special Control register (SCU_GMAC_SC)

Address Offset: 0xBEC

Bit	Field	Access	Initial	Description
31	–	–	0	Reserved
30	CAN0SRAMRET	RW	0	CAN0 SRAM retention mode control 0 : Normal mode 1 : Retention mode
29	CAN1SRAMRET	RW	0	CAN1 SRAM retention mode control 0 : Normal mode 1 : Retention mode
28	ETHSRAMRET	RW	0	Ethernet MAC SRAM retention mode control 0 : Normal mode 1 : Retention mode
27:18	–	–	0	Reserved
17	ETHMAC_HP_TX_POLL	RW	0	High priority TX poll demand, on cycle pulse activation with the system clock domain 0: Clear 1: SET
16	ETHMAC_NP_TX_POLL	RW	0	Normal priority TX poll demand, on cycle pulse activation with the system clock domain 0: Clear 1: SET
15:0	–	–	0	Reserved

3.3.33 SCU SRAM1~3 & BKPSRAM controller sideband signal register (SCU_SRAM1~3CTRL & SCU_BKPSRAMCTRL)

Address Offset: 0xBF0, 0xBF4, 0xBF8, 0xBFC

SRAM1CTRL (0xBF0): SRAM1 32KB 0x2000_0000 (Aliasing 0x1000_0000)

SRAM2CTRL (0xBF4): SRAM2 96KB 0x2000_8000 (Aliasing 0x2000_8000)

SRAM3CTRL (0xBF8): SRAM3 32KB 0x2002_0000 (Aliasing 0x2002_0000)

BKPSRAMCTRL (0xBFC): BKPSRAM 4KB 0x4005_1000

Bit	Field	Access	Initial	Description
31:28	–	–	0	Reserved
27:16	TIMEOUT	RW	0x4	Time-out counter to trigger SRAM enter the power-saving mode
15:12	SLPRESDLY	RW	0	Resuming delay time for SRAM sleep mode returns to normal mode (Time for the power recovery)
11:8	RETRESDLY	RW	0x2	Resuming delay time for SRAM retention mode returns to normal mode (Time for the power recovery)
7:4	NAPRESDLY	RW	0	Resuming delay time for SRAM nap mode returns to normal mode (Time for the power recovery)
3	–	–	0	Reserved
2	AUTOSLPEN	RW	0	SRAM automatic sleep function. When the SRAM is idle, it automatically enters sleep mode. 0: Disable 1: Enable
1	AUTORETEN	RW	0	SRAM automatic retention function. When the SRAM is idle, it automatically enters retention mode. 0: Disable 1: Enable
0	AUTONAPEN	RW	0	SRAM automatic nap function. When the SRAM is idle, it automatically enters nap mode. 0: Disable 1: Enable

3.4 ALWAYS-ON DOMAIN REGISTERS

Base Address: 0x4001 B000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	ALWAYSON_BKPREG0	Backup register 0
0x0004	ALWAYSON_BKPREG1	Backup register 1
0x0008	ALWAYSON_BKPREG2	Backup register 2
0x000C	ALWAYSON_BKPREG3	Backup register 3
0x0010	ALWAYSON_BKPREG4	Backup register 4
0x0014	ALWAYSON_BKPREG5	Backup register 5
0x0018	ALWAYSON_BKPREG6	Backup register 6
0x001C	ALWAYSON_BKPREG7	Backup register 7
0x0020	ALWAYSON_BKPREG8	Backup register 8
0x0024	ALWAYSON_BKPREG9	Backup register 9
0x0028	ALWAYSON_BKPREG10	Backup register 10
0x002C	ALWAYSON_BKPREG11	Backup register 11
0x0030	ALWAYSON_BKPREG12	Backup register 12
0x0034	ALWAYSON_BKPREG13	Backup register 13
0x0038	ALWAYSON_BKPREG14	Backup register 14
0x003C	ALWAYSON_BKPREG15	Backup register 15
0x0040	ALWAYSON_BKPREG16	Backup register 16
0x0044	ALWAYSON_BKPREG17	Backup register 17
0x0048	ALWAYSON_BKPREG18	Backup register 18
0x004C	ALWAYSON_BKPREG19	Backup register 19
0x0050 – 0x00FC	-	Reserved
0x0100	ALWAYSON_RSTST	System Reset Status register
0x0104	ALWAYSON_LVDCTRL	LVD Control register
0x0108	ALWAYSON_OSCCTL	Oscillator Control register
0x010C	ALWAYSON_OSCRDY	Oscillator Ready Flag register
0x0110	ALWAYSON_POR_MISC	POR Miscellaneous Control register
0x0114 – 0x01FC	-	Reserved
0x0200	ALWAYSON_P0_AFIO0	GPIO0.0~GPIO0.2 Alternate Function Control register
0x0204	ALWAYSON_P0_AFIO1	GPIO0.10~GPIO0.11 Alternate Function Control register
0x0208	ALWAYSON_P0_AFIO2	GPIO0.12~GPIO0.15 Alternate Function Control register
0x020C	ALWAYSON_P0_AFIO3	GPIO0.20 Alternate Function Control register
0x0210	ALWAYSON_P1_AFIO0	GPIO1.0~GPIO1.5 Alternate Function Control register
0x0214	ALWAYSON_P1_AFIO1	GPIO1.6~GPIO1.11 Alternate Function Control register
0x0218	ALWAYSON_P1_AFIO2	GPIO1.12~GPIO1.15 Alternate Function Control register
0x021C	-	Reserved
0x0220	ALWAYSON_P2_AFIO0	GPIO2.0~GPIO2.5 Alternate Function Control register
0x0224	ALWAYSON_P2_AFIO1	GPIO2.6~GPIO2.11 Alternate Function Control register
0x0228	ALWAYSON_P2_AFIO2	GPIO2.12~GPIO2.15 Alternate Function Control register
0x022C	-	Reserved
0x0230	ALWAYSON_P3_AFIO0	GPIO3.0~GPIO3.5 Alternate Function Control register
0x0234	ALWAYSON_P3_AFIO1	GPIO3.6~GPIO3.11 Alternate Function Control register
0x0238	ALWAYSON_P3_AFIO2	GPIO3.12~GPIO3.13 Alternate Function Control register
0x023C	-	Reserved
0x0240	ALWAYSON_GPNOV	GPIO0~3 Non-overlap Control register
0x0244 – 0x03D8	-	Reserved
0x03DC	ALWAYSON_OSCMISC	Oscillator Miscellaneous Control register

3.4.1 Backup register n (ALWAYSON_BKPREGn) (n=0~19)

Address Offset: 0x00+n*4

Bit	Field	Access	Initial	Description
31:0	BKPDATA	RW	0	Backup data

3.4.2 System Reset Status register (ALWAYSON_RSTST)

Address Offset: 0x100

Bit	Field	Access	Initial	Description
31:3	–	–	0	Reserved
2	LVDF	RW1C	0	LVD reset flag or interrupt flag 0: No interrupt flag occurred 1: LVD interrupt flag occurred This flag is inactive if Always-on domain registers clock is disabled.
1:0	–	–	0	Reserved

3.4.3 LVD control register (ALWAYSON_LVDCTRL)

Address Offset: 0x104

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

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15	LVDEN	RW	1	LVD function enable 0: Disable 1: Enable * Always disable in Deep sleep/Deep power-down mode
14	LVDRSTEN	RW	0	LVD Reset enable 0: Flag mode. Interrupt control by NVIC interrupt enable. 1: Reset mode. Reset Events.
13:2	–	–	0	Reserved
1:0	LVDLVL	RW	0x3	LVD reset/interrupt level 0: 2.40V 1: 2.60V 2: 2.80V 3: 3.00V

✷ ***Note: 1. After setting the new level, user needs to wait 50us to enable the LVD interrupt/reset function. Avoid accidentally triggering interrupt/reset before LVD functionality is stable.***
2. After setting LVD disable, user needs to wait 20us to take effect.

3.4.4 Oscillator Control register (ALWAYSON_OSCCTL)

Address Offset: 0x0108

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

Bit	Field	Access	Initial	Description
31:7	–	–	0	Reserved
6	EHSINROFF	RW	0	Internal resistor of EHS XTAL control 0: Enable internal resistor. EHS with internal resistor 1: Disable internal resistor. EHS without internal resistor
5	–	–	0	Reserved
4	EHSEN	RW	0	External high-speed clock enable 0: Disable EHS XTAL. HW will disable clock filter automatically. 1: Enable EHS XTAL. HW will enable clock filter automatically. Note: This bit can NOT be cleared if the EHS XTAL is selected as system clock or is selected to become the system clock.
3:0	–	–	0	Reserved

3.4.5 Oscillator Ready Flag register (ALWAYSON_OSCRDY)

Address Offset: 0x10C

Bit	Field	Access	Initial	Description
31:5	–	–	0	Reserved
4	EHSRDY	R	0	EHS XTAL ready flag 0: EHS Xtal is Not Ready 1: EHS Xtal is Ready
3:0	–	–	0	Reserved

3.4.6 POR Miscellaneous Control register (ALWAYSON_POR_MISC)

Address Offset: 0x110

Bit	Field	Access	Initial	Description
31:5	–	–	0	Reserved
16	PORRSTF	RW1C	1	POR reset flag 0: No POR occurred 1: POR occurred
15:5	–	–	0	Reserved
4	ELSRDY	R	0	ELS XTAL ready flag 0: ELS Xtal is Not Ready 1: ELS Xtal is Ready
3	ELSFLOFF	RW	1	External low-speed oscillator clock filter control 0: Enable clock filter. ELS with clock filter 1: Disable clock filter. ELS without clock filter ELSFLOFF can only be switched when ELSEN = 0, and cannot be switched at the same time as ELSEN.
2	ELSEN	RW	0	External low-speed oscillator enable 0: Disable External 32.768 KHz oscillator 1: Enable External 32.768 KHz oscillator
1	RTCCLKSEL	RW	0	RTC counter clock source 0: RTCCLK=ELS 1: RTCCLK=ILRC 32kHz
0	RTCCLKEN	RW	0	Enable clock for RTC 0: Disable

Bit	Field	Access	Initial	Description
				1: Enable

3.4.7 GPIO0.0~GPIO0.2 Alternate Function Control register (ALWAYSON_P0_AFIO0)

Address Offset: 0x200

Bit	Field	Access	Initial	Description
31:15	–	–	0	Reserved
14:10	IO2	RW	0	Alternate function of P0.2 (Reference PIN Mux table) 0: P0.2 1: URXD2 2: MI0 3: SDA1 4: CAN_RX1 8: CT16B0_PWM0 9: CT16B5_PWM3 10: CT16B0_PWM3N 11: CT16B2_PWM3 12: CT16B8_PWM2 13: LCM_AD2 14: ETH_RMII_TX_EN 15: ETH_MII_TX_EN 16: SO0 18: IRDA_RXDL2 20: SDIO_D2 Other: Reserved
9:5	IO1	RW	0	Alternate function of P0.1 (Reference PIN Mux table) 0: P0.1 1: UTXD2 2: URXD2 3: SEL0 4: SCK1 5: SCL1 6: BCLK1 8: CT16B0_PWM1 9: CT16B0_PWM2N 10: CT16B0_PWM3 11: CT16B2_PWM2 12: CT16B8_PWM1 13: LCM_AD1 14: CAN_TX0 15: ETH_MII_RX_ER 16: IRDA_TXD2 18: IRDA_RXDL2 20: SDIO_D1 Other: Reserved
4:0	IO0	RW	0	Alternate function of P0.0 (Reference PIN Mux table) 0: P0.0 1: SCK0 2: MO2 3: DOUT2 6: CT16B0_PWM2 7: CT16B0_PWM3N 8: CT16B0_PWM1N 9: CT16B2_PWM3 10: CT16B0_PWM0 11: CT16B2_PWM0 12: CT16B8_PWM0 13: LCM_AD0 14: CAN_RX0 16: SI2 20: SDIO_D0

Bit	Field	Access	Initial	Description
				Other: Reserved

3.4.8 GPIO0.10~GPIO0.11 Alternate Function Control register (ALWAYSON_P0_AFIO1)

Address Offset: 0x204

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO11	RW	0	Alternate function of P0.11 (Reference PIN Mux table) 0: P0.11 1: URXD0 2: UCTS3 3: SDA0 6: CT16B2_CAP0 7: CT16B1_PWM3 8: CT16B2_PWM1 9: CT16B3_PWM1N 10: CT16B4_PWM1 11: CT16B4_PWM0N 12: CT16B8_PWM6 18: IRDA_RXDL0 Other: Reserved
24:20	IO10	RW	0	Alternate function of P0.10 (Reference PIN Mux table) 0: P0.10 1: UTXD0 2: UTXD4 3: SCL0 4: CAN_TX1 6: CT16B0_BRK 7: CT16B1_PWM2 8: CT16B2_PWM0 9: CT16B3_PWM0N 10: CT16B4_PWM0 11: CT16B4_PWM1N 12: CT16B8_PWM5 14: LCM_CS 15: CLKOUT3 16: IRDA_TXD0 17: IRDA_TXD4 Other: Reserved
19:0	–	–	0	Reserved

3.4.9 GPIO0.12~GPIO0.15 Alternate Function Control register (ALWAYSON_P0_AFIO2)

Address Offset: 0x208

Bit	Field	Access	Initial	Description
31:20	–	–	0	Reserved
19:15	IO15	RW	0	Alternate function of P0.15 (Reference PIN Mux table) 0: P0.15 1: UTXD5 3: SCL0 7: CT16B0_PWM0 10: CT16B5_PWM0 12: CT16B8_PWM5 14: SDIO_D6 16: IRDA_TXD5 Other: Reserved
14:10	IO14	RW	0	Alternate function of P0.14 (Reference PIN Mux table)

Bit	Field	Access	Initial	Description
				0: P0.14 1: URXD5 2: SCK1 3: SDA0 5: BCLK1 7: CT16B0_PWM1 10: CT16B5_PWM1 12: CT16B8_PWM6 14: SDIO_D7 18: IRDA_RXDL5 Other: Reserved
9:5	IO13	RW	0	Alternate function of P0.13 (Reference PIN Mux table) 0: P0.13 1: SEL1 6: CT16B4_CAP0 7: CT16B0_PWM2 10: CT16B5_PWM2 12: CT16B8_PWM7 14: SDIO_D0 Other: Reserved
4:0	IO12	RW	0	Alternate function of P0.12 (Reference PIN Mux table) 0: P0.12 1: SCK1 3: SDA2 7: CT16B0_PWM3 8: CT16B0_PWM0N 10: CT16B5_PWM3 12: CT16B8_PWM8 14: SDIO_D1 15: CLKOUT2 Other: Reserved

3.4.10 GPIO0.20 Alternate Function Control register (ALWAYSON_P0_AFIO3)

Address Offset: 0x20C

Bit	Field	Access	Initial	Description
31:15	–	–	0	Reserved
14:10	IO20	RW	0	Alternate function of P0.20 (Reference PIN Mux table) 0: P0.20 5: SDA1 12: CT16B8_PWM1 14: LCM_CS Other: Reserved
9:0	–	–	0	Reserved

3.4.11 GPIO1.0~GPIO1.5 Alternate Function Control register (ALWAYSON_P1_AFIO0)

Address Offset: 0x210

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO5	RW	0	Alternate function of P1.5 (Reference PIN Mux table) 0: P1.5 1: UTXD4 2: SEL1 3: SDA0 4: SDA1 5: WS1

Bit	Field	Access	Initial	Description
				7: CAN_TX1 8: CT16B2_PWM1 9: CT16B2_PWM3 10: CT16B3_PWM1 11: CT16B4_PWM0 12: CT16B8_PWM9 14: SDIO_D5 16: IRDA_TXD4 Other: Reserved
24:20	IO4	RW	0	Alternate function of P1.4 (Reference PIN Mux table) 0: P1.4 1: UTXD2 2: URXD4 3: SCL0 7: CAN_RX1 8: CT16B2_PWM0 9: CT16B2_PWM2 10: CT16B3_PWM0 11: CT16B4_PWM1 12: CT16B8_PWM8 14: SDIO_D4 15: ETH_MII_TXD3 16: IRDA_TXD2 18: IRDA_RXDL4 Other: Reserved
19:15	IO3	RW	0	Alternate function of P1.3 (Reference PIN Mux table) 0: P1.3 1: URXD2 2: DIN0 8: CAN_STBY1 10: CT16B3_PWM0N 11: CT16B4_PWM1N 12: CT16B8_PWM7 18: IRDA_RXDL2 Other: Reserved
14:10	IO2	RW	0	Alternate function of P1.2 (Reference PIN Mux table) 0: P1.2 1: URXD4 2: MO0 3: MO2 4: SDA0 5: DOUT0 6: DOUT1 7: DOUT2 8: CAN_STBY0 9: CAN_RX1 10: CT16B1_PWM1 11: CT16B4_PWM0N 12: CT16B5_PWM1 16: SI0 17: SI2 18: IRDA_RXDL4 Other: Reserved
9:5	IO1	RW	0	Alternate function of P1.1 (Reference PIN Mux table) 0: P1.1 1: URXD4 2: MI0 3: MI2 4: SCL0 5: SDA2 7: DIN1 8: CAN_TX0 9: CT16B1_PWM0 10: CT16B5_PWM0 12: CT16B8_PWM4 14: LCM_CS

Bit	Field	Access	Initial	Description
				16: SO0 17: SO2 18: IRDA_RXDL4 Other: Reserved
4:0	IO0	RW	0	Alternate function of P1.0 (Reference PIN Mux table) 0: P1.0 1: UTXD4 2: URXD0 3: SCK0 4: SCK2 5: SDA1 6: BCLK0 7: BCLK2 8: CAN_RX0 9: CT16B2_PWM1 10: CT16B5_PWM2 12: CT16B8_PWM3 15: SWO 16: IRDA_TXD4 18: IRDA_RXDL0 Other: Reserved

3.4.12 GPIO1.6~GPIO1.11 Alternate Function Control register (ALWAYSON_P1_AFIO1)

Address Offset: 0x214

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO11	RW	0	Alternate function of P1.11 (Reference PIN Mux table) 0: P1.11 1: URXD4 2: URTS2 3: MO1 4: SDA1 11: CT16B8_PWM2 12: LCM_A0 14: SDIO_CMD 16: SI1 18: IRDA_RXDL4 Other: Reserved
24:20	IO10	RW	0	Alternate function of P1.10 (Reference PIN Mux table) 0: P1.10 1: UTXD4 2: MI1 3: MO2 4: SCL1 5: SDA1 6: DIN1 7: DOUT2 9: CT16B2_PWM3 12: CT16B8_PWM1 14: SDIO_CK 16: SO1 17: SI2 18: IRDA_TXD4 Other: Reserved
19:15	IO9	RW	0	Alternate function of P1.9 (Reference PIN Mux table) 0: P1.9 1: UTXD1 2: URXD3 3: SEL1 4: SDA1

Bit	Field	Access	Initial	Description
				5: SCL1 6: WS1 7: CAN_STBY0 9: CT16B5_PWM0 12: CT16B8_PWM9 16: IRDA_TXD1 18: IRDA_RXDL3 Other: Reserved
14:10	IO8	RW	0	Alternate function of P1.8 (Reference PIN Mux table) 0: P1.8 1: URXD1 2: URTS3 3: UTXD0 4: SEL0 5: SCK1 6: SEL2 7: SCL1 8: WS0 9: WS2 10: CT16B2_PWM0 12: CT16B8_PWM10 13: LCM_AD3 16: IRDA_TXD0 18: IRDA_RXDL1 Other: Reserved
9:5	IO7	RW	0	Alternate function of P1.7 (Reference PIN Mux table) 0: P1.7 1: UTXD2 2: UTXD3 3: SEL0 4: SCK2 6: BCLK2 8: CT16B2_PWM1 9: CT16B5_PWM1 12: CT16B8_PWM11 14: SDIO_D2 16: IRDA_TXD2 17: IRDA_TXD3 Other: Reserved
4:0	IO6	RW	0	Alternate function of P1.6 (Reference PIN Mux table) 0: P1.6 1: URXD2 2: URXD3 3: MI2 4: BCLK1 8: CT16B2_PWM2 9: CT16B5_PWM2 12: CT16B8_PWM0 14: SDIO_D3 16: SO2 18: IRDA_RXDL2 20: IRDA_RXDL3 Other: Reserved

3.4.13 GPIO1.12~GPIO1.15 Alternate Function Control register (ALWAYSON_P1_AFIO2)

Address Offset: 0x218

Bit	Field	Access	Initial	Description
31:20	–	–	0	Reserved
19:15	IO15	RW	0	Alternate function of P1.15 (Reference PIN Mux table) 0: P1.15

Bit	Field	Access	Initial	Description
				2: MO1 3: DOUT1 4: SCL2 6: CT16B5_CAP0 7: CT16B3_PWM1 8: CT16B0_PWM2N 9: CT16B1_PWM3 10: CT16B3_PWM0N 11: CT16B4_PWM1 12: CT16B8_PWM4 14: LCM_AD7 16: SI1 Other: Reserved
14:10	IO14	RW	0	Alternate function of P1.14 (Reference PIN Mux table) 0: P1.14 1: URTS2 2: MI1 3: SDA1 4: SDA2 6: CAN_STBY0 8: CT16B0_PWM1N 9: CT16B1_PWM2 10: CT16B3_PWM0 11: CT16B4_PWM0 12: CT16B8_PWM3 14: LCM_AD6 16: SO1 Other: Reserved
9:5	IO13	RW	0	Alternate function of P1.13 (Reference PIN Mux table) 0: P1.13 1: UCTS2 2: SCK1 3: SCL1 4: SCL2 5: BCLK1 6: CAN_TX0 7: CAN_TX1 8: CT16B0_PWM0N 9: CT16B1_PWM1 10: CT16B3_PWM1N 11: CT16B4_PWM1N 13: LCM_AD5 14: ETH_RMII_TXD1 15: ETH_MII_TXD1 20: SDIO_CMD Other: Reserved
4:0	IO12	RW	0	Alternate function of P1.12 (Reference PIN Mux table) 0: P1.12 1: SEL1 2: SCK2 3: WS1 4: BCLK2 5: CAN_RX0 6: CAN_RX1 7: CT16B0_BRK 8: CT16B0_PWM0 9: CT16B1_PWM0 10: CT16B3_PWM1 11: CT16B4_PWM1 12: CT16B5_PWM0 13: LCM_AD4 14: ETH_RMII_TXD0 15: ETH_MII_TXD0 20: SDIO_CK Other: Reserved

3.4.14 GPIO2.0~GPIO2.5 Alternate Function Control register (ALWAYSON_P2_AFIO0)

Address Offset: 0x220

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO5	RW	0	Alternate function of P2.5 (Reference PIN Mux table) 0: P2.5 1: URXD5 2: SCK0 3: BCLK0 6: CT16B0_PWM1 7: CT16B0_PWM0N 8: CT16B2_PWM0 9: CT16B4_PWM1 12: ADC0_AIN5 13: LCM_AD11 18: IRDA_RXDL5 20: SDIO_D3 Other: Reserved
24:20	IO4	RW	0	Alternate function of P2.4 (Reference PIN Mux table) 0: P2.4 1: UTXD5 2: SEL0 3: SEL2 4: WS0 5: WS2 6: CT16B0_PWM1N 7: CT16B1_PWM1 8: CT16B3_PWM1N 9: CT16B4_PWM1N 12: ADC0_AIN4 14: LCM_AD10 16: IRDA_TXD5 Other: Reserved
19:15	IO3	RW	0	Alternate function of P2.3 (Reference PIN Mux table) 0: P2.3 1: URXD1 7: CT16B2_PWM3 8: CT16B3_PWM1 11: CT16B5_PWM3 12: ADC0_AIN3 15: ETH_MII_COL 18: IRDA_RXDL1 Other: Reserved
14:10	IO2	RW	0	Alternate function of P2.2 (Reference PIN Mux table) 0: P2.2 1: UTXD1 6: CAN_TX1 7: CT16B2_PWM2 8: CT16B3_PWM0 11: CT16B5_PWM2 12: ADC0_AIN2 14: LCM_CS 15: ETH_MDIO 16: IRDA_TXD1 Other: Reserved
9:5	IO1	RW	0	Alternate function of P2.1 (Reference PIN Mux table) 0: P2.1 1: URXD3 2: UTXD0 3: UTXD3 4: URTS1

Bit	Field	Access	Initial	Description
				5: SDA1 6: CAN_RX1 7: CT16B2_PWM1 8: CT16B3_PWM0N 11: CT16B5_PWM1 12: ADC0_AIN1 13: SDIO_CMD 14: ETH_RMII_REF_CLK 15: ETH_MII_RX_CLK 16: IRDA_TXD0 17: IRDA_TXD3 18: IRDA_RXDL3 Other: Reserved
4:0	IO0	RW	0	Alternate function of P2.0 (Reference PIN Mux table) 0: P2.0 1: UTXD3 2: URXD0 3: UCTS1 5: SCL1 6: CAN_STBY1 7: CT16B2_PWM0 9: CT16B1_PWM0 11: CT16B5_PWM0 12: ADC0_AIN0 13: SDIO_CK 14: ETH_MII_CRS 16: IRDA_TXD3 18: IRDA_RXDL0 Other: Reserved

3.4.15 GPIO2.6~GPIO2.11 Alternate Function Control register (ALWAYSON_P2_AFIO1)

Address Offset: 0x224

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO11	RW	0	Alternate function of P2.11 (Reference PIN Mux table) 0: P2.11 1: URXD5 2: MO2 3: MO1 4: SDA2 5: DOUT2 6: DOUT1 12: ADC0_AIN11 13: LCM_AD17 15: ETH_MDC 16: SI2 17: SI1 18: IRDA_RXDL5 Other: Reserved
24:20	IO10	RW	0	Alternate function of P2.10 (Reference PIN Mux table) 0: P2.10 1: UTXD5 2: SCL2 12: ADC0_AIN10 13: LCM_AD16 16: IRDA_TXD5 Other: Reserved
19:15	IO9	RW	0	Alternate function of P2.9 (Reference PIN Mux table) 0: P2.9 1: URTS2

Bit	Field	Access	Initial	Description
				2: SCK1 3: BCLK1 6: CT16B1_PWM3 7: CT16B0_PWM1 8: CT16B0_PWM2N 11: CT16B5_PWM3 12: ADC0_AIN9 13: LCM_AD15 14: CAN_STBY0 15: ETH_MII_RXD3 20: SDIO_D7 Other: Reserved
14:10	IO8	RW	0	Alternate function of P2.8 (Reference PIN Mux table) 0: P2.8 1: URXD1 2: MO2 3: DIN0 5: DOUT2 7: CT16B0_PWM1N 8: CT16B0_PWM2 11: CT16B5_PWM2 12: ADC0_AIN8 13: LCM_AD14 14: CAN_STBY1 15: ETH_MII_RXD2 16: SI2 18: IRDA_RXDL1 20: SDIO_D6 Other: Reserved
9:5	IO7	RW	0	Alternate function of P2.7 (Reference PIN Mux table) 0: P2.7 1: MO0 2: DOUT0 6: CT16B0_PWM0N 7: CT16B0_PWM2 8: CT16B1_PWM2 9: CT16B4_PWM0 11: CT16B5_PWM1 12: ADC0_AIN7 13: LCM_AD13 14: ETH_MII_RX_DV 15: ETH_RMII_CRS_DV 16: SI0 20: SDIO_D5 Other: Reserved
4:0	IO6	RW	0	Alternate function of P2.6 (Reference PIN Mux table) 0: P2.6 1: UCTS2 2: MI0 5: CT16B8_PWM2 6: CT16B0_PWM0 7: CT16B0_PWM2N 8: CT16B3_PWM0 9: CT16B4_PWM0N 10: CT16B4_PWM1 11: CT16B5_PWM0 12: ADC0_AIN6 13: LCM_AD12 15: CT16B0_BRK 16: SO0 20: SDIO_D4 Other: Reserved

3.4.16 GPIO2.12~GPIO2.15 Alternate Function Control register (ALWAYSON_P2_AFIO2)

Address Offset: 0x228

Bit	Field	Access	Initial	Description
31:20	–	–	0	Reserved
19:15	IO15	RW	0	Alternate function of P2.15 (Reference PIN Mux table) 0: P2.15 1: URXD2 8: CT16B0_PWM3N 9: CT16B3_PWM1 12: ADC0_AIN15 14: ETH_RMII_RXD1 15: ETH_MII_RXD1 18: IRDA_RXDL2 Other: Reserved
14:10	IO14	RW	0	Alternate function of P2.14 (Reference PIN Mux table) 0: P2.14 1: UTXD2 8: CT16B0_PWM3 9: CT16B3_PWM0 12: ADC0_AIN14 14: ETH_RMII_RXD0 15: ETH_MII_RXD0 16: IRDA_TXD2 Other: Reserved
9:5	IO13	RW	0	Alternate function of P2.13 (Reference PIN Mux table) 0: P2.13 1: MO1 2: DOUT1 12: ADC0_AIN13 15: ETH_MII_TX_CLK 16: SI1 Other: Reserved
4:0	IO12	RW	0	Alternate function of P2.12 (Reference PIN Mux table) 0: P2.12 1: MI1 8: CT16B0_PWM1 12: ADC0_AIN12 15: ETH_MII_TXD2 16: SO1 Other: Reserved

3.4.17 GPIO3.0~GPIO3.5 Alternate Function Control register (ALWAYSON_P3_AFIO0)

Address Offset: 0x230

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO5	RW	0	Alternate function of P3.5 (Reference PIN Mux table) 0: SWDIO 1: P3.5 2: MI2 16: SO2 Other: Reserved
24:20	IO4	RW	0	Alternate function of P3.4 (Reference PIN Mux table) 0: P3.4 1: URXD1 2: URXD5 3: URTS0 4: MI1

Bit	Field	Access	Initial	Description
				5: SDA1 6: DIN0 7: CAN_TX0 13: SDIO_D3 15: ETH_TXER 16: SO1 18: IRDA_RXDL1 20: IRDA_RXDL5 Other: Reserved
19:15	IO3	RW	0	Alternate function of P3.3 (Reference PIN Mux table) 0: P3.3 1: UCTS0 2: UTXD5 3: SEL1 4: SCL1 5: DOUT0 6: WS1 7: CAN_RX0 8: CT16B0_PWM3 13: SDIO_D2 15: ETH_RMII_RX_ER 16: IRDA_TXD5 Other: Reserved
14:10	IO2	RW	0	Alternate function of P3.2 (Reference PIN Mux table) 0: P3.2 1: URXD0 2: MO0 3: MO1 4: SDA0 5: WS0 6: DOUT1 7: CAN_TX1 8: CT16B0_PWM2 13: SDIO_D1 16: SI0 17: SI1 18: IRDA_RXDL0 Other: Reserved
9:5	IO1	RW	0	Alternate function of P3.1 (Reference PIN Mux table) 0: P3.1 1: UTXD0 2: MI0 3: SCK1 4: SCL0 5: BCLK0 6: BCLK1 7: CAN_RX1 8: CT16B0_PWM1 13: SDIO_D0 16: SO0 17: IRDA_TXD0 Other: Reserved
4:0	IO0	RW	0	Alternate function of P3.0 (Reference PIN Mux table) 0: P3.0 1: UTXD1 2: SCK0 3: SCL2 7: CAN_STBY1 8: CT16B0_PWM0 13: LCM_A0 (RS) 15: CLKOUT1 16: IRDA_TXD1 Other: Reserved

3.4.18 GPIO3.6~GPIO3.11 Alternate Function Control register (ALWAYSON_P3_AFIO1)

Address Offset: 0x234

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO11	RW	0	Alternate function of P3.11 (Reference PIN Mux table) 0: P3.11 1: LXOUT 13: LCM_OE Other: Reserved
24:20	IO10	RW	0	Alternate function of P3.10 (Reference PIN Mux table) 0: P3.10 1: LXIN 13: LCM_WE Other: Reserved
19:15	IO9	RW	0	Alternate function of P3.9 (Reference PIN Mux table) 0: P3.9 6: CT16B0_CAP0 12: CT16B8_PWM11 14: LCM_AD9 15: ETH_PHY_PDN Other: Reserved
14:10	IO8	RW	0	Alternate function of P3.8 (Reference PIN Mux table) 0: P3.8 6: CT16B1_CAP0 12: CT16B8_PWM10 14: LCM_AD8 15: ETH_PHY_LINKSTS Other: Reserved
9:5	IO7	RW	0	Alternate function of P3.7 (Reference PIN Mux table) 0: P3.7 1: External reset Other: Reserved
4:0	IO6	RW	0	Alternate function of P3.6 (Reference PIN Mux table) 0: SWCLK 1: P3.6 2: UTXD1 3: MO2 4: DOUT2 16: SI2 17: IRDA_TXD1 Other: Reserved

3.4.19 GPIO3.12~GPIO3.13 Alternate Function Control register (ALWAYSON_P3_AFIO2)

Address Offset: 0x238

Bit	Field	Access	Initial	Description
31:10	–	–	0	Reserved
9:5	IO13	RW	0	Alternate function of P3.13 (Reference PIN Mux table) 0: P3.13 1: XOUT 2: URXD3 3: SCL0 4: SCL1 18: IRDA_RXDL3 Other: Reserved
4:0	IO12	RW	0	Alternate function of P3.12 (Reference PIN Mux table) 0: P3.12 1: XIN

Bit	Field	Access	Initial	Description
				2: UTXD3 3: SDA0 4: SDA1 16: IRDA_TXD3 Other: Reserved

3.4.20 GPIO0~3 Non-overlap Control register (ALWAYSON_GPNOV)

Address Offset: 0x240

Bit	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3	GPIO3	RW	1	GPIO3.0~3.13 Non-overlap control 0: Disable 1: Enable
2	GPIO2	RW	1	GPIO2.0~2.15 Non-overlap control 0: Disable 1: Enable
1	GPIO1	RW	1	GPIO1.0~1.15 Non-overlap control 0: Disable 1: Enable
0	GPIO0	RW	1	GPIO0.0~0.15 and GPIO0.20 Non-overlap control 0: Disable 1: Enable

3.4.21 Oscillator Miscellaneous Control register (ALWAYSON_OSCMISC)

Address Offset: 0x3DC

Bit	Field	Access	Initial	Description
31:6	–	–	0	Reserved
5:4	EHSFREQ[1:0]	RW	0x1	Frequency range of EHS XTAL 0: Reserved 1: 12MHz ~ 16MHz 2: 1MHz ~ 12MHz 3: 16MHz ~ 25MHz
3:0	–	–	0xE	Reserved

4 SYSTEM OPERATION MODE

4.1 OVERVIEW

The chip builds in four operating mode 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
- Deep power down mode

4.2 NORMAL MODE

In Normal mode, the ARM Cortex-M4F core, memories, and peripherals are clocked by the system clock. The SCU_AHBCLKG, SCU_APB0CLKG and SCU_APB1CLKG 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 three special modes of processor power reduction: Sleep mode, Deep-sleep mode and Deep power-down mode. The SCU_PWRMODE 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 and Deep power-down 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-M4F core is stopped and execution of instructions is suspended.

Peripheral functions, if selected to be clocked in SCU_SLP_AHBCLKG, SCU_SLP_APB0CLKG and SCU_SLP_APB1CLKG 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.

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

The INT_WAKEUP bit of SCU_RIS will be set after CPU wake-up from sleep mode. If WAKEUP_EINT bit of SCU_IE is set, system will enter SCU interrupt subroutine (ISR) after CPU wake-up.

The RESET pin has keep functionality in Sleep mode.

4.3.1.1 Entering Sleep Mode

The sleep mode is entered by using the following steps:

1. CPU programs SLEEPDEEP for SCR register (0xE000ED10 bit 2) to control the processor deep sleep as its low power mode.
2. CPU programs NVIC (0xE000E100~) corresponding bit to enable wakeup interrupt source
   - System Tick timer must be disabled to avoid wake-up system.
   - Disable other peripheral interrupt to avoid wake-up system.
3. CPU programs SCU_SLP_AHBCLKG Bit[31:0] : AHB clock sleep Control Register
   - Each bit controls one HCLK enabled or gated.
   - Enabled or gated signal will not change until SCU_PWRMODE bit[3] is set.
   - Some wake up events need specified clocks, if clocks are gated off, wake up interrupts will not occur.
   - CPU FCLK must be on
4. CPU programs SCU_SLP_APB0CLKG and SCU_SLP_APB1CLKG Bit[31:0] : APB clock sleep Control Register
   - Each bit controls one PCLK enabled or gated.
   - Enabled or gated signal will not change until SCU_PWRMODE bit[3] is set.
   - Some wake up events need specified clocks, if clocks are gated off, wake up interrupts will not occur
5. If DMA operating needs to access flash, EFC_STB_OFF bit of SCU_PWRMODE is set before CPU enters sleep mode.
6. CPU programs SCU_SLP_WAKUPEN : wake-up event enable register
   - Enable NVIC interrupt event enable
     - Bit1 : NVIC interrupt as 0x1 (default on)
7. CPU programs SCU_PWRMODE : power mode register
   - SLEEP (Bit[3]) as 1
8. CPU sets itself to enter "wait for interrupt" state by WFI instruction
   - CPU enter WFI
9. System controller waits for sideband signal from CPU
10. System controller updates all clock enable that set in SCU_SLP_AHBCLKG, SCU_SLP_APB0CLKG and SCU_SLP_APB1CLKG
11. All respective clocks are enabled or gated as CPU programmed
12. WIC waits for wake-up interrupt trigger

4.3.1.2 Exiting Sleep Mode

The chip wakes up from sleep mode in the following steps:

1. An interrupt occurs to wake up the system.
2. Wake-up Interrupt Controller got relative wakeup source and issue wakeup signal to system controller.
3. System controller updates all clock enable that set in SCU_AHBCLKG, SCU_APB0CLKG and SCU_APB1CLKG.
4. CPU wake-up, enter corresponding interrupt subroutine (ISR)

✷ ***Note: If a wake-up event occurs at the same time as entering sleep mode, the WFI instruction may be ignored and the CPU will execute the next program. When this occurs, the SLEEP bit cannot be cleared automatically. In this case, the PWRMODECLR bit can be used to clear it.***

4.3.2 DEEP-SLEEP MODE

In Deep-sleep mode, the system clock to the ARM Cortex-M4F 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 GPIO interrupt to the ARM core. Furthermore, the interrupts corresponding to each input must be enabled in the NVIC.

Wake up the chip from Deep sleep mode by a wakeup event occurs.

The INT_DS_WAKEUP bit of SCU_RIS will be set after CPU wake-up from deep sleep mode. If DS_WAKEUP_EINT bit of SCU_IE is set, system will enter SCU interrupt subroutine (ISR) after CPU wake-up.

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

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.

4.3.2.1 Entering Deep Sleep Mode

The deep sleep mode is entered by using the following steps:

1. CPU programs NVIC (0xE000E100~) corresponding bit to enable/disable wakeup interrupt source
   - GPIO0~3 interrupt
   - SCU interrupt (RTC alarm)
   - WOL interrupt
   - System Tick timer must be disabled to avoid wake-up system.
   - Disable other peripheral interrupt to avoid wake-up system.
2. Clear bit 17~24 of SCU_BTUP_STS to avoid wake-up system.
3. CPU programs SCU_BTUP_CTRL
   - Bit[16] = 0 (MUST)
   - Bit[17] = 1 to enable wake-up from RTC interrupt (RTC clock must be presented)
   - Bit[18] = 1 to enable wake-up by GPIO0 interrupt (Only non_clk_mode)
   - Bit[19] = 1 to enable wake-up by GPIO1 interrupt (Only non_clk_mode)
   - Bit[20] = 1 to enable wake-up by GPIO2 interrupt (Only non_clk_mode)
   - Bit[21] = 1 to enable wake-up by GPIO3 interrupt (Only non_clk_mode)
   - Bit[23] = 1 to enable wake-up by Ethernet WOL

4. CPU programs SRAM retention enable bits.
   - AUTOSLPEN (SRAM1CTRL[2]) as 1
   - AUTOSLPEN (SRAM2CTRL[2]) as 1
   - AUTOSLPEN (SRAM3CTRL[2]) as 1
   - AUTOSLPEN (BKPSRAMCTRL[2]) as 1
   - CAN0SRAMRET (SCU_GMAC_SC[30]) as 1
   - CAN1SRAMRET (SCU_ GMAC_SC[29]) as 1
   - ETHSRAMRET (SCU_ GMAC_SC[28]) as 1
5. CPU programs SCU_PWRMODE
   - DSLEEP (Bit[2]) as 1
   - EFC_STB_OFF (Bit[31]) as 0
6. CPU sets itself to enter “wait for interrupt” state by WFI instruction
   - CPU enter WFI
7. System controller waits for sideband signal from CPU
8. System controller controls followings in the correct sequence
   - PLL off (if enable)
   - IHRC off (if enable)
   - EHS off (if enable)
   - ILRC off (if RTC clock source is ILRC, ILRC should be ON)
9. SCU enter Deep Sleep mode

4.3.2.2 Exiting Deep Sleep Mode

The chip wakes up from deep sleep mode in the following steps:

1. System controller detects wake-up events that are enabled in SCU_BTUP_CTRL
   - ILRC on
2. System controller enables IHRC/PLL/EHS
   - Restore IHRC if it was enabled before entering deep sleep mode
   - Restore EHS if it was enabled before entering deep sleep mode
   - Restore PLL if it was enabled before entering deep sleep mode
3. The corresponding wakeup event flag of SCU_BTUP_STS will be set.
4. System controller wake-up Flash from deep sleep mode to active mode
5. System controller updates all clock enable that set in SCU_AHBCLKG, SCU_APB0CLKG and SCU_APB1CLKG.
6. CPU wakeup, enter corresponding interrupt subroutine (ISR)
7. Clear the wakeup event flag of SCU_BTUP_STS.
8. CPU disable SRAM retention.
   - AUTOSLPEN (SRAM1CTRL[2]) as 0
   - AUTOSLPEN (SRAM2CTRL[2]) as 0
   - AUTOSLPEN (SRAM3CTRL[2]) as 0
   - AUTOSLPEN (BKPSRAMCTRL[2]) as 0
   - CAN0SRAMRET (SCU_GMAC_SC[30]) as 0
   - CAN1SRAMRET (SCU_GMAC_SC[29]) as 0
   - ETHSRAMRET (SCU_GMAC_SC[28]) as 0

✸ ***Note:***

1. If a wake-up event occurs at the same time as entering deep sleep mode, the WFI instruction may be ignored and the CPU will execute the next program. When this occurs, the DSLEEP bit cannot be cleared automatically. In this case, the PWRMODECLR bit can be used to clear it.

2. Before clearing DSLEEP bit through PWRMODECLR bit, DS_WAKEUP_EINT bit must be cleared to avoid interrupts caused by INT_DS_WAKEUP flag generated after clearing.

4.3.3 Deep Power-Down mode

In Deep power-down mode, power (Turn off the on-chip voltage regulator) and clocks are shut off to the entire chip with the exception of the GPIO pins.

The processor state and registers, peripheral registers, and internal SRAM values are not retained. However, the chip can retain data in twenty BACKUP registers and 4KB BKP SRAM, and the status of all GPIO pins can also be latched. The user must program the GPIO registers for each pin to set the appropriate GPIO status (input pull-up, input pull-down, input floating, output high, and output low, open-drain). After entering deep power-down mode, the system will automatically set the GPIO_HOLD bit of the SCU_BTUP_STS register and latch the status of all GPIO pins.

All GPIO pins can be 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 power-down mode when GPIO indicates a GPIOn wakeup event to system controller. The on-chip voltage regulator will be turned on, and when the core voltage reaches the power-on-reset (POR) trip point, a system reset will be triggered and the chip re-boots.

Once the chip has rebooted, the user can read SCU_BTUP_STS register to verify that the reset was caused by a wake-up event from Deep power-down and was not a cold reset. If MCU wakes up from Deep power-down mode, the user must program GPIO registers with the same settings (input pull-up, input pull-down, input floating, output high, and output low) and then clear GPIO_HOLD bit of SCU_BTUP_STS register to release the status of all GPIO pins.

Wake up the chip from Deep sleep mode by a wakeup event (GPIO or RTC) occurs.

The RESET pin has keep functionality in Deep power-down mode.

The advantage of deep power-down mode is that can power off most of the logic circuits, clock circuits and memory, thereby gaining far greater power savings than deep-sleep mode. Since the settings cannot be retained when the power is turned off, the system needs to be reset after wakeup.

4.3.3.1 Entering Deep Power-Down Mode

The deep power-down mode is entered by using the following steps:

1. Clear bit 17~23 of SCU_BTUP_STS to avoid wake-up system.
2. CPU programs SCU_BTUP_CTRL
   - Bit[16] = 0 (MUST)
   - Bit[17] = 1 to enable wake-up from RTC interrupt (RTC clock must be presented)
   - Bit[18] = 1 to enable wake-up by GPIO0 interrupt (Only non_clk_mode)
   - Bit[19] = 1 to enable wake-up by GPIO1 interrupt (Only non_clk_mode)
   - Bit[20] = 1 to enable wake-up by GPIO2 interrupt (Only non_clk_mode)
   - Bit[21] = 1 to enable wake-up by GPIO3 interrupt (Only non_clk_mode)
3. (Optional) Save data to be retained during Deep power-down to the backup registers or backup SRAM.
4. CPU programs SCU_PWRMODE
   - DPD(Bit[1]) as 1
5. CPU sets itself to enter “wait for interrupt” state by WFI instruction
   - CPU enter WFI
6. System controller waits for sideband signal from CPU
7. Hardware sets GPIO_HOLD bit of SCU_BTUP_STS to latch the status of all GPIO pins.
8. System controller controls internal power switches and isolation cells in correct sequence
   - GPIO output was held
   - Most of the logic circuits, clock circuits and memory are power-off
   - ILRC off (if RTC clock source is ILRC, ILRC should be ON)
9. SCU enter Deep power down mode

4.3.3.2 Exiting Deep Power-Down Mode

The chip wakes up from deep power-down mode in the following steps:

1. System controller detects wake-up events that are enabled in SCU_BTUP_CTRL
   - ILRC turn on
   - Resume the power of shut down area
2. System controller enables IHRC/EHS

- Set system clock to IHRC
- Restore EHS if needed

3. A system reset will be triggered and the chip reboots.
   - CPU will boot-up from 0x0000_0000 address
   - All registers will be reset, except the Backup registers, SCU 0x000~0x004 registers, RTC registers, ALWAYSON registers and GPIO registers.
   - The DPDWKF bit of SCU_BTUP_STS will be set.
   - The corresponding wakeup event flag of SCU_BTUP_STS will be set.
4. Clear the DPDWKF bit of SCU_BTUP_STS.
5. Clear the wakeup event flag of SCU_BTUP_STS.
6. (Optional) Read the stored data in the backup registers or backup SRAM.
7. Setup the same GPIO status of all GPIO pins as the last status.
8. Clear GPIO_HOLD bit of SCU_BTUP_STS to release the status of all GPIO pins.

✸ ***Note:***

1. If a wake-up event occurs at the same time as entering deep power-down mode, the WFI instruction may be ignored and the CPU will execute the next program. When this occurs, the DPD bit cannot be cleared automatically. In this case, the PWRMODECLR bit can be used to clear it.

2. When the WFI instruction is ignored, the system should think of it as abnormal or re-execute into deep power-down mode.

4.4 SRAM POWER SAVING

4.4.1 Automatic Power Saving

The system provides automatic sleep/retention/nap mode to save power. If this function is enabled, when the SRAM is not being read for more than the time, the SRAM will automatically sleep to save power.

The SRAM automatic sleep/retention/nap modes are switched on and off using the AUTOSLPEN/AUTORETEN/AUTONAPEN bits. (SCU SRAM1~3CTRL[2:0] & BKPSRAMCTRL[2:0])

The time-out counter is programmed by TIMEOUT bits. (SCU SRAM1~3CTRL[27:16] & BKPSRAMCTRL[27:16])

The correlative resume counter is programmed by SLPRESDLY, RETRESDLY and NAPRESDLY bits. (SCU SRAM1~3CTRL[15:4] & BKPSRAMCTRL[15:4])

✷ ***Note: We strongly recommend using the default values for time-out and resume, and using auto-sleep mode to save power.***

SRAM auto-power saving mode control flow can be separated into three parts:

1. When an unused SRAM accepts new AHB command, system controller will wake up SRAM and wait for resuming delay cycles (default 2 cycles) to read/write SRAM.
2. If there is no effective AHB command, it will decrease the down count timer (default 4cycles) and check the timer status. SRAM will enter the power saving mode when the timer counter is equal to zero.
3. If SRAM enters power saving mode, the system controller will turn off the SRAM clock to achieve much lower power consumption.

4.4.2 Peripheral SRAM Power Saving

The system controller unit can support the peripheral SRAM power-down mode function. The SRAM power-down mode function provides a method which forces some peripheral SRAMs to enter the power-saving mode by the retention control signal.

The Ethernet MAC SRAM power-down mode is switched on and off using the ETHSRAMRET bits. (SCU_ GMAC_SC[28])
The CAN1 SRAM power-down mode is switched on and off using the CAN1SRAMRET bits. (SCU_ GMAC_SC[29])
The CAN0 SRAM power-down mode is switched on and off using the CAN0SRAMRET bits. (SCU_ GMAC_SC[30])

Power-down mode can be turned on or off only when the SRAM clock is enable. The SRAM clocks are controlled by ETHSRAMEN, CAN1SRAMCLKEN and CAN0SRAMCLKEN bits. (SCU AHBCLKG[14], SCU_APB0CLKG[14:13])

For Ethernet MAC and CAN Bus SRAM retention mode control, when Ethernet function or CAN bus function are not used, FW can directly put SRAM to retention mode by setting SCU_GMAC_SC register,
− bit[30] : CAN0SRAMRET = 0x1
− bit[29] : CAN1SRAMRET = 0x1
− bit[28] : ETHSRAMRET = 0x1

Peripheral SRAM	Retention control	SRAM clock control
Ethernet MAC SRAM	ETHSRAMRET	ETHSRAMEN
CAN1 SRAM	CAN1SRAMRET	CAN1SRAMCLKEN
CAN0 SRAM	CAN0SRAMRET	CAN0SRAMCLKEN

4.5 WAKEUP

4.5.1 OVERVIEW

Under sleep mode and deep sleep 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 alarm interrupt, Ethernet WOL interrupt and the RESET pin.

In deep power-down mode, wake-up is through GPIO and RTC wake-up events. But after waking up, the reset procedure will start.

4.5.2 WAKEUP TIME

4.5.2.1 Sleep mode

When the system is in sleep mode, the CPU, NVIC and some peripheral clocks will be stopped, and the flash will be turned off, but the Wake-up Interrupt Controller clock will be retained for receiving interrupt wakeup. When MCU is waken up from sleep mode, MCU waits for 91T HCLK + Twake. This time is for recovering the CPU, NVIC and peripheral clocks and flash warm up. After the wakeup time, the system goes into the normal mode.

- ***Note:***
  1. ***The Twake time is flash warm-up time. Typical value is 15us.***
  2. ***If the flash keeps on while in sleep mode. The wake-up time can be shortened to 23T HCLK.***

4.5.2.2 Deep Sleep Mode

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 3T ILRC + 85T HCLK + Twake and IHRC/EHS/PLL warm up time. After the wakeup time, the system goes into the normal mode.

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

The Warm up time of EHS X'tal = 2T ILRC + 1/F_{EHS} * 4096 (sec) + high clock start-up time

- Example: F_{EHS}=16MHz, ILRC=32K and HCLK=EHS/1, the wakeup time from deep sleep mode is as the following.

The total Wakeup time = 5T ILRC + 85T HCLK + 1/F_{EHS} * 4096 + oscillator start-up time + Twake

= 417.563 us + oscillator start-up time + Twake ((F_{EHS} = 16MHz)

The value of the IHRC warm up time is as the following.

The Warm up time of IHRC = 1T ILRC + 1/F_{IHRC} * 256 (sec)

- Example: F_{IHRC}=12MHz, ILRC=32K and HCLK=IHRC/1, the wakeup time is as the following.

The total Wakeup time = 4T ILRC + 85T HCLK + 1/F_{IHRC} * 256 + Twake

= 153.417 us + Twake (F_{IHRC} = 12MHz)

If the PLL reference clock is EHS, the value of the external high-speed clock oscillator and PLL warm-up time from Deep sleep mode is as the following.

The Warm up time of PLL (ref. EHS) = 5T ILRC + 1/F_{EHS} * 4096 (sec) + high clock start-up time

- Example: F_{EHS}=16MHz, ILRC=32K, PLL output = 192MHz and HCLK=PLL/1, the wakeup time from Deep sleep mode is as the following.

The total Wakeup time = 8T ILRC + 85T HCLK + 1/F_{EHS} * 4096 + oscillator start-up time + Twake

= 506.443 us + oscillator start-up time + Twake ((F_{EHS} = 16MHz, PLL = 192MHz)

If the PLL reference clock is IHRC, the value of the IHRC and PLL warm-up time from Deep sleep mode is as the following.

The Warm up time of PLL (ref. IHRC) = 5T ILRC + 1/F_{IHRC} * 256 (sec)

- Example: F_{IHRC}=12MHz, ILRC=32K, PLL output = 192MHz and HCLK=PLL/1, the wakeup time is as the following.

The total Wakeup time = 8T ILRC + 85T HCLK + 1/F_{IHRC} * 256 + Twake

= 271.776 us + Twake (F_{IHRC} = 12MHz, PLL = 192MHz)

✸ ***Note:***
1. ***The Twake time is flash warm-up time. Typical value is 15us.***
2. ***The high clock start-up time is depended on the VDD and oscillator type of high clock.***

4.5.2.3 Deep Power-Down Mode

When the system is in deep power-down mode, the high clock will stop or power off. When MCU is waken up from deep power-down mode, MCU waits for 12T ILRC + 18200T HCLK and IHRC warm up time. After the wakeup time, the system reboots into normal mode.

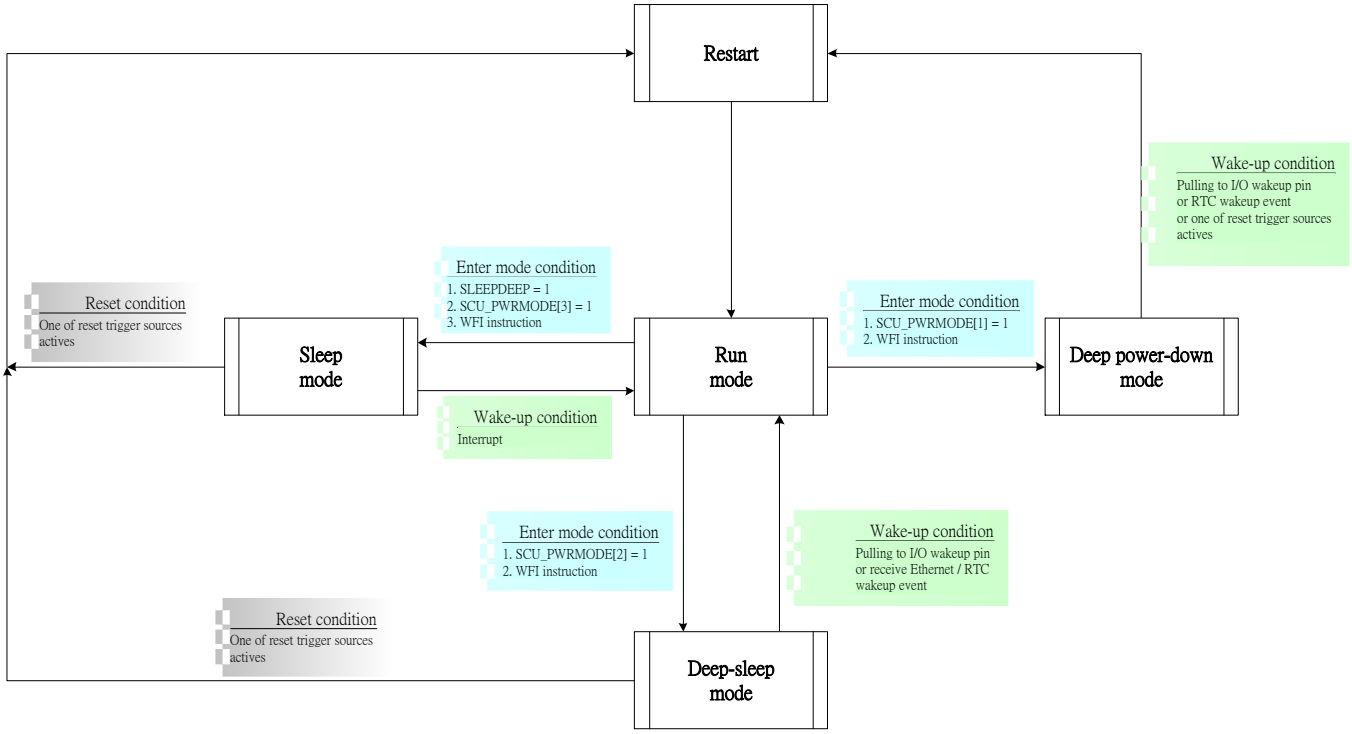
- Example: The default HCLK is IHRC/1. F_{IHRC}=12MHz and ILRC=32K, the wakeup time is as the following.

The total Wakeup time = 12T ILRC + 18200T HCLK + 1/F_{IHRC} * 256

= 1913 us (F_{IHRC} = 12MHz)

✸ ***Note: If EHS is enabled before entering Deep power-down mode. EHS will restart after waking up from deep power-down mode.***

4.6 STATE MACHINE OF PMU



➢ GPIO can only use non_clk_mode to wake up from deep sleep and deep power-down modes.

4.7 OPERATION MODE COMPARSION TABLE

Operation Mode	Normal Mode	Low-Power Mode		
		Sleep Mode	Deep Sleep Mode	Deep power-down
IHRC	By IHRCEN		Disable	Power off
ILRC	ON		Enable if used as clock source of RTC	Enable if used as clock source of RTC
EHS X'TAL	By EHSEN		Disable	Disable
ELS X'TAL	By ELSEN		By ELSEN	By ELSEN
PLL	By PLLEN		Disable	Power off
Cortex-M4	Running	Stop	Stop	Power off
Flash ROM	Enable	By EFC_STB_OFF and FLASHEN	Disable	Power off
RAM	Enable	By SRAMnEN	Power saving by AUTOSLPEN	Power off
Backup RAM	By BKPSRAMEN		Power saving by AUTOSLPEN	Maintain
Backup register	By ALWAYSONREGEN		Maintain	Maintain
ADC	By ADEN		Disable	Power off
LVD	By LVDEN		Disable	Power off
Ethernet	By ETHMAC_MACCTRL[3:0]		Wake-On-LAN wake-up By ETHMAC_WOLCTRL[6:0]	Power off
RTC	By RTC_EN		By RTCEN	By RTCEN
Peripherals	By Enable bit of each peripherals		Disable HCLK	Power off
IO status	-	Maintained	Maintained	Maintained
Wakeup Source	N/A	All interrupts, RESET pin	GPIO interrupt[1], RTC alarm, Ethernet wake-on-LAN wakeup RESET pin	GPIO interrupt[1], RTC alarm, RESET pin

[1] Only non_clk_mode.

4.8 PMU REGISTERS

Base Address: 0x4001 F000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x020	SCU_PWRMODE	SCU Power Mode register
0x128	SCU_PWRMODEMISC	SCU Power Mode Clear register

4.8.1 SCU Power Mode register (SCU_PWRMODE)

Address Offset: 0x20

The power control register selects whether one of the system controlled power-down modes (Sleep mode, Deep-sleep mode or Deep power-down mode) and provides the flags 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	Field	Access	Initial	Description
31	EFC_STB_OFF	RW	0	eFlash stand-by command off 0: eFlash enters standby mode in all power down mode and frequency change sequence. 1: eFlash normal run
30:29	–	–	0	Reserved
28	FCS_PLLRSTOFF	RW	0	PLL resets in the frequency change sequence 0: Reset PLL for locking 1: keep PLL active in FCS
27:25	–	–	0	Reserved
24	REMAP	RW	0	Target remap programmable register After issuing the reboot function for the changed address remapping of the BOOT_CODE/USER_CODE. 0: Shadow memory is BOOT CODE 1: Shadow memory is USER CODE
23:11	–	–	0	Reserved
10	PERRST	RWS	0	Peripheral reset command 0: No effect 1: Issue a reset to initiate multiple peripherals The bit is set to issue a reset signal that the interval can be adjusted by programming PERRSTCTL, and be cleared by hardware after the reset is done.
9:8	–	–	0	Reserved
7	REBOOT	RWS	0	Reboot command 0: No effect 1: Issue a reset to initiate the CPU and bus state
6	FCS	RWS	0	Frequency Change Sequence (FCS) 0: No effect 1: Enter the frequency change sequence cleared after finishing FCS. The command does not execute with the SLEEP, DSLEEP, or DPD command. It can be clear to 1'b0 when PWRMODECLR bit is set to 1'b0.
5:4	–	–	0	Reserved
3	SLEEP	RWS	0	Sleep-mode sequence

Bit	Field	Access	Initial	Description
				0: Not sleep mode 1: Enter the sleep mode This bit will be automatically cleared after waking up. The command does not execute with the FCS, DSLEEP, or DPD command. It can be clear to 1'b0 when PWRMODECLR bit is set to 1'b0.
2	DSLEEP	RWS	0	Deep sleep mode sequence 0: Not Deep sleep mode 1: Enter the Deep sleep mode This bit will be automatically cleared after waking up. The command does not execute with the FCS, SLEEP, or DPD command. It can be clear to 1'b0 when PWRMODECLR bit is set to 1'b0.
1	DPD	RWS	0	Deep power-down mode sequence 0: Not Deep power-down mode 1: Enter the Deep power-down mode This bit will be automatically cleared after waking up. The command does not execute with the FCS, SLEEP, or DSLEEP command. It can be clear to 1'b0 when PWRMODECLR bit is set to 1'b0.
0	–	–	0	Reserved

4.8.2 SCU Power Mode Clear register (SCU_PWRMODEMISC)

Address Offset: 0x128

Bit	Field	Access	Initial	Description
31:13	–	–	0	Reserved
12	PWRMODECLR	W	1	Always 1'b1 when reading. Writing 0 will clear FCS/SLEEP/DSLEEP/DPD bits of PWRMODE to 1'b0.
11:0	–	–	0	Reserved

5 GENERAL PURPOSE I/O PORT (GPIO)

5.1 OVERVIEW

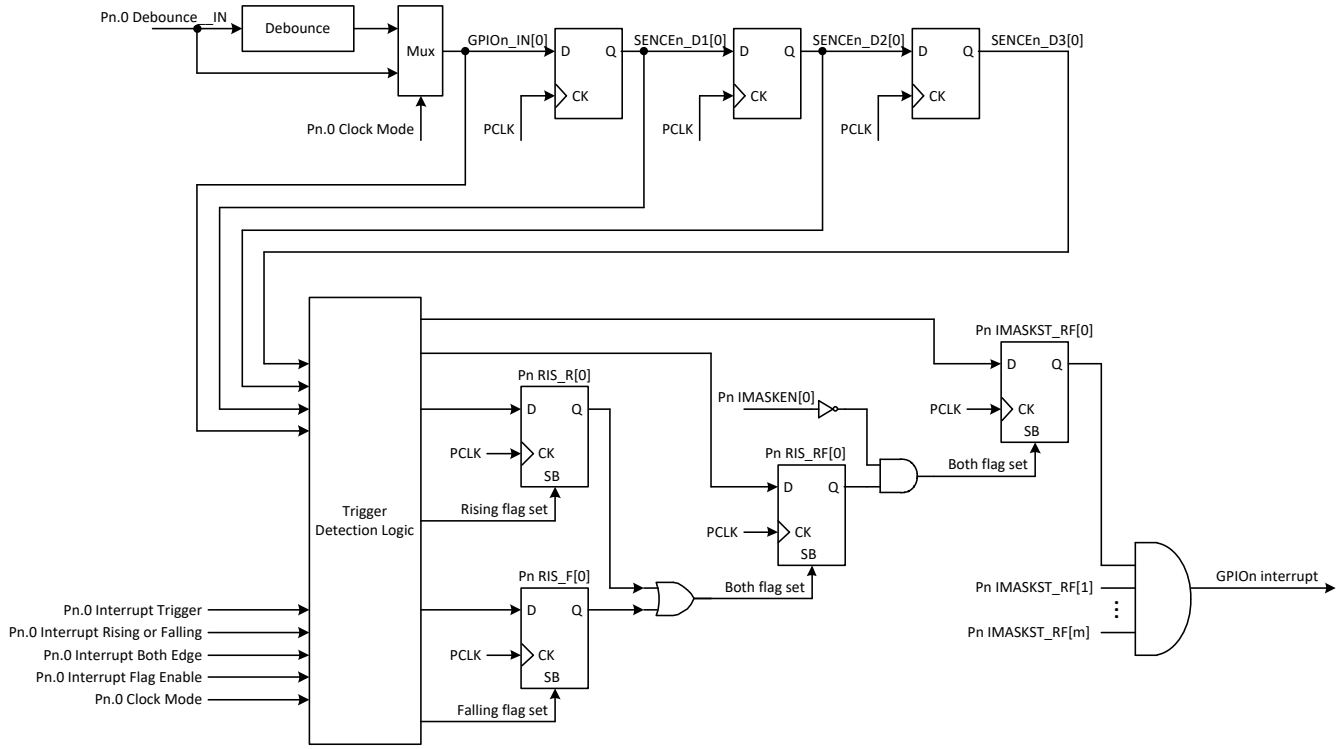
Digital ports can be configured input/output by SW.

- 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, and high/low level.
- Individual interrupt levels can be programmed.
- 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.
- All GPIO pins are inputs and floating by default.

5.2 FEATURES

- Up to 55 independent input, output, and output enable buses for bidirectional I/O pins
- Each port can individually trigger GPIO interrupt when programmed as input pin.
- Each port of interrupt generation can be triggered at rising edge, falling edge, both edges, or at high/low level.
- Each port can be pulled high or pulled low.
- Each port can choose pre-scaled or PCLK clock source.
- Output data bit can be separately set or cleared.
- All ports are set as inputs upon hardware reset.
- If the clock is turned off, rising and falling edges can be detect. Rising and falling interrupt status of each port can be independently masked.

5.3 TRIGGER DETECTION



In the trigger detection block, the GPIO_IN will be delayed by 3 cycles for edge or level detection. The GPIO_IN, SENCE_D1, SENCE_D2, SENCE_D3, Interrupt Trigger, Interrupt Rising or Falling, Interrupt Both Edge, Interrupt Flag Enable and Clock Mode are used to generate trigger flag for trigger detection.

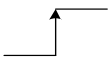
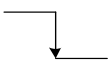
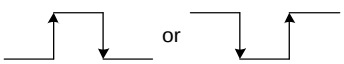
In the normal mode interrupt state will be triggered when high level, low level, rising edge, falling edge or both edge are detected. The non-clock mode interrupt state is only triggered on rising edge, falling edge or both edge.

The rising flag is triggered when a high level or rising edge is detected. The falling flag is triggered when a low level or rising edge is detected.

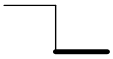
The rising flag set and falling flag set are generated to set RIS_R and RIS_F high. The flag states of rising edge or falling edge can become RIS_RF signal.

If IMASKEN is 1, it will mask the RIS_RF, and does not set the IMASKST_RF. Any IMASKST_RF bits of GPIOn will be generated GPIOn interrupt signal.

5.3.1 Edge Trigger

CLKMODE	ITRIG	IBETRIG	IRFTRIG	IE	Trigger Waveform	RIS_R	RIS_F	RIS_RF
Normal mode/ Non-clock mode	0	0	0	1		1	0	1
	0	0	1	1		0	1	1
	0	1	Don't care	1		1	1	1

5.3.2 Level Trigger

CLKMODE	ITRIG	IBETRIG	IRFTRIG	IE	Trigger Waveform	RIS_R	RIS_F	RIS_RF
Normal mode	1	Don't care	0	1		1	0	1
	1	Don't care	1	1		0	1	1

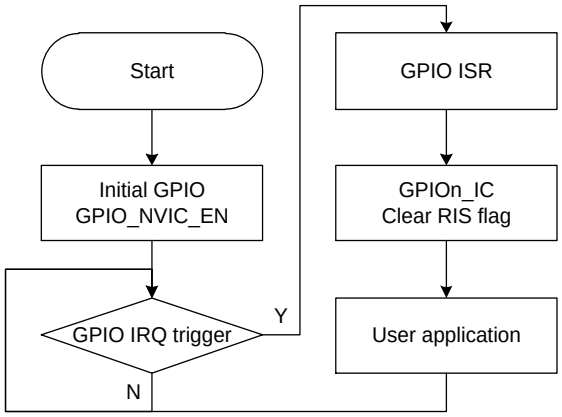
✸ ***Note: Level trigger only works in GPIO normal mode.***

5.4 INTERRUPT SUBROUTINE

5.4.1 Normal Mode

The GPIO normal mode is used in normal and sleep mode.

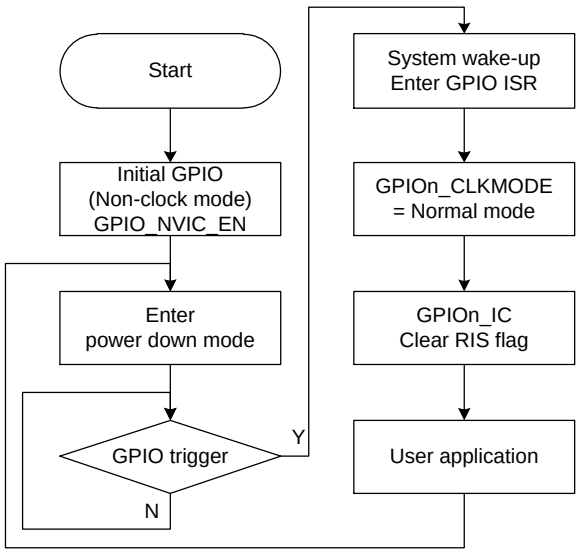
The RIS_RF, RIS_R and RIS_F flags are clear by GPIOn_IC bit. User can use GPIOn_IC_R/GPIOn_IC_F bit to clear only RIS_R/RIS_F flag. The IMASKST_RF can be clear by GPIOn_IC, GPIOn_IC_R or GPIOn_IC_F bit.



5.4.2 Non-Clock Mode

The GPIO non-clock mode is used in deep sleep and deep power-down mode. After wake-up, please switch to normal mode in GPIOn ISR first for clear flag.

The RIS_RF, RIS_R and RIS_F flags are clear by GPIOn_IC bit. User can use GPIOn_IC_R/GPIOn_IC_F bit to clear only RIS_R/RIS_F flag. The IMASKST_RF can be clear by GPIOn_IC, GPIOn_IC_R or GPIOn_IC_F bit.

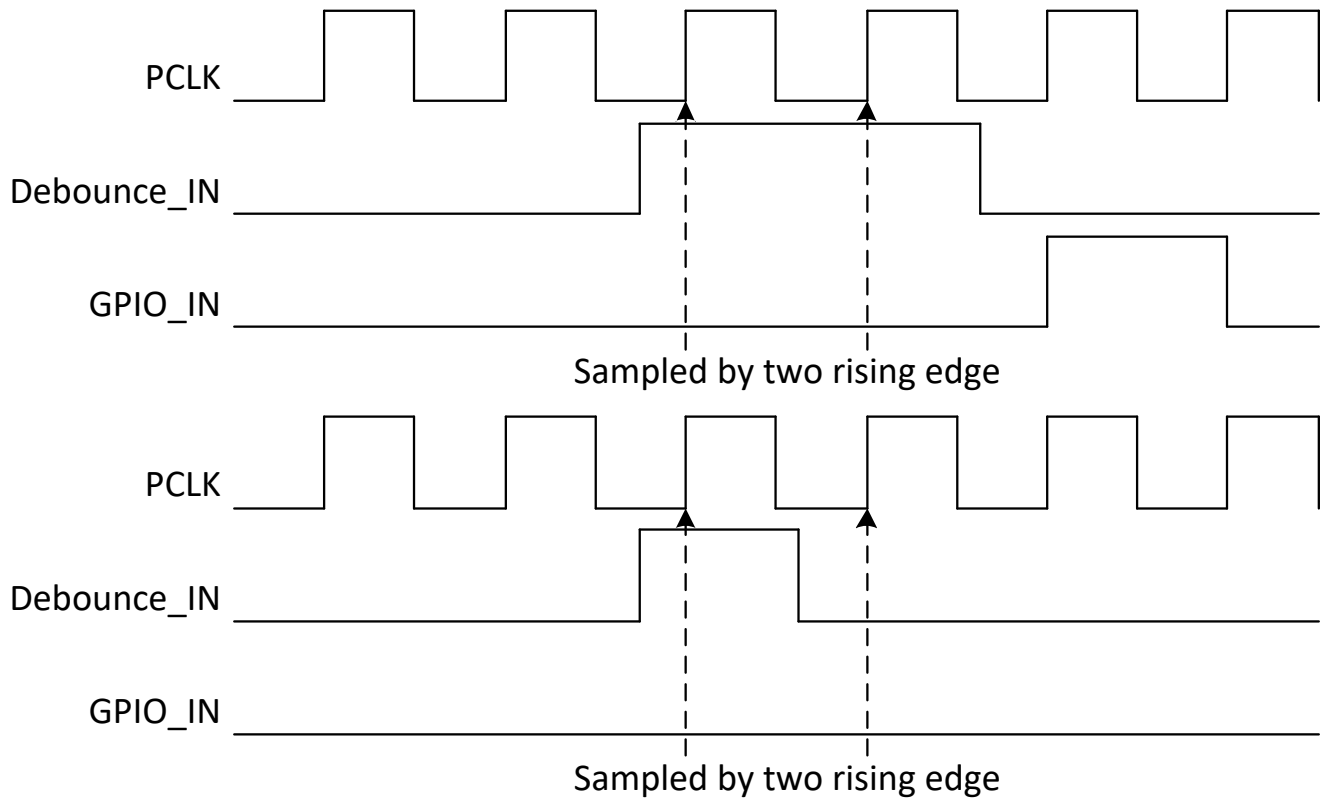


✷ ***Note: Clear bit only works in GPIO normal mode.***

5.5 INPUT DE-BOUNCE

This BNEN register controls the de-bounce function. If BNEN is enabled, the trigger detection will be sampled by the bounce clock.

The general-purpose input/output controller is capable of filtering out the external bounce which is sampled less than two continuous PCLK (or de-bounce clock) rising edges when the trigger function is enabled. It is recommended that the duration of the trigger signal is longer than two clock periods (or de-bounce clock period) to guarantee that the general-purpose input/output controller will sample the trigger signal by two rising edges of clock.



✷ ***Note: GPIO de-bounce function is only applicable to Normal mode (Non-Clock mode is invalid).***

5.6 GPIO REGISTERS

Base Address: 0x4003 7000 (GPIO 0)
0x4003 8000 (GPIO 1)
0x4003 9000 (GPIO 2)
0x4003 A000 (GPIO 3)

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	GPIOn_DATAOUT	GPIO Data-out register
0x0004	GPIOn_DATAIN	GPIO Data-in register
0x0008	GPIOn_DIR	GPIO Pin Direction register
0x000C	-	Reserved
0x0010	GPIOn_BSET	GPIO Data Bit Set register
0x0014	GPIOn_BCLR	GPIO Data Bit Clear register
0x0018	GPIOn_PULLEN	GPIO Pin Pull Enable register
0x001C	GPIOn_PUTYPE	GPIO Pull Type register
0x0020	GPIOn_IE	GPIO Interrupt Flag Enable register
0x0024 – 0x0028	-	Reserved
0x002C	GPIOn_IMASKEN	GPIO Interrupt Mask Enable register
0x0030	GPIOn_IC	GPIO Interrupt Clear register
0x0034	GPIOn_ITRIG	GPIO Interrupt Trigger register
0x0038	GPIOn_IBETRIG	GPIO Interrupt Both Edge Trigger register
0x003C	GPIOn_IRFTRIG	GPIO Interrupt Rising or Falling Edge Trigger register
0x0040	GPIOn_BNEN	GPIO Bounce Enable register
0x0044	GPIOn_BNCLKPRE	GPIO Bounce Clock Prescaler register
0x0048	GPIOn_CLKMODE	GPIO Clock Mode register
0x004C	GPIOn_RIS_R	GPIO Interrupt Raw State of Rising Edge register
0x0050	GPIOn_RIS_F	GPIO Interrupt Raw State of Falling Edge register
0x0054	GPIOn_RIS_RF	GPIO Interrupt Raw State of Both Edge register
0x0058	GPIOn_IC_R	GPIO Interrupt Clear of Rising Edge register
0x005C	GPIOn_IC_F	GPIO Interrupt Clear of Falling Edge register
0x0060	GPIOn_IMASKST_RF	GPIO Interrupt Masked State register

➢ ***Note: GPIO0/1/2/3 have different number of pins. The bit number of the correlative register is also different. But the bit position is the same.***

GPIO0: P0.0~P0.2, P0.10~P0.15, P0.20
GPIO1: P1.0~P1.15
GPIO2: P2.0~P2.15
GPIO3: P3.0~P3.13

5.6.1 GPIO Port n Data Out register (GPIOn_DATAOUT) (n=0,1,2,3)

Address offset: 0x00

The DATAOUT register is the GPIO data out register. When DIR indicates the pin is an input, the DATAOUT register can hold the data.

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	DATA20	RW	0	Data out of Pn.20 0: Pn.20 is 0 1: Pn.20 is 1
19:16	–	–	0	Reserved

Bit	Field	Access	Initial	Description
15	DATA15	RW	0	Data out of Pn.15 0: Pn.15 is 0 1: Pn.15 is 1
14	DATA14	RW	0	Data out of Pn.14 0: Pn.14 is 0 1: Pn.14 is 1
13	DATA13	RW	0	Data out of Pn.13 0: Pn.13 is 0 1: Pn.13 is 1
12	DATA12	RW	0	Data out of Pn.12 0: Pn.12 is 0 1: Pn.12 is 1
11	DATA11	RW	0	Data out of Pn.11 0: Pn.11 is 0 1: Pn.11 is 1
10	DATA10	RW	0	Data out of Pn.10 0: Pn.10 is 0 1: Pn.10 is 1
9	DATA9	RW	0	Data out of Pn.9 0: Pn.9 is 0 1: Pn.9 is 1
8	DATA8	RW	0	Data out of Pn.8 0: Pn.8 is 0 1: Pn.8 is 1
7	DATA7	RW	0	Data out of Pn.7 0: Pn.7 is 0 1: Pn.7 is 1
6	DATA6	RW	0	Data out of Pn.6 0: Pn.6 is 0 1: Pn.6 is 1
5	DATA5	RW	0	Data out of Pn.5 0: Pn.5 is 0 1: Pn.5 is 1
4	DATA4	RW	0	Data out of Pn.4 0: Pn.4 is 0 1: Pn.4 is 1
3	DATA3	RW	0	Data out of Pn.3 0: Pn.3 is 0 1: Pn.3 is 1
2	DATA2	RW	0	Data out of Pn.2 0: Pn.2 is 0 1: Pn.2 is 1
1	DATA1	RW	0	Data out of Pn.1 0: Pn.1 is 0 1: Pn.1 is 1
0	DATA0	RW	0	Data out of Pn.0 0: Pn.0 is 0 1: Pn.0 is 1

5.6.2 GPIO Port n Data In register (GPIOn_DATAIN) (n=0,1,2,3)

Address offset: 0x04

The DATAIN register is the GPIO data in register. When DIR indicates the pin is an output, the DATAIN register will be a “don't care” register.

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	DATA20	R	0	Data In of Pn.20 0: Pn.20 is 0 1: Pn.20 is 1
19:16	–	–	0	Reserved

Bit	Field	Access	Initial	Description
15	DATA15	R	0	Data In of Pn.15 0: Pn.15 is 0 1: Pn.15 is 1
14	DATA14	R	0	Data In of Pn.14 0: Pn.14 is 0 1: Pn.14 is 1
13	DATA13	R	0	Data In of Pn.13 0: Pn.13 is 0 1: Pn.13 is 1
12	DATA12	R	0	Data In of Pn.12 0: Pn.12 is 0 1: Pn.12 is 1
11	DATA11	R	0	Data In of Pn.11 0: Pn.11 is 0 1: Pn.11 is 1
10	DATA10	R	0	Data In of Pn.10 0: Pn.10 is 0 1: Pn.10 is 1
9	DATA9	R	0	Data In of Pn.9 0: Pn.9 is 0 1: Pn.9 is 1
8	DATA8	R	0	Data In of Pn.8 0: Pn.8 is 0 1: Pn.8 is 1
7	DATA7	R	0	Data In of Pn.7 0: Pn.7 is 0 1: Pn.7 is 1
6	DATA6	R	0	Data In of Pn.6 0: Pn.6 is 0 1: Pn.6 is 1
5	DATA5	R	0	Data In of Pn.5 0: Pn.5 is 0 1: Pn.5 is 1
4	DATA4	R	0	Data In of Pn.4 0: Pn.4 is 0 1: Pn.4 is 1
3	DATA3	R	0	Data In of Pn.3 0: Pn.3 is 0 1: Pn.3 is 1
2	DATA2	R	0	Data In of Pn.2 0: Pn.2 is 0 1: Pn.2 is 1
1	DATA1	R	0	Data In of Pn.1 0: Pn.1 is 0 1: Pn.1 is 1
0	DATA0	R	0	Data In of Pn.0 0: Pn.0 is 0 1: Pn.0 is 1

5.6.3 GPIO Port n Direction register (GPIOn_DIR) (n=0,1,2,3)

Address offset: 0x08

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

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	DIR20	RW	0	Direction of Pn.20

Bit	Field	Access	Initial	Description
				0: Input 1: Output
19:16	–	–	0	Reserved
15	DIR15	RW	0	Direction of Pn.15 0: Input 1: Output
14	DIR14	RW	0	Direction of Pn.14 0: Input 1: Output
13	DIR13	RW	0	Direction of Pn.13 0: Input 1: Output
12	DIR12	RW	0	Direction of Pn.12 0: Input 1: Output
11	DIR11	RW	0	Direction of Pn.11 0: Input 1: Output
10	DIR10	RW	0	Direction of Pn.10 0: Input 1: Output
9	DIR9	RW	0	Direction of Pn.9 0: Input 1: Output
8	DIR8	RW	0	Direction of Pn.8 0: Input 1: Output
7	DIR7	RW	0	Direction of Pn.7 0: Input 1: Output
6	DIR6	RW	0	Direction of Pn.6 0: Input 1: Output
5	DIR5	RW	0	Direction of Pn.5 0: Input 1: Output
4	DIR4	RW	0	Direction of Pn.4 0: Input 1: Output
3	DIR3	RW	0	Direction of Pn.3 0: Input 1: Output
2	DIR2	RW	0	Direction of Pn.2 0: Input 1: Output
1	DIR1	RW	0	Direction of Pn.1 0: Input 1: Output
0	DIR0	RW	0	Direction of Pn.0 0: Input 1: Output

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

Address offset: 0x10

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	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	BSET20	W	0	Set Pn.20

Bit	Field	Access	Initial	Description
				0: No effect 1: Set Pn.20 to 1
19:16	–	–	0	Reserved
15	BSET15	W	0	Set Pn.15 0: No effect 1: Set Pn.15 to 1
14	BSET14	W	0	Set Pn.14 0: No effect 1: Set Pn.14 to 1
13	BSET13	W	0	Set Pn.13 0: No effect 1: Set Pn.13 to 1
12	BSET12	W	0	Set Pn.12 0: No effect 1: Set Pn.12 to 1
11	BSET11	W	0	Set Pn.11 0: No effect 1: Set Pn.11 to 1
10	BSET10	W	0	Set Pn.10 0: No effect 1: Set Pn.10 to 1
9	BSET9	W	0	Set Pn.9 0: No effect 1: Set Pn.9 to 1
8	BSET8	W	0	Set Pn.8 0: No effect 1: Set Pn.8 to 1
7	BSET7	W	0	Set Pn.7 0: No effect 1: Set Pn.7 to 1
6	BSET6	W	0	Set Pn.6 0: No effect 1: Set Pn.6 to 1
5	BSET5	W	0	Set Pn.5 0: No effect 1: Set Pn.5 to 1
4	BSET4	W	0	Set Pn.4 0: No effect 1: Set Pn.4 to 1
3	BSET3	W	0	Set Pn.3 0: No effect 1: Set Pn.3 to 1
2	BSET2	W	0	Set Pn.2 0: No effect 1: Set Pn.2 to 1
1	BSET1	W	0	Set Pn.1 0: No effect 1: Set Pn.1 to 1
0	BSET0	W	0	Set Pn.0 0: No effect 1: Set Pn.0 to 1

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

Address offset: 0x14

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	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	BCLR20	W	0	Clear Pn.20 0: No effect

Bit	Field	Access	Initial	Description
				1: Clear Pn.19
19:16	–	–	0	Reserved
15	BCLR15	W	0	Clear Pn.15 0: No effect 1: Clear Pn.15
14	BCLR14	W	0	Clear Pn.14 0: No effect 1: Clear Pn.14
13	BCLR13	W	0	Clear Pn.13 0: No effect 1: Clear Pn.13
12	BCLR12	W	0	Clear Pn.12 0: No effect 1: Clear Pn.12
11	BCLR11	W	0	Clear Pn.11 0: No effect 1: Clear Pn.11
10	BCLR10	W	0	Clear Pn.10 0: No effect 1: Clear Pn.10
9	BCLR9	W	0	Clear Pn.9 0: No effect 1: Clear Pn.9
8	BCLR8	W	0	Clear Pn.8 0: No effect 1: Clear Pn.8
7	BCLR7	W	0	Clear Pn.7 0: No effect 1: Clear Pn.7
6	BCLR6	W	0	Clear Pn.6 0: No effect 1: Clear Pn.6
5	BCLR5	W	0	Clear Pn.5 0: No effect 1: Clear Pn.5
4	BCLR4	W	0	Clear Pn.4 0: No effect 1: Clear Pn.4
3	BCLR3	W	0	Clear Pn.3 0: No effect 1: Clear Pn.3
2	BCLR2	W	0	Clear Pn.2 0: No effect 1: Clear Pn.2
1	BCLR1	W	0	Clear Pn.1 0: No effect 1: Clear Pn.1
0	BCLR0	W	0	Clear Pn.0 0: No effect 1: Clear Pn.0

5.6.6 GPIO Port n Pull Enable register (GPIOn_PULLEN) (n=0,1,2,3)

Address offset: 0x18

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	PE20	RW	0	Pn.20 Pull function enable bit 0: Disable 1: Enable
19:16	–	–	0	Reserved
15	PE15	RW	0	Pn.15 Pull function enable bit

Bit	Field	Access	Initial	Description
				0: Disable 1: Enable
14	PE14	RW	0	Pn.14 Pull function enable bit 0: Disable 1: Enable
13	PE13	RW	0	Pn.13 Pull function enable bit 0: Disable 1: Enable
12	PE12	RW	0	Pn.12 Pull function enable bit 0: Disable 1: Enable
11	PE11	RW	0	Pn.11 Pull function enable bit 0: Disable 1: Enable
10	PE10	RW	0	Pn.10 Pull function enable bit 0: Disable 1: Enable
9	PE9	RW	0	Pn.9 Pull function enable bit 0: Disable 1: Enable
8	PE8	RW	0	Pn.8 Pull function enable bit 0: Disable 1: Enable
7	PE7	RW	0	Pn.7 Pull function enable bit 0: Disable 1: Enable
6	PE6	RW	0	Pn.6 Pull function enable bit 0: Disable 1: Enable
5	PE5	RW	0	Pn.5 Pull function enable bit 0: Disable 1: Enable
4	PE4	RW	0	Pn.4 Pull function enable bit 0: Disable 1: Enable
3	PE3	RW	0	Pn.3 Pull function enable bit 0: Disable 1: Enable
2	PE2	RW	0	Pn.2 Pull function enable bit 0: Disable 1: Enable
1	PE1	RW	0	Pn.1 Pull function enable bit 0: Disable 1: Enable
0	PE0	RW	0	Pn.0 Pull function enable bit 0: Disable 1: Enable

5.6.7 GPIO Port n Pull Type register (GPIOn_PUTYPE) (n=0,1,2,3)

Address offset: 0x1C

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	TYPE20	RW	0	Pull Type of Pn.20 0: Pull down 1: Pull up
19:16	–	–	0	Reserved
15	TYPE15	RW	0	Pull Type of Pn.15 0: Pull down 1: Pull up
14	TYPE14	RW	0	Pull Type of Pn.14 0: Pull down

Bit	Field	Access	Initial	Description
				1: Pull up
13	TYPE13	RW	0	Pull Type of Pn.13 0: Pull down 1: Pull up
12	TYPE12	RW	0	Pull Type of Pn.12 0: Pull down 1: Pull up
11	TYPE11	RW	0	Pull Type of Pn.11 0: Pull down 1: Pull up
10	TYPE10	RW	0	Pull Type of Pn.10 0: Pull down 1: Pull up
9	TYPE9	RW	0	Pull Type of Pn.9 0: Pull down 1: Pull up
8	TYPE8	RW	0	Pull Type of Pn.8 0: Pull down 1: Pull up
7	TYPE7	RW	0	Pull Type of Pn.7 0: Pull down 1: Pull up
6	TYPE6	RW	0	Pull Type of Pn.6 0: Pull down 1: Pull up
5	TYPE5	RW	0	Pull Type of Pn.5 0: Pull down 1: Pull up
4	TYPE4	RW	0	Pull Type of Pn.4 0: Pull down 1: Pull up
3	TYPE3	RW	0	Pull Type of Pn.3 0: Pull down 1: Pull up
2	TYPE2	RW	0	Pull Type of Pn.2 0: Pull down 1: Pull up
1	TYPE1	RW	0	Pull Type of Pn.1 0: Pull down 1: Pull up
0	TYPE0	RW	0	Pull Type of Pn.0 0: Pull down 1: Pull up

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

Address offset: 0x20

This register controls the enable or disable trigger detection logics. It is a mask of the trigger detection logic. When the pin direction is the input and the trigger detection is enabled, the pin can accept the trigger from pad. The sensed state is stored in the RIS_R or RIS_F register (Masked by IE). Before turning on IE, programmers can clear the flag state by writing a '1' to IC to ensure the initial state.

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	IE20	RW	0	Interrupt flag on Pn.20 enable 0: Disable 1: Enable
19:16	–	–	0	Reserved
15	IE15	RW	0	Interrupt flag on Pn.15 enable 0: Disable 1: Enable

Bit	Field	Access	Initial	Description
14	IE14	RW	0	Interrupt flag on Pn.14 enable 0: Disable 1: Enable
13	IE13	RW	0	Interrupt flag on Pn.13 enable 0: Disable 1: Enable
12	IE12	RW	0	Interrupt flag on Pn.11 enable 0: Disable 1: Enable
11	IE11	RW	0	Interrupt flag on Pn.11 enable 0: Disable 1: Enable
10	IE10	RW	0	Interrupt flag on Pn.10 enable 0: Disable 1: Enable
9	IE9	RW	0	Interrupt flag on Pn.9 enable 0: Disable 1: Enable
8	IE8	RW	0	Interrupt flag on Pn.8 enable 0: Disable 1: Enable
7	IE7	RW	0	Interrupt flag on Pn.7 enable 0: Disable 1: Enable
6	IE6	RW	0	Interrupt flag on Pn.6 enable 0: Disable 1: Enable
5	IE5	RW	0	Interrupt flag on Pn.5 enable 0: Disable 1: Enable
4	IE4	RW	0	Interrupt flag on Pn.4 enable 0: Disable 1: Enable
3	IE3	RW	0	Interrupt flag on Pn.3 enable 0: Disable 1: Enable
2	IE2	RW	0	Interrupt flag on Pn.2 enable 0: Disable 1: Enable
1	IE1	RW	0	Interrupt flag on Pn.1 enable 0: Disable 1: Enable
0	IE0	RW	0	Interrupt flag on Pn.0 enable 0: Disable 1: Enable 0

5.6.9 GPIO Port n Interrupt Mask Enable register (GPIOn_IMASKEN) (n=0,1,2,3)

Address offset: 0x2C

This register is the mask register of the interrupt detection. It masks the RIS register. For example, if IE[0] = ‘1’, RIS[0] = ‘1’, and IMASKEN[0] = ‘1’, IMASKST [0] will never be changed to ‘1’.

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	IM20E	RW	0	Interrupt Mask enable bit of Pn.20 0: Disable 1: Enable
19:16	–	–	0	Reserved
15	IM15E	RW	0	Interrupt Mask enable bit of Pn.15 0: Disable 1: Enable
14	IM14E	RW	0	Interrupt Mask enable bit of Pn.14

Bit	Field	Access	Initial	Description
				0: Disable 1: Enable
13	IM13E	RW	0	Interrupt Mask enable bit of Pn.13 0: Disable 1: Enable
12	IM12E	RW	0	Interrupt Mask enable bit of Pn.12 0: Disable 1: Enable
11	IM11E	RW	0	Interrupt Mask enable bit of Pn.11 0: Disable 1: Enable
10	IM10E	RW	0	Interrupt Mask enable bit of Pn.10 0: Disable 1: Enable
9	IM9E	RW	0	Interrupt Mask enable bit of Pn.9 0: Disable 1: Enable
8	IM8E	RW	0	Interrupt Mask enable bit of Pn.8 0: Disable 1: Enable
7	IM7E	RW	0	Interrupt Mask enable bit of Pn.7 0: Disable 1: Enable
6	IM6E	RW	0	Interrupt Mask enable bit of Pn.6 0: Disable 1: Enable
5	IM5E	RW	0	Interrupt Mask enable bit of Pn.5 0: Disable 1: Enable
4	IM4E	RW	0	Interrupt Mask enable bit of Pn.4 0: Disable 1: Enable
3	IM3E	RW	0	Interrupt Mask enable bit of Pn.3 0: Disable 1: Enable
2	IM2E	RW	0	Interrupt Mask enable bit of Pn.2 0: Disable 1: Enable
1	IM1E	RW	0	Interrupt Mask enable bit of Pn.1 0: Disable 1: Enable
0	IM0E	RW	0	Interrupt Mask enable bit of Pn.0 0: Disable 1: Enable

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

Address offset: 0x30

✸ ***Note: GPIOn_IC can clear GPIOn_RIS_R/GPIOn_RIS_F/GPIOn_RIS_RF/GPIOn_IMASKST_RF status.***

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	IC20	W	0	Pn.20 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.20
19:16	–	–	0	Reserved
15	IC15	W	0	Pn.15 interrupt flag clear

Bit	Field	Access	Initial	Description
				0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.15
14	IC14	W	0	Pn.14 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.14
13	IC13	W	0	Pn.13 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.13
12	IC12	W	0	Pn.12 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.12
11	IC11	W	0	Pn.11 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.11
10	IC10	W	0	Pn.10 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.10
9	IC9	W	0	Pn.9 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.9
8	IC8	W	0	Pn.8 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.8
7	IC7	W	0	Pn.7 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.7
6	IC6	W	0	Pn.6 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.6
5	IC5	W	0	Pn.5 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.5
4	IC4	W	0	Pn.4 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.4
3	IC3	W	0	Pn.3 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.3
2	IC2	W	0	Pn.2 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.2
1	IC1	W	0	Pn.1 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.1
0	IC0	W	0	Pn.0 interrupt flag clear 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.0

5.6.11 GPIO Port n Interrupt Trigger register (GPIOn_ITRIG) (n=0,1,2,3)

Address offset: 0x34

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	ITRIG20	RW	0	ITRIG of Pn.20 0: Pn.20 interrupt will be triggered at the edge 1: Pn.20 interrupt will be triggered at the level
19:16	–	–	0	Reserved
15	ITRIG15	RW	0	ITRIG of Pn.15 0: Pn.15 interrupt will be triggered at the edge 1: Pn.15 interrupt will be triggered at the level
14	ITRIG14	RW	0	ITRIG of Pn.14

Bit	Field	Access	Initial	Description
				0: Pn.14 interrupt will be triggered at the edge 1: Pn.14 interrupt will be triggered at the level
13	ITRIG13	RW	0	ITRIG of Pn.13 0: Pn.13 interrupt will be triggered at the edge 1: Pn.13 interrupt will be triggered at the level
12	ITRIG12	RW	0	ITRIG of Pn.12 0: Pn.12 interrupt will be triggered at the edge 1: Pn.12 interrupt will be triggered at the level
11	ITRIG11	RW	0	ITRIG of Pn.11 0: Pn.11 interrupt will be triggered at the edge 1: Pn.11 interrupt will be triggered at the level
10	ITRIG10	RW	0	ITRIG of Pn.10 0: Pn.10 interrupt will be triggered at the edge 1: Pn.10 interrupt will be triggered at the level
9	ITRIG9	RW	0	ITRIG of Pn.9 0: Pn.9 interrupt will be triggered at the edge 1: Pn.9 interrupt will be triggered at the level
8	ITRIG8	RW	0	ITRIG of Pn.8 0: Pn.8 interrupt will be triggered at the edge 1: Pn.8 interrupt will be triggered at the level
7	ITRIG7	RW	0	ITRIG of Pn.7 0: Pn.7 interrupt will be triggered at the edge 1: Pn.7 interrupt will be triggered at the level
6	ITRIG6	RW	0	ITRIG of Pn.6 0: Pn.6 interrupt will be triggered at the edge 1: Pn.6 interrupt will be triggered at the level
5	ITRIG5	RW	0	ITRIG of Pn.5 0: Pn.5 interrupt will be triggered at the edge 1: Pn.5 interrupt will be triggered at the level
4	ITRIG4	RW	0	ITRIG of Pn.4 0: Pn.4 interrupt will be triggered at the edge 1: Pn.4 interrupt will be triggered at the level
3	ITRIG3	RW	0	ITRIG of Pn.3 0: Pn.3 interrupt will be triggered at the edge 1: Pn.3 interrupt will be triggered at the level
2	ITRIG2	RW	0	ITRIG of Pn.2 0: Pn.2 interrupt will be triggered at the edge 1: Pn.2 interrupt will be triggered at the level
1	ITRIG1	RW	0	ITRIG of Pn.1 0: Pn.1 interrupt will be triggered at the edge 1: Pn.1 interrupt will be triggered at the level
0	ITRIG0	RW	0	ITRIG of Pn.0 0: Pn.0 interrupt will be triggered at the edge 1: Pn.0 interrupt will be triggered at the level

5.6.12 GPIO Port n Interrupt Both Edge Trigger register (GPIOn_IBETRIG) (n=0,1,2,3)

Address offset: 0x38

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	IBE20	RW	0	Interrupt Both Edge Trigger of Pn.20 0: Pn.20 interrupt is triggered by the single edge 1: Pn.20 interrupt is triggered by both edges
19:16	–	–	0	Reserved
15	IBE15	RW	0	Interrupt Both Edge Trigger of Pn.15 0: Pn.15 interrupt is triggered by the single edge 1: Pn.15 interrupt is triggered by both edges
14	IBE14	RW	0	Interrupt Both Edge Trigger of Pn.14 0: Pn.14 interrupt is triggered by the single edge 1: Pn.14 interrupt is triggered by both edges

Bit	Field	Access	Initial	Description
13	IBE13	RW	0	Interrupt Both Edge Trigger of Pn.13 0: Pn.13 interrupt is triggered by the single edge 1: Pn.13 interrupt is triggered by both edges
12	IBE12	RW	0	Interrupt Both Edge Trigger of Pn.12 0: Pn.12 interrupt is triggered by the single edge 1: Pn.12 interrupt is triggered by both edges
11	IBE11	RW	0	Interrupt Both Edge Trigger of Pn.11 0: Pn.11 interrupt is triggered by the single edge 1: Pn.11 interrupt is triggered by both edges
10	IBE10	RW	0	Interrupt Both Edge Trigger of Pn.10 0: Pn.10 interrupt is triggered by the single edge 1: Pn.10 interrupt is triggered by both edges
9	IBE9	RW	0	Interrupt Both Edge Trigger of Pn.9 0: Pn.9 interrupt is triggered by the single edge 1: Pn.9 interrupt is triggered by both edges
8	IBE8	RW	0	Interrupt Both Edge Trigger of Pn.8 0: Pn.8 interrupt is triggered by the single edge 1: Pn.8 interrupt is triggered by both edges
7	IBE7	RW	0	Interrupt Both Edge Trigger of Pn.7 0: Pn.7 interrupt is triggered by the single edge 1: Pn.7 interrupt is triggered by both edges
6	IBE6	RW	0	Interrupt Both Edge Trigger of Pn.6 0: Pn.6 interrupt is triggered by the single edge 1: Pn.6 interrupt is triggered by both edges
5	IBE5	RW	0	Interrupt Both Edge Trigger of Pn.5 0: Pn.5 interrupt is triggered by the single edge 1: Pn.5 interrupt is triggered by both edges
4	IBE4	RW	0	Interrupt Both Edge Trigger of Pn.4 0: Pn.4 interrupt is triggered by the single edge 1: Pn.4 interrupt is triggered by both edges
3	IBE3	RW	0	Interrupt Both Edge Trigger of Pn.3 0: Pn.3 interrupt is triggered by the single edge 1: Pn.3 interrupt is triggered by both edges
2	IBE2	RW	0	Interrupt Both Edge Trigger of Pn.2 0: Pn.2 interrupt is triggered by the single edge 1: Pn.2 interrupt is triggered by both edges
1	IBE1	RW	0	Interrupt Both Edge Trigger of Pn.1 0: Pn.1 interrupt is triggered by the single edge 1: Pn.1 interrupt is triggered by both edges
0	IBE0	RW	0	Interrupt Both Edge Trigger of Pn.0 0: Pn.0 interrupt is triggered by the single edge 1: Pn.0 interrupt is triggered by both edges

5.6.13 GPIO Port n Interrupt Rising or Falling Edge Trigger register (GPIOn_IRFTRIG) (n=0,1,2,3)

Address offset: 0x3C

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	IRF20	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.20 0: Pn.20 interrupt is triggered at the rising edge/high level 1: Pn.20 interrupt is triggered at the falling edge/low level
19:16	–	–	0	Reserved
15	IRF15	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.15 0: Pn.15 interrupt is triggered at the rising edge/high level 1: Pn.15 interrupt is triggered at the falling edge/low level
14	IRF14	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.14 0: Pn.14 interrupt is triggered at the rising edge/high level 1: Pn.14 interrupt is triggered at the falling edge/low level
13	IRF13	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.13 0: Pn.13 interrupt is triggered at the rising edge/high level

Bit	Field	Access	Initial	Description
				1: Pn.13 interrupt is triggered at the falling edge/low level
12	IRF12	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.12 0: Pn.12 interrupt is triggered at the rising edge/high level 1: Pn.12 interrupt is triggered at the falling edge/low level
11	IRF11	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.11 0: Pn.11 interrupt is triggered at the rising edge/high level 1: Pn.11 interrupt is triggered at the falling edge/low level
10	IRF10	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.10 0: Pn.10 interrupt is triggered at the rising edge/high level 1: Pn.10 interrupt is triggered at the falling edge/low level
9	IRF9	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.9 0: Pn.9 interrupt is triggered at the rising edge/high level 1: Pn.9 interrupt is triggered at the falling edge/low level
8	IRF8	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.8 0: Pn.8 interrupt is triggered at the rising edge/high level 1: Pn.8 interrupt is triggered at the falling edge/low level
7	IRF7	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.7 0: Pn.7 interrupt is triggered at the rising edge/high level 1: Pn.7 interrupt is triggered at the falling edge/low level
6	IRF6	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.6 0: Pn.6 interrupt is triggered at the rising edge/high level 1: Pn.6 interrupt is triggered at the falling edge/low level
5	IRF5	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.5 0: Pn.5 interrupt is triggered at the rising edge/high level 1: Pn.5 interrupt is triggered at the falling edge/low level
4	IRF4	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.4 0: Pn.4 interrupt is triggered at the rising edge/high level 1: Pn.4 interrupt is triggered at the falling edge/low level
3	IRF3	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.3 0: Pn.3 interrupt is triggered at the rising edge/high level 1: Pn.3 interrupt is triggered at the falling edge/low level
2	IRF2	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.2 0: Pn.2 interrupt is triggered at the rising edge/high level 1: Pn.2 interrupt is triggered at the falling edge/low level
1	IRF1	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.1 0: Pn.1 interrupt is triggered at the rising edge/high level 1: Pn.1 interrupt is triggered at the falling edge/low level
0	IRF0	RW	0	Interrupt Rising or Falling Edge Trigger selection of Pn.0 0: Pn.0 interrupt is triggered at the rising edge/high level 1: Pn.0 interrupt is triggered at the falling edge/low level

5.6.14 GPIO Port n Bounce Enable register (GPIOn_BNEN) (n=0,1,2,3)

Address offset: 0x40

This register controls the de-bounce function. If BNEN is enabled, the trigger detection will be sampled by the bounce clock.

The general-purpose input/output controller is capable of filtering out the external bounce which is sampled less than two continuous PCLK (or de-bounce clock) rising edges when the trigger function is enabled. It is recommended that the duration of the trigger signal is longer than two clock periods (or de-bounce clock period) to guarantee that the general-purpose input/output controller will sample the trigger signal by two rising edges of clock.

✸ ***Note: GPIO de-bounce function is only applicable to Normal mode (Non-Clock mode is invalid).***

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	BN20EN	RW	0	Bounce enable bit of Pn.20

Bit	Field	Access	Initial	Description
				0: Disable 1: Enable
19:16	–	–	0	Reserved
15	BN15EN	RW	0	Bounce enable bit of Pn.15 0: Disable 1: Enable
14	BN14EN	RW	0	Bounce enable bit of Pn.14 0: Disable 1: Enable
13	BN13EN	RW	0	Bounce enable bit of Pn.13 0: Disable 1: Enable
12	BN12EN	RW	0	Bounce enable bit of Pn.12 0: Disable 1: Enable
11	BN11EN	RW	0	Bounce enable bit of Pn.11 0: Disable 1: Enable
10	BN10EN	RW	0	Bounce enable bit of Pn.10 0: Disable 1: Enable
9	BN9EN	RW	0	Bounce enable bit of Pn.9 0: Disable 1: Enable
8	BN8EN	RW	0	Bounce enable bit of Pn.8 0: Disable 1: Enable
7	BN7EN	RW	0	Bounce enable bit of Pn.7 0: Disable 1: Enable
6	BN6EN	RW	0	Bounce enable bit of Pn.6 0: Disable 1: Enable
5	BN5EN	RW	0	Bounce enable bit of Pn.5 0: Disable 1: Enable
4	BN4EN	RW	0	Bounce enable bit of Pn.4 0: Disable 1: Enable
3	BN3EN	RW	0	Bounce enable bit of Pn.3 0: Disable 1: Enable
2	BN2EN	RW	0	Bounce enable bit of Pn.2 0: Disable 1: Enable
1	BN1EN	RW	0	Bounce enable bit of Pn.1 0: Disable 1: Enable
0	BN0EN	RW	0	Bounce enable bit of Pn.0 0: Disable 1: Enable

5.6.15 GPIO Port n Bounce Clock Prescaler register (GPIOn_BNCLKPRE) (n=0,1,2,3)

Address offset: 0x44

The BNCLKPRE register is an automatic register to indicate for the bounce timer. It can extend PCLK to the PRE cycles, which can be used to adjust the trigger sample clock period in different machines. The reset value is 0x7D0, which means that if the APB clock frequency is 66 MHz, the de-bounce clock will be divided by (0x7D0 + 1) to 32.98 kHz. Programmers can adjust this register to fit different systems.

Bit	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23:0	PRE	RW	0x7D0	Bounce clock pre-scaler value

5.6.16 GPIO Port n Clock Mode register (GPIOn_CLKMODE) (n=0,1,2,3)

Address offset: 0x48

Bit	Field	Access	Initial	Description
31:1	–	–	0	Reserved
0	MODE	RW	0	GPIO clock mode 0: Normal mode 1: Non-clock mode

5.6.17 GPIO Port n Interrupt Raw State of Rising Edge register (GPIOn_RIS_R) (n=0,1,2,3)

Address offset: 0x4C

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	IF20	R	0	[In Non-clock mode] Pn.20 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.20 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.20 1: Interrupt requirements met on Pn.20
19:16	–	–	0	Reserved
15	IF15	R	0	[In Non-clock mode] Pn.15 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.15 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.15 1: Interrupt requirements met on Pn.15
14	IF14	R	0	[In Non-clock mode] Pn.14 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.14 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.14 1: Interrupt requirements met on Pn.14
13	IF13	R	0	[In Non-clock mode] Pn.13 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.13 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.13 1: Interrupt requirements met on Pn.13
12	IF12	R	0	[In Non-clock mode] Pn.12 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.12 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.12 1: Interrupt requirements met on Pn.12
11	IF11	R	0	[In Non-clock mode] Pn.11 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.11 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.11 1: Interrupt requirements met on Pn.11
10	IF10	R	0	[In Non-clock mode] Pn.10 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.10 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.10

				1: Interrupt requirements met on Pn.10
9	IF9	R	0	[In Non-clock mode] Pn.9 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.9 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.9 1: Interrupt requirements met on Pn.9
8	IF8	R	0	[In Non-clock mode] Pn.8 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.8 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.8 1: Interrupt requirements met on Pn.8
7	IF7	R	0	[In Non-clock mode] Pn.7 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.7 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.7 1: Interrupt requirements met on Pn.7
6	IF6	R	0	[In Non-clock mode] Pn.6 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.6 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.6 1: Interrupt requirements met on Pn.6
5	IF5	R	0	[In Non-clock mode] Pn.5 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.5 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.5 1: Interrupt requirements met on Pn.5
4	IF4	R	0	[In Non-clock mode] Pn.4 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.4 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.4 1: Interrupt requirements met on Pn.4
3	IF3	R	0	[In Non-clock mode] Pn.3 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.3 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.3 1: Interrupt requirements met on Pn.3
2	IF2	R	0	[In Non-clock mode] Pn.2 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.2 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.2 1: Interrupt requirements met on Pn.2
1	IF1	R	0	[In Non-clock mode] Pn.1 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.1 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.1 1: Interrupt requirements met on Pn.1
0	IF0	R	0	[In Non-clock mode] Pn.0 rising edge raw interrupt flag in non-clock mode [In Normal mode] Pn.0 rising edge or high level raw interrupt flag in normal mode 0: No interrupt on Pn.0 1: Interrupt requirements met on Pn.0

5.6.18 GPIO Port n Interrupt Raw State of Falling Edge register (GPIOn_RIS_F) (n=0,1,2,3)

Address offset: 0x50

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	IF20	R	0	[In Non-clock mode] Pn.20 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.20 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.20 1: Interrupt requirements met on Pn.20
19:16	–	–	0	Reserved
15	IF15	R	0	[In Non-clock mode] Pn.15 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.15 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.15 1: Interrupt requirements met on Pn.15
14	IF14	R	0	[In Non-clock mode] Pn.14 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.14 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.14 1: Interrupt requirements met on Pn.14
13	IF13	R	0	[In Non-clock mode] Pn.13 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.13 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.13 1: Interrupt requirements met on Pn.13
12	IF12	R	0	[In Non-clock mode] Pn.12 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.12 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.12 1: Interrupt requirements met on Pn.12
11	IF11	R	0	[In Non-clock mode] Pn.11 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.11 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.11 1: Interrupt requirements met on Pn.11
10	IF10	R	0	[In Non-clock mode] Pn.10 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.10 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.10 1: Interrupt requirements met on Pn.10
9	IF9	R	0	[In Non-clock mode] Pn.9 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.9 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.9 1: Interrupt requirements met on Pn.9
8	IF8	R	0	[In Non-clock mode] Pn.8 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.8 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.8 1: Interrupt requirements met on Pn.8
7	IF7	R	0	[In Non-clock mode] Pn.7 falling edge raw interrupt flag in non-clock mode [In Normal mode]

Bit	Field	Access	Initial	Description
				Pn.7 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.7 1: Interrupt requirements met on Pn.7
6	IF6	R	0	[In Non-clock mode] Pn.6 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.6 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.6 1: Interrupt requirements met on Pn.6
5	IF5	R	0	[In Non-clock mode] Pn.5 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.5 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.5 1: Interrupt requirements met on Pn.5
4	IF4	R	0	[In Non-clock mode] Pn.4 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.4 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.4 1: Interrupt requirements met on Pn.4
3	IF3	R	0	[In Non-clock mode] Pn.3 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.3 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.3 1: Interrupt requirements met on Pn.3
2	IF2	R	0	[In Non-clock mode] Pn.2 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.2 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.2 1: Interrupt requirements met on Pn.2
1	IF1	R	0	[In Non-clock mode] Pn.1 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.1 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.1 1: Interrupt requirements met on Pn.1
0	IF0	R	0	[In Non-clock mode] Pn.0 falling edge raw interrupt flag in non-clock mode [In Normal mode] Pn.0 falling edge or low level raw interrupt flag in normal mode 0: No interrupt on Pn.0 1: Interrupt requirements met on Pn.0

5.6.19 GPIO Port n Interrupt Raw State of Both Edge register (GPIOn_RIS_RF) (n=0,1,2,3)

Address offset: 0x54

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	IF20	R	0	[In Non-clock mode] Pn.20 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.20 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.20 1: Interrupt requirements met on Pn.20
19:16	–	–	0	Reserved
15	IF15	R	0	[In Non-clock mode] Pn.15 both edge raw interrupt flag in non-clock mode [In Normal mode]

Bit	Field	Access	Initial	Description
				Pn.15 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.15 1: Interrupt requirements met on Pn.15
14	IF14	R	0	[In Non-clock mode] Pn.14 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.14 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.14 1: Interrupt requirements met on Pn.14
13	IF13	R	0	[In Non-clock mode] Pn.13 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.13 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.13 1: Interrupt requirements met on Pn.13
12	IF12	R	0	[In Non-clock mode] Pn.12 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.12 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.12 1: Interrupt requirements met on Pn.12
11	IF11	R	0	[In Non-clock mode] Pn.11 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.11 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.11 1: Interrupt requirements met on Pn.11
10	IF10	R	0	[In Non-clock mode] Pn.10 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.10 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.10 1: Interrupt requirements met on Pn.10
9	IF9	R	0	[In Non-clock mode] Pn.9 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.9 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.9 1: Interrupt requirements met on Pn.9
8	IF8	R	0	[In Non-clock mode] Pn.8 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.8 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.8 1: Interrupt requirements met on Pn.8
7	IF7	R	0	[In Non-clock mode] Pn.7 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.7 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.7 1: Interrupt requirements met on Pn.7
6	IF6	R	0	[In Non-clock mode] Pn.6 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.6 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.6 1: Interrupt requirements met on Pn.6
5	IF5	R	0	[In Non-clock mode] Pn.5 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.5 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.5 1: Interrupt requirements met on Pn.5
4	IF4	R	0	[In Non-clock mode] Pn.4 both edge raw interrupt flag in non-clock mode

Bit	Field	Access	Initial	Description
				[In Normal mode] Pn.4 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.4 1: Interrupt requirements met on Pn.4
3	IF3	R	0	[In Non-clock mode] Pn.3 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.3 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.3 1: Interrupt requirements met on Pn.3
2	IF2	R	0	[In Non-clock mode] Pn.2 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.2 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.2 1: Interrupt requirements met on Pn.2
1	IF1	R	0	[In Non-clock mode] Pn.1 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.1 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.1 1: Interrupt requirements met on Pn.1
0	IF0	R	0	[In Non-clock mode] Pn.0 both edge raw interrupt flag in non-clock mode [In Normal mode] Pn.0 both edge or H/L level raw interrupt flag in normal mode 0: No interrupt on Pn.0 1: Interrupt requirements met on Pn.0

5.6.20 GPIO Port n Interrupt Clear of Rising Edge register (GPIOn_IC_R) (n=0,1,2,3)

Address offset: 0x58

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	IC20	W	0	Pn.20 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.20
19:16	–	–	0	Reserved
15	IC15	W	0	Pn.15 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.15
14	IC14	W	0	Pn.14 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.14
13	IC13	W	0	Pn.13 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.13
12	IC12	W	0	Pn.12 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.12
11	IC11	W	0	Pn.11 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.11
10	IC10	W	0	Pn.10 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.10
9	IC9	W	0	Pn.9 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.9
8	IC8	W	0	Pn.8 rising edge interrupt flag clear in non-clock mode

Bit	Field	Access	Initial	Description
				0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.8
7	IC7	W	0	Pn.7 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.7
6	IC6	W	0	Pn.6 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.6
5	IC5	W	0	Pn.5 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.5
4	IC4	W	0	Pn.4 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.4
3	IC3	W	0	Pn.3 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.3
2	IC2	W	0	Pn.2 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.2
1	IC1	W	0	Pn.1 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.1
0	IC0	W	0	Pn.0 rising edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.0

5.6.21 GPIO Port n Interrupt Clear of Falling Edge register (GPIOn_IC_F) (n=0,1,2,3)

Address offset: 0x5C

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	IC20	W	0	Pn.20 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.20
19:16	–	–	0	Reserved
15	IC15	W	0	Pn.15 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.15
14	IC14	W	0	Pn.14 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.14
13	IC13	W	0	Pn.13 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.13
12	IC12	W	0	Pn.12 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.12
11	IC11	W	0	Pn.11 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.11
10	IC10	W	0	Pn.10 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.10
9	IC9	W	0	Pn.9 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.9
8	IC8	W	0	Pn.8 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.8

Bit	Field	Access	Initial	Description
7	IC7	W	0	Pn.7 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.7
6	IC6	W	0	Pn.6 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.6
5	IC5	W	0	Pn.5 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.5
4	IC4	W	0	Pn.4 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.4
3	IC3	W	0	Pn.3 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.3
2	IC2	W	0	Pn.2 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.2
1	IC1	W	0	Pn.1 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.1
0	IC0	W	0	Pn.0 falling edge interrupt flag clear in non-clock mode 0: No effect 1: Clear interrupt flag and interrupt masked state of Pn.0

5.6.22 GPIO Port n Interrupt Masked State register (GPIOn_IMASKST_RF) (n=0,1,2,3)

Address offset: 0x60

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	IMS20	R	0	Interrupt Masked State of Pn.20 in non-clock mode 0: Pn.20 interrupt is not detected or is masked 1: Pn.20 interrupt is detected and not masked
19:16	–	–	0	Reserved
15	IMS15	R	0	Interrupt Masked State of Pn.15 in non-clock mode 0: Pn.15 interrupt is not detected or is masked 1: Pn.15 interrupt is detected and not masked
14	IMS14	R	0	Interrupt Masked State of Pn.14 in non-clock mode 0: Pn.14 interrupt is not detected or is masked 1: Pn.14 interrupt is detected and not masked
13	IMS13	R	0	Interrupt Masked State of Pn.13 in non-clock mode 0: Pn.13 interrupt is not detected or is masked 1: Pn.13 interrupt is detected and not masked
12	IMS12	R	0	Interrupt Masked State of Pn.12 in non-clock mode 0: Pn.12 interrupt is not detected or is masked 1: Pn.12 interrupt is detected and not masked
11	IMS11	R	0	Interrupt Masked State of Pn.11 in non-clock mode 0: Pn.11 interrupt is not detected or is masked 1: Pn.11 interrupt is detected and not masked
10	IMS10	R	0	Interrupt Masked State of Pn.10 in non-clock mode 0: Pn.10 interrupt is not detected or is masked 1: Pn.10 interrupt is detected and not masked
9	IMS9	R	0	Interrupt Masked State of Pn.9 in non-clock mode 0: Pn.9 interrupt is not detected or is masked 1: Pn.9 interrupt is detected and not masked
8	IMS8	R	0	Interrupt Masked State of Pn.8 in non-clock mode 0: Pn.8 interrupt is not detected or is masked 1: Pn.8 interrupt is detected and not masked
7	IMS7	R	0	Interrupt Masked State of Pn.7 in non-clock mode 0: Pn.7 interrupt is not detected or is masked

Bit	Field	Access	Initial	Description
				1: Pn.7 interrupt is detected and not masked
6	IMS6	R	0	Interrupt Masked State of Pn.6 in non-clock mode 0: Pn.6 interrupt is not detected or is masked 1: Pn.6 interrupt is detected and not masked
5	IMS5	R	0	Interrupt Masked State of Pn.5 in non-clock mode 0: Pn.5 interrupt is not detected or is masked 1: Pn.5 interrupt is detected and not masked
4	IMS4	R	0	Interrupt Masked State of Pn.4 in non-clock mode 0: Pn.4 interrupt is not detected or is masked 1: Pn.4 interrupt is detected and not masked
3	IMS3	R	0	Interrupt Masked State of Pn.3 in non-clock mode 0: Pn.3 interrupt is not detected or is masked 1: Pn.3 interrupt is detected and not masked
2	IMS2	R	0	Interrupt Masked State of Pn.2 in non-clock mode 0: Pn.2 interrupt is not detected or is masked 1: Pn.2 interrupt is detected and not masked
1	IMS1	R	0	Interrupt Masked State of Pn.1 in non-clock mode 0: Pn.1 interrupt is not detected or is masked 1: Pn.1 interrupt is detected and not masked
0	IMS0	R	0	Interrupt Masked State of Pn.0 in non-clock mode 0: Pn.0 interrupt is not detected or is masked 1: Pn.0 interrupt is detected and not masked

6 GPIO ALTERNATE FUNCTION (AFIO)

6.1 OVERVIEW

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

6.2 FEATURES

- Flexible assignment of peripheral functions to desired pins.
- Supported functions are XTAL, RESET, SWD, SPI, I2C, I2S, UART, CAN, Ethernet, ADC, LCM, SDIO, Capture, and PWM.

6.3 ALTERNATE FUNCTION MAPPING

PIN	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9
P0.0	P0.0	SCK0	MO2	DOUT2			B0_PWM2	B0_PWM3N	B0_PWM1N	B2_PWM3
P0.1	P0.1	UTXD2	URXD2	SEL0	SCK1	SCL1	BCLK1		B0_PWM1	B0_PWM2N
P0.2	P0.2	URXD2	MI0	SDA1	CAN_RX1				B0_PWM0	B5_PWM3
P0.10	P0.10	UTXD0	UTXD4	SCL0	CAN_TX1		B0_BRK	B1_PWM2	B2_PWM0	B3_PWM0N
P0.11	P0.11	URXD0	UCTS3	SDA0			B2_CAP0	B1_PWM3	B2_PWM1	B3_PWM1N
P0.12	P0.12	SCK1		SDA2				B0_PWM3	B0_PWM0N	
P0.13	P0.13	SEL1					B4_CAP0	B0_PWM2		
P0.14	P0.14	URXD5	SCK1	SDA0		BCLK1		B0_PWM1		
P0.15	P0.15	UTXD5		SCL0				B0_PWM0		
P0.20	P0.20					SDA1				

PIN	AF10	AF11	AF12	AF13	AF14	AF15	AF16	AF17	AF18	AF19	AF20
P0.0	B0_PWM0	B2_PWM0	B8_PWM0	LCM_AD0	CAN_RX0		SI2				SDIO_D0
P0.1	B0_PWM3	B2_PWM2	B8_PWM1	LCM_AD1	CAN_TX0	ETH_MII_RX_ER	IRDA_TXD2		IRDA_RXDL2		SDIO_D1
P0.2	B0_PWM3N	B2_PWM3	B8_PWM2	LCM_AD2	ETH_RMII_TX_EN	ETH_MII_TX_EN	SO0		IRDA_RXDL2		SDIO_D2
P0.10	B4_PWM0	B4_PWM1N	B8_PWM5		LCM_CS	**CLKOUT3**	IRDA_TXD0	IRDA_TXD4			
P0.11	B4_PWM1	B4_PWM0N	B8_PWM6						IRDA_RXDL0		
P0.12	B5_PWM3		B8_PWM8		SDIO_D1	**CLKOUT2**					
P0.13	B5_PWM2		B8_PWM7		SDIO_D0						
P0.14	B5_PWM1		B8_PWM6		SDIO_D7				IRDA_RXDL5		
P0.15	B5_PWM0		B8_PWM5		SDIO_D6		IRDA_TXD5				
P0.20			B8_PWM11		LCM_CS						

SWD SPI I2C I2S UART CAN ETHERNET ADC LCM SDIO

PIN	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9
P1.0	P1.0	UTXD4	URXD0	SCK0	SCK2	SDA1	BCLK0	BCLK2	CAN_RX0	B2_PWM1
P1.1	P1.1	URXD4	MI0	MI2	SCL0	SDA2		DIN1	CAN_TX0	B1_PWM0
P1.2	P1.2	URXD4	MO0	MO2	SDA0	DOUT0	DOUT1	DOUT2	CAN_STBY0	CAN_RX1
P1.3	P1.3	URXD2	DIN0						CAN_STBY1	
P1.4	P1.4	UTXD2	URXD4	SCL0				CAN_RX1	B2_PWM0	B2_PWM2
P1.5	P1.5	UTXD4	SEL1	SDA0	SDA1	WS1		CAN_TX1	B2_PWM1	B2_PWM3
P1.6	P1.6	URXD2	URXD3	MI2	BCLK1				B2_PWM2	B5_PWM2
P1.7	P1.7	UTXD2	UTXD3	SEL0	SCK2		BCLK2		B2_PWM1	B5_PWM1
P1.8	P1.8	URXD1	URTS3	UTXD0	SEL0	SCK1	SEL2	SCL1	WS0	WS2
P1.9	P1.9	UTXD1	URXD3	SEL1	SDA1	SCL1	WS1	CAN_STBY0		B5_PWM0
P1.10	P1.10	UTXD4	MI1	MO2	SCL1	SDA1	DIN1	DOUT2		B2_PWM3
P1.11	P1.11	URXD4	URTS2	MO1	SDA1					
P1.12	P1.12	SEL1	SCK2	WS1	BCLK2	CAN_RX0	CAN_RX1	B0_BRK	B0_PWM0	B1_PWM0
P1.13	P1.13	UCTS2	SCK1	SCL1	SCL2	BCLK1	CAN_TX0	CAN_TX1	B0_PWM0N	B1_PWM1
P1.14	P1.14	URTS2	MI1	SDA1	SDA2		CAN_STBY0		B0_PWM1N	B1_PWM2
P1.15	P1.15		MO1	DOUT1	SCL2		B5_CAP0	B3_PWM1	B0_PWM2N	B1_PWM3

PIN	AF10	AF11	AF12	AF13	AF14	AF15	AF16	AF17	AF18	AF19	AF20
P1.0	B5_PWM2		B8_PWM3			SWO	IRDA_TXD4		IRDA_RXDL0		
P1.1	B5_PWM0		B8_PWM4		LCM_CS		SO0	SO2	IRDA_RXDL4		
P1.2	B1_PWM1	B4_PWM0N	B5_PWM1				SI0	SI2	IRDA_RXDL4		
P1.3	B3_PWM0N	B4_PWM1N	B8_PWM7						IRDA_RXDL2		
P1.4	B3_PWM0	B4_PWM1	B8_PWM8		SDIO_D4	ETH_MII_TXD3	IRDA_TXD2		IRDA_RXDL4		
P1.5	B3_PWM1	B4_PWM0	B8_PWM9		SDIO_D5		IRDA_TXD4				
P1.6			B8_PWM0		SDIO_D3		SO2		IRDA_RXDL2		IRDA_RXDL3
P1.7			B8_PWM11		SDIO_D2		IRDA_TXD2	IRDA_TXD3			
P1.8	B2_PWM0		B8_PWM10	LCM_AD3			IRDA_TXD0		IRDA_RXDL1		
P1.9			B8_PWM9				IRDA_TXD1		IRDA_RXDL3		
P1.10			B8_PWM1		SDIO_CK		SO1	SI2	IRDA_TXD4		
P1.11		B8_PWM2	LCM_A0		SDIO_CMD		SI1		IRDA_RXDL4		
P1.12	B3_PWM1	B4_PWM1	B5_PWM0	LCM_AD4	ETH_RMII_TXD0	ETH_MII_TXD0					SDIO_CK
P1.13	B3_PWM1N	B4_PWM1N		LCM_AD5	ETH_RMII_TXD1	ETH_MII_TXD1					SDIO_CMD
P1.14	B3_PWM0	B4_PWM0	B8_PWM3		LCM_AD6		SO1				
P1.15	B3_PWM0N	B4_PWM1	B8_PWM4		LCM_AD7		SI1				

SWD SPI I2C I2S UART CAN ETHERNET ADC LCM SDIO

PIN	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9
P2.0	P2.0	UTXD3	URXD0	UCTS1		SCL1	CAN_STBY1	B2_PWM0		B1_PWM0
P2.1	P2.1	URXD3	UTXD0	UTXD3	URTS1	SDA1	CAN_RX1	B2_PWM1	B3_PWM0N	
P2.2	P2.2	UTXD1					CAN_TX1	B2_PWM2	B3_PWM0	
P2.3	P2.3	URXD1						B2_PWM3	B3_PWM1	
P2.4	P2.4	UTXD5	SEL0	SEL2	WS0	WS2	B0_PWM1N	B1_PWM1	B3_PWM1N	B4_PWM1N
P2.5	P2.5	URXD5	SCK0	BCLK0			B0_PWM1	B0_PWM0N	B2_PWM0	B4_PWM1
P2.6	P2.6	UCTS2	MI0			B8_PWM2	B0_PWM0	B0_PWM2N	B3_PWM0	B4_PWM0N
P2.7	P2.7	MO0	DOUT0				B0_PWM0N	B0_PWM2	B1_PWM2	B4_PWM0
P2.8	P2.8	URXD1	MO2	DIN0		DOUT2		B0_PWM1N	B0_PWM2	
P2.9	P2.9	URTS2	SCK1	BCLK1			B1_PWM3	B0_PWM1	B0_PWM2N	
P2.10	P2.10	UTXD5	SCL2							
P2.11	P2.11	URXD5	MO2	MO1	SDA2	DOUT2	DOUT1			
P2.12	P2.12	MI1							B0_PWM1	
P2.13	P2.13	MO1	DOUT1							
P2.14	P2.14	UTXD2							B0_PWM3	B3_PWM0
P2.15	P2.15	URXD2							B0_PWM3N	B3_PWM1

PIN	AF10	AF11	AF12	AF13	AF14	AF15	AF16	AF17	AF18	AF19	AF20
P2.0		B5_PWM0	ADC0_AIN0	SDIO_CK	ETH_MII_CRS		IRDA_TXD3		IRDA_RXDL0		
P2.1		B5_PWM1	ADC0_AIN1	SDIO_CMD	ETH_RMII_REF_CLK	ETH_MII_RX_CLK	IRDA_TXD0	IRDA_TXD3	IRDA_RXDL3		
P2.2		B5_PWM2	ADC0_AIN2		LCM_CS	ETH_MDIO	IRDA_TXD1				
P2.3		B5_PWM3	ADC0_AIN3			ETH_MII_COL			IRDA_RXDL1		
P2.4			ADC0_AIN4		LCM_AD10		IRDA_TXD5				
P2.5			ADC0_AIN5	LCM_AD11					IRDA_RXDL5		SDIO_D3
P2.6	B4_PWM1	B5_PWM0	ADC0_AIN6	LCM_AD12		B0_BRK	SO0				SDIO_D4
P2.7		B5_PWM1	ADC0_AIN7	LCM_AD13	ETH_MII_RX_DV	ETH_RMII_CRS_DV	SI0				SDIO_D5
P2.8		B5_PWM2	ADC0_AIN8	LCM_AD14	CAN_STBY1	ETH_MII_RXD2	SI2		IRDA_RXDL1		SDIO_D6
P2.9		B5_PWM3	ADC0_AIN9	LCM_AD15	CAN_STBY0	ETH_MII_RXD3					SDIO_D7
P2.10			ADC0_AIN10	LCM_AD16			IRDA_TXD5				
P2.11			ADC0_AIN11	LCM_AD17		ETH_MDC	SI2	SI1	IRDA_RXDL5		
P2.12			ADC0_AIN12			ETH_MII_TXD2	SO1				
P2.13			ADC0_AIN13			ETH_MII_TX_CLK	SI1				
P2.14			ADC0_AIN14		ETH_RMII_RXD0	ETH_MII_RXD0	IRDA_TXD2				
P2.15			ADC0_AIN15		ETH_RMII_RXD1	ETH_MII_RXD1			IRDA_RXDL2		

SWD SPI I2C I2S UART CAN ETHERNET ADC LCM SDIO

PIN	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9
P3.0	P3.0	UTXD1	SCK0	SCL2				CAN_STBY1	B0_PWM0	
P3.1	P3.1	UTXD0	MI0	SCK1	SCL0	BCLK0	BCLK1	CAN_RX1	B0_PWM1	
P3.2	P3.2	URXD0	MO0	MO1	SDA0	WS0	DOUT1	CAN_TX1	B0_PWM2	
P3.3	P3.3	UCTS0	UTXD5	SEL1	SCL1	DOUT0	WS1	CAN_RX0	B0_PWM3	
P3.4	P3.4	URXD1	URXD5	URTS0	MI1	SDA1	DIN0	CAN_TX0		
P3.5	SWDIO	P3.5	MI2							
P3.6	SWCLK	P3.6	UTXD1	MO2	DOUT2					
P3.7	P3.7	RESET								
P3.8	P3.8						B1_CAP0			
P3.9	P3.9						B0_CAP0			
P3.10	P3.10	LXIN								
P3.11	P3.11	LXOUT								
P3.12	P3.12	XIN	UTXD3	SDA0	SDA1					
P3.13	P3.13	XOUT	URXD3	SCL0	SCL1					

PIN	AF10	AF11	AF12	AF13	AF14	AF15	AF16	AF17	AF18	AF19	AF20
P3.0				LCM_A0 (RS)		**CLKOUT1**	IRDA_TXD1				
P3.1				SDIO_D0			SO0	IRDA_TXD0			
P3.2				SDIO_D1			SI0	SI1	IRDA_RXDL0		
P3.3				SDIO_D2		ETH_RMII_RX_ER	IRDA_TXD5				
P3.4				SDIO_D3		ETH_TXER	SO1		IRDA_RXDL1		IRDA_RXDL5
P3.5							SO2				
P3.6							SI2	IRDA_TXD1			
P3.7											
P3.8			B8_PWM10		LCM_AD8	ETH_PHY_LINKSTS					
P3.9			B8_PWM11		LCM_AD9	ETH_PHY_PDN					
P3.10				LCM_WE							
P3.11				LCM_OE							
P3.12							IRDA_TXD3				
P3.13									IRDA_RXDL3		

SWD SPI I2C I2S UART CAN ETHERNET ADC LCM SDIO

✸ ***Note: Don't switch the AFIO function, after the relative GPIO interrupts are set enable.***

6.4 ALTERNATE FUNCTION REGISTERS

Base Address: 0x4001 B000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0200	ALWAYSON_P0_AFIO0	GPIO0.0~GPIO0.2 Alternate Function Control register
0x0204	ALWAYSON_P0_AFIO1	GPIO0.10~GPIO0.11 Alternate Function Control register
0x0208	ALWAYSON_P0_AFIO2	GPIO0.12~GPIO0.15 Alternate Function Control register
0x020C	ALWAYSON_P0_AFIO3	GPIO0.20 Alternate Function Control register
0x0210	ALWAYSON_P1_AFIO0	GPIO1.0~GPIO1.5 Alternate Function Control register
0x0214	ALWAYSON_P1_AFIO1	GPIO1.6~GPIO1.11 Alternate Function Control register
0x0218	ALWAYSON_P1_AFIO2	GPIO1.12~GPIO1.15 Alternate Function Control register
0x021C	-	Reserved
0x0220	ALWAYSON_P2_AFIO0	GPIO2.0~GPIO2.5 Alternate Function Control register
0x0224	ALWAYSON_P2_AFIO1	GPIO2.6~GPIO2.11 Alternate Function Control register
0x0228	ALWAYSON_P2_AFIO2	GPIO2.12~GPIO2.15 Alternate Function Control register
0x022C	-	Reserved
0x0230	ALWAYSON_P3_AFIO0	GPIO3.0~GPIO3.5 Alternate Function Control register
0x0234	ALWAYSON_P3_AFIO1	GPIO3.6~GPIO3.11 Alternate Function Control register
0x0238	ALWAYSON_P3_AFIO2	GPIO3.12~GPIO3.13 Alternate Function Control register

6.4.1 GPIO0.0~GPIO0.2 Alternate Function Control register (ALWAYSON_P0_AFIO0)

Address Offset: 0x200

Bit	Field	Access	Initial	Description
31:15	–	–	0	Reserved
14:10	IO2	RW	0	Alternate function of P0.2 (Reference PIN Mux table) 0: P0.2 1: URXD2 2: MI0 3: SDA1 4: CAN_RX1 8: CT16B0_PWM0 9: CT16B5_PWM3 10: CT16B0_PWM3N 11: CT16B2_PWM3 12: CT16B8_PWM2 13: LCM_AD2 14: ETH_RMII_TX_EN 15: ETH_MII_TX_EN 16: SO0 18: IRDA_RXDL2 20: SDIO_D2 Other: Reserved
9:5	IO1	RW	0	Alternate function of P0.1 (Reference PIN Mux table) 0: P0.1 1: UTXD2 2: URXD2 3: SEL0 4: SCK1 5: SCL1 6: BCLK1 8: CT16B0_PWM1 9: CT16B0_PWM2N 10: CT16B0_PWM3 11: CT16B2_PWM2 12: CT16B8_PWM1

Bit	Field	Access	Initial	Description
				13: LCM_AD1 14: CAN_TX0 15: ETH_MII_RX_ER 16: IRDA_TXD2 18: IRDA_RXDL2 20: SDIO_D1 Other: Reserved
4:0	IO0	RW	0	Alternate function of P0.0 (Reference PIN Mux table) 0: P0.0 1: SCK0 2: MO2 3: DOUT2 6: CT16B0_PWM2 7: CT16B0_PWM3N 8: CT16B0_PWM1N 9: CT16B2_PWM3 10: CT16B0_PWM0 11: CT16B2_PWM0 12: CT16B8_PWM0 13: LCM_AD0 14: CAN_RX0 16: SI2 20: SDIO_D0 Other: Reserved

6.4.2 GPIO0.10~GPIO0.11 Alternate Function Control register (ALWAYSON_P0_AFIO1)

Address Offset: 0x204

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO11	RW	0	Alternate function of P0.11 (Reference PIN Mux table) 0: P0.11 1: URXD0 2: UCTS3 3: SDA0 6: CT16B2_CAP0 7: CT16B1_PWM3 8: CT16B2_PWM1 9: CT16B3_PWM1N 10: CT16B4_PWM1 11: CT16B4_PWM0N 12: CT16B8_PWM6 18: IRDA_RXDL0 Other: Reserved
24:20	IO10	RW	0	Alternate function of P0.10 (Reference PIN Mux table) 0: P0.10 1: UTXD0 2: UTXD4 3: SCL0 4: CAN_TX1 6: CT16B0_BRK 7: CT16B1_PWM2 8: CT16B2_PWM0 9: CT16B3_PWM0N 10: CT16B4_PWM0 11: CT16B4_PWM1N 12: CT16B8_PWM5 14: LCM_CS 15: CLKOUT3 16: IRDA_TXD0 17: IRDA_TXD4 Other: Reserved

Bit	Field	Access	Initial	Description
19:0	–	–	0	Reserved

6.4.3 GPIO0.12~GPIO0.15 Alternate Function Control register (ALWAYSON_P0_AFIO2)

Address Offset: 0x208

Bit	Field	Access	Initial	Description
31:20	–	–	0	Reserved
19:15	IO15	RW	0	Alternate function of P0.15 (Reference PIN Mux table) 0: P0.15 1: UTXD5 3: SCL0 7: CT16B0_PWM0 10: CT16B5_PWM0 12: CT16B8_PWM5 14: SDIO_D6 16: IRDA_TXD5 Other: Reserved
14:10	IO14	RW	0	Alternate function of P0.14 (Reference PIN Mux table) 0: P0.14 1: URXD5 2: SCK1 3: SDA0 5: BCLK1 7: CT16B0_PWM1 10: CT16B5_PWM1 12: CT16B8_PWM6 14: SDIO_D7 18: IRDA_RXDL5 Other: Reserved
9:5	IO13	RW	0	Alternate function of P0.13 (Reference PIN Mux table) 0: P0.13 1: SEL1 6: CT16B4_CAP0 7: CT16B0_PWM2 10: CT16B5_PWM2 12: CT16B8_PWM7 14: SDIO_D0 Other: Reserved
4:0	IO12	RW	0	Alternate function of P0.12 (Reference PIN Mux table) 0: P0.12 1: SCK1 3: SDA2 7: CT16B0_PWM3 8: CT16B0_PWM0N 10: CT16B5_PWM3 12: CT16B8_PWM8 14: SDIO_D1 15: CLKOUT2 Other: Reserved

6.4.4 GPIO0.20 Alternate Function Control register (ALWAYSON_P0_AFIO3)

Address Offset: 0x20C

Bit	Field	Access	Initial	Description
31:15	–	–	0	Reserved
14:10	IO20	RW	0	Alternate function of P0.20 (Reference PIN Mux table) 0: P0.20 5: SDA1 12: CT16B8_PWM1 14: LCM_CS Other: Reserved
9:0	–	–	0	Reserved

6.4.5 GPIO1.0~GPIO1.5 Alternate Function Control register (ALWAYSON_P1_AFIO0)

Address Offset: 0x210

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO5	RW	0	Alternate function of P1.5 (Reference PIN Mux table) 0: P1.5 1: UTXD4 2: SEL1 3: SDA0 4: SDA1 5: WS1 7: CAN_TX1 8: CT16B2_PWM1 9: CT16B2_PWM3 10: CT16B3_PWM1 11: CT16B4_PWM0 12: CT16B8_PWM9 14: SDIO_D5 16: IRDA_TXD4 Other: Reserved
24:20	IO4	RW	0	Alternate function of P1.4 (Reference PIN Mux table) 0: P1.4 1: UTXD2 2: URXD4 3: SCL0 7: CAN_RX1 8: CT16B2_PWM0 9: CT16B2_PWM2 10: CT16B3_PWM0 11: CT16B4_PWM1 12: CT16B8_PWM8 14: SDIO_D4 15: ETH_MII_TXD3 16: IRDA_TXD2 18: IRDA_RXDL4 Other: Reserved
19:15	IO3	RW	0	Alternate function of P1.3 (Reference PIN Mux table) 0: P1.3 1: URXD2 2: DIN0 8: CAN_STBY1 10: CT16B3_PWM0N 11: CT16B4_PWM1N 12: CT16B8_PWM7 18: IRDA_RXDL2 Other: Reserved

Bit	Field	Access	Initial	Description
14:10	IO2	RW	0	Alternate function of P1.2 (Reference PIN Mux table) 0: P1.2 1: URXD4 2: MO0 3: MO2 4: SDA0 5: DOUT0 6: DOUT1 7: DOUT2 8: CAN_STBY0 9: CAN_RX1 10: CT16B1_PWM1 11: CT16B4_PWM0N 12: CT16B5_PWM1 16: SI0 17: SI2 18: IRDA_RXDL4 Other: Reserved
9:5	IO1	RW	0	Alternate function of P1.1 (Reference PIN Mux table) 0: P1.1 1: URXD4 2: MI0 3: MI2 4: SCL0 5: SDA2 7: DIN1 8: CAN_TX0 9: CT16B1_PWM0 10: CT16B5_PWM0 12: CT16B8_PWM4 14: LCM_CS 16: SO0 17: SO2 18: IRDA_RXDL4 Other: Reserved
4:0	IO0	RW	0	Alternate function of P1.0 (Reference PIN Mux table) 0: P1.0 1: UTXD4 2: URXD0 3: SCK0 4: SCK2 5: SDA1 6: BCLK0 7: BCLK2 8: CAN_RX0 9: CT16B2_PWM1 10: CT16B5_PWM2 12: CT16B8_PWM3 15: SWO 16: IRDA_TXD4 18: IRDA_RXDL0 Other: Reserved

6.4.6 GPIO1.6~GPIO1.11 Alternate Function Control register (ALWAYSON_P1_AFIO1)

Address Offset: 0x214

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO11	RW	0	Alternate function of P1.11 (Reference PIN Mux table) 0: P1.11 1: URXD4 2: URTS2 3: MO1 4: SDA1 11: CT16B8_PWM2 12: LCM_A0 14: SDIO_CMD 16: SI1 18: IRDA_RXDL4 Other: Reserved
24:20	IO10	RW	0	Alternate function of P1.10 (Reference PIN Mux table) 0: P1.10 1: UTXD4 2: MI1 3: MO2 4: SCL1 5: SDA1 6: DIN1 7: DOUT2 9: CT16B2_PWM3 12: CT16B8_PWM1 14: SDIO_CK 16: SO1 17: SI2 18: IRDA_TXD4 Other: Reserved
19:15	IO9	RW	0	Alternate function of P1.9 (Reference PIN Mux table) 0: P1.9 1: UTXD1 2: URXD3 3: SEL1 4: SDA1 5: SCL1 6: WS1 7: CAN_STBY0 9: CT16B5_PWM0 12: CT16B8_PWM9 16: IRDA_TXD1 18: IRDA_RXDL3 Other: Reserved
14:10	IO8	RW	0	Alternate function of P1.8 (Reference PIN Mux table) 0: P1.8 1: URXD1 2: URTS3 3: UTXD0 4: SEL0 5: SCK1 6: SEL2 7: SCL1 8: WS0 9: WS2 10: CT16B2_PWM0 12: CT16B8_PWM10 13: LCM_AD3 16: IRDA_TXD0 18: IRDA_RXDL1 Other: Reserved

Bit	Field	Access	Initial	Description
9:5	IO7	RW	0	Alternate function of P1.7 (Reference PIN Mux table) 0: P1.7 1: UTXD2 2: UTXD3 3: SEL0 4: SCK2 6: BCLK2 8: CT16B2_PWM1 9: CT16B5_PWM1 12: CT16B8_PWM11 14: SDIO_D2 16: IRDA_TXD2 17: IRDA_TXD3 Other: Reserved
4:0	IO6	RW	0	Alternate function of P1.6 (Reference PIN Mux table) 0: P1.6 1: URXD2 2: URXD3 3: MI2 4: BCLK1 8: CT16B2_PWM2 9: CT16B5_PWM2 12: CT16B8_PWM0 14: SDIO_D3 16: SO2 18: IRDA_RXDL2 20: IRDA_RXDL3 Other: Reserved

6.4.7 GPIO1.12~GPIO1.15 Alternate Function Control register (ALWAYSON_P1_AFIO2)

Address Offset: 0x218

Bit	Field	Access	Initial	Description
31:20	–	–	0	Reserved
19:15	IO15	RW	0	Alternate function of P1.15 (Reference PIN Mux table) 0: P1.15 2: MO1 3: DOUT1 4: SCL2 6: CT16B5_CAP0 7: CT16B3_PWM1 8: CT16B0_PWM2N 9: CT16B1_PWM3 10: CT16B3_PWM0N 11: CT16B4_PWM1 12: CT16B8_PWM4 14: LCM_AD7 16: SI1 Other: Reserved
14:10	IO14	RW	0	Alternate function of P1.14 (Reference PIN Mux table) 0: P1.14 1: URTS2 2: MI1 3: SDA1 4: SDA2 6: CAN_STBY0 8: CT16B0_PWM1N 9: CT16B1_PWM2 10: CT16B3_PWM0 11: CT16B4_PWM0 12: CT16B8_PWM3

Bit	Field	Access	Initial	Description
				14: LCM_AD6 16: SO1 Other: Reserved
9:5	IO13	RW	0	Alternate function of P1.13 (Reference PIN Mux table) 0: P1.13 1: UCTS2 2: SCK1 3: SCL1 4: SCL2 5: BCLK1 6: CAN_TX0 7: CAN_TX1 8: CT16B0_PWM0N 9: CT16B1_PWM1 10: CT16B3_PWM1N 11: CT16B4_PWM1N 13: LCM_AD5 14: ETH_RMII_TXD1 15: ETH_MII_TXD1 20: SDIO_CMD Other: Reserved
4:0	IO12	RW	0	Alternate function of P1.12 (Reference PIN Mux table) 0: P1.12 1: SEL1 2: SCK2 3: WS1 4: BCLK2 5: CAN_RX0 6: CAN_RX1 7: CT16B0_BRK 8: CT16B0_PWM0 9: CT16B1_PWM0 10: CT16B3_PWM1 11: CT16B4_PWM1 12: CT16B5_PWM0 13: LCM_AD4 14: ETH_RMII_TXD0 15: ETH_MII_TXD0 20: SDIO_CK Other: Reserved

6.4.8 GPIO2.0~GPIO2.5 Alternate Function Control register (ALWAYSON_P2_AFIO0)

Address Offset: 0x220

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO5	RW	0	Alternate function of P2.5 (Reference PIN Mux table) 0: P2.5 1: URXD5 2: SCK0 3: BCLK0 6: CT16B0_PWM1 7: CT16B0_PWM0N 8: CT16B2_PWM0 9: CT16B4_PWM1 12: ADC0_AIN5 13: LCM_AD11 18: IRDA_RXDL5 20: SDIO_D3 Other: Reserved
24:20	IO4	RW	0	Alternate function of P2.4 (Reference PIN Mux table)

Bit	Field	Access	Initial	Description
				0: P2.4 1: UTXD5 2: SEL0 3: SEL2 4: WS0 5: WS2 6: CT16B0_PWM1N 7: CT16B1_PWM1 8: CT16B3_PWM1N 9: CT16B4_PWM1N 12: ADC0_AIN4 14: LCM_AD10 16: IRDA_TXD5 Other: Reserved
19:15	IO3	RW	0	Alternate function of P2.3 (Reference PIN Mux table) 0: P2.3 1: URXD1 7: CT16B2_PWM3 8: CT16B3_PWM1 11: CT16B5_PWM3 12: ADC0_AIN3 15: ETH_MII_COL 18: IRDA_RXDL1 Other: Reserved
14:10	IO2	RW	0	Alternate function of P2.2 (Reference PIN Mux table) 0: P2.2 1: UTXD1 6: CAN_TX1 7: CT16B2_PWM2 8: CT16B3_PWM0 11: CT16B5_PWM2 12: ADC0_AIN2 14: LCM_CS 15: ETH_MDIO 16: IRDA_TXD1 Other: Reserved
9:5	IO1	RW	0	Alternate function of P2.1 (Reference PIN Mux table) 0: P2.1 1: URXD3 2: UTXD0 3: UTXD3 4: URTS1 5: SDA1 6: CAN_RX1 7: CT16B2_PWM1 8: CT16B3_PWM0N 11: CT16B5_PWM1 12: ADC0_AIN1 13: SDIO_CMD 14: ETH_RMII_REF_CLK 15: ETH_MII_RX_CLK 16: IRDA_TXD0 17: IRDA_TXD3 18: IRDA_RXDL3 Other: Reserved
4:0	IO0	RW	0	Alternate function of P2.0 (Reference PIN Mux table) 0: P2.0 1: UTXD3 2: URXD0 3: UCTS1 5: SCL1 6: CAN_STBY1 7: CT16B2_PWM0 9: CT16B1_PWM0 11: CT16B5_PWM0 12: ADC0_AIN0/AVREFH

Bit	Field	Access	Initial	Description
				13: SDIO_CK 14: ETH_MII_CRS 16: IRDA_TXD3 18: IRDA_RXDL0 Other: Reserved

6.4.9 GPIO2.6~GPIO2.11 Alternate Function Control register (ALWAYSON_P2_AFIO1)

Address Offset: 0x224

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO11	RW	0	Alternate function of P2.11 (Reference PIN Mux table) 0: P2.11 1: URXD5 2: MO2 3: MO1 4: SDA2 5: DOUT2 6: DOUT1 12: ADC0_AIN11 13: LCM_AD17 15: ETH_MDC 16: SI2 17: SI1 18: IRDA_RXDL5 Other: Reserved
24:20	IO10	RW	0	Alternate function of P2.10 (Reference PIN Mux table) 0: P2.10 1: UTXD5 2: SCL2 12: ADC0_AIN10 13: LCM_AD16 16: IRDA_TXD5 Other: Reserved
19:15	IO9	RW	0	Alternate function of P2.9 (Reference PIN Mux table) 0: P2.9 1: URTS2 2: SCK1 3: BCLK1 6: CT16B1_PWM3 7: CT16B0_PWM1 8: CT16B0_PWM2N 11: CT16B5_PWM3 12: ADC0_AIN9 13: LCM_AD15 14: CAN_STBY0 15: ETH_MII_RXD3 20: SDIO_D7 Other: Reserved
14:10	IO8	RW	0	Alternate function of P2.8 (Reference PIN Mux table) 0: P2.8 1: URXD1 2: MO2 3: DIN0 5: DOUT2 7: CT16B0_PWM1N 8: CT16B0_PWM2 11: CT16B5_PWM2 12: ADC0_AIN8 13: LCM_AD14 14: CAN_STBY1

Bit	Field	Access	Initial	Description
				15: ETH_MII_RXD2 16: SI2 18: IRDA_RXDL1 20: SDIO_D6 Other: Reserved
9:5	IO7	RW	0	Alternate function of P2.7 (Reference PIN Mux table) 0: P2.7 1: MO0 2: DOUT0 6: CT16B0_PWM0N 7: CT16B0_PWM2 8: CT16B1_PWM2 9: CT16B4_PWM0 11: CT16B5_PWM1 12: ADC0_AIN7 13: LCM_AD13 14: ETH_MII_RX_DV 15: ETH_RMII_CRS_DV 16: SI0 20: SDIO_D5 Other: Reserved
4:0	IO6	RW	0	Alternate function of P2.6 (Reference PIN Mux table) 0: P2.6 1: UCTS2 2: MI0 5: CT16B8_PWM2 6: CT16B0_PWM0 7: CT16B0_PWM2N 8: CT16B3_PWM0 9: CT16B4_PWM0N 10: CT16B4_PWM1 11: CT16B5_PWM0 12: ADC0_AIN6 13: LCM_AD12 15: CT16B0_BRK 16: SO0 20: SDIO_D4 Other: Reserved

6.4.10 GPIO2.12~GPIO2.15 Alternate Function Control register (ALWAYSON_P2_AFIO2)

Address Offset: 0x228

Bit	Field	Access	Initial	Description
31:20	–	–	0	Reserved
19:15	IO15	RW	0	Alternate function of P2.15 (Reference PIN Mux table) 0: P2.15 1: URXD2 8: CT16B0_PWM3N 9: CT16B3_PWM1 12: ADC0_AIN15 14: ETH_RMII_RXD1 15: ETH_MII_RXD1 18: IRDA_RXDL2 Other: Reserved
14:10	IO14	RW	0	Alternate function of P2.14 (Reference PIN Mux table) 0: P2.14 1: UTXD2 8: CT16B0_PWM3 9: CT16B3_PWM0 12: ADC0_AIN14 14: ETH_RMII_RXD0

Bit	Field	Access	Initial	Description
				15: ETH_MII_RXD0 16: IRDA_TXD2 Other: Reserved
9:5	IO13	RW	0	Alternate function of P2.13 (Reference PIN Mux table) 0: P2.13 1: MO1 2: DOUT1 12: ADC0_AIN13 15: ETH_MII_TX_CLK 16: SI1 Other: Reserved
4:0	IO12	RW	0	Alternate function of P2.12 (Reference PIN Mux table) 0: P2.12 1: MI1 8: CT16B0_PWM1 12: ADC0_AIN12 15: ETH_MII_TXD2 16: SO1 Other: Reserved

6.4.11 GPIO3.0~GPIO3.5 Alternate Function Control register (ALWAYSON_P3_AFIO0)

Address Offset: 0x230

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO5	RW	0	Alternate function of P3.5 (Reference PIN Mux table) 0: SWDIO 1: P3.5 2: MI2 16: SO2 Other: Reserved
24:20	IO4	RW	0	Alternate function of P3.4 (Reference PIN Mux table) 0: P3.4 1: URXD1 2: URXD5 3: URTS0 4: MI1 5: SDA1 6: DIN0 7: CAN_TX0 13: SDIO_D3 15: ETH_TXER 16: SO1 18: IRDA_RXDL1 20: IRDA_RXDL5 Other: Reserved
19:15	IO3	RW	0	Alternate function of P3.3 (Reference PIN Mux table) 0: P3.3 1: UCTS0 2: UTXD5 3: SEL1 4: SCL1 5: DOUT0 6: WS1 7: CAN_RX0 8: CT16B0_PWM3 13: SDIO_D2 15: ETH_RMII_RX_ER 16: IRDA_TXD5 Other: Reserved
14:10	IO2	RW	0	Alternate function of P3.2 (Reference PIN Mux table)

Bit	Field	Access	Initial	Description
				0: P3.2 1: URXD0 2: MO0 3: MO1 4: SDA0 5: WS0 6: DOUT1 7: CAN_TX1 8: CT16B0_PWM2 13: SDIO_D1 16: SI0 17: SI1 18: IRDA_RXDL0 Other: Reserved
9:5	IO1	RW	0	Alternate function of P3.1 (Reference PIN Mux table) 0: P3.1 1: UTXD0 2: MI0 3: SCK1 4: SCL0 5: BCLK0 6: BCLK1 7: CAN_RX1 8: CT16B0_PWM1 13: SDIO_D0 16: SO0 17: IRDA_TXD0 Other: Reserved
4:0	IO0	RW	0	Alternate function of P3.0 (Reference PIN Mux table) 0: P3.0 1: UTXD1 2: SCK0 3: SCL2 7: CAN_STBY1 8: CT16B0_PWM0 13: LCM_A0 (RS) 15: CLKOUT1 16: IRDA_TXD1 Other: Reserved

6.4.12 GPIO3.6~GPIO3.11 Alternate Function Control register (ALWAYSON_P3_AFIO1)

Address Offset: 0x234

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:25	IO11	RW	0	Alternate function of P3.11 (Reference PIN Mux table) 0: P3.11 1: LXOUT 13: LCM_OE Other: Reserved
24:20	IO10	RW	0	Alternate function of P3.10 (Reference PIN Mux table) 0: P3.10 1: LXIN 13: LCM_WE Other: Reserved
19:15	IO9	RW	0	Alternate function of P3.9 (Reference PIN Mux table) 0: P3.9 6: CT16B0_CAP0 12: CT16B8_PWM11 14: LCM_AD9 15: ETH_PHY_PDN

Bit	Field	Access	Initial	Description
				Other: Reserved
14:10	IO8	RW	0	Alternate function of P3.8 (Reference PIN Mux table) 0: P3.8 6: CT16B1_CAP0 12: CT16B8_PWM10 14: LCM_AD8 15: ETH_PHY_LINKSTS Other: Reserved
9:5	IO7	RW	0	Alternate function of P3.7 (Reference PIN Mux table) 0: P3.7 1: External reset Other: Reserved
4:0	IO6	RW	0	Alternate function of P3.6 (Reference PIN Mux table) 0: SWCLK 1: P3.6 2: UTXD1 3: MO2 4: DOUT2 16: SI2 17: IRDA_TXD1 Other: Reserved

6.4.13 GPIO3.12~GPIO3.13 Alternate Function Control register (ALWAYSON_P3_AFIO2)

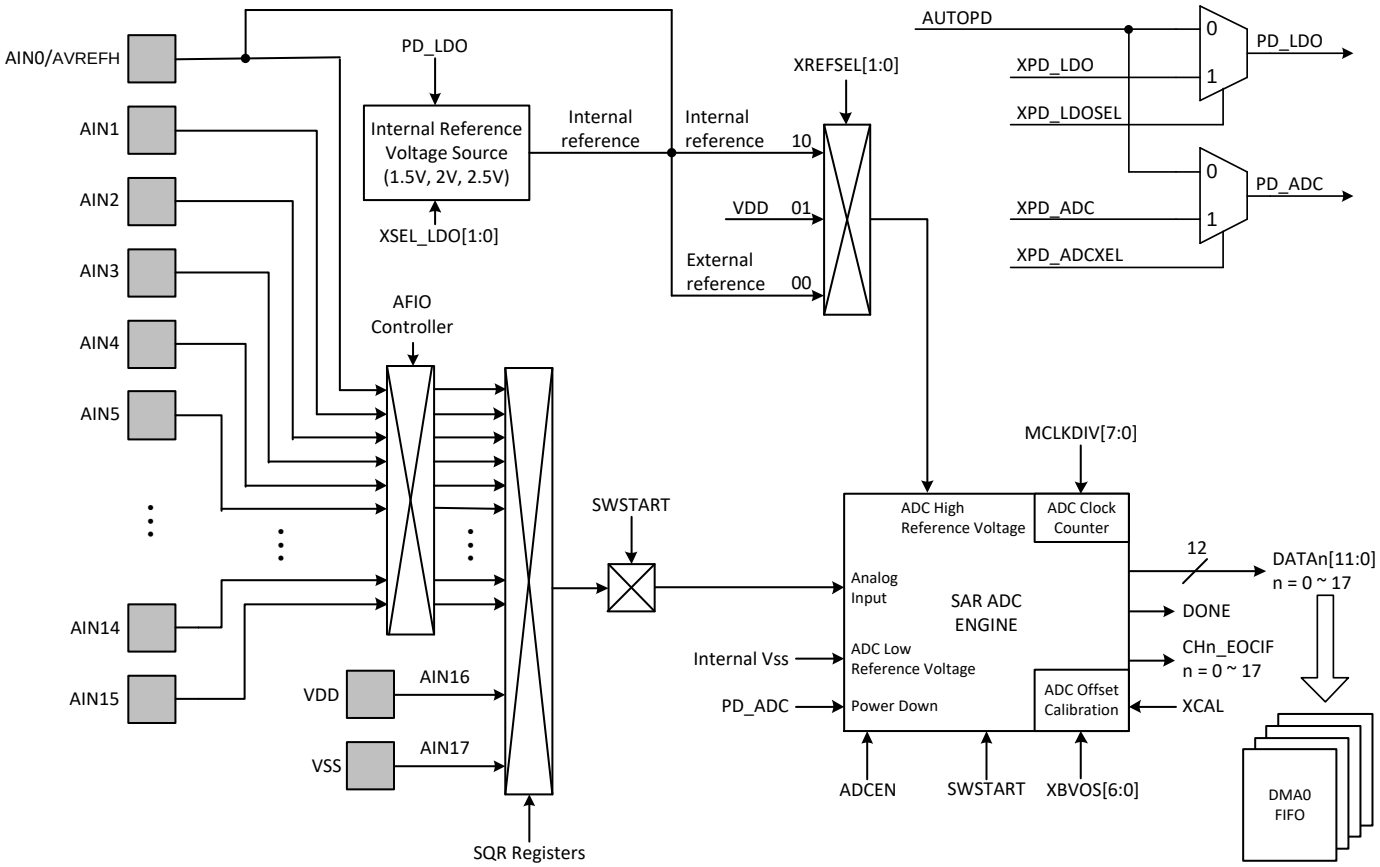
Address Offset: 0x238

Bit	Field	Access	Initial	Description
31:10	–	–	0	Reserved
9:5	IO13	RW	0	Alternate function of P3.13 (Reference PIN Mux table) 0: P3.13 1: XOUT 2: URXD3 3: SCL0 4: SCL1 18: IRDA_RXDL3 Other: Reserved
4:0	IO12	RW	0	Alternate function of P3.12 (Reference PIN Mux table) 0: P3.12 1: XIN 2: UTXD3 3: SDA0 4: SDA1 16: IRDA_TXD3 Other: Reserved

7 16+2 CHANNEL ANALOG TO DIGITAL CONVERTOR (ADC)

7.1 OVERVIEW

This analog to digital converter (ADC) has 16 external input sources, one internal channel to internal voltage source (VDD), one internal channel to internal voltage source (VSS), with up to 4096-step resolution to transfer analog signal into 12-bits digital data. The sequence of ADC operation is to select input source (AIN0 ~ AIN17) and SWSTART bit to "1" to start conversion. When the conversion is complete, the ADC circuit will set DONE bit to "1" and final value output in ADC_DATA register.



Use ADC_SQR registers to select AIN pin, the analog signal inputs to ADC engine.

The ADC converting rate can be selected by MCLKDIV[7:0] bits.

The AIN0 can be internal 1.5V or 2V or 2.5V input channel when internal reference voltage is enabled. In this time ADC reference voltage must be internal VDD, not internal 1.5V or 2V or 1.5V. AIN0 can be a good battery detector for battery system. To select appropriate internal level and compare value, a high performance and cheaper low battery detector is built in the system.

The ADC reference high voltage includes three source, one is internal VDDA (XREFSEL[1:0]=01b), one is internal reference voltage (1.5V or 2V or 2.5V, XREFSEL[1:0]=10b), and the other one is external reference voltage input pin from AIN0/P2.0 pin (XREFSEL[1:0]=00b).

Internal reference voltage and external reference voltage cannot exist at the same time, and AIN0 will be occupied when

used. At this time, AIN0/P2.0 must be floating.

ADC provides different power–down control signals to save power. the power-down control signal is selected by selection signals.

ADC supports up to four DMA channels. Each DMA channel consists of four registers, data port register, function control register, interrupt enable register, and interrupt status register. Each DMA channel has 16-frame DMA FIFO.

The ADC clock is enabled by the ADC0CLKEN bit of SCU_APB1CLKG registers. Enabled by ADC0CLKEN bit of SCU_SLP_APB1CLKG in sleep mode.

✸ ***Note:***

1. ***ADC_MCLK 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 + 2.4V.***
4. ***The internal reference voltage output needs C_{REF} 1uF in P2.0 pin.***
5. ***ADC programming notice***
   - ***Set ADC input pin I/O direction as input mode.***
   - ***Disable pull-up/down resistor of ADC input pin.***
   - ***The Alternate function of GPIO is selected as the ADC channel.***
   - ***Disable ADC (set ADCEN = "0") before enter low-power (Sleep/Deep-sleep/Deep-Power-down) mode to save power consumption .***
   - ***Delay power enable time after enable ADC (set ADCEN = "1") to wait ADC circuit ready for conversion.***

7.2 FEATURES

1. Sampling rate up to 1Msps
2. Each channel has an independent data buffer
3. 4 sets 16-frame DMA FIFO (2 data each frame, 16x2x12 bits)
4. Offset calibration.
5. Programmable sampling time
6. Conversion can be triggered by SW
7. Single step, single scan, and continuous scan conversion modes
8. Scan sequence
9. Automatic power-down
10. Data watchdog for range checking (Up to 36 threshold detect sets)
11. Supply 4 DMA channels

7.3 ADC Channel

PIN	Channel selection : SQCn[1]	ADC Channel	PIN	Channel selection : SQCn[1]	ADC Channel
P2.0	0x00	ADC0_AIN0/AVREFH	P2.9	0x09	ADC0_AIN9
P2.1	0x01	ADC0_AIN1	P2.10	0x0A	ADC0_AIN10
P2.2	0x02	ADC0_AIN2	P2.11	0x0B	ADC0_AIN11
P2.3	0x03	ADC0_AIN3	P2.12	0x0C	ADC0_AIN12
P2.4	0x04	ADC0_AIN4	P2.13	0x0D	ADC0_AIN13
P2.5	0x05	ADC0_AIN5	P2.14	0x0E	ADC0_AIN14
P2.6	0x06	ADC0_AIN6	P2.15	0x0F	ADC0_AIN15
P2.7	0x07	ADC0_AIN7	-	0x10	VDD
P2.8	0x08	ADC0_AIN8	-	0x11	VSS

[1] Select the ADC channel by ADC_SQR registers.

7.4 ADC CONVERTING TIME

The ADC converting time is from SWSTART=1 (Start to ADC convert) to DONE=1 (End of ADC convert). The converting time duration is depend on ADC resolution and ADC clock rate.

ADC clock source is controlled by MCLKDIV[7:0] 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.

ADC conversion clock MCLK = ADC Data Bus clock PCLK / (MCLKDIV+1)

12-bit ADC conversion time = Conversion cycle / MCLK sec

If Conversion cycle is 16 clock cycles

MCLKDIV [7:0]	ADC Data Bus clock (PCLK)	Pre-scaler	ADC Conversion Clock (MCLK)	ADC Conversion Time (us)	ADC Conversion Rate (KHz)
5	96 MHz	PCLK/6	16 MHz	1	1000
23	96 MHz	PCLK/24	4 MHz	4	250
95	96 MHz	PCLK/96	1 MHz	16	62.5
255	96 MHz	PCLK/256	37.5 KHz	42.66	23.4375
2	48 MHz	PCLK/3	16 MHz	1	1000
255	48 MHz	PCLK/256	18.75 KHz	85.33	11.71875
4	75 MHz	PCLK/5	15 MHz	1.066	937.5
3	50 MHz	PCLK/4	12.5 MHz	1.28	781.25
0	12 MHz	PCLK/1	12MHz	1.33	750

✸ ***Note: The conversion cycle is controlled by timing configuration.***

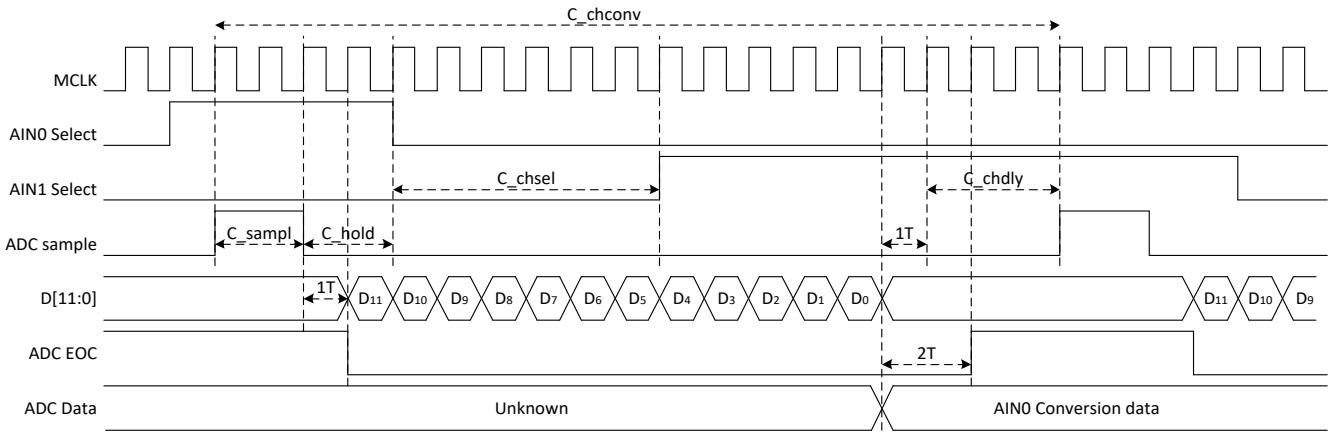
7.4.1 Timing Configuration

The ADC supports seven timing parameters to configure conversion timing.

7.4.1.1 Conversion timing

C_chconv = C_sampl + 1*T_{MCLK} + C_dataout (12*T_{MCLK}) + 1*T_{MCLK} + C_chdly

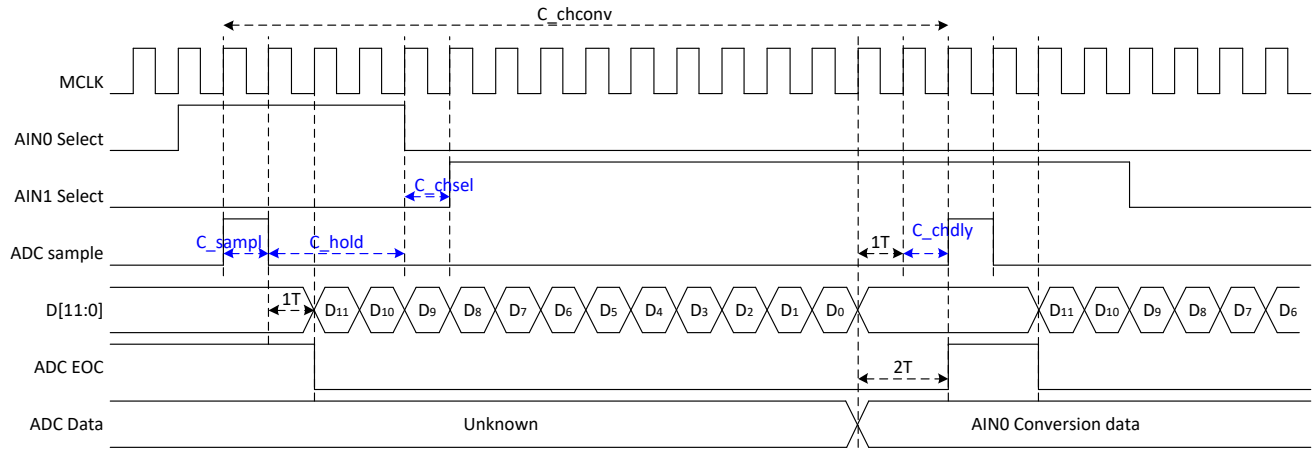
MCLK can be adjusted by the MCLK_DIV register.



Symbol	Description	Register	Bit	Default (unit MCLK)
C_sampl	Channel sample cycle	ADC_SMPR	SAMPLTime[3:0]	0x0 (1T)
C_hold	Channel hold cycle	ADC_SMPR	HOLDTime[3:0]	0x2 (3T)
C_chsel	Channel change discharge cycle for next channel	ADC_SMPR	CHSELTime[2:0]	0x0 (1T)
C_chdly	Delay time cycle for each channel	ADC_SMPR	CHDLYTime[7:0]	0x00 (1T)

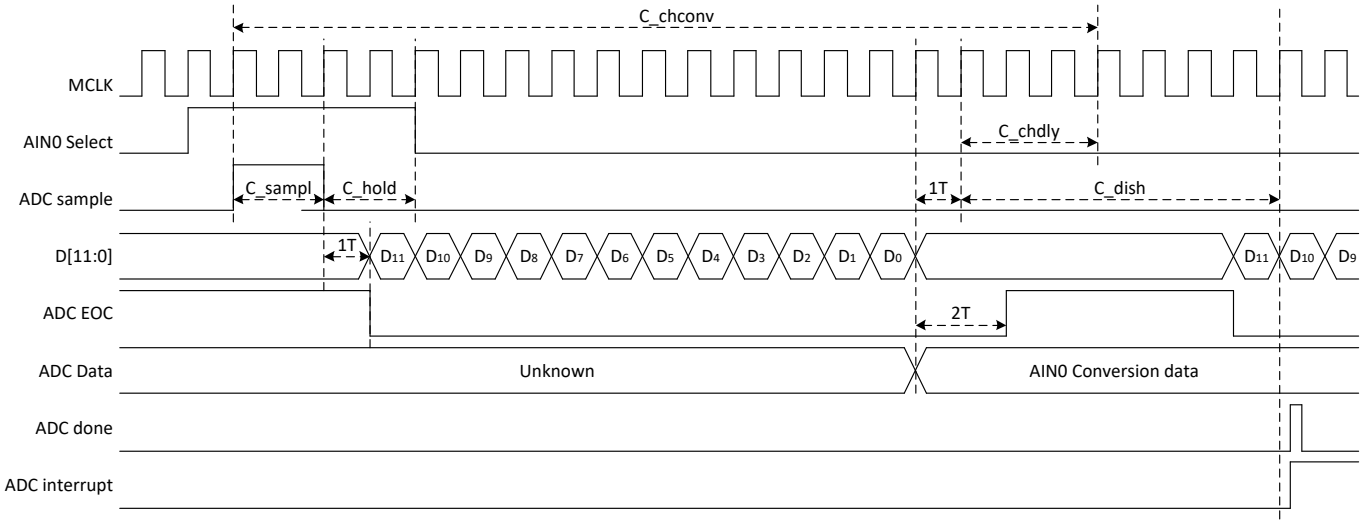
7.4.1.2 Conversion Rate 1MSPS

A conversion cycle takes 1T+1T+12T+1T+1T=16T (SAMPLTime = 0 and CHDLYTime = 0) to achieve 1MSPS when MCLK is 16MHz.



7.4.1.3 Discharge Time

The scan discharge cycle is a programmable timing parameter to reserve for discharging capacitors of the pre-amplifier circuit in each ADC scan, please refer to the ADC_DISCHTIME register for the detailed information. The discharge time will be decided by the system application and ADC specification. An example timing chart is shown in the following figure, the discharge cycles, C_dish, are programmed as 8 cycles of ADC clock. For the ADC controller interrupt, the event will be triggered when each scanning cycle is finished, the interrupt will be issued after discharging time.

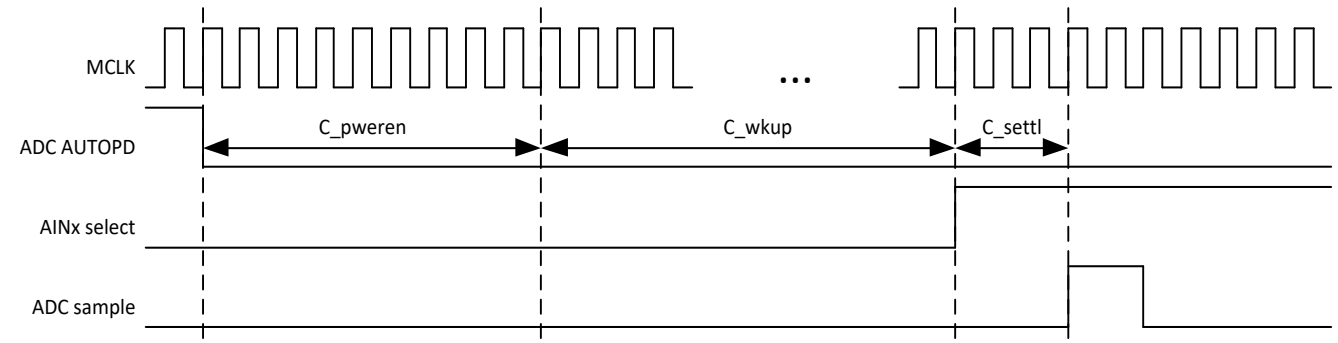


Symbol	Description	Register	Bit	Default (unit MCLK)
C_dish	Sensing discharge cycle after sequence end	ADC_DISCHTIME	DISCHTIME[15:0]	0x0000 (1T)

7.4.1.4 Power Enable Time

ADC needs to wait the wakeup time when ADC is turned on. An example of timing chart is shown in the following figure. The ADC controller provides a programmable PWRENTIME parameter to adjust the ADC wake-up time based on the ADC clock.

The ADC auto power down (AUTOPD) is low when ADC_EN = 0. The ADC auto power enable is high when ADC_EN = 1 and SWSTART = 1.



Symbol	Description	Register	Bit	Default (unit MCLK)
C_pweren	ADC power enable cycle	ADC_TPARM	PWRENTIME[15:0]	0x0020 (21T)
C_wkup	ADC wakeup cycles of ADC conversion	ADC_TPARM	WKUPTIME[18:16]	0x3 (unit conversion cycle)
C_settl	Channel settling cycle for first channel	ADC_SMPR	SETTLTime[7:0]	0x00 (1T)

7.4.1.5 Minimum timing cycle

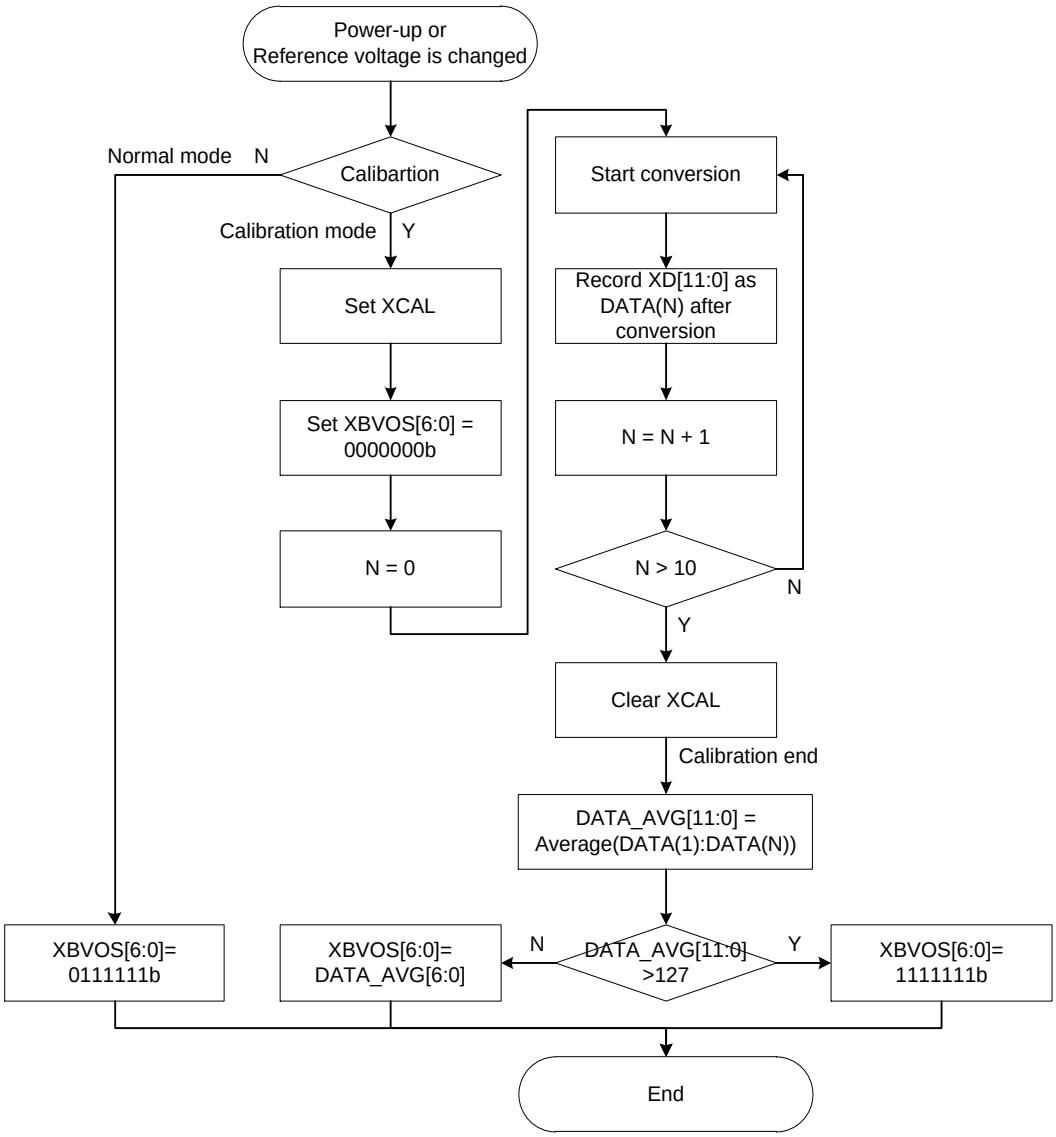
Symbol	Description	Register	Bit	Value	Signal cycle (unit MCLK)
C_sampl	Channel sample cycle	ADC_SMPR	SAMPLTime[3:0]	0	1
C_hold	Channel hold cycle	ADC_SMPR	HOLDTime[3:0]	0	1
C_chsel	Channel change discharge cycle for next channel	ADC_SMPR	CHSELTime[2:0]	0	1
C_chdly	Delay time cycle for each channel	ADC_SMPR	CHDLYTime[7:0]	0	1
C_dish	Sensing discharge cycle after sequence end	ADC_DISCHTIME	DISCHTIME[11:0]	0	1
C_pweren	ADC power enable cycle	ADC_TPARM	PWRENTIME[11:0]	0	1
C_settl	Channel settling cycle for first channel	ADC_SMPR	SETTLTime[7:0]	0	1

Symbol	Description	Register	Bit	Value	Signal cycle (unit conversion cycle)
C_wkup	ADC wakeup cycles of ADC conversion	ADC_TPARM	WKUPTIME[18:16]	0	0

7.5 OFFSET CALIBRATION

The ADC builds in ADC offset calibration. Implement ADC offset calibration through firmware. ADC offset adjust range = 0~127 count controlled by XBVOS[6:0].

If the reference source changes, it must be recalibrated. Entering and exiting power-down mode also requires recalibration.



7.6 ADC CONTROL NOTICE

7.6.1 ADC Signal

The ADC high reference voltage includes internal VDD33A/2.5V/2V/1.5V and external reference voltage source from AIN0/AVREFH pin controlled by XREFSEL[1:0] bits. If XREFSEL[1:0]=10b, ADC reference voltage is from internal voltage source; if XREFSEL[1:0] =01b, ADC reference voltage is from VDD; if XREFSEL[1:0] =00b, ADC reference voltage is from external voltage source (AIN0/AVREFH).

ADC reference voltage range limitation is "**(ADC high reference voltage – low reference voltage) ≧ 2.4V**". ADC low reference voltage is VSS = 0V. So **ADC high reference voltage range is 2.4V~VDD**. The range is ADC external high reference voltage range.

- **ADC Internal Low Reference Voltage = 0V**
- **ADC Internal High Reference Voltage = 2.5V/2V/1.5V (XREFSEL[1:0] = 10b)**
- **ADC Internal High Reference Voltage = VDD (XREFSEL[1:0] = 01b)**
- **ADC External High Reference Voltage = 2.4V~VDD (XREFSEL[1:0] = 00b)**

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**

XREFSEL[1:0]	XSEL_LDO[1:0]	ADC high reference
10b	00b	Internal 2.5V
10b	01b	Internal 2.0V
10b	10b	Internal 1.5V
01b	Don't care	VDD
00b	Don't care	External 2.4V~VDD

✷ ***Note: Internal reference voltage and external reference voltage cannot exist at the same time, and AIN0 will be occupied when used. At this time, AIN0 must be floating.***

7.6.2 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:** Enable ADC. ADCEN is ADC control bit to control. ADCEN = 1 is to enable ADC. ADCEN = 0 is to disable ADC. **When ADCEN is enabled, the system must be delay 3*ADC conversion cycle to be the ADC warm-up time by program, and then set SWSTART to do ADC converting. The 3*ADC conversion cycle delay time is necessary after ADCEN setting (not SWSTART setting), or the ADC converting result would be error.** Normally, the ADCEN is set one time when the system under normal run condition, and do the delay time only one time.

- **Step 2:** If the ADC high reference voltage is from external voltage source, set the XREFSEL[1:0] = 00b. The ADC external high reference voltage inputs from AIN0 pin. **It is necessary to set AIN0 as input mode without pull-up resistor.**

- **Step 3:** Set MISC register for calibration mode. Start offset calibration.

- **Step 4:** Select the ADC input pin by ADC_SQR registers. **When one AIN pin is selected to be analog signal input pin, it is necessary to setup the pin as input mode and disable the pull-up resistor by program.**

- **Step 5:** Start to execute ADC conversion by setting SWSTART = 1.

- **Step 5:** Wait the end of ADC converting through checking ADC_DONEIF = 1 or CHn_EOCIF = 1. If ADC interrupt function is enabled, the program executes ADC interrupt service when ADC interrupt occurrence.

7.6.3 ADC PIN CONFIGURATION

ADC input pins are shared with Port 2 digital I/O pins. ADC channel selection is through scan sequence registers. The conversion order is based on the arrangement of the sequence register. The value of scan sequence points to the ADC input channel directly, SQCn[4:0]=0000b selects AIN0, SQCn[4:0]=0001b selects AIN1, etc.

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 Port 2 will encounter above current leakage situation.

The P2.0/AIN0 can be ADC external high reference voltage input pin when XREFSEL[1:0] = 00b. In the condition, P2.0 GPIO mode must be set as input mode and inactive (no pull-down/up resistor enabled, ADC channel) with GPIO2_DIR, GPIO2_PULLEN and GPIO2 alternate function register by program.

Only one pin of Port 2 can be configured as ADC input in the same time. The pins of Port 2 configured as ADC input channel must be set as input mode, inactive (no pull-down/pull-up resistor enabled, ADC channel) with GPIO2_DIR, GPIO2_PULLEN and GPIO2 alternate function register by program to avoid current leakage.

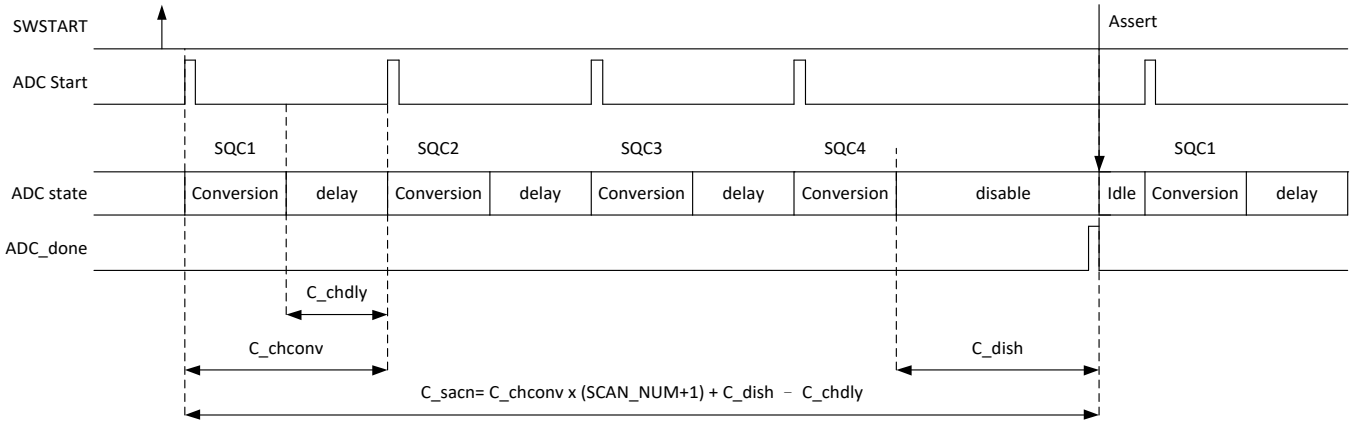
✸ ***Note: The GPIO mode of ADC input channels used must be set as input mode and inactive (no pull-down/pull-up resistor enabled, Schmitt trigger disabled) with GPIO2_DIR, GPIO2_PULLEN and GPIO2 alternate function register by program.***

7.7 ADC CONVERSION MODES

The ADC can either converts a single channel or a sequence of multiple channels, determined by SCANNUM bits in ADC_CTRL register. The ADC will convert only one channel selected by SQC1[2:0] bits in ADC_SQR0 register if SCANNUM bits is 0, and convert a sequence of multiple channels if SCANNUM bits is not zero. User can select the number of scan conversion channels and the scan sequence by programming SCNNUM bits in ADC_CTRL register and ADC_SQR registers.

Besides, the ADC supports either Single Step mode, Single Scan mode or Continuous Scan mode, determined by COVNMODE[1:0] bits in ADC_CTRL register.

If the ADCEN bit is cleared, the continuous mode will be stop. When the ADCEN bit is cleared during the ADC conversion, it is asserted when the ADC controller is already at the idle state.



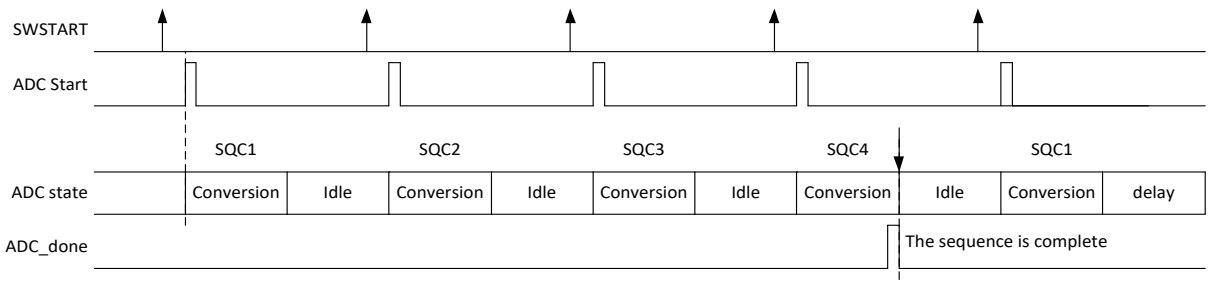
7.7.1 Single Mode

Set COVNMODE to 0 and set SWSTART bit to 1 to start the ADC conversion in Single Step mode. the selected channels will be converted once per trigger until the numver of scan conversion is done.

When each conversion is complete, the converted data are stored in the independent buffer ADC_DATAn[11:0]. The CHn_EOCIF 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 CHn_EOCIE bit are set.

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

The following figure is an example of single-step conversion mode when SCANNUM = 3.



✸ ***Note:***
1. ***The conversion order is based on the arrangement of the sequence register.***
2. ***If the user wants to convert single channel fixedly, SCANNUM should be set to 0. At this time, SQC1 is the target channel of the conversion.***

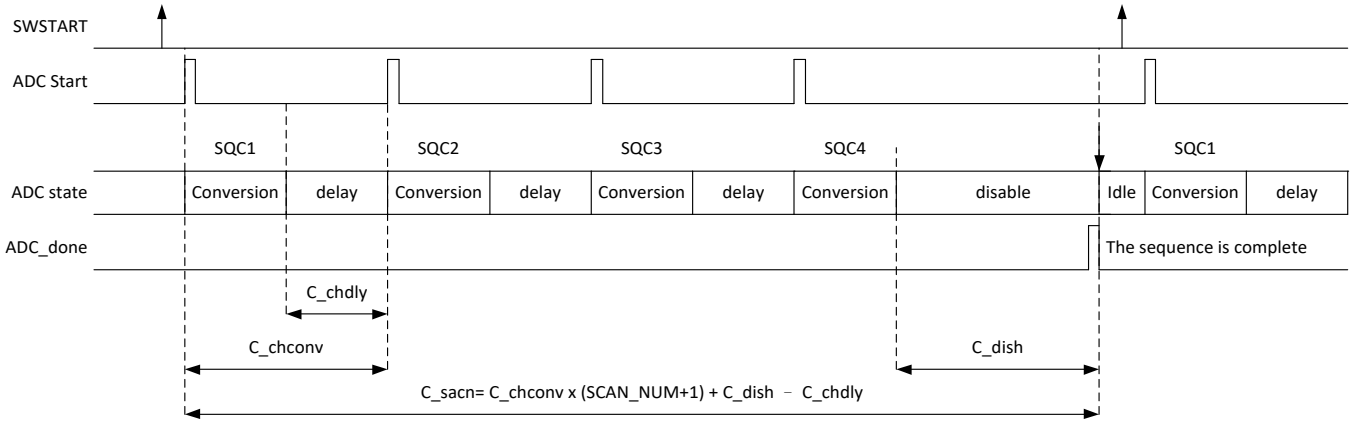
7.7.2 Single Scan Mode

Set COVNMODE to 1 and set SWSTART bit to 1 to start the ADC conversion in Single Scan mode. the selected channels will be converted until the number of scan conversion is done.

When each conversion is complete, the converted data are stored in the independent buffer ADC_DATAn[11:0]. The CHn_EOCIF 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 CHn_EOCIE bit are set.

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

The following figure is an example of single-scan conversion mode when SCANNUM = 3.



✸ ***Note: The conversion order is based on the arrangement of the sequence register.***

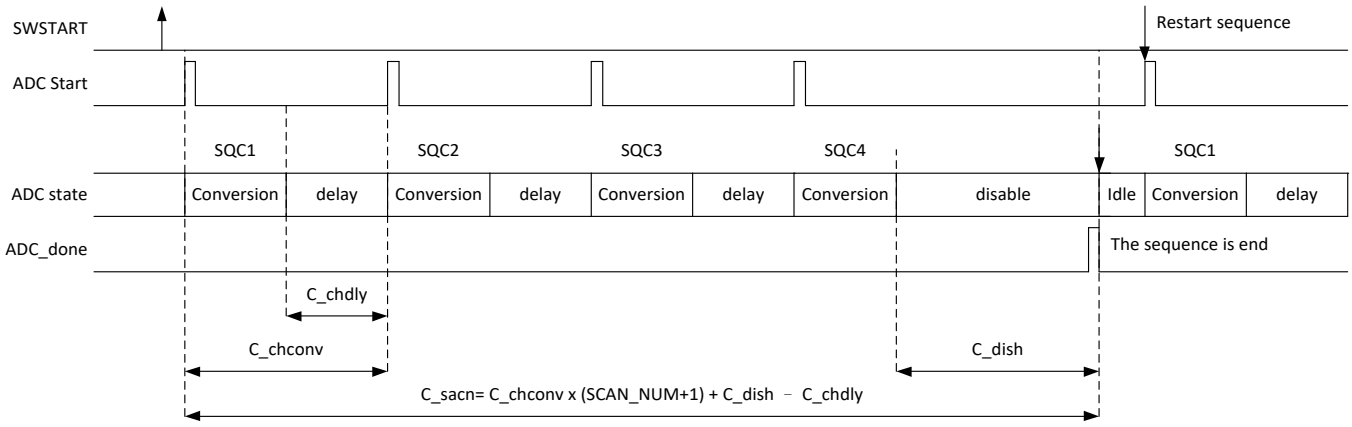
7.7.3 Continuous Scan Mode

Set COVNMODE to 2 and set SWSTART bit to 1 to start the ADC conversion in Continuous Scan 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 ADCEN bit is cleared to stop the continuous scan mode.

When each conversion is complete, the converted data are stored in the independent buffer ADC_DATAn[11:0]. The CHn_EOCIF 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 CHn_EOCIE bit are set.

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

The following figure is an example of single-scan conversion mode when SCANNUM = 3.

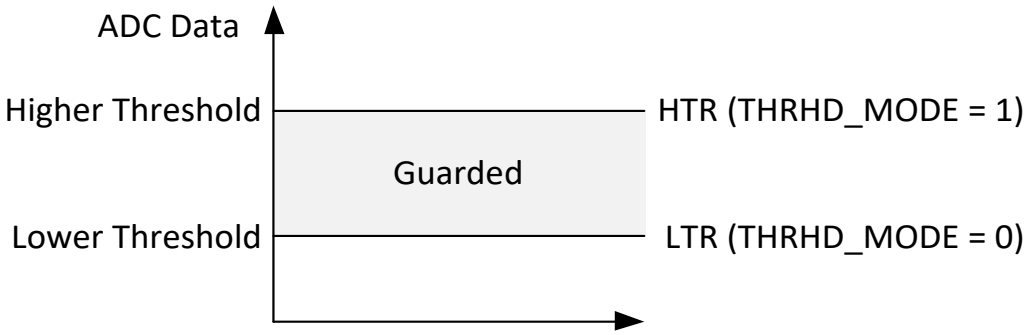


✸ ***Note:***
1. ***The conversion order is based on the arrangement of the sequence register.***
2. ***If the user wants to convert single channel fixedly, SCANNUM should be set to 0. At this time, SQC1 is the target channel of the conversion.***

7.8 ADC WINDOW WATCHDOG

The ADC window watchdog (AWW) will monitor whether the measured data is inside or outside of the high threshold value and low threshold value in ADC_THRHOLD registers. The THRDn_IF bit is set to 1 to notify the system when a desired condition is matched, and the ADC interrupt is generated if the THRDn_IE bit is set.

The ADC controller supports up to 18 programming registers providing up to 36 threshold values for threshold detection.



7.8.1 Data > High Threshold

1. n = 0 ~ 35, x = 0 ~ 17
2. THRHDn_MODE = 1b (HTR): DATAx > THRHDn_VALUE
3. THRHDn_CH (Detect Channel): AINx
4. Other detect condition don't use (LTR = 0x000)

Comparison Register Settings	Output Code (DATAx)	THRDn_IF Effects
	0xFFF	**THRDn_IF = 1**
	…	
	0x081	
HTRn, THRHDn_VALUE [11:0] = 0x080	0x080	THRDn_IF Not Affected

	0x07F	
	…	
	0x001	
Other LTR, THRHD_VALUE[11:0] = 0x000	0x000	

7.8.2 Data < Low Threshold

1. n = 0 ~ 35, x = 0 ~ 17
2. THRHDn_MODE = 0b (LTR): DATAx < THRHDn_VALUE
3. THRHDn_CH (Detect Channel): AINx
4. Other detect condition don't use (HTR = 0xFFF)

Comparison Register Settings	Output Code (DATAx)	THRDn_IF Effects
Other HTR, THRHD_VALUE[11:0] = 0xFFF	0xFFF	THRDn_IF Not Affected
	0xFFE	
	…	
	0x041	
LTRn, THRHDn_VALUE [11:0] = 0x040	0x040	
	0x03F	**THRDn_IF = 1**
	…	
	0x000	

7.8.3 High Threshold < Data < Low Threshold

1. n = 0 ~ 35, m = 0 ~ 35, x = 0 ~ 17
2. THRHDn_MODE = 0b (LTR): DATAx < THRHDn_VALUE
3. THRHDn_CH (Detect Channel): AINx
4. THRHDm_MODE = 1b (HTR): DATAx > THRHDm_VALUE
5. THRHDm_CH (Detect Channel): AINx
6. Other detect condition don't use (HTR = 0xFFF or LTR = 0x000)

Comparison Register Settings	Output Code (DATAx)	THRDn_IF Effects
	0xFFF	THRDn_IF Not Affected, **THRDm_IF = 1**
	…	
	0x081	
LTRn, THRHDn_VALUE [11:0] = 0x080	0x080	
	0x07F	**THRDn_IF = 1, THRDm_IF = 1**
	…	
	0x041	
HTRm, THRHDm_VALUE [11:0] = 0x040	0x040	THRDm_IF Not Affected, **THRDn_IF = 1**
	0x03F	
	…	
	0x000	

7.9 POWER DOWN MODE

ADC supports auto power-down mode and single power-down mode to save power consumption. When auto power down mode, the XPD, XPD_ADC and XPD_LDO can assert together.

7.9.1 Power-down Control

ADC provides different power–down control signals to save power. the power-down control signal is selected by selection signals.

The ADC controller uses PD_ADC to control the power-down mode of ADC conversion engine. The ADC conversion engine enters power-down mode when PD_ADC is high. The signal source of PD_ADC can choose AUTOPD signal or XPD_ADC bit.

The internal LDO supplies 1.5V/2V/2.5V for internal reference voltage. The ADC controller uses PD_LDO to control the power-down mode of internal LDO. The internal LDO enters power-down mode when PD_LDO is high. The signal source of PD_LDO can choose AUTOPD signal or XPD_LDO bit.

Power down control	PD
PD_ADC	Power-down control signal for ADC core Tie to logic '0' to enter the normal operating mode Tie to logic '1' to enter the power-down mode(all MUX will be off)
PD_LDO	Power-down control signal for internal reference LDO Tie to logic '0' to enter the normal operating mode Tie to logic '1' to enter the power-down mode

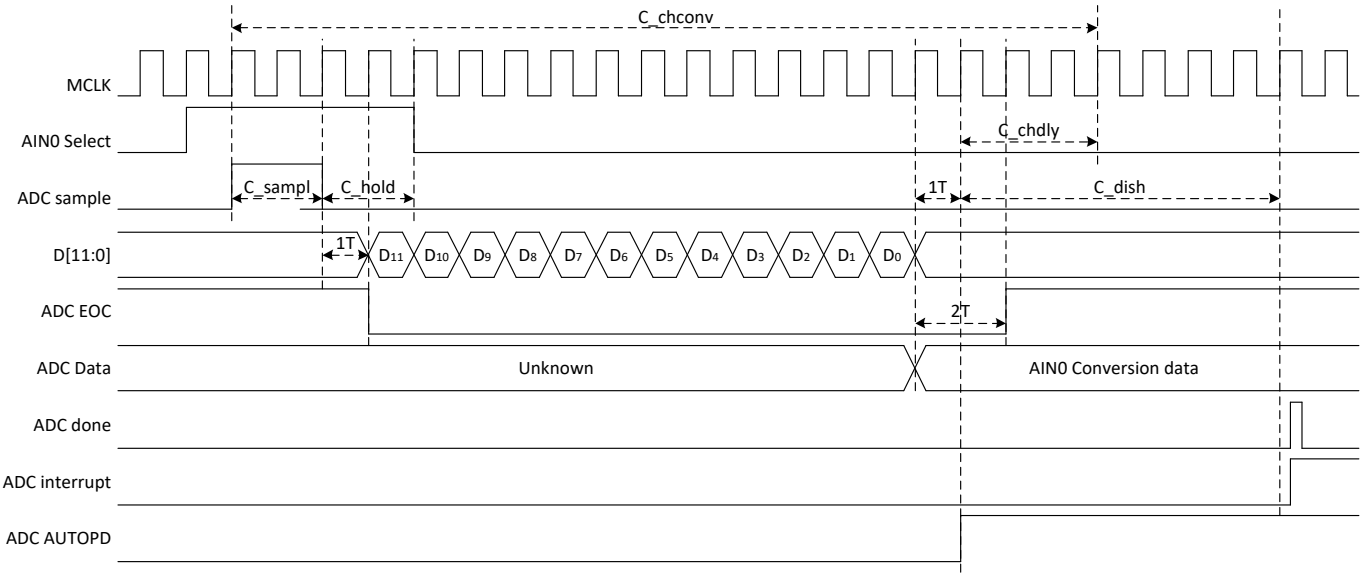
Power down select	PD_ADC	PD_LDO
XPD_ADCSEL = 0	AUTOPD	
XPD_ADCSEL = 1	XPD_ADC	
XPD_LDOSEL = 0		AUTOPD
XPD_LDOSEL = 1		XPD_LDO

XPD_ADC	XPD_LDO	AOTOPWDN	Power down select		ADC State
0	0	0	XPD_LDOSEL	X	ADC Operation mode with internal reference. (Reference from internal LDO ref. or VDD)
			XPD_ADCSEL	X	
0	1	0	XPD_LDOSEL	1	ADC Operation mode without internal reference. (Reference from external ref. or VDD)
			XPD_ADCSEL	X	
1	0	0	XPD_LDOSEL	1	Only internal LDO reference is active
			XPD_ADCSEL	X	
1	1	0	XPD_LDOSEL	1	Power-down mode (Whole ADC and internal LDO power-down)
			XPD_ADCSEL	1	
0	0	1	XPD_LDOSEL	0	ADC Operation mode with internal reference. ADC automatically enters the power-down mode when ADC scan conversion finishes. (Whole ADC and internal LDO power-down)
			XPD_ADCSEL	0	
0	1	1	XPD_LDOSEL	1	ADC Operation mode without internal reference. ADC automatically enters the power-down mode when ADC scan conversion finishes. (Whole ADC and internal LDO power-down)
			XPD_ADCSEL	0	

7.9.2 Auto Power down Mode

When AUTOPWDN of ADC_CTRL register is set and the conversion mode is single step or single scan conversion mode, the ADC will assert power down signal after ADC conversion is done.

- The ADC auto power down (AUTOPD) is low when ADC_EN = 0. The ADC auto power enable is high when ADC_EN = 1 and SWSTART = 1.
- When AUTOPWDN = 1, AUTOPD will be 1 after the conversion is completed. At this time, the controller needs to re-power on when it receives the next trigger, and it needs to re-execute power enable and wake-up time.
- When AUTOPWDN = 0, AUTOPD will be 0 after ADC_EN is clear.
- The effect of AUTOPWDN cannot be changed until the next trigger is received.



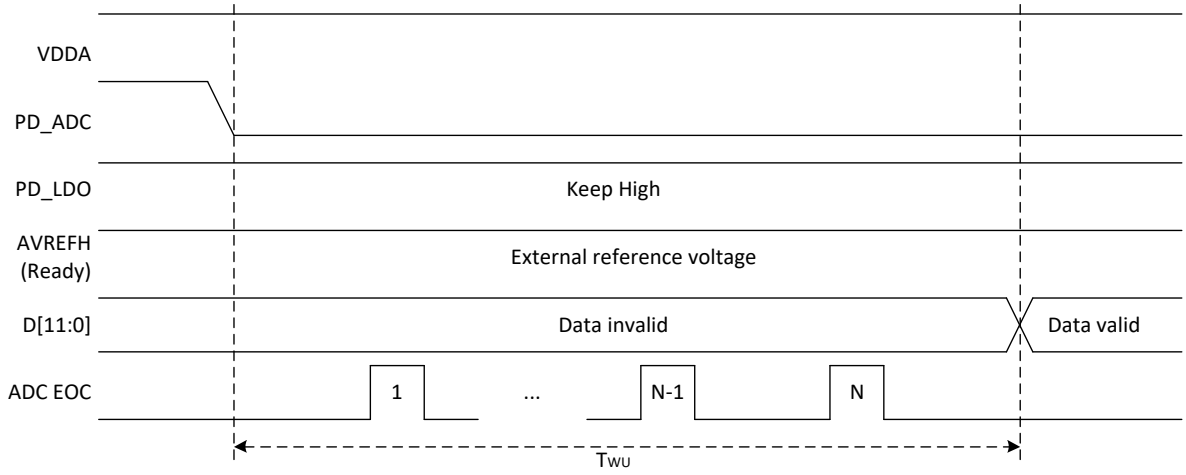
Symbol	Description	Register	Bit	Default (unit MCLK)
C_chdly	Delay time cycle for each channel	ADC_SMPR	CHDLYTime[7:0]	0x00 (1T)
C_dish	Sensing discharge cycle after sequence end	ADC_DISCHTIME	DISCHTIME[15:0]	0x0000 (1T)

7.9.3 Wake-up

Wakeup control by de-assert power down signal. The wake-up sequence depends on the reference source.

7.9.3.1 Wake-up with External Reference/VDD

- XPD_LDO is fixed value during ADC conversion.
- XREFSEL[1:0] = 2'b00, external reference pin
- XREFSEL[1:0] = 2'b01, VDD pin

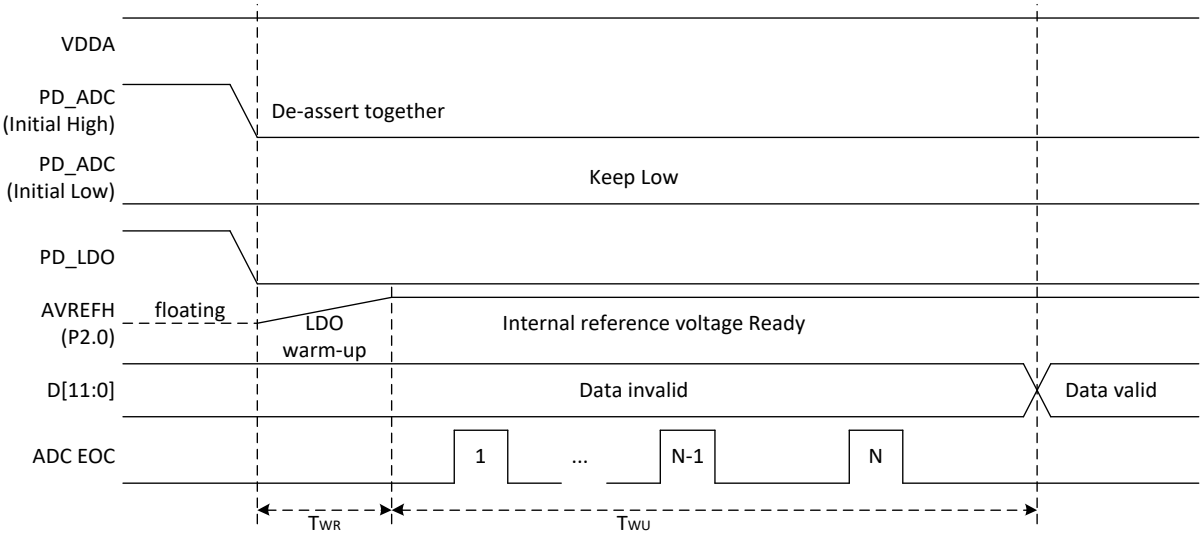


Symbol	Description	Wake-up Timing	Register value
C_pweren	ADC power enable cycle	T_{WR} = 0 ms	0 (unit MCLK)
C_wkup	ADC wakeup cycles of ADC conversion	T_{WU} = 3 conversion cycle (Minimum)	0x3 (unit conversion cycle)

✷ ***Note: At least 3 idle conversion cycles are required for the ADC engine warm-up.***

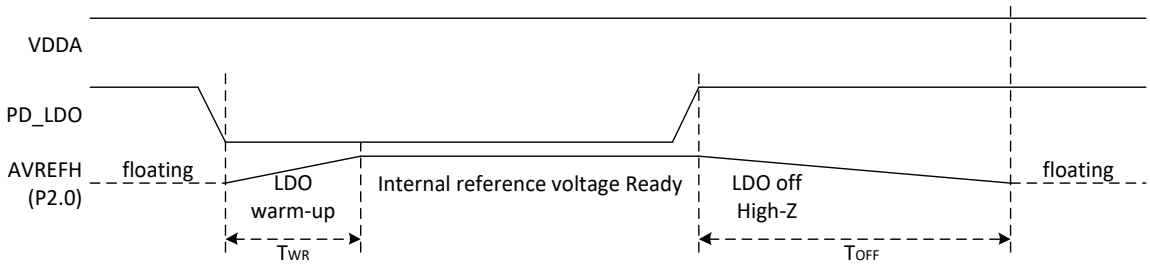
7.9.3.2 Wake-up with Internal Reference

- PD_ADC/PD_LDO can de-assert together.
- XREFSEL[1:0] = 2'b10, internal reference from internal LDO



Symbol	Description	Wake-up Timing	Register value
C_pweren	ADC power enable cycle	T_{WR} = 0.5 ms (Maximum)	0x1F40@16MHz (unit MCLK)
C_wkup	ADC wakeup cycles of ADC conversion	T_{WU} = 3 conversion cycle (Minimum)	0x3 (unit conversion cycle)

- Internal LDO on/off time



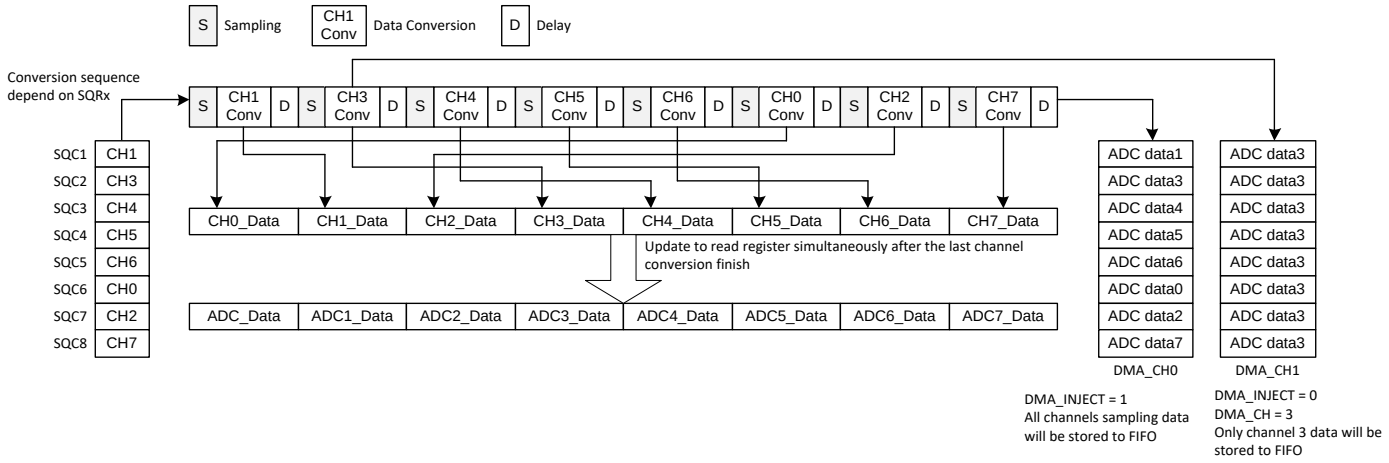
Symbol	Description	Condition	Timing
T_{WR}	Internal reference wake-up time	PD_LDO = 0 PD_ADC = 0 C_{AVREFH} = 1uF	0.5 ms (Maximum)
T_{OFF}	Internal reference off time	PD_LDO = 1 C_{AVREFH} = 1uF Output loading 1mA~10mA	5.5 ms (Maximum)

7.10 ADC DMA BUFFER

ADC supports up to four DMA channels. Each DMA channel consists of four registers, data port register, function control register, interrupt enable register, and interrupt status register. Each DMA channel has 16-frame DMA FIFO.

User can determine the scan sequence by programming the scan sequence register. When ADC conversion starts, the controller will latch the ADC data to DMA FIFO. Users can select the data that is injected from conversion data or from specific ADC channel to DMA FIFO by programming DMA_INJECT bit.

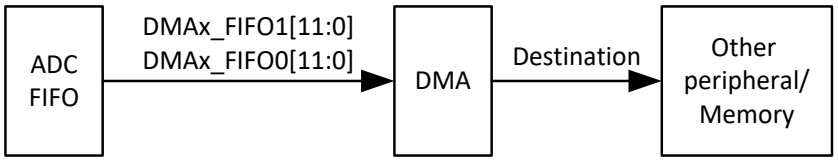
The following figure is an example that FIFO of DMA channel 0 is set to be injected from all ADC channels. FIFO of DMA channel 1 is set to be inject from ADC channel 3. DMA_CH in DMA control register selects the specific ADC channel to be latched.



✸ ***Note: ADC DMA FIFO is 16 layers, each layer can store two ADC data***

7.11 DMA MODE

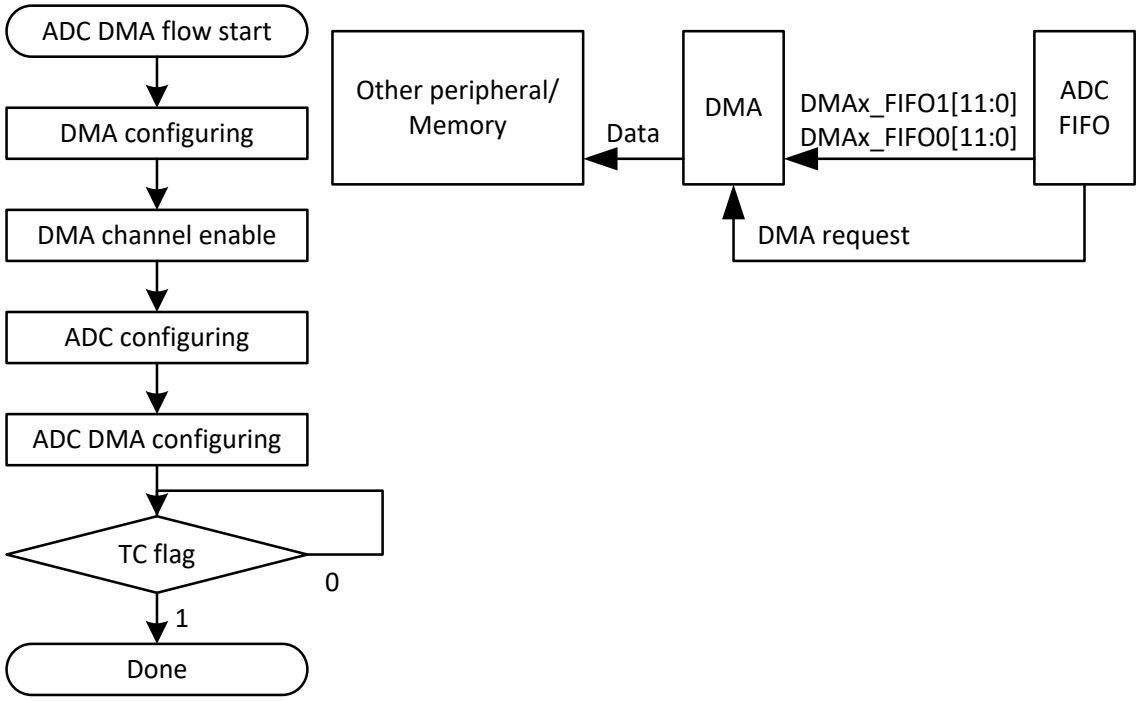
The ADC DMA mode is to use DMA engine to move data out ADC. Before the DMA transfer start, DMA engine must be set up first. In ADC DMA Mode, DMA receive data from ADC and send to other peripherals or memories.



✸ ***Note:***
1. ***The burst data cannot be larger than the FIFO size of ADC.***
2. ***ADC FIFO threshold level = Burst size.***

7.11.1 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.



7.11.2 Configuration

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

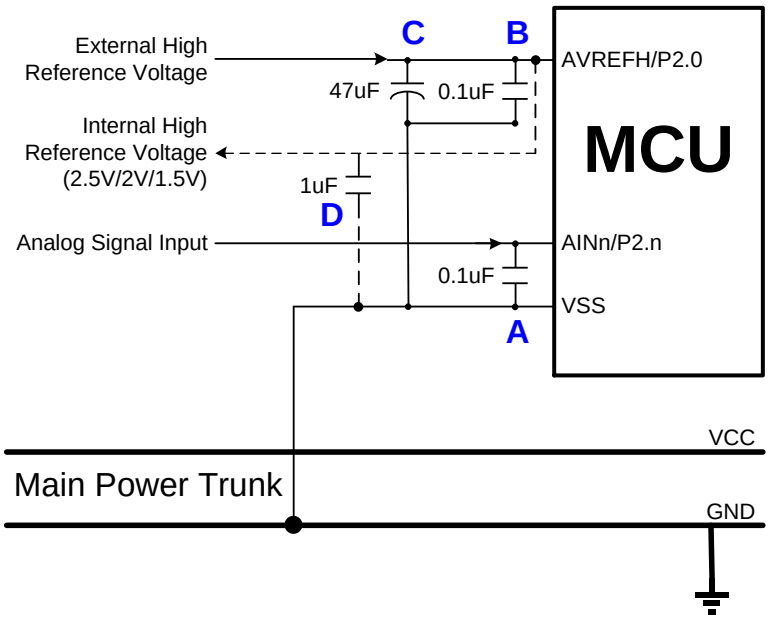
Because ADC DMA data port register has odd conversion data and even conversion data, the DMA transfer width is 32 bits and fetch two data as the same time.

Configuration	Register
SRCADDR = ADCn_DMAmDAT	DMA_Cn_SRCADDR
SRC_RS = ADC0_m SRC_HE=1	DMA_Cn_CFG
Set SRCAD_CTL=10 (Fixed) Set MODE = 1 (Peripheral)	DMA_Cn_CSR
DMA_EN = 1	ADCn_DMAmCTRL

DMAC setting for ADC data buffer (12-bit)				Recommend setting for ADC register (assume ADC FIFO depth == 8 x 12-bit)
Cn_CSR **SRC_WIDTH**	Cn_CSR **SRC_SIZE**	Cn_SIZE **TOT_SIZE**	Cn_CSR **DST_WIDTH**	**DMA_THD**
0x2 (word)	0x1 (burst=4)	N (word)	0x2 (word)	0x1 (8 data)
0x2 (word)	0x2 (burst=8)	N (word)	0x2 (word)	0x2 (16 data)
0x2 (word)	0x3 (burst=16)	N (word)	0x2 (word)	0x3 (full, 32 data)

✸ ***Note: Each Word has two 12-bit ADC data (odd and even).***

7.12 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.

If the ADC high reference voltage is from internal voltage source (2.5V/2V/1.5V), the internal high reference is connected to AVREFH pin (P2.0). The internal high reference source must be through a 1uF ”D” capacitor. 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.

7.13 ADC REGISTERS

Base Address: 0x4002 7000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	ADCn_DATA0	ADC Channel 0 Sensing Data register
0x0004	ADCn_DATA1	ADC Channel 1 Sensing Data register
0x0008	ADCn_DATA2	ADC Channel 2 Sensing Data register
0x000C	ADCn_DATA3	ADC Channel 3 Sensing Data register
0x0010	ADCn_DATA4	ADC Channel 4 Sensing Data register
0x0014	ADCn_DATA5	ADC Channel 5 Sensing Data register
0x0018	ADCn_DATA6	ADC Channel 6 Sensing Data register
0x001C	ADCn_DATA7	ADC Channel 7 Sensing Data register
0x0020	ADCn_DATA8	ADC Channel 8 Sensing Data register
0x0024	ADCn_DATA9	ADC Channel 9 Sensing Data register
0x0028	ADCn_DATA10	ADC Channel 10 Sensing Data register
0x002C	ADCn_DATA11	ADC Channel 11 Sensing Data register
0x0030	ADCn_DATA12	ADC Channel 12 Sensing Data register
0x0034	ADCn_DATA13	ADC Channel 13 Sensing Data register
0x0038	ADCn_DATA14	ADC Channel 14 Sensing Data register
0x003C	ADCn_DATA15	ADC Channel 15 Sensing Data register
0x0040	ADCn_DATA16	ADC Channel 16 Sensing Data register
0x0044	ADCn_DATA17	ADC Channel 17 Sensing Data register
0x0048 – 0x00FC	–	Reserved
0x0100	ADCn_CTRL	ADC Control register
0x0104	ADCn_MISC	ADC Miscellaneous Control register
0x0108	ADCn_IE	ADC Interrupt Enable register
0x010C	ADCn_RIS	ADC Raw Interrupt Status register
0x0110	ADCn_TPARM	ADC Timing Parameter register
0x0114	ADCn_SMPR	ADC Sampling Timing Parameter Control register
0x0118	ADCn_DISCHTIME	ADC Sensing Discharge Time register
0x011C	ADCn_PRE	ADC Clock Prescaler Control register
0x0120	ADCn_SQR0	ADC Scan Sequence register 0
0x0124	ADCn_SQR1	ADC Scan Sequence register 1
0x0128	ADCn_SQR2	ADC Scan Sequence register 2
0x012C	ADCn_SQR3	ADC Scan Sequence register 3
0x0130	ADCn_SQR4	ADC Scan Sequence register 4
0x0140	ADCn_EOCIE	ADC EOC Interrupt Enable register
0x0144	ADCn_THRDIE0	ADC Threshold Interrupt Enable register 0
0x0148	ADCn_THRDIE1	ADC Threshold Interrupt Enable register 1
0x014C	–	Reserved
0x0150	ADCn_EOCIS	ADC EOC Interrupt Status register
0x0154	ADCn_THRDIS0	ADC Threshold Interrupt Status register 0
0x0158	ADCn_THRDIS1	ADC Threshold Interrupt Status register 1
0x015C – 0x01FC	–	Reserved
0x0200	ADCn_DMA0DAT	ADC DMA0 Data Port register
0x0204	ADCn_DMA0CTRL	ADC DMA0 Control register
0x0208	ADCn_DMA0IE	ADC DMA0 Interrupt Enable register
0x020C	ADCn_DMA0IS	ADC DMA0 Interrupt Status register
0x0210	ADCn_DMA1DAT	ADC DMA1 Data Port register
0x0214	ADCn_DMA1CTRL	ADC DMA1 Control register
0x0218	ADCn_DMA1IE	ADC DMA1 Interrupt Enable register
0x021C	ADCn_DMA1IS	ADC DMA1 Interrupt Status register
0x0220	ADCn_DMA2DAT	ADC DMA2 Data Port register
0x0224	ADCn_DMA2CTRL	ADC DMA2 Control register
0x0228	ADCn_DMA2IE	ADC DMA2 Interrupt Enable register
0x022C	ADCn_DMA2IS	ADC DMA2 Interrupt Status register
0x0230	ADCn_DMA3DAT	ADC DMA3 Data Port register

0x0234	ADCn_DMA3CTRL	ADC DMA3 Control register
0x0238	ADCn_DMA3IE	ADC DMA3 Interrupt Enable register
0x023C	ADCn_DMA3IS	ADC DMA3 Interrupt Status register
0x0240 – 0x02FC	–	Reserved
0x0300	ADCn_THRHOLD0	ADC Threshold Detect 0 Programming register
0x0304	ADCn_THRHOLD1	ADC Threshold Detect 1 Programming register
0x0308	ADCn_THRHOLD2	ADC Threshold Detect 2 Programming register
0x030C	ADCn_THRHOLD3	ADC Threshold Detect 3 Programming register
0x0310	ADCn_THRHOLD4	ADC Threshold Detect 4 Programming register
0x0314	ADCn_THRHOLD5	ADC Threshold Detect 5 Programming register
0x0318	ADCn_THRHOLD6	ADC Threshold Detect 6 Programming register
0x031C	ADCn_THRHOLD7	ADC Threshold Detect 7 Programming register
0x0320	ADCn_THRHOLD8	ADC Threshold Detect 8 Programming register
0x0324	ADCn_THRHOLD9	ADC Threshold Detect 9 Programming register
0x0328	ADCn_THRHOLD10	ADC Threshold Detect 10 Programming register
0x032C	ADCn_THRHOLD11	ADC Threshold Detect 11 Programming register
0x0330	ADCn_THRHOLD12	ADC Threshold Detect 12 Programming register
0x0334	ADCn_THRHOLD13	ADC Threshold Detect 13 Programming register
0x0338	ADCn_THRHOLD14	ADC Threshold Detect 14 Programming register
0x033C	ADCn_THRHOLD15	ADC Threshold Detect 15 Programming register
0x0340	ADCn_THRHOLD16	ADC Threshold Detect 16 Programming register
0x0344	ADCn_THRHOLD17	ADC Threshold Detect 17 Programming register
0x0348	ADCn_THRHOLD18	ADC Threshold Detect 18 Programming register
0x034C	ADCn_THRHOLD19	ADC Threshold Detect 19 Programming register
0x0350	ADCn_THRHOLD20	ADC Threshold Detect 20 Programming register
0x0354	ADCn_THRHOLD21	ADC Threshold Detect 21 Programming register
0x0358	ADCn_THRHOLD22	ADC Threshold Detect 22 Programming register
0x035C	ADCn_THRHOLD23	ADC Threshold Detect 23 Programming register
0x0360	ADCn_THRHOLD24	ADC Threshold Detect 24 Programming register
0x0364	ADCn_THRHOLD25	ADC Threshold Detect 25 Programming register
0x0368	ADCn_THRHOLD26	ADC Threshold Detect 26 Programming register
0x036C	ADCn_THRHOLD27	ADC Threshold Detect 27 Programming register
0x0370	ADCn_THRHOLD28	ADC Threshold Detect 28 Programming register
0x0374	ADCn_THRHOLD29	ADC Threshold Detect 29 programming register
0x0378	ADCn_THRHOLD30	ADC Threshold Detect 30 Programming register
0x037C	ADCn_THRHOLD31	ADC Threshold Detect 31 Programming register
0x0380	ADCn_THRHOLD32	ADC Threshold Detect 32 Programming register
0x0384	ADCn_THRHOLD33	ADC Threshold Detect 33 Programming register
0x0388	ADCn_THRHOLD34	ADC Threshold Detect 34 Programming register
0x038C	ADCn_THRHOLD35	ADC Threshold Detect 35 Programming register

7.13.1 ADC n Channel m Sensing Data register (ADCn_DATAm) (n=0/m=0~17)

Address Offset: 0x00, 0x04, …, 0x44

✸ ***Note: The initial value of ADC Data buffer after reset is unknown.***

Bit	Field	Access	Initial	Description
31:12	–	–	0	Reserved
11:0	DATA	R	0	ADC channel n converter data

7.13.2 ADC n Control register (ADCn_CTRL) (n=0)

Address Offset: 0x100

✸ ***Note:***

1. ***When the Alternate function of GPIO is selected as the ADC channel. 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. ***If P2.0 is used as external reference voltage input pin or internal reference voltage output pin, user should set P2.0 as input mode inactive (no pull-down/up resistor enabled, Schmitt trigger disabled).***
3. ***If AIN0 channel is selected as internal 1.5V/2V/2.5V input channel, there is no any input pin from outside. In this time ADC reference voltage must be internal VDD, Not internal 1.5V/2V/2.5V.***
4. ***The pins of Port 2 configured as ADC input channel must be set as input mode, inactive (no pull-down/pull-up resistor enabled, ADC channel) with GPIO2_DIR, GPIO2_PULLEN and GPIO2 alternate function register by program to avoid current leakage.***

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20:16	SCANNUM	RW	0x11	ADC scan conversion channel numbers = SCANNUM[4:0] + 1 in each scan The number of the scan channels will be latched to execute conversion after software programming trigger. In the latching interval, any register programming will not be immediately updated until next trigger event. After ADCEN is set, the scan channels will be fixed, it will not be changed by re-programming the register. Valid value: 0~17
15:13	–	–	0	Reserved
12	AUTOPWDN	RW	0	ADC conversion mode 0: ADC still keeps powering on after finishing ADC conversion 1: ADC automatically enters the power-down mode when ADC scan conversion finishes If users want to power down ADC by software, users can set the bit to1'b1 and clear ADCEN to shut ADC power down. AUTOPWDN only supports when COVNMODE is set to single step and single scan conversion mode. The auto power-down mode will be latched to execute conversion after software programming trigger. In the latching interval, any register programming will not be immediately updated until next trigger event.
11:8	COVNMODE	RW	0	ADC conversion mode 0: Single step conversion mode 1: Single scan conversion mode 2: Continuous scan conversion mode Other: Reserved If write the COVNMODE when ADC is converting, the conversion mode will not change before converting of current channel is complete.
7:5	–	–	0	Reserved
4	SWSTART	W	0	Software programming to trigger ADC conversion, always 0 for read
3:1	–	–	0	Reserved
0	ADCEN	RW	0	ADC control enable 0: Disable 1: Enable Controller will enable the ADC hard macro to enter the normal operation mode if it is set. After ADC enable, ADC will execute the wakeup procedure. User could check the ADC_DONEIF by set SWSTART once, to make sure the wakeup procedure is complete.

7.13.3 ADC n Miscellaneous Control register (ADCn_MISC) (n=0)

Address Offset: 0x104

Bit	Field	Access	Initial	Description
31	XCAL	RW	0	Calibration control signal. It is high in calibration mode. 0: ADC normal conversion mode 1: ADC calibration mode
30:24	XBVOS	RW	0x3F	Offset calibration code option
23:22	–	–	0	Reserved
21:16	XTEST	RW	0x08	Test pin
15:14	–	–	0	Reserved
13:12	XREFSEL	RW	0x1	ADC reference option select 0: ADC in external reference mode (VSS~AVREFH) 1: ADC in VDD/GND reference mode (VSS~VDD) 2: ADC in internal reference mode (VSS~IREF) 3: Reserved
11:10	–	–	0	Reserved
9:8	XSEL_LDO	RW	0	Internal reference LDO option select 0: Internal reference voltage= 2.5V 1: Internal reference voltage = 2.0V 2: Internal reference voltage = 1.5V Other: Reserved
7	–	–	0	Reserved
6	XPD_ADCSEL	RW	0	Whole ADC power-down control signal select 0: From ADCEN and AUTOPWDN bits 1: From XPD_ADC bit
5	XPD_LDOSEL	RW	1	Internal Reference LDO power-down control signal select 0: From ADCEN and AUTOPWDN bits 1: From XPD_LDO bit
4:3	–	–	0	Reserved
2	XPD_ADC	RW	1	Power-down control signal for ADC core 0: Normal operating mode 1: Power-down mode
1	XPD_LDO	RW	1	Power-down control signal for internal reference LDO 0: Normal operating mode 1: Power-down mode
0	–	–	0	Reserved

7.13.4 ADC n Interrupt Enable register (ADCn_IE) (n=0)

Address offset: 0x108

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1	ADC_STOPIE	RW	0	ADC scan conversion stop interrupt enable 0: Disable 1: Enable
0	ADC_DONEIE	RW	0	ADC scan conversion finish interrupt enable 0: Disable 1: Enable

7.13.5 ADC n Raw Interrupt Status register (ADCn_RIS) (n=0)

Address offset: 0x10C

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	DMACH3IF	RW1C	0	DMA Channel 3 interrupt flag DMACH3IF will not change, before DMA_OVRIE or DMA_UDRIE are enable.
6	DMACH2IF	RW1C	0	DMA Channel 2 interrupt flag DMACH2IF will not change, before DMA_OVRIE or DMA_UDRIE are enable.
5	DMACH1IF	RW1C	0	DMA Channel 1 interrupt flag DMACH1IF will not change, before DMA_OVRIE or DMA_UDRIE are enable.
4	DMACH0IF	RW1C	0	DMA Channel 0 interrupt flag DMACH0IF will not change, before DMA_OVRIE or DMA_UDRIE are enable.
3:2	–	–	0	Reserved
1	ADC_STOPIF	RW1C	0	ADC converter stop interrupt flag When the ADCEN bit is set to 1'b0 during the ADC conversion, it is asserted when the ADC controller is already at the idle state.
0	ADC_DONEIF	RW1C	0	ADC converter done interrupt flag It is asserted when the ADC conversion is done.

7.13.6 ADC n Timing Parameter register (ADCn_TPARM) (n=0)

Address offset: 0x110

Bit	Field	Access	Initial	Description
31:19	–	–	0	Reserved
18:16	WKUPTIME	RW	0x3	ADC wakeup cycles of ADC conversion 0: Disable 1: Waiting 1 ADC conversion cycle 2: Waiting 2 ADC conversion cycle 3: Waiting 3 ADC conversion cycle 4: Waiting 4 ADC conversion cycle 5: Waiting 5 ADC conversion cycle 6: Waiting 6 ADC conversion cycle 7: Waiting 7 ADC conversion cycle *Should not write the value less than 3 to WKUPTIME.
15:0	PWRENTIME	RW	0x20	ADC power enable cycle number of MCLK C_pweren = (PWRENTIME +1) * T_{MCLK} It should be programmed before the ADCEN bit is active. When the ADCEN bit is active, the value should be frozen.

7.13.7 ADC n Sampling Timing Parameter Control register (ADCn_SMPR) (n=0)

Address Offset: 0x114

Bit	Field	Access	Initial	Description
31:24	CHDLYTIME	RW	0	Delay time cycle number of MCLK for each channel sampling C_chdly = (CHDLYTIME +1) * T_{MCLK} It should be programmed before the ADCEN bit is active. When the ADCEN bit is active, the value should be frozen.
23	–	–	0	Reserved
22:20	CHSELTIME	RW	0	Analog input channel change discharge cycle number of MCLK C_chsel = (CHSELTIME +1) * T_{MCLK} It should be programmed before the ADCEN bit is active. When the ADCEN bit is active, the value should be frozen.
19:16	HOLDTIME	RW	0x2	Analog input channel hold cycle number of MCLK C_hold = HOLDTIME * T_{MCLK} It should be programmed before the ADCEN bit is active. When the

Bit	Field	Access	Initial	Description
				ADCEN bit is active, the value should be frozen.
15:12	–	–	0	Reserved
11:8	SAMPLTIME	RW	0	Analog input channel sample clock high level cycle number of MCLK C_sampl = (SAMPLTIME +1) * T_{MCLK} It should be programmed before the ADCEN bit is active. When the ADCEN bit is active, the value should be frozen.
7:0	SETTLTIME	RW	0x0	Analog input channel settling cycle number of MCLK C_settl = (SETTLTIME +1) * T_{MCLK} It should be programmed before the ADCEN bit is active. When the ADCEN bit is active, the value should be frozen.

7.13.8 ADC n Sensing Discharge Timr register (ADCn_DISCHTIME) (n=0)

Address Offset: 0x118

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	DISCHTIME	RW	0	Sensing Discharge cycle number of MCLK C_disch = (DISCHTIME +1) * T_{MCLK} It should be programmed before any ADCEN bit is active. When ADCEN bit is active, the value should be frozen.

7.13.9 ADC n Clock Prescaler Control register (ADCn_PRE) (n=0)

Address Offset: 0x11C

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	MCLKDIV	RW	0x5	MCLK divider factor. MCLK = PCLK/(MCLKDIV+1) MCLKDIV = 0 is invalid. The valid range is 0x01~0xFF.

7.13.10 ADC n Scan Sequence register 0 (ADCn_SQR0) (n=0)

Address Offset: 0x120

Bit	Field	Access	Initial	Description
31:29	–	–	0	Reserved
28:24	SQC4	RW	0x3	The 4th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
23:21	–	–	0	Reserved
20:16	SQC3	RW	0x2	The 3rd conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
15:13	–	–	0	Reserved
12:8	SQC2	RW	1	The 2nd conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
7:5	–	–	0	Reserved
4:0	SQC1	RW	0	The 1st conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden

7.13.11 ADC n Scan Sequence register 1 (ADCn_SQR1) (n=0)

Address Offset: 0x124

Bit	Field	Access	Initial	Description
31:29	–	–	0	Reserved
28:24	SQC8	RW	0x7	The 8th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
23:21	–	–	0	Reserved

Bit	Field	Access	Initial	Description
20:16	SQC7	RW	0x6	The 7th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
15:13	–	–	0	Reserved
12:8	SQC6	RW	0x5	The 6th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
7:5	–	–	0	Reserved
4:0	SQC5	RW	0x4	The 5th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden

7.13.12 ADC n Scan Sequence register 2 (ADCn_SQR2) (n=0)

Address Offset: 0x128

Bit	Field	Access	Initial	Description
31:29	–	–	0	Reserved
28:24	SQC12	RW	0xB	The 12th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
23:21	–	–	0	Reserved
20:16	SQC11	RW	0xA	The 11th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
15:13	–	–	0	Reserved
12:8	SQC10	RW	0x9	The 10th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
7:5	–	–	0	Reserved
4:0	SQC9	RW	0x8	The 9th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden

7.13.13 ADC n Scan Sequence register 3 (ADCn_SQR3) (n=0)

Address Offset: 0x12C

Bit	Field	Access	Initial	Description
31:29	–	–	0	Reserved
28:24	SQC16	RW	0xF	The 16th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
23:21	–	–	0	Reserved
20:16	SQC15	RW	0xE	The 15th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
15:13	–	–	0	Reserved
12:8	SQC14	RW	0xD	The 14th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
7:5	–	–	0	Reserved
4:0	SQC13	RW	0xC	The 13th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden

7.13.14 ADC n Scan Sequence register 4 (ADCn_SQR4) (n=0)

Address Offset: 0x130

Bit	Field	Access	Initial	Description
31:29	–	–	0	Reserved
28:24	–	–	0x13	Reserved
23:21	–	–	0	Reserved
20:16	–	–	0x12	Reserved
15:13	–	–	0	Reserved
12:8	SQC18	RW	0x11	The 18th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden
7:5	–	–	0	Reserved

Bit	Field	Access	Initial	Description
4:0	SQC17	RW	0x10	The 17th conversion channel selection for the scan mode 5'h0~5'h11: ADC channel 0~17, Others: forbidden

7.13.15 ADC n EOC Interrupt Enable register (ADCn_EOCIE) (n=0)

Address Offset: 0x140

Bit	Field	Access	Initial	Description
31:18	–	–	0	Reserved
17	CH17_EOCIE	RW	0	ADC channel 17 EOC interrupt enable 0: Disable 1: Enable
16	CH16_EOCIE	RW	0	ADC channel 16 EOC interrupt enable 0: Disable 1: Enable
15	CH15_EOCIE	RW	0	ADC channel 15 EOC interrupt enable 0: Disable 1: Enable
14	CH14_EOCIE	RW	0	ADC channel 14 EOC interrupt enable 0: Disable 1: Enable
13	CH13_EOCIE	RW	0	ADC channel 13 EOC interrupt enable 0: Disable 1: Enable
12	CH12_EOCIE	RW	0	ADC channel 12 EOC interrupt enable 0: Disable 1: Enable
11	CH11_EOCIE	RW	0	ADC channel 11 EOC interrupt enable 0: Disable 1: Enable
10	CH10_EOCIE	RW	0	ADC channel 10 EOC interrupt enable 0: Disable 1: Enable
9	CH9_EOCIE	RW	0	ADC channel 9 EOC interrupt enable 0: Disable 1: Enable
8	CH8_EOCIE	RW	0	ADC channel 8 EOC interrupt enable 0: Disable 1: Enable
7	CH7_EOCIE	RW	0	ADC channel 7 EOC interrupt enable 0: Disable 1: Enable
6	CH6_EOCIE	RW	0	ADC channel 6 EOC interrupt enable 0: Disable 1: Enable
5	CH5_EOCIE	RW	0	ADC channel 5 EOC interrupt enable 0: Disable 1: Enable
4	CH4_EOCIE	RW	0	ADC channel 4 EOC interrupt enable 0: Disable 1: Enable
3	CH3_EOCIE	RW	0	ADC channel 3 EOC interrupt enable 0: Disable 1: Enable
2	CH2_EOCIE	RW	0	ADC channel 2 EOC interrupt enable 0: Disable 1: Enable
1	CH1_EOCIE	RW	0	ADC channel 1 EOC interrupt enable 0: Disable 1: Enable
0	CH0_EOCIE	RW	0	ADC channel 0 EOC interrupt enable 0: Disable 1: Enable

7.13.16 ADC n Threshold Interrupt Enable register 0 (ADCn_THRDIE0) (n=0)

Address Offset: 0x144

Bit	Field	Access	Initial	Description
31	THRD31_IE	RW	0	Threshold detection 31 interrupt enable 0: Disable 1: Enable
30	THRD30_IE	RW	0	Threshold detection 30 interrupt enable 0: Disable 1: Enable
29	THRD29_IE	RW	0	Threshold detection 29 interrupt enable 0: Disable 1: Enable
28	THRD28_IE	RW	0	Threshold detection 28 interrupt enable 0: Disable 1: Enable
27	THRD27_IE	RW	0	Threshold detection 27 interrupt enable 0: Disable 1: Enable
26	THRD26_IE	RW	0	Threshold detection 26 interrupt enable 0: Disable 1: Enable
25	THRD25_IE	RW	0	Threshold detection 25 interrupt enable 0: Disable 1: Enable
24	THRD24_IE	RW	0	Threshold detection 24 interrupt enable 0: Disable 1: Enable
23	THRD23_IE	RW	0	Threshold detection 23 interrupt enable 0: Disable 1: Enable
22	THRD22_IE	RW	0	Threshold detection 22 interrupt enable 0: Disable 1: Enable
21	THRD21_IE	RW	0	Threshold detection 21 interrupt enable 0: Disable 1: Enable
20	THRD20_IE	RW	0	Threshold detection 20 interrupt enable 0: Disable 1: Enable
19	THRD19_IE	RW	0	Threshold detection 19 interrupt enable 0: Disable 1: Enable
18	THRD18_IE	RW	0	Threshold detection 18 interrupt enable 0: Disable 1: Enable
17	THRD17_IE	RW	0	Threshold detection 17 interrupt enable 0: Disable 1: Enable
16	THRD16_IE	RW	0	Threshold detection 16 interrupt enable 0: Disable 1: Enable
15	THRD15_IE	RW	0	Threshold detection 15 interrupt enable 0: Disable 1: Enable
14	THRD14_IE	RW	0	Threshold detection 14 interrupt enable 0: Disable 1: Enable
13	THRD13_IE	RW	0	Threshold detection 13 interrupt enable 0: Disable 1: Enable
12	THRD12_IE	RW	0	Threshold detection 12 interrupt enable

Bit	Field	Access	Initial	Description
				0: Disable 1: Enable
11	THRD11_IE	RW	0	Threshold detection 11 interrupt enable 0: Disable 1: Enable
10	THRD10_IE	RW	0	Threshold detection 10 interrupt enable 0: Disable 1: Enable
9	THRD9_IE	RW	0	Threshold detection 9 interrupt enable 0: Disable 1: Enable
8	THRD8_IE	RW	0	Threshold detection 8 interrupt enable 0: Disable 1: Enable
7	THRD7_IE	RW	0	Threshold detection 7 interrupt enable 0: Disable 1: Enable
6	THRD6_IE	RW	0	Threshold detection 6 interrupt enable 0: Disable 1: Enable
5	THRD5_IE	RW	0	Threshold detection 5 interrupt enable 0: Disable 1: Enable
4	THRD4_IE	RW	0	Threshold detection 4 interrupt enable 0: Disable 1: Enable
3	THRD3_IE	RW	0	Threshold detection 3 interrupt enable 0: Disable 1: Enable
2	THRD2_IE	RW	0	Threshold detection 2 interrupt enable 0: Disable 1: Enable
1	THRD1_IE	RW	0	Threshold detection 1 interrupt enable 0: Disable 1: Enable
0	THRD0_IE	RW	0	Threshold detection 0 interrupt enable 0: Disable 1: Enable

7.13.17 ADC n Threshold Interrupt Enable register 1 (ADCn_THRDIE1) (n=0)

Address Offset: 0x148

Bit	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3	THRD35_IE	RW	0	Threshold detection 35 interrupt enable 0: Disable 1: Enable
2	THRD34_IE	RW	0	Threshold detection 34 interrupt enable 0: Disable 1: Enable
1	THRD33_IE	RW	0	Threshold detection 33 interrupt enable 0: Disable 1: Enable
0	THRD32_IE	RW	0	Threshold detection 32 interrupt enable 0: Disable 1: Enable

7.13.18 ADC n EOC Interrupt Status register (ADCn_EOCIS) (n=0)

Address Offset: 0x150

Bit	Field	Access	Initial	Description
31:18	–	–	0	Reserved
17	CH17_EOCIF	RW1C	0	ADC channel 17 EOC interrupt flag
16	CH16_EOCIF	RW1C	0	ADC channel 16 EOC interrupt flag
15	CH15_EOCIF	RW1C	0	ADC channel 15 EOC interrupt flag
14	CH14_EOCIF	RW1C	0	ADC channel 14 EOC interrupt flag
13	CH13_EOCIF	RW1C	0	ADC channel 13 EOC interrupt flag
12	CH12_EOCIF	RW1C	0	ADC channel 12 EOC interrupt flag
11	CH11_EOCIF	RW1C	0	ADC channel 11 EOC interrupt flag
10	CH10_EOCIF	RW1C	0	ADC channel 10 EOC interrupt flag
9	CH9_EOCIF	RW1C	0	ADC channel 9 EOC interrupt flag
8	CH8_EOCIF	RW1C	0	ADC channel 8 EOC interrupt flag
7	CH7_EOCIF	RW1C	0	ADC channel 7 EOC interrupt flag
6	CH6_EOCIF	RW1C	0	ADC channel 6 EOC interrupt flag
5	CH5_EOCIF	RW1C	0	ADC channel 5 EOC interrupt flag
4	CH4_EOCIF	RW1C	0	ADC channel 4 EOC interrupt flag
3	CH3_EOCIF	RW1C	0	ADC channel 3 EOC interrupt flag
2	CH2_EOCIF	RW1C	0	ADC channel 2 EOC interrupt flag
1	CH1_EOCIF	RW1C	0	ADC channel 1 EOC interrupt flag
0	CH0_EOCIF	RW1C	0	ADC channel 0 EOC interrupt flag

7.13.19 ADC n Threshold Interrupt Status register 0 (ADCn_THRDIS0) (n=0)

Address Offset: 0x154

Bit	Field	Access	Initial	Description
31	THRD31_IF	RW1C	0	Threshold detection 31 interrupt flag
30	THRD30_IF	RW1C	0	Threshold detection 30 interrupt flag
29	THRD29_IF	RW1C	0	Threshold detection 29 interrupt flag
28	THRD28_IF	RW1C	0	Threshold detection 28 interrupt flag
27	THRD27_IF	RW1C	0	Threshold detection 27 interrupt flag
26	THRD26_IF	RW1C	0	Threshold detection 26 interrupt flag
25	THRD25_IF	RW1C	0	Threshold detection 25 interrupt flag
24	THRD24_IF	RW1C	0	Threshold detection 24 interrupt flag
23	THRD23_IF	RW1C	0	Threshold detection 23 interrupt flag
22	THRD22_IF	RW1C	0	Threshold detection 22 interrupt flag
21	THRD21_IF	RW1C	0	Threshold detection 21 interrupt flag
20	THRD20_IF	RW1C	0	Threshold detection 20 interrupt flag
19	THRD19_IF	RW1C	0	Threshold detection 19 interrupt flag
18	THRD18_IF	RW1C	0	Threshold detection 18 interrupt flag
17	THRD17_IF	RW1C	0	Threshold detection 17 interrupt flag
16	THRD16_IF	RW1C	0	Threshold detection 16 interrupt flag
15	THRD15_IF	RW1C	0	Threshold detection 15 interrupt flag
14	THRD14_IF	RW1C	0	Threshold detection 14 interrupt flag
13	THRD13_IF	RW1C	0	Threshold detection 13 interrupt flag
12	THRD12_IF	RW1C	0	Threshold detection 12 interrupt flag
11	THRD11_IF	RW1C	0	Threshold detection 11 interrupt flag
10	THRD10_IF	RW1C	0	Threshold detection 10 interrupt flag
9	THRD9_IF	RW1C	0	Threshold detection 9 interrupt flag
8	THRD8_IF	RW1C	0	Threshold detection 8 interrupt flag
7	THRD7_IF	RW1C	0	Threshold detection 7 interrupt flag
6	THRD6_IF	RW1C	0	Threshold detection 6 interrupt flag
5	THRD5_IF	RW1C	0	Threshold detection 5 interrupt flag
4	THRD4_IF	RW1C	0	Threshold detection 4 interrupt flag
3	THRD3_IF	RW1C	0	Threshold detection 3 interrupt flag
2	THRD2_IF	RW1C	0	Threshold detection 2 interrupt flag

Bit	Field	Access	Initial	Description
1	THRD1_IF	RW1C	0	Threshold detection 1 interrupt flag
0	THRD0_IF	RW1C	0	Threshold detection 0 interrupt flag

7.13.20 ADC n Threshold Interrupt Status register 1 (ADCn_THRDIS1) (n=0)

Address Offset: 0x158

Bit	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3	THRD35_IF	RW1C	0	Threshold detection 35 interrupt flag
2	THRD34_IF	RW1C	0	Threshold detection 34 interrupt flag
1	THRD33_IF	RW1C	0	Threshold detection 33 interrupt flag
0	THRD32_IF	RW1C	0	Threshold detection 32 interrupt flag

7.13.21 ADC n DMA 0~3 Data Port register (ADCn_DMA0~3DAT) (n=0)

Address Offset: 0x200, 0x210, 0x220, 0x230

Each data port register consist of two conversion data, which represents the even and odd data in FIFO. The number of FIFO layer is 16, it can store 32 conversion data at most. DMA_DAT register will not store new data after storing 32 conversion data.

Bit	Field	Access	Initial	Description
31:28	–	–	0	Reserved
27:16	DMA_FIFO1	R	0	Even conversion data
15:12	–	–	0	Reserved
11:0	DMA_FIFO0	R	0	Odd conversion data

7.13.22 ADC n DMA0~3 Control register (ADCn_DMA0~3CTRL) (n=0)

Address Offset: 0x204, 0x214, 0x224, 0x234

Bit	Field	Access	Initial	Description
31:26	–	–	0	Reserved
25:24	DMA_INJECT	RW	0	ADC DMA injected data mode 0: Injected data by DMA_CH select 1: Injected data for all conversion Other: Reserved
23:17	–	–	0	Reserved
16	DMA_EN	RW	0	DMA enable 0: Disable 1: Enable
15:10	–	–	0	Reserved
9:8	DMA_THD	RW	0x1	DMA threshold configuration 0: Issue DMA request after 4 data in FIFO 1: Issue DMA request after 8 data in FIFO 2: Issue DMA request after 16 data in FIFO 3: Issue DMA request after full
7:5	–	–	0	Reserved
4:0	DMA_CH	RW	0	ADC channel select for DMA 0: ADC channel 0 for DMA 1: ADC channel 1 for DMA 2: ADC channel 2 for DMA 3: ADC channel 3 for DMA 4: ADC channel 4 for DMA 5: ADC channel 5 for DMA 6: ADC channel 6 for DMA 7: ADC channel 7 for DMA 8: ADC channel 8 for DMA

Bit	Field	Access	Initial	Description
				9: ADC channel 9 for DMA 10: ADC channel 10 for DMA 11: ADC channel 11 for DMA 12: ADC channel 12 for DMA 13: ADC channel 13 for DMA 14: ADC channel 14 for DMA 15: ADC channel 15 for DMA 16: ADC channel 16 for DMA 17: ADC channel 17 for DMA Other: Reserved

7.13.23 ADC n DMA0~3 Interrupt Enable register (ADCn_DMA0~3IE) (n=0)

Address Offset: 0x208, 0x218, 0x228, 0x238

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1	DMA_OVRIE	RW	0	DMA overrun interrupt enable 0: Disable 1: Enable
0	DMA_UDRIE	RW	0	DMA underrun interrupt enable 0: Disable 1: Enable

7.13.24 ADC n DMA0~3 Interrupt Status register (ADCn_DMA0~3IS) (n=0)

Address Offset: 0x20C, 0x21C, 0x22C, 0x23C

Bit	Field	Access	Initial	Description
31:29	–	–	0	Reserved
28:24	FIFO_DAT_CNT	R	0	DMA FIFO count After store 32 data, the FIFO_DAT_CNT will become 0. If FIFO_DAT_CNT=1, it will not become 0 after reading DMA_DAT. If FIFO_DAT_CNT=2, it will become 0 after reading DMA_DAT.
23:19	–	–	0	Reserved
18	WPTR_FULL	R	0	DMA FIFO full
17	FIFO_HFULL	R	0	DMA FIFO full according to DMA_THD
16	RPTR_EMPTY	R	1	DMA FIFO empty
15:2	–	–	0	Reserved
1	DMA_OVRIF	RW1C	0	DMA FIFO overrun interrupt flag
0	DMA_UDRIF	RW1C	0	DMA FIFO underrun interrupt flag

7.13.25 ADC n Threshold Detect 0~35 Programming register (ADCn_THRHOLD0~35) (n=0)

Address Offset: 0x300, 0x304, …., 0x38C

Bit	Field	Access	Initial	Description
31	THRHD_MODE	RW	0	Threshold detecting mode 0: Under the threshold mode for LTR 1: Over the threshold mode for HTR
30:21	–	–	0	Reserved
20:16	THRHD_CH	RW	0	Threshold detect for ADC channels 0: Threshold detect for ADC channel 0 1: Threshold detect for ADC channel 1

Bit	Field	Access	Initial	Description
				2: Threshold detect for ADC channel 2 3: Threshold detect for ADC channel 3 4: Threshold detect for ADC channel 4 5: Threshold detect for ADC channel 5 6: Threshold detect for ADC channel 6 7: Threshold detect for ADC channel 7 8: Threshold detect for ADC channel 8 9: Threshold detect for ADC channel 9 10: Threshold detect for ADC channel 10 11: Threshold detect for ADC channel 11 12: Threshold detect for ADC channel 12 13: Threshold detect for ADC channel 13 14: Threshold detect for ADC channel 14 15: Threshold detect for ADC channel 15 16: Threshold detect for ADC channel 16 17: Threshold detect for ADC channel 17 Other: Reserved Each ADC channel can set two threshold detect programming registers for HTR and LTR.
15:12	–	–	0	Reserved
11:0	THRHD_VAL	RW	0	Threshold detect value programming register(HTR/LTR) **Higher threshold register (HTR):** When THRHD_MODE = 1, the register (THRHD_VAL) is a higher threshold programmable register for over threshold detection function. If all bits of THRHD_VAL are 1, it can disable the threshold detection function. **Lower threshold register (LTR):** When THRHD_MODE = 0, the register (THRHD_VAL) is a lower threshold programmable register for under threshold detection function. If all bits of THRHD_VAL are 0, it can disable the threshold detection function.

8 16-BIT TIMER WITH CAPTURE FUNCTION

8.1 OVERVIEW

The microcontroller builds in nine 16-bit timers (CT16B0~CT16B8). 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	CT16B2	CT16B3	CT16B4	CT16B5
Clock Source	HCLK PLLCLK					HCLK PLLCLK ELS
Counter Mode	Up/Down/ Center-aligned counting	Up/Down/ Center-aligned counting	Up/Down/ Center-aligned counting	Up-counting	Up-counting	Up/Down/ Center-aligned counting
PWM	PWM0/1/2/3+ PWM0N/1N/2N/3N	PWM0/1/2/3	PWM0/1/2/3	PWM0/1+ PWM0N/1N	PWM0/1+ PWM0N/1N	PWM0/1/2/3
PCLK	max 192MHz	max 192MHz	max 192MHz	max 192MHz	max 192MHz	max 192MHz

	CT16B6	CT16B7	CT16B8
Clock Source	HCLK PLLCLK		
Counter Mode	Up-counting	Up-counting	Up-counting
PWM	-	-	PWM0/1/…/11
PCLK	max 192MHz	max 192MHz	max 192MHz

The 16-bit timer (CT16B0~CT16B8) clock is enabled by CT16BnCLKEN bit of SCU APB Clock Control register (SCU_APB0CLKG, SCU_APB1CLKG). Enabled by CT16BnCLKEN bit of SCU APB Sleep Mode Clock Control register (SCU_SLP_APB0CLKG, SCU_SLP_APB1CLKG) in sleep mode.

The 16-bit timer (CT16B0~CT16B8) PCLK is selected by CT16BnCLKSEL bit of SCU_CLKSEL register.

8.2 FEATURES

- Seven 16-bit counter/timers with a programmable 8-bit prescaler.
- Two 16-bit timers with a programmable 8-bit prescaler. (CT16B6/CT16B7)
- Counter or timer operation
- Five 16-bit capture channels (CT16B0/1/2/4/5 CAP0) 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.
- 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 12 (CT16B8), 4 (CT16B0, CT16B1, CT16B2, CT16B5), or 2 (CT16B3, CT16B4) 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 inverse waveform of the PWM signals, and builds in programmable dead-band function.
- For CT16B3 and CT16B4, up to 2 complete inverse waveform of the PWM signals, and builds in programmable dead-band function.
- Break function for triggers to stop PWM. (CT16B0)
- Supply DMA transfer

8.3 PIN DESCRIPTION

Pin Name	Type	Description	GPIO Configuration
CT16Bn_BRK	I	Break signal input	Depends on AFIO register
CT16Bn_CAP0	I	Capture channel input 0	Depends on AFIO register
CT16Bn_PWMx	O	Output channel x of Match/PWM output.	Depends on AFIO register
CT16Bn_PWMxN	O	Inverse Output channel of Match/PWMx output.	Depends on AFIO register

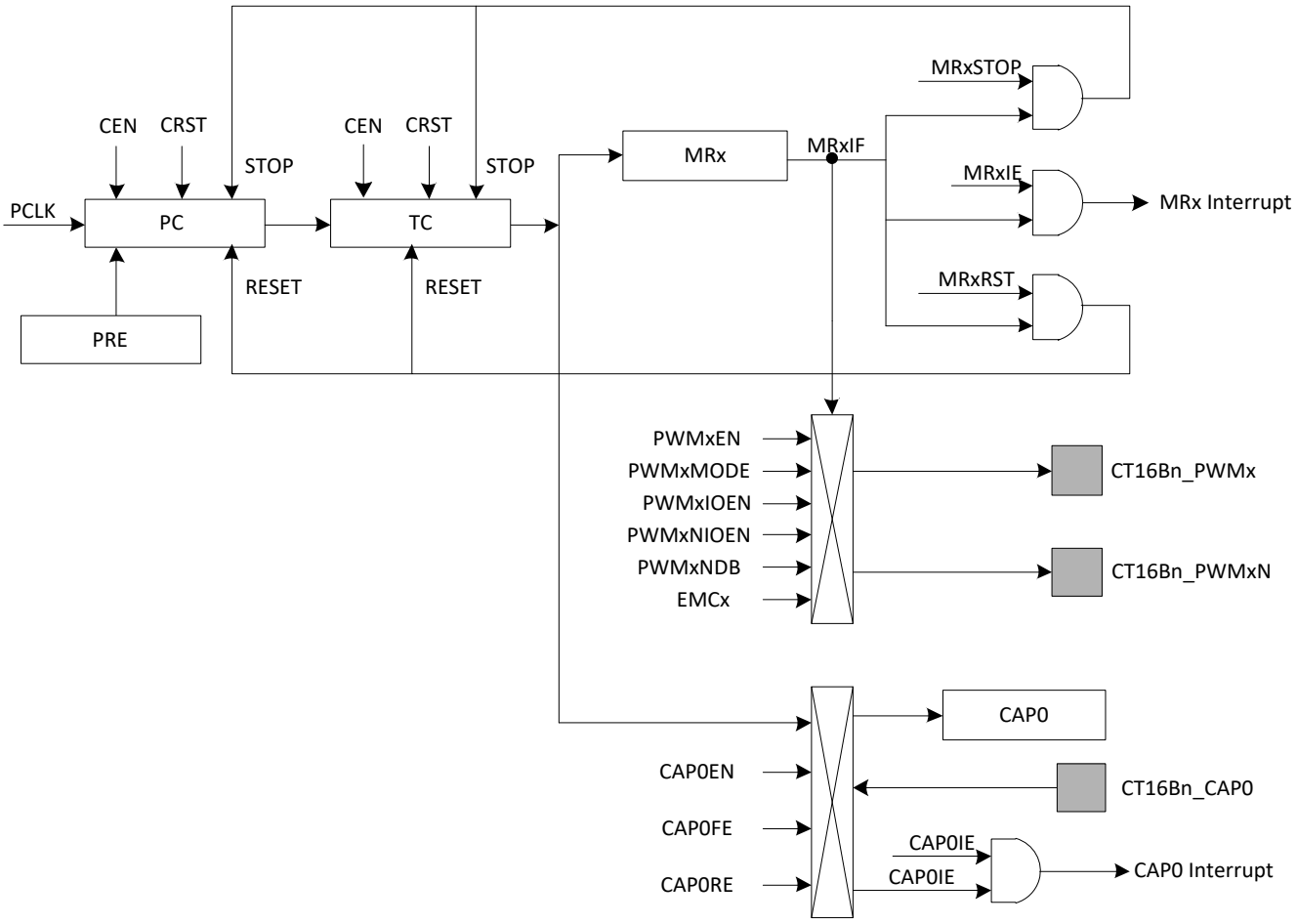
Pin Name	B0	B1	B2	B3	B4	B5	B8
CT16Bn_BRK	O	-	-	-	-	-	-
CT16Bn_CAP0	O	O	O		O	O	-
CT16Bn_PWM0	O	O	O	O	O	O	O
CT16Bn_PWM1	O	O	O	O	O	O	O
CT16Bn_PWM2	O	O	O	-	-	O	O
CT16Bn_PWM3	O	O	O	-	-	O	O
CT16Bn_PWM0N	O	-	-	O	O	-	-
CT16Bn_PWM1N	O	-	-	O	O	-	-
CT16Bn_PWM2N	O	-	-	-	-	-	-
CT16Bn_PWM3N	O	-	-	-	-	-	-

CT16Bn_PWM4	-	-	-	-	-	-	O
CT16Bn_PWM5	-	-	-	-	-	-	O
CT16Bn_PWM6	-	-	-	-	-	-	O
CT16Bn_PWM7	-	-	-	-	-	-	O
CT16Bn_PWM8	-	-	-	-	-	-	O
CT16Bn_PWM9	-	-	-	-	-	-	O
CT16Bn_PWM10	-	-	-	-	-	-	O
CT16Bn_PWM11	-	-	-	-	-	-	O

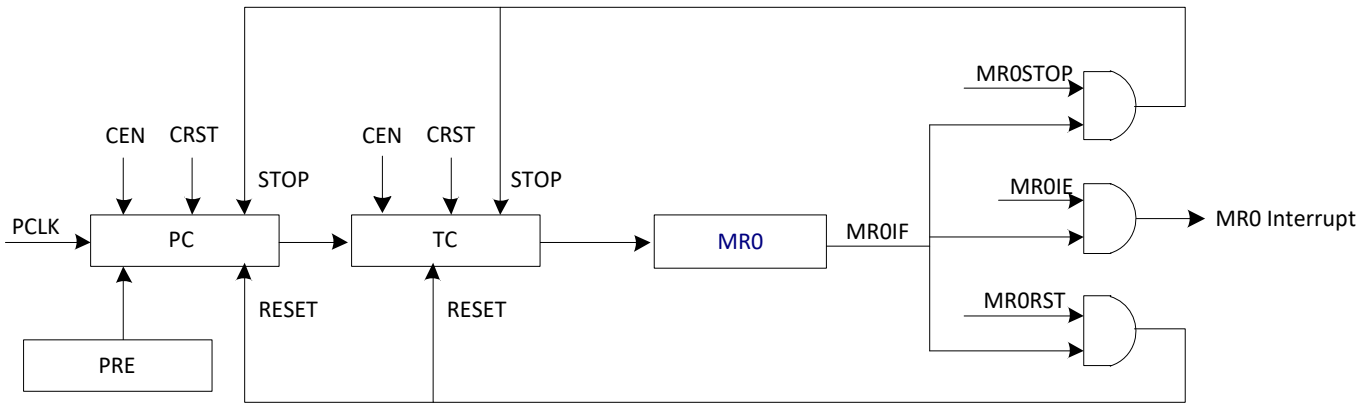
O: supply

8.4 BLOCK DIAGRAM

8.4.1 CT16B0~CT16B5, CT16B8



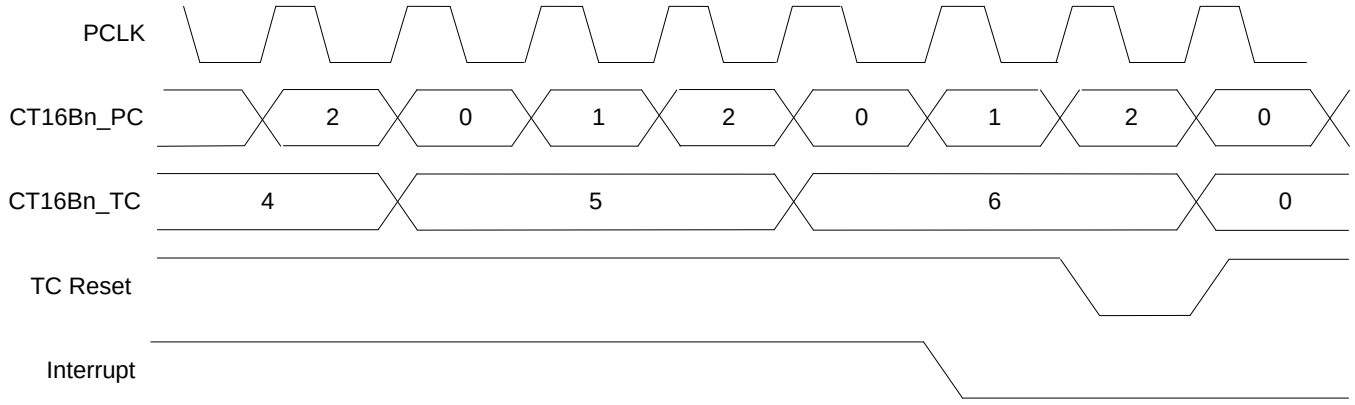
8.4.2 CT16B6~CT16B7



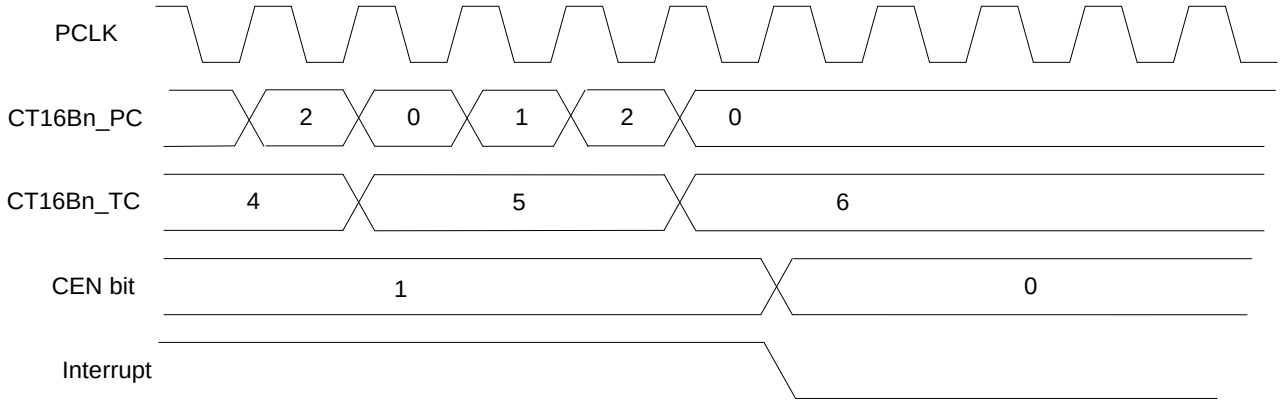
8.5 TIMER OPERATION

8.5.1 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.



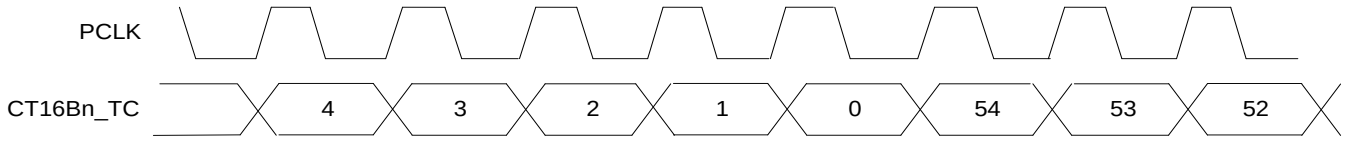
The following figure shows a timer configured to stop and generate an interrupt on match in Edge-aligned up-counting mode. The CT16Bn_PRE register is set to 2, and the CT16Bn_MRx register is set to 6. In the next clock after the timer reaches the match value, the CEN bit in CT16Bn_TMRCTRL register is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



8.5.2 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.

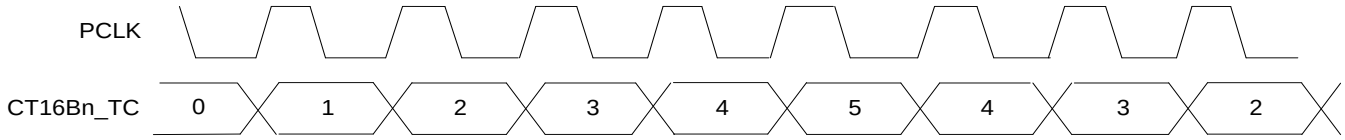
The following figure shows a timer configured to reset the count in Edge-aligned down-counting mode. The CT16Bn_PRE register is set to 0, and the CT16Bn_MR9 register is set to 54. After TC reaches 0, the timer count is reset and loaded from the value of CT16Bn_MR9.



8.5.3 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.



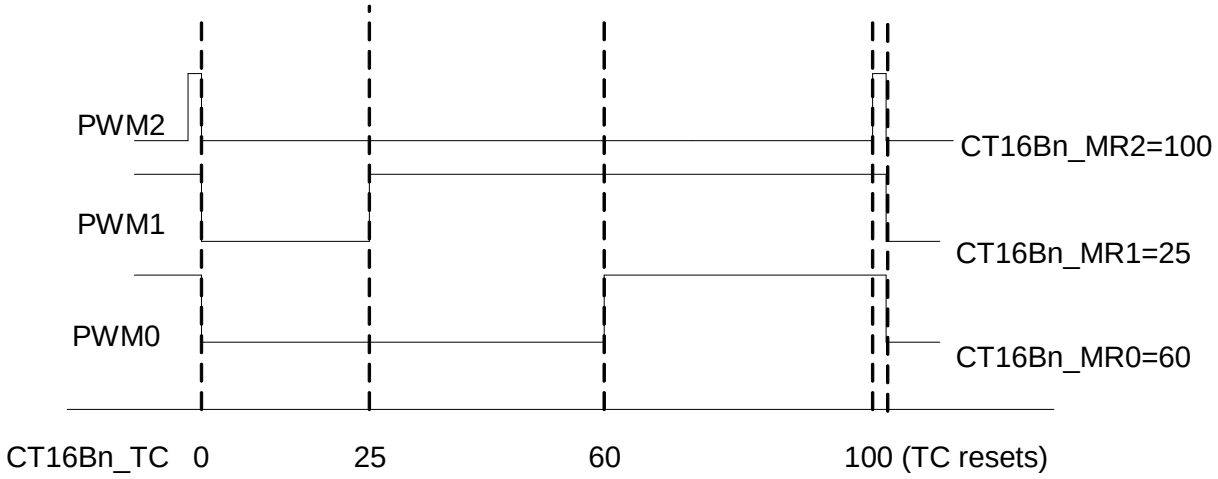
8.6 PWM

8.6.1 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,

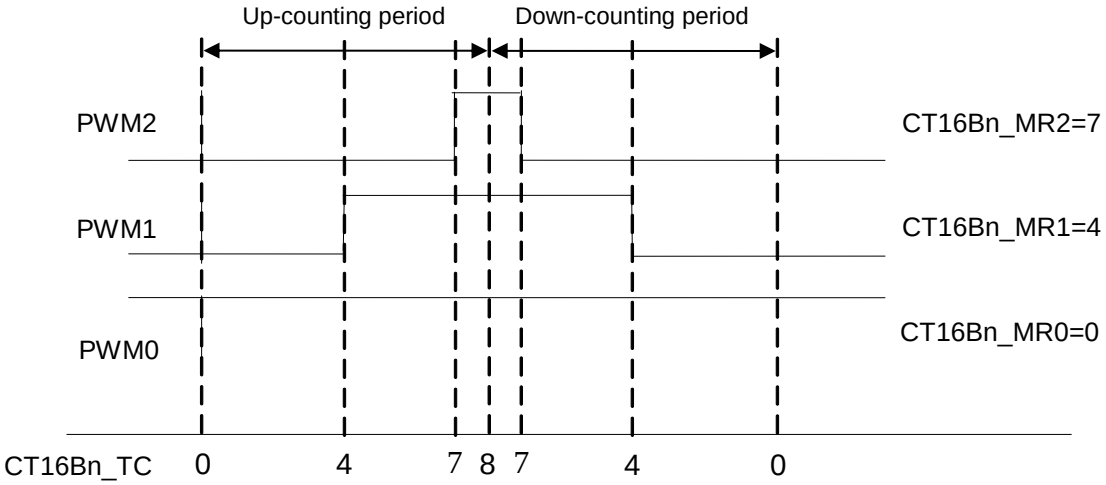
1. All single edge controlled PWM outputs go LOW at the beginning of each PWM cycle (timer is set to zero) unless their match value in CT16Bn_MR0~3 registers is equal to zero.
2. Each PWM output will go HIGH when its match value is reached. If no match occurs, the PWM output remains continuously LOW.
3. 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.
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 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.
5. 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.



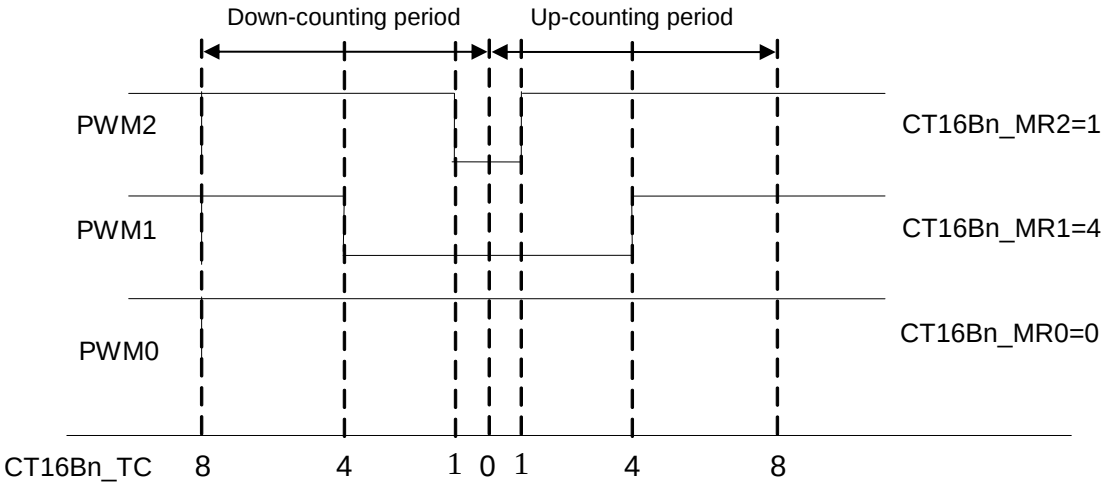
✷ ***Note: When the match outputs are selected to perform as PWM outputs, the timer reset (MRnRST) and timer stop (MRnSTOP) bits in CT16Bn_MCTRL register must be set to zero except for the match register setting the PWM cycle length. For this register, set the MRnRST bit to one to enable the timer reset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



8.6.2 PWM Mode 2

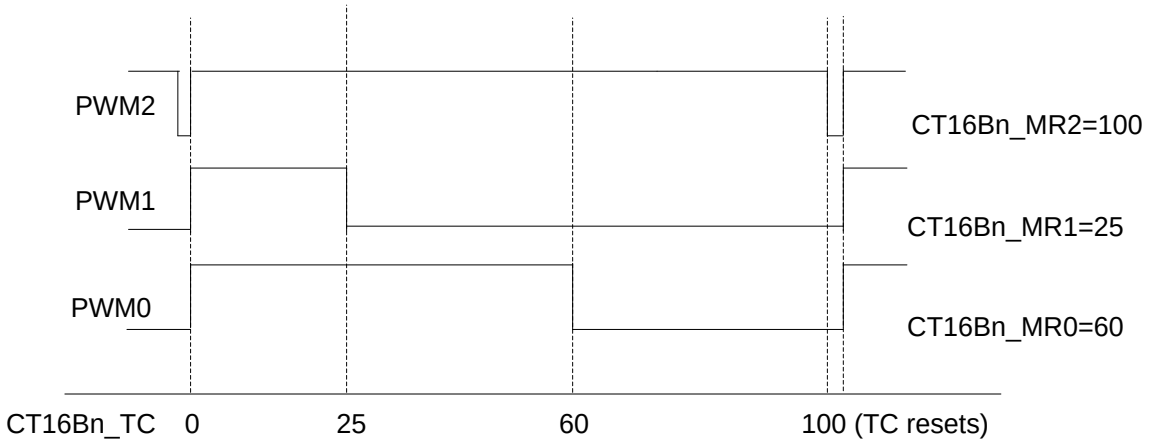
- PWMn is 1 when TC<MRn during Up-counting period
- PWMn is 1 when TC≤MRn during Down-counting period
- Not support in Center-aligned counting mode

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 MRnRST bit to one to enable the timer reset when the timer value matches the value of the corresponding match register.***

8.7 INVERSE PWM OUTPUT WITH DEAD-BAND PERIOD

The CT16B2_PWMm builds in inverse output function controlled by PWMmNIOEN[1:0] bits in CT16Bn_PWMCTRL register.

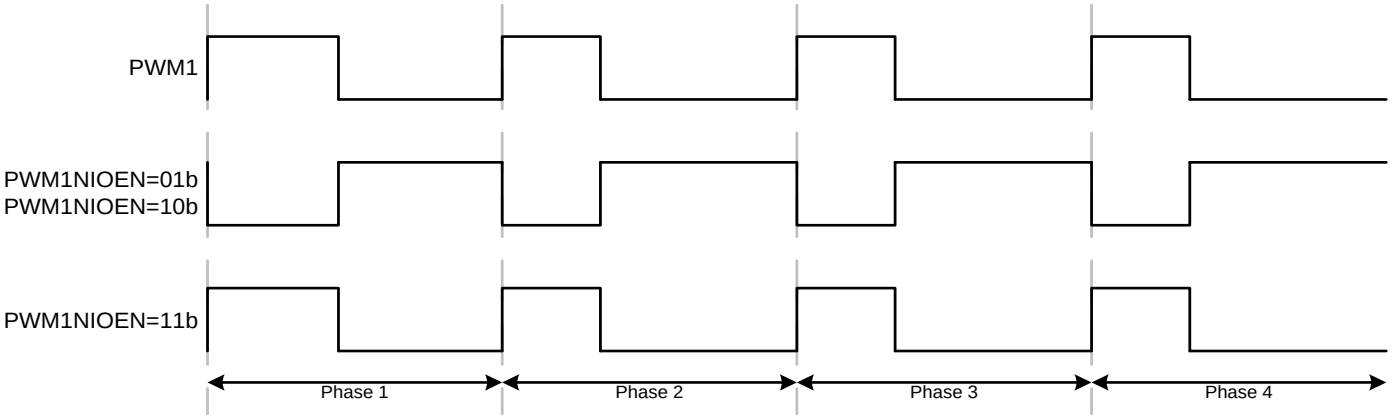
- When PWMmNIOEN[1:0] = 00b, PWMmN pin is GPIO mode.
- When PWMmNIOEN[1:0] = 01b, PWMmN pin changes to inverse PWM output pin which outputs the inverse PWM signal of PWMm with dead-band period, but same High signal during dead-band period.
- When PWMmNIOEN[1:0] = 10b, PWMmN pin changes to inverse PWM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

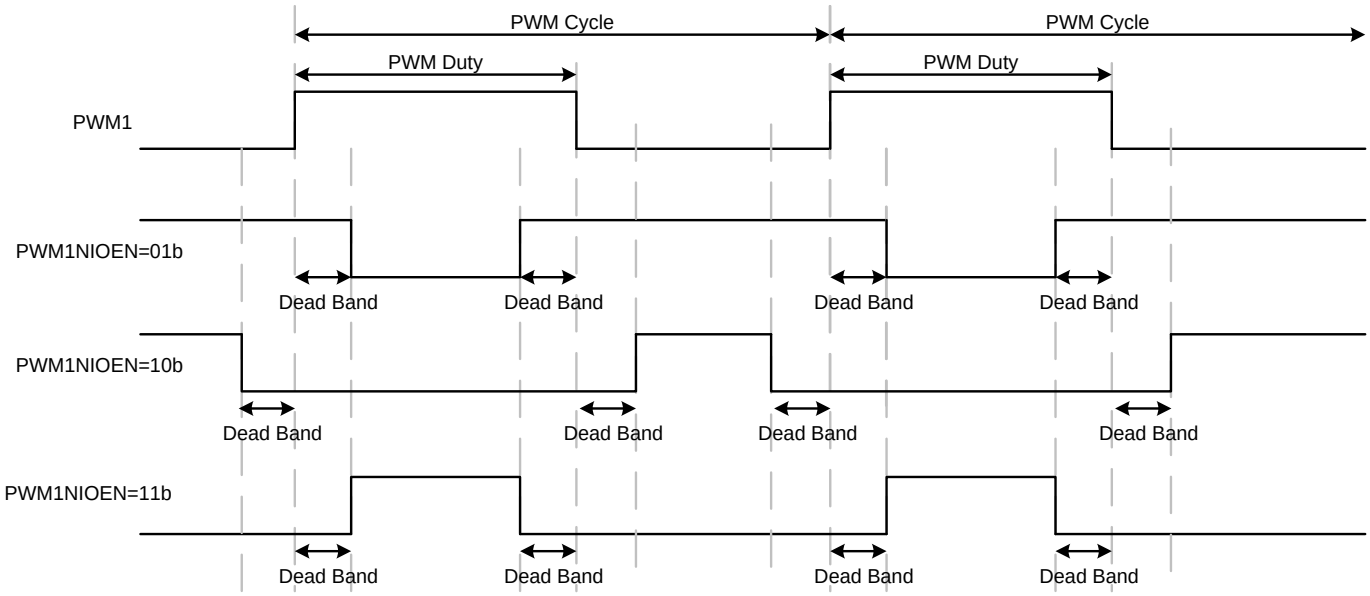
If DB = 1, the dead-band period is set as 1*CT16Bn_PCLK*(PR+1), 1*CT16Bn_PCLK*(PR+1) dead-band is in the left side of PWM high pulse, and the other side also includes one dead-band duration, so the total dead-band period is 2*CT16Bn_PCLK*(PR+1).

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:***

1. ***After the PWMN pin switches to GPIO mode (PWMnNIOEN=0) and before switching to PWMN mode (PWMnNIOEN=1), it is recommended to reinitialize the Timer to reset the internal state machine to avoid PWMN initial state errors.***
2. ***If the bead-band period is longer than PWM duty, the PWM1N is no output.***
3. ***When the dead-band function is enabled in Center-aligned mode, and MR9RST=1, CT16Bn_PWMxN will always output "0"***
4. ***When the dead zone function is enabled in Center alignment mode, the MRn setting range is DB <= MRn <= MR9.***
5. ***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.***

8.8 Break function (Only for CT16B0)

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 be selected by BRKSEL bits to select break pin (BRK).

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.

If break source is break pin (BRK), the break level can be selected as low or high.

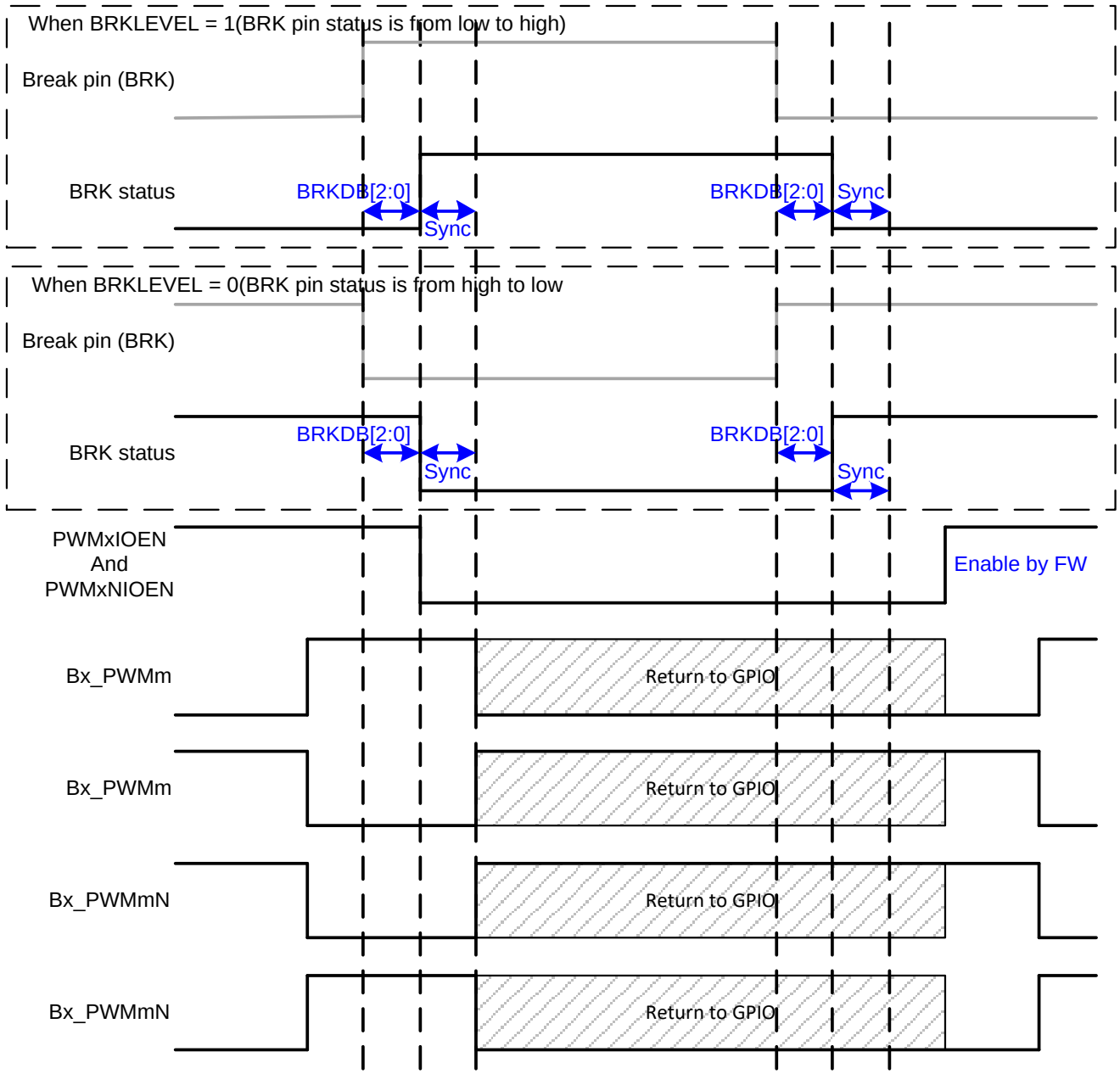
Besides, the break pin signal is through a de-bounce circuit to filter comparator transient status. The de-bounce time is controlled by BRKDB[2:0] bits.

when break condition occurs, the BRKIF is set. The BRKIF = 1 condition makes the interrupt service executed when BRKIE bit is set.

✸ ***Note:***

1. ***When the break condition is matched, PWM0IOEN~PWM3IOEN returns 00b (GPIO mode), and PWM0NIOEN~PWM3NIOEN also returns 00b (GPIO mode).***
2. ***If the length of Break signal is less than 2T BUSCLK, it may not be detected. It usually happens when the PCLK of de-bounce is smaller than BUSCLK.***

8.8.1 The break condition is break pin

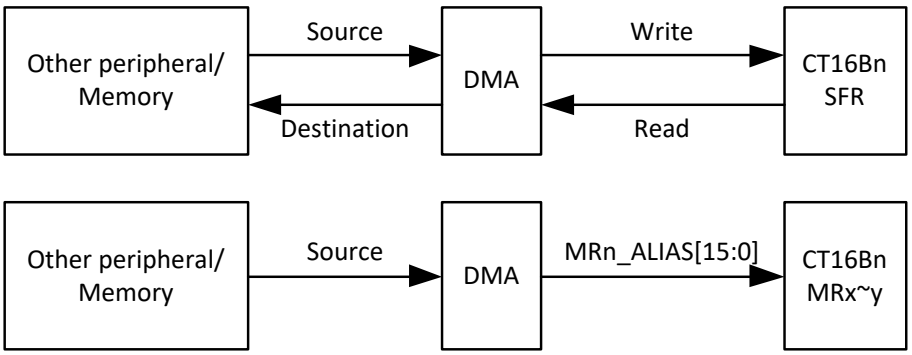


✷ ***Note:***

1. ***The break signal is through a de-bounce circuit to filter break pin transient status. The de-bounce time is controlled by BRKDB[2:0] bits.***
2. ***The Synchronize time is 4T PCLK + 4T BUSCLK.***

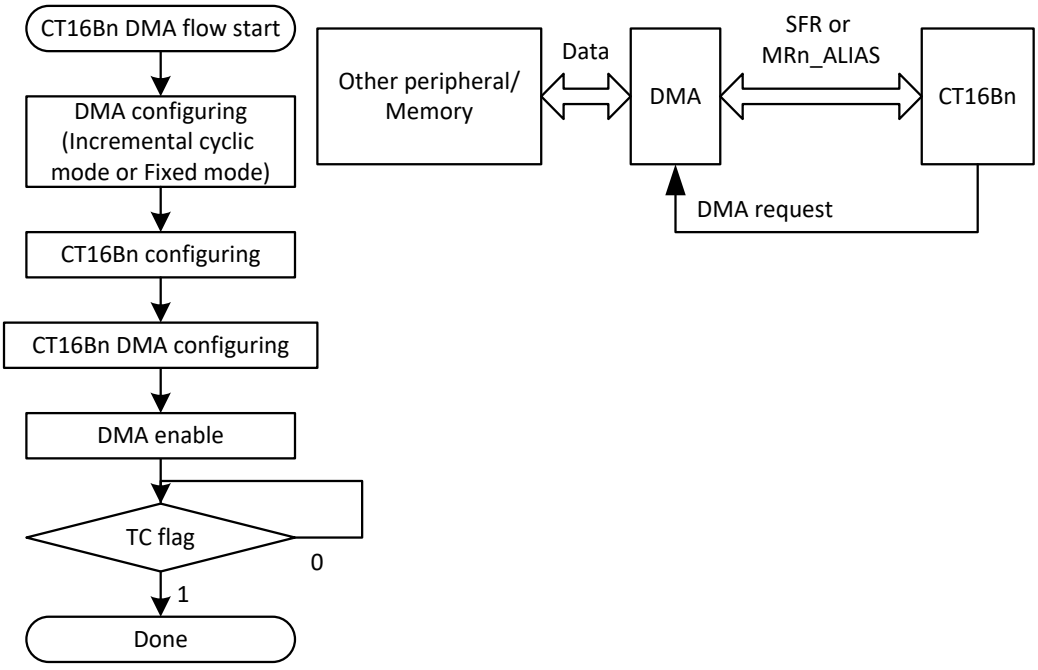
8.9 DMA MODE

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.



8.9.1 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.



➢ ***Note: If the interrupt flag is used to trigger a DMA request, it will be cleared by the DMA controller acknowledge signal and no interrupt will be issued.***
However, an interrupt may be raised before clearing. It is recommended to disable the interrupt enable to ensure that the interrupt vector will not be entered by mistake.

8.9.2 Configuration

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

Configuration	Register
SRCADDR = CT16Bn_BASE + **0x2C (CAP0)**	DMA_Cn_SRCADDR
SRC_RS = **CT16Bn_CAP0** SRC_HE=1	DMA_Cn_CFG
SRCAD_CTL=10 (Fixed) MODE = 1 (Peripheral)	DMA_Cn_CSR
DMA_CAP0 = 1	CT16n_DMA

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)

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

Configuration	Register
DSTADDR = CT16Bn_BASE + **0x40 (MR9)**	DMA_Cn_DSTADDR
DST_RS = **CT16Bn_MR9** DST_HE=1	DMA_Cn_CFG
DSTCAD_CTL=10 (Fixed) MODE = 1 (Peripheral)	DMA_Cn_CSR
DMA_MR9 = 1	CT16n_DMA

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)

8.9.2.3 Write CT16B0/1/2/5 MR0~3_ALIAS buffers

Configuration	Register
DSTADDR = CT16Bn_BASE + **0xD4 (DMAMRA1)**	DMA_Cn_DSTADDR
DST_RS = CT16Bn_MRx **(Use MRx request)** DST_HE=1	DMA_Cn_CFG
DSTCAD_CTL=11 **(Incremental cyclic mode)** MODE = 1 (Peripheral)	DMA_Cn_CSR
DMA_MRx = 1	CT16n_DMA

DMAC setting for CT16Bn MR0~3_ALIAS buffers			
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/CT16B2/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)
10. Set IC=1 (if CT16Bn_CEN = enable before step 8)
    I. If DMA is enabled when CT16B is counting(CEN = 1), it is recommended that IC should be set 1 after setting the CT16B DMA mode register to avoid DMA incorrect actions.
    II. CT16B DMA mode register should be disable after DMA TC(Transfer count) event flag is issued.

8.9.2.4 Write CT16B3/4 MR0~1_ALIAS buffers

Configuration	Register
DSTADDR = CT16Bn_BASE + **0xD4 (DMAMRA1)**	DMA_Cn_DSTADDR
DST_RS = CT16Bn_MRx **(Use MRx request)** DST_HE=1	DMA_Cn_CFG
DSTCAD_CTL=10 **(Fixed mode)** MODE = 1 (Peripheral)	DMA_Cn_CSR
DMA_MRx = 1	CT16n_DMA

DMAC setting for CT16Bn MR0~1_ALIAS buffers			
Cn_CSR **SRC_WIDTH**	Cn_CSR **SRC_SIZE**	Cn_SIZE **TOT_SIZE**	Cn_CSR **DST_WIDTH**
0x2 (word)	0x0 (burst=1)	N (word)	0x2 (word)
0x0 (byte)	0x1 (burst=4)	N (byte)	0x2 (word)

DMA update MR flow for CT16B3/CT16B4
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=010, DST_WIDTH=001
6. Set SRC_SIZE=000 (burst 1)
7. Set SRCAD_CTL=00, DSTCAD_CTL=11
8. Set CH_EN to Enable channel
9. Set DMA mode register (CT16n_DMA)

8.9.2.5 Write CT16B8 MR0~11_ALIAS buffers

Configuration	Register
DSTADDR = CT16Bn_BASE + **0xD4 (DMAMRA1)**	DMA_Cn_DSTADDR
DST_RS = CT16Bn_MRx **(Use MRx request)** DST_HE=1	DMA_Cn_CFG
DSTCAD_CTL=11 **(Incremental cyclic mode)** MODE = 1 (Peripheral)	DMA_Cn_CSR
DMA_MRx = 1	CT16n_DMA

DMAC setting for CT16Bn MRn_ALIAS[15:0]			
Cn_CSR **SRC_WIDTH**	Cn_CSR **SRC_SIZE**	Cn_SIZE **TOT_SIZE**	Cn_CSR **DST_WIDTH**
0x2 (word)	0x2 (burst=8)	N (word)	0x2 (word)
0x1 (half-word)	0x3 (burst=16)	N (half-word)	0x1 (half-word)
0x0 (byte)	0x32 (burst=32)	N (byte)	0x1 (half-word)

DMA update MR flow for CT16B8
11. Assign a DMA channel
12. Set SrcAddr, DstAddr = CT16Bn_BASE + 0xD4
13. Set TOT_SIZE
14. Set DST_RS, DST_HE=1
15. Set SRC_WIDTH=001, DST_WIDTH=001
16. Set SRC_SIZE=003 (burst 16)
17. Set SRCAD_CTL=00, DSTCAD_CTL=11
18. Set CH_EN to Enable channel
19. Set DMA mode register (CT16n_DMA)

> ***Note: DMA controller will move 16 half-words (or 8 words, 32 bytes) each time, but CT16B8 only has 12 MR buffers. Therefore, the 4 idle half-words (or 2 words, 8 bytes) are added to each 12 source data to meet the number of DMA burst sizes each time.***

8.10 CT16Bn REGISTERS

Base Address: 0x4002 0000 (CT16B0)
0x4002 1000 (CT16B1)
0x4002 2000 (CT16B2)
0x4002 3000 (CT16B3)
0x4002 D000 (CT16B4)
0x4002 E000 (CT16B5)
0x4000 0000 (CT16B6)
0x4000 1000 (CT16B7)
0x4002 F000 (CT16B8)

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	n	Description
0x0000	CT16Bn_TMRCTRL	0/1/2/3/4/5/6/7	CT16Bn Timer Control register
0x0004	CT16Bn_TC	0/1/2/3/4/5/6/7	CT16Bn Timer Counter register
0x0008	CT16Bn_PRE	0/1/2/3/4/5/6/7	CT16Bn Prescale register
0x000C	CT16Bn_PC	0/1/2/3/4/5/6/7	CT16Bn Prescale Counter register
0x0010	CT16Bn_CNTCTRL	0/1/2/4/5	CT16Bn Counter Control register
0x0014	CT16Bn_MCTRL	0/1/2/3/4/5/6/7	CT16Bn Match Control register
0x0018	CT16Bn_MR0	0/1/2/3/4/5/6/7	CT16Bn MR0 register
0x001C	CT16Bn_MR1	0/1/2/3/4/5	CT16Bn MR1 register
0x0020	CT16Bn_MR2	0/1/2/5	CT16Bn MR2 register
0x0024	CT16Bn_MR3	0/1/2/5	CT16Bn MR3 register
0x0028	CT16Bn_CAPCTRL	0/1/2/4/5	CT16Bn Capture Control register
0x002C	CT16Bn_CAP0	0/1/2/4/5	CT16Bn CAP0 register
0x0030	CT16Bn_EM	0/1/2/3/4/5	CT16Bn External Match register
0x0034	CT16Bn_PWMCTRL	0/1/2/3/4/5	CT16Bn PWM Control register
0x0038	CT16Bn_RIS	0/1/2/3/4/5/6/7	CT16Bn Raw Interrupt Status register
0x003C	CT16Bn_IC	0/1/2/3/4/5/6/7	CT16Bn Interrupt Clear register
0x0040	CT16Bn_MR9	0/1/2/3/4/5	CT16Bn MR9 register
0x0044	CT16Bn_PWM0NDB	0/3/4	CT16Bn PWM0N Dead-band Period register
0x0048	CT16Bn_PWM1NDB	0/3/4	CT16Bn PWM1N Dead-band Period register
0x004C	CT16Bn_PWM2NDB	0	CT16Bn PWM2N Dead-band Period register
0x0050	CT16Bn_PWM3NDB	0	CT16Bn PWM3N Dead-band Period register
0x0054 – 0x0070	–		Reserved
0x0074	CT16Bn_LOADCTRL	0/1/2/5	CT16Bn PWM Load Mode Control register
0x0078 – 0x00CC	–		Reserved
0x00D0	CT16Bn_DMA	0/1/2/3/4/5/6/7	CT16Bn DMA mode register
0x00D4	CT16Bn_DMAMRA1	0/1/2/3/4/5	CT16Bn DMA MR Alias register 1
0x00D8	CT16Bn_DMAMRA2	0/1/2/5	CT16Bn DMA MR Alias register 2
0x00DC	–		Reserved
0x00E0	CT16Bn_BRKCTRL	0	CT16Bn Break Function Control register

Offset	Register	Description
0x0000	CT16B8_TMRCTRL	CT16Bn Timer Control register
0x0004	CT16B8_TC	CT16Bn Timer Counter register
0x0008	CT16B8_PRE	CT16Bn Prescale register
0x000C	CT16B8_PC	CT16Bn Prescale Counter register
0x0010	–	Reserved
0x0014	CT16B8_MCTRL	CT16Bn Match Control register
0x0018	CT16B8_MCTRL2	CT16Bn Match Control register
0x001C	CT16B8_MR0	CT16Bn MR0 register
0x0020	CT16B8_MR1	CT16Bn MR1 register
0x0024	CT16B8_MR2	CT16Bn MR2 register
0x0028	CT16B8_MR3	CT16Bn MR3 register
0x002C	CT16B8_MR4	CT16Bn MR4 register
0x0030	CT16B8_MR5	CT16Bn MR5 register

0x0034	CT16B8_MR6	CT16Bn MR6 register
0x0038	CT16B8_MR7	CT16Bn MR7 register
0x003C	CT16B8_MR8	CT16Bn MR8 register
0x0040	CT16B8_MR9	CT16Bn MR9 register
0x0044	CT16B8_MR10	CT16Bn MR10 register
0x0048	CT16B8_MR11	CT16Bn MR11 register
0x004C	CT16B8_MR_PERIOD	CT16B8 MR_PERIOD register
0x0050 – 0x0054	–	Reserved
0x0058	CT16B8_EM	CT16Bn External Match register
0x005C	CT16B8_EMC	CT16Bn External Match Control register
0x0060	CT16B8_PWMCTRL	CT16Bn PWM Control register
0x0064	CT16B8_PWMENB	CT16Bn PWM Enable register
0x0068	CT16B8_PWMIOENB	CT16Bn PWM IO Enable register
0x006C	CT16B8_RIS	CT16Bn Raw Interrupt Status register
0x0070	CT16B8_IC	CT16Bn Interrupt Clear register
0x0074 – 0x00CC	–	Reserved
0x00D0	CT16B8_DMA	CT16Bn DMA mode register
0x00D4	CT16B8_DMAMRA1	CT16Bn DMA MR Alias register 1
0x00D8	CT16B8_DMAMRA2	CT16Bn DMA MR Alias register 2
0x00DC	CT16B8_DMAMRA3	CT16Bn DMA MR Alias register 3
0x00E0	CT16B8_DMAMRA4	CT16Bn DMA MR Alias register 4
0x00E4	CT16B8_DMAMRA5	CT16Bn DMA MR Alias register 5
0x00E8	CT16B8_DMAMRA6	CT16Bn DMA MR Alias register 6

8.10.1 CT16Bn Timer Control register (CT16Bn_TMRCTRL) (n=0,1,2,5)

Address Offset: 0x00

✸ ***Note:***
1. ***CEN bit shall be set at last!***
2. ***In order to initial TC and PC correctly, SW shall reset TC and PC by setting CRST to 1, and then enable counter by setting CEN to 1.***
3. ***The following register values must be defined in the initialization phase, and the values cannot be changed after CEN = 1:***
   - ***CM[2:0], CLKSEL, TC[15:0], PC[7:0], CTM[1:0]***
   - ***PWMnMODE[1:0], DB[9:0], LOADCTRL, DMA, BRKCTRL, MR9[15:0]***

Bit	Field	Access	Initial	Description
31:7	–	–	0	Reserved
6:4	CM	RW	0	Counting mode selection 0: Edge-aligned Up-counting mode 1: Edge-aligned Down-counting mode 2: Center-aligned mode 1. The match interrupt flag is set during the down-counting period 4: Center-aligned mode 2. The match interrupt flag is set during the up-counting period 6: Center-aligned mode 3. The match interrupt flag is set during both up-counting and down-counting period Other: Reserved
3:2	–	–	0	Reserved
1	CRST	RW	0	Counter Reset 0: Disable Counter 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.
0	CEN	RW	0	Counter enable 0: Disable counter 1: Enable Timer Counter and Prescale Counter for counting.

8.10.2 CT16Bn Timer Control register (CT16Bn_TMRCTRL) (n=3,4,6,7,8)

Address Offset: 0x00

✸ ***Note:***
1. ***CEN bit shall be set at last!***
2. ***In order to initial TC and PC correctly, SW shall reset TC and PC by setting CRST to 1, and then enable counter by setting CEN to 1.***
3. ***The following register values must be defined in the initialization phase, and the values cannot be changed after CEN = 1:***
   - ***CM[2:0], CLKSEL, TC[15:0], PC[7:0], CTM[1:0]***
   - ***PWMnMODE[1:0], DB[9:0], LOADCTRL, DMA, MR9[15:0]***

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1	CRST	RW	0	Counter Reset 0: Disable Counter 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.
0	CEN	RW	0	Counter enable 0: Disable counter 1: Enable Timer Counter and Prescale Counter for counting.

8.10.3 CT16Bn Timer Counter register (CT16Bn_TC) (n=0,1,2,3,4,5,6,7,8)

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	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	TC	RW	0	Timer Counter

8.10.4 CT16Bn Prescale register (CT16Bn_PRE) (n=0,1,2,3,4,5,6,7,8)

Address Offset: 0x08

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

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	PRE	RW	0	Prescaler value

8.10.5 CT16Bn Prescale Counter register (CT16Bn_PC) (n=0,1,2,3,4,5,6,7,8)

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	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	PC	RW	0	Prescaler Counter

8.10.6 CT16Bn Count Control register (CT16Bn_CNTCTRL) (n=0,1,2,4,5)

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 to 0 of Capture Control (CAPCTRL) register and bit 7 to 0 of Prescale (PRE) register must be set to 0x0.***

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1:0	CTM	RW	0	Counter/Timer Mode This field selects which rising PCLK edges can increment Timer's PC, or clear PC and increment TC. 0: Timer Mode - every rising PCLK edge 1: Counter Mode - TC is incremented on rising edges on the CAP0 input selected 2: Counter Mode - TC is incremented on falling edges on the CAP0 input selected 3: Counter Mode - TC is incremented on both edges on the CAP0 input selected

8.10.7 CT16Bn Match Control register (CT16Bn_MCTRL) (n=0,1,2,5)

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	Field	Access	Initial	Description
31	MR9STOP	RW	0	Stop TC and PC and clear CEN bit when MR9 matches TC 0: Disable 1: Stop TC and PC and clear CEN bit when MR9 matches TC
30	MR9RST	RW	0	Enable reset TC when MR9 matches TC 0: Disable 1: Reset TC when MR9 matches TC
29	MR9IE	RW	0	Enable generating an interrupt based on CM[2:0] when MR9 matches the value in the TC 0: Disable 1: Generating an interrupt when MR9 matches TC
28:12	–	–	0	Reserved
11	MR3STOP	RW	0	Stop TC and PC and clear CEN bit when MR3 matches TC 0: Disable 1: Stop TC and PC and clear CEN bit when MR3 matches TC
10	MR3RST	RW	0	Enable reset TC when MR3 matches TC 0: Disable 1: Reset TC when MR3 matches TC
9	MR3IE	RW	0	Enable generating an interrupt when MR3 matches TC 0: Disable 1: Generating an interrupt when MR3 matches TC
8	MR2STOP	RW	0	Stop TC and PC and clear CEN bit when MR2 matches TC 0: Disable 1: Stop TC and PC and clear CEN bit when MR2 matches TC
7	MR2RST	RW	0	Enable reset TC when MR2 matches TC 0: Disable 1: Reset TC when MR2 matches TC
6	MR2IE	RW	0	Enable generating an interrupt when MR2 matches TC 0: Disable 1: Generating an interrupt when MR2 matches TC
5	MR1STOP	RW	0	Stop TC and PC and clear CEN bit when MR1 matches TC 0: Disable 1: Stop TC and PC and clear CEN bit when MR1 matches TC
4	MR1RST	RW	0	Enable reset TC when MR1 matches TC 0: Disable 1: Reset TC when MR1 matches TC
3	MR1IE	RW	0	Enable generating an interrupt when MR1 matches TC 0: Disable 1: Generating an interrupt when MR1 matches TC
2	MR0STOP	RW	0	Stop TC and PC and clear CEN bit when MR0 matches TC 0: Disable 1: Stop TC and PC and clear CEN bit when MR0 matches TC
1	MR0RST	RW	0	Enable reset TC when MR0 matches TC 0: Disable 1: Reset TC when MR0 matches TC
0	MR0IE	RW	0	Enable generating an interrupt when MR0 matches TC 0: Disable 1: Generating an interrupt when MR0 matches TC

8.10.8 CT16Bn Match Control register (CT16Bn_MCTRL) (n=3,4)

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	Field	Access	Initial	Description
31	MR9STOP	RW	0	Stop TC and PC and clear CEN bit when MR9 matches TC 0: Disable 1: Stop TC and PC and clear CEN bit when MR9 matches TC
30	MR9RST	RW	0	Enable reset TC when MR9 matches TC 0: Disable 1: Reset TC when MR9 matches TC
29	MR9IE	RW	0	Enable generating an interrupt based on CM[2:0] when MR9 matches the value in the TC 0: Disable 1: Generating an interrupt when MR9 matches TC
28:6	–	–	0	Reserved
5	MR1STOP	RW	0	Stop TC and PC and clear CEN bit when MR1 matches TC 0: Disable 1: Stop TC and PC and clear CEN bit when MR1 matches TC
4	MR1RST	RW	0	Enable reset TC when MR1 matches TC 0: Disable 1: Reset TC when MR1 matches TC
3	MR1IE	RW	0	Enable generating an interrupt when MR1 matches TC 0: Disable 1: Generating an interrupt when MR1 matches TC
2	MR0STOP	RW	0	Stop TC and PC and clear CEN bit when MR0 matches TC 0: Disable 1: Stop TC and PC and clear CEN bit when MR0 matches TC
1	MR0RST	RW	0	Enable reset TC when MR0 matches TC 0: Disable 1: Reset TC when MR0 matches TC
0	MR0IE	RW	0	Enable generating an interrupt when MR0 matches TC 0: Disable 1: Generating an interrupt when MR0 matches TC

8.10.9 CT16Bn Match Control register (CT16Bn_MCTRL) (n=6,7)

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	Field	Access	Initial	Description
31:3	–	–	0	Reserved
2	MR0STOP	RW	0	Stop TC and PC and clear CEN bit when MR0 matches TC 0: Disable 1: Stop TC and PC and clear CEN bit when MR0 matches TC
1	MR0RST	RW	0	Enable reset TC when MR0 matches TC 0: Disable 1: Reset TC when MR0 matches TC
0	MR0IE	RW	0	Enable generating an interrupt based on CM[2:0] when MR0 matches the value in the TC 0: Disable 1: Generating an interrupt when MR0 matches TC

8.10.10 CT16Bn Match Control register (CT16Bn_MCTRL) (n=8)

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	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29	MR9STOP	RW	0	Stop TC and PC and clear CEN bit when MR9 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR9 matches TC
28	MR9RST	RW	0	Enable reset TC when MR9 matches TC 0: Disable reset TC 1: Reset TC when MR9 matches TC
27	MR9IE	RW	0	Enable generating an interrupt based on CM[2:0] when MR9 matches the value in the TC 0: Disable generating an interrupt 1: Generating an interrupt when MR9 matches TC
26	MR8STOP	RW	0	Stop TC and PC and clear CEN bit when MR8 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR8 matches TC
25	MR8RST	RW	0	Enable reset TC when MR8 matches TC 0: Disable reset TC 1: Reset TC when MR8 matches TC
24	MR8IE	RW	0	Enable generating an interrupt when MR8 matches TC 0: Disable generating an interrupt

Bit	Field	Access	Initial	Description
				1: Generating an interrupt when MR8 matches TC
23	MR7STOP	RW	0	Stop TC and PC and clear CEN bit when MR7 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR7 matches TC
22	MR7RST	RW	0	Enable reset TC when MR7 matches TC 0: Disable reset TC 1: Reset TC when MR7 matches TC
21	MR7IE	RW	0	Enable generating an interrupt when MR7 matches TC 0: Disable generating an interrupt 1: Generating an interrupt when MR7 matches TC
20	MR6STOP	RW	0	Stop TC and PC and clear CEN bit when MR6 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR6 matches TC
19	MR6RST	RW	0	Enable reset TC when MR6 matches TC 0: Disable reset TC 1: Reset TC when MR6 matches TC
18	MR6IE	RW	0	Enable generating an interrupt when MR6 matches TC 0: Disable generating an interrupt 1: Generating an interrupt when MR6 matches TC
17	MR5STOP	RW	0	Stop TC and PC and clear CEN bit when MR5 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR5 matches TC
16	MR5RST	RW	0	Enable reset TC when MR5 matches TC 0: Disable reset TC 1: Reset TC when MR5 matches TC
15	MR5IE	RW	0	Enable generating an interrupt when MR5 matches TC 0: Disable generating an interrupt 1: Generating an interrupt when MR5 matches TC
14	MR4STOP	RW	0	Stop TC and PC and clear CEN bit when MR4 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR4 matches TC
13	MR4RST	RW	0	Enable reset TC when MR4 matches TC 0: Disable reset TC 1: Reset TC when MR4 matches TC
12	MR4IE	RW	0	Enable generating an interrupt when MR4 matches TC 0: Disable generating an interrupt 1: Generating an interrupt when MR4 matches TC
11	MR3STOP	RW	0	Stop TC and PC and clear CEN bit when MR3 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR3 matches TC
10	MR3RST	RW	0	Enable reset TC when MR3 matches TC 0: Disable reset TC 1: Reset TC when MR3 matches TC
9	MR3IE	RW	0	Enable generating an interrupt when MR3 matches TC 0: Disable generating an interrupt 1: Generating an interrupt when MR3 matches TC
8	MR2STOP	RW	0	Stop TC and PC and clear CEN bit when MR2 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR2 matches TC
7	MR2RST	RW	0	Enable reset TC when MR2 matches TC 0: Disable reset TC 1: Reset TC when MR2 matches TC
6	MR2IE	RW	0	Enable generating an interrupt when MR2 matches TC 0: Disable generating an interrupt 1: Generating an interrupt when MR2 matches TC
5	MR1STOP	RW	0	Stop TC and PC and clear CEN bit when MR1 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR1 matches TC
4	MR1RST	RW	0	Enable reset TC when MR1 matches TC 0: Disable reset TC 1: Reset TC when MR1 matches TC
3	MR1IE	RW	0	Enable generating an interrupt when MR1 matches TC 0: Disable generating an interrupt 1: Generating an interrupt when MR1 matches TC

Bit	Field	Access	Initial	Description
2	MR0STOP	RW	0	Stop TC and PC and clear CEN bit when MR0 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR0 matches TC
1	MR0RST	RW	0	Enable reset TC when MR0 matches TC 0: Disable reset TC 1: Reset TC when MR0 matches TC
0	MR0IE	RW	0	Enable generating an interrupt when MR0 matches TC 0: Disable generating an interrupt 1: Generating an interrupt when MR0 matches TC

8.10.11 CT16Bn Match Control register 2 (CT16Bn_MCTRL2) (n=8)

Address Offset: 0x18

- ***Note: When the dead-band function is enabled in Center-aligned mode, and MR12RST=1, CT16B1_PWMxN will always output "0"***
- ***Note: When the dead-band function is enabled***
  - System will reset TC refer to MRPERIODRST ONLY in Up-counting mode
  - In Down counting mode, TC[15:0] will be reloaded from CT16Bn_ MRPERIOD after resetting counter
  - System will reset TC refer to MRPERIODRST ONLY in Center-aligned mode

Bit	Field	Access	Initial	Description
31	MRPERIODSTOP	RW	0	Stop MR_PERIOD: TC and PC will stop and CEN bit will be cleared if MR_PERIOD matches TC 0: Disable stop TC and PC 1: TC and PC will stop and CEN bit will be cleared if MR_PERIOD matches TC
30	MRPERIODRST	RW	0	Enable reset TC when MR_PERIOD matches TC 0: Disable reset TC 1: Reset TC when MR_PERIOD matches TC
29	MRPERIODIE	RW	0	Enable generating an interrupt based on CM[2:0] when MR_PERIOD matches the value in the TC 0: Disable generating an interrupt 1: Generating an interrupt based on CM[2:0] when MR_PERIOD matches the value in the TC
28:6	–	–	0	Reserved
5	MR11STOP	RW	0	Stop TC and PC and clear CEN bit when MR11 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR11 matches TC
4	MR11RST	RW	0	Enable reset TC when MR11 matches TC 0: Disable reset TC 1: Reset TC when MR11 matches TC
3	MR11IE	RW	0	Enable generating an interrupt when MR11 matches TC 0: Disable generating an interrupt 1: Generating an interrupt when MR11 matches TC
2	MR10STOP	RW	0	Stop TC and PC and clear CEN bit when MR10 matches TC 0: Disable stop TC and PC 1: Stop TC and PC and clear CEN bit when MR10 matches TC
1	MR10RST	RW	0	Enable reset TC when MR10 matches TC 0: Disable reset TC 1: Reset TC when MR10 matches TC
0	MR10IE	RW	0	Enable generating an interrupt when MR10 matches TC 0: Disable generating an interrupt 1: Generating an interrupt when MR10 matches TC

8.10.12 CT16Bn Match register 0 (CT16Bn_MR0) (n=0,1,2,3,4,5,6,7)

Address Offset: 0x18

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	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	MR	RW	0	Timer counter match value

8.10.13 CT16Bn Match register 1 (CT16Bn_MR1) (n=0,1,2,3,4,5)

Address Offset: 0x1C

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	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	MR	RW	0	Timer counter match value

8.10.14 CT16Bn Match register 2~3 (CT16Bn_MR2~3) (n=0,1,2,5)

Address Offset: 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	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	MR	RW	0	Timer counter match value

8.10.15 CT16Bn Match register 9 (CT16Bn_MR9) (n=0,1,2,3,4,5)

Address Offset: 0x40

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	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	MR	RW	0	Timer counter match value

8.10.16 CT16Bn Match register 0~11 (CT16Bn_MR0~11) (n=8)

Address Offset: 0x1C, 0x20, 0x24, 0x28, 0x2C, 0x30, 0x34, 0x38, 0x3C, 0x40, 0x44, 0x48

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	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	MR	RW	0	Timer counter match value

8.10.17 CT16Bn Match for Period register (CT16Bn_MR_PERIOD) (n=8)

Address Offset: 0x4C

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	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	MR	RW	0	Timer counter match value

8.10.18 CT16Bn Capture Control register (CT16Bn_CAPCTRL) (n=0,1,2,4,5)

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:***
1. ***HW will switch I/O Configuration directly when CAP0EN=1.***
2. ***If Counter mode is selected in the CNTCTRL register, CAPCTRL[2:0] must be programmed as 0x0.***

Bit	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3	CAP0EN	RW	0	CAP0 function enable 0: Disable 1: Enable.
2	CAP0IE	RW	0	Interrupt on CT16Bn_CAP0 event: a CAP0 load due to a CT16Bn_CAP0 event will generate an interrupt. 0: Disable 1: Enable
1	CAP0FE	RW	0	Capture on CT16Bn_CAP0 signal 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
0	CAP0RE	RW	0	Capture on CT16Bn_CAP0 signal 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

8.10.19 CT16Bn Capture 0 register (CT16Bn_CAP0) (n=0,1,2,4,5)

Address Offset: 0x2C

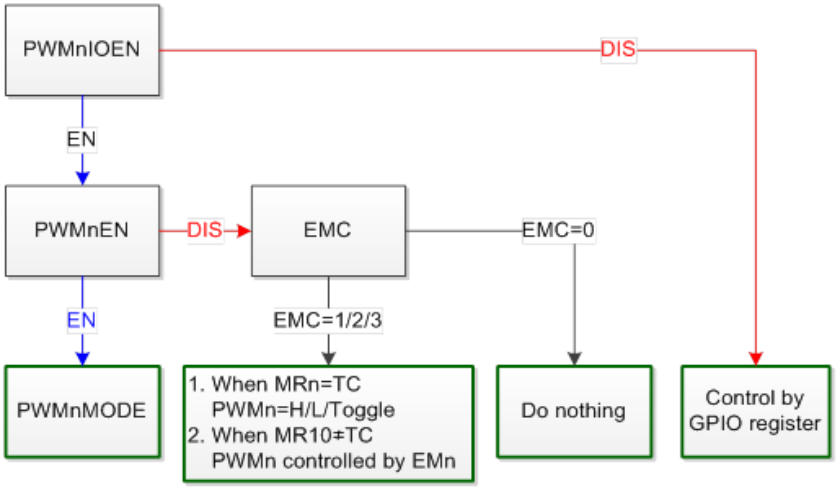
Each Capture register is associated with a device pin and may be loaded with the counter/timer value when a specified event occurs on that pin. The settings in the Capture Control register determine whether the capture function is enabled, and whether a capture event happens on the rising edge of the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

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	CAP0	R	0	Timer counter capture value

8.10.20 CT16Bn External Match register (CT16Bn_EM) (n=0,1,2,5)

Address Offset: 0x30

The External Match register provides both control and status of CT16Bn_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.

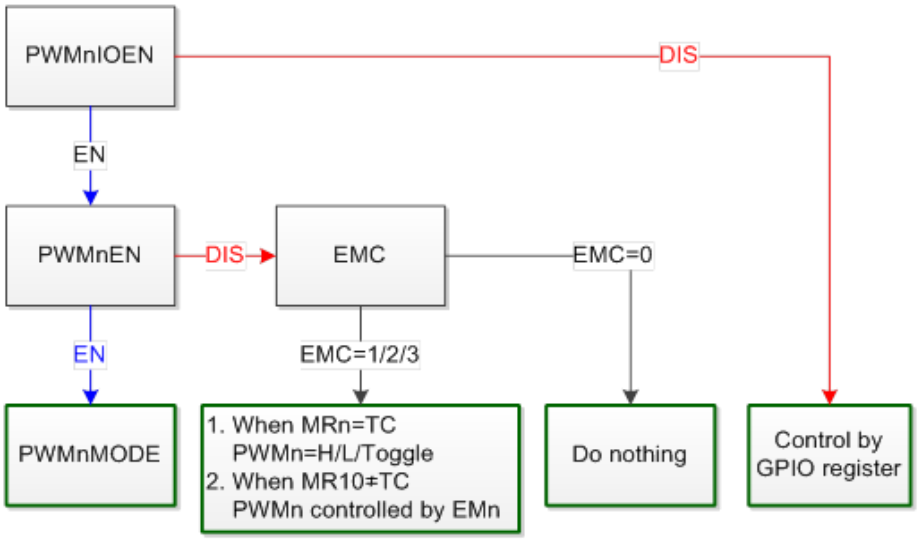


Bit	Field	Access	Initial	Description
31:12	–	–	0	Reserved
11:10	EMC3	RW	0	CT16Bn_PWM3 functionality when MR3=TC 0: Do nothing 1: CT16Bn_PWM3 pin is LOW 2: CT16Bn_PWM3 pin is HIGH 3: Toggle CT16Bn_PWM3 pin
9:8	EMC2	RW	0	CT16Bn_PWM2 functionality when MR2=TC 0: Do nothing 1: CT16Bn_PWM2 pin is LOW 2: CT16Bn_PWM2 pin is HIGH 3: Toggle CT16Bn_PWM2 pin
7:6	EMC1	RW	0	CT16Bn_PWM1 functionality when MR1=TC 0: Do nothing 1: CT16Bn_PWM1 pin is LOW 2: CT16Bn_PWM1 pin is HIGH 3: Toggle CT16Bn_PWM1 pin
5:4	EMC0	RW	0	CT16Bn_PWM0 functionality when MR0=TC 0: Do nothing 1: CT16Bn_PWM0 pin is LOW 2: CT16Bn_PWM0 pin is HIGH 3: Toggle CT16Bn_PWM0 pin
3	EM3	RW	0	When EMC3≠00b and MR3≠TC, this bit will drive the state of CT16Bn_PWM3 output 0: Drive Low 1: Drive High
2	EM2	RW	0	When EMC2≠00b and MR2≠TC, this bit will drive the state of CT16Bn_PWM2 output 0: Drive Low 1: Drive High
1	EM1	RW	0	When EMC1≠00b and MR1≠TC, this bit will drive the state of CT16Bn_PWM1 output 0: Drive Low 1: Drive High
0	EM0	RW	0	When EMC0≠00b and MR0≠TC, this bit will drive the state of CT16Bn_PWM0 output 0: Drive Low 1: Drive High

8.10.21 CT16Bn External Match register (CT16Bn_EM) (n=3,4)

Address Offset: 0x30

The External Match register provides both control and status of CT16Bn_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.

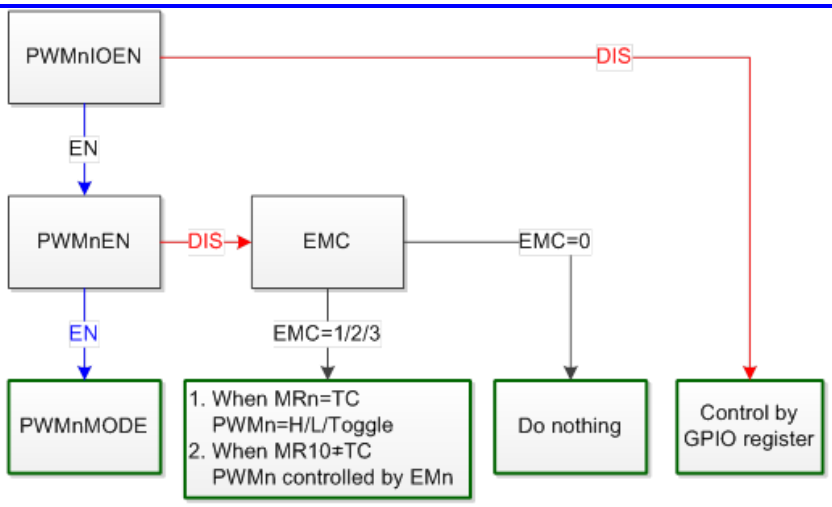


Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:6	EMC1	RW	0	CT16Bn_PWM1 functionality when MR1=TC 0: Do nothing 1: CT16Bn_PWM1 pin is LOW 2: CT16Bn_PWM1 pin is HIGH 3: Toggle CT16Bn_PWM1 pin
5:4	EMC0	RW	0	CT16Bn_PWM0 functionality when MR0=TC 0: Do nothing 1: CT16Bn_PWM0 pin is LOW 2: CT16Bn_PWM0 pin is HIGH 3: Toggle CT16Bn_PWM0 pin
3:2	–	–	0	Reserved
1	EM1	RW	0	When EMC1≠00b and MR1≠TC, this bit will drive the state of CT16Bn_PWM1 output 0: Drive Low 1: Drive High
0	EM0	RW	0	When EMC0≠00b and MR0≠TC, this bit will drive the state of CT16Bn_PWM0 output 0: Drive Low 1: Drive High

8.10.22 CT16Bn External Match register (CT16Bn_EM) (n=8)

Address Offset: 0x58

The External Match register provides both control and status of CT16B8_PWM[11:0]. If the match outputs are configured as PWM output, the function of the external match registers is determined by the PWM rules.



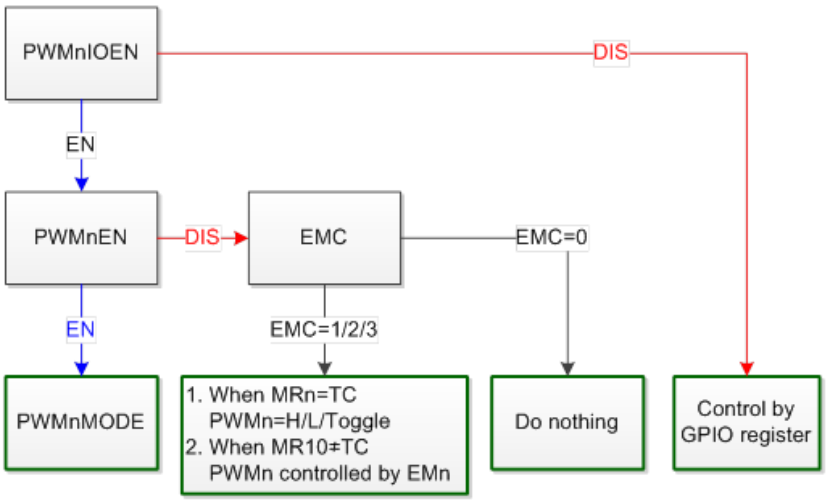
Bit	Field	Access	Initial	Description
31:12	–	–	0	Reserved
11	EM11	RW	0	When EMC11≠00b and MR11≠TC, this bit will drive the state of CT16Bn_PWM11 output 0: Drive Low 1: Drive High
10	EM10	RW	0	When EMC10≠00b and MR10≠TC, this bit will drive the state of CT16Bn_PWM10 output 0: Drive Low 1: Drive High
9	EM9	RW	0	When EMC9≠00b and MR9≠TC, this bit will drive the state of CT16Bn_PWM9 output 0: Drive Low 1: Drive High
8	EM8	RW	0	When EMC8≠00b and MR8≠TC, this bit will drive the state of CT16Bn_PWM8 output 0: Drive Low 1: Drive High
7	EM7	RW	0	When EMC7≠00b and MR7≠TC, this bit will drive the state of CT16Bn_PWM7 output 0: Drive Low 1: Drive High
6	EM6	RW	0	When EMC6≠00b and MR6≠TC, this bit will drive the state of CT16Bn_PWM6 output 0: Drive Low 1: Drive High
5	EM5	RW	0	When EMC5≠00b and MR5≠TC, this bit will drive the state of CT16Bn_PWM5 output 0: Drive Low 1: Drive High
4	EM4	RW	0	When EMC4≠00b and MR4≠TC, this bit will drive the state of CT16Bn_PWM4 output 0: Drive Low 1: Drive High
3	EM3	RW	0	When EMC3≠00b and MR3≠TC, this bit will drive the state of CT16Bn_PWM3 output 0: Drive Low 1: Drive High
2	EM2	RW	0	When EMC2≠00b and MR2≠TC, this bit will drive the state of CT16Bn_PWM2 output 0: Drive Low 1: Drive High
1	EM1	RW	0	When EMC1≠00b and MR1≠TC, this bit will drive the state of CT16Bn_PWM1 output 0: Drive Low 1: Drive High
0	EM0	RW	0	When EMC0≠00b and MR0≠TC, this bit will drive the state of

Bit	Field	Access	Initial	Description
				CT16Bn_PWM0 output 0: Drive Low 1: Drive High

8.10.23 CT16Bn External Match Control register (CT16Bn_EMC) (n=8)

Address Offset: 0x5C

The External Match Control register provides control of CT16B8_PWM[11:0]. If the match outputs are configured as PWM output, the function of the external match registers is determined by the PWM rules.



Bit	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23:22	EMC11	RW	0	CT16Bn_PWM11 functionality when MR11=TC 0: Do nothing 1: CT16Bn_PWM11 pin is LOW 2: CT16Bn_PWM11 pin is HIGH 3: Toggle CT16Bn_PWM11 pin
21:20	EMC10	RW	0	CT16Bn_PWM10 functionality when MR10=TC 0: Do nothing 1: CT16Bn_PWM10 pin is LOW 2: CT16Bn_PWM10 pin is HIGH 3: Toggle CT16Bn_PWM10 pin
19:18	EMC9	RW	0	CT16Bn_PWM9 functionality when MR9=TC 0: Do nothing 1: CT16Bn_PWM9 pin is LOW 2: CT16Bn_PWM9 pin is HIGH 3: Toggle CT16Bn_PWM9 pin
17:16	EMC8	RW	0	CT16Bn_PWM8 functionality when MR8=TC 0: Do nothing 1: CT16Bn_PWM8 pin is LOW 2: CT16Bn_PWM8 pin is HIGH 3: Toggle CT16Bn_PWM8 pin
15:14	EMC7	RW	0	CT16Bn_PWM7 functionality when MR7=TC 0: Do nothing 1: CT16Bn_PWM7 pin is LOW 2: CT16Bn_PWM7 pin is HIGH 3: Toggle CT16Bn_PWM7 pin
13:12	EMC6	RW	0	CT16Bn_PWM6 functionality when MR6=TC 0: Do nothing 1: CT16Bn_PWM6 pin is LOW 2: CT16Bn_PWM6 pin is HIGH 3: Toggle CT16Bn_PWM6 pin

Bit	Field	Access	Initial	Description
11:10	EMC5	RW	0	CT16Bn_PWM5 functionality when MR5=TC 0: Do nothing 1: CT16Bn_PWM5 pin is LOW 2: CT16Bn_PWM5 pin is HIGH 3: Toggle CT16Bn_PWM5 pin
9:8	EMC4	RW	0	CT16Bn_PWM4 functionality when MR4=TC 0: Do nothing 1: CT16Bn_PWM4 pin is LOW 2: CT16Bn_PWM4 pin is HIGH 3: Toggle CT16Bn_PWM4 pin
7:6	EMC3	RW	0	CT16Bn_PWM3 functionality when MR3=TC 0: Do nothing 1: CT16Bn_PWM3 pin is LOW 2: CT16Bn_PWM3 pin is HIGH 3: Toggle CT16Bn_PWM3 pin
5:4	EMC2	RW	0	CT16Bn_PWM2 functionality when MR2=TC 0: Do nothing 1: CT16Bn_PWM2 pin is LOW 2: CT16Bn_PWM2 pin is HIGH 3: Toggle CT16Bn_PWM2 pin
3:2	EMC1	RW	0	CT16Bn_PWM1 functionality when MR1=TC 0: Do nothing 1: CT16Bn_PWM1 pin is LOW 2: CT16Bn_PWM1 pin is HIGH 3: Toggle CT16Bn_PWM1 pin
1:0	EMC0	RW	0	CT16Bn_PWM0 functionality when MR0=TC 0: Do nothing 1: CT16Bn_PWM0 pin is LOW 2: CT16Bn_PWM0 pin is HIGH 3: Toggle CT16Bn_PWM0 pin

8.10.24 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 independently 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 on the CT16Bn_PWMCTRL[3:0] outputs. 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	Field	Access	Initial	Description
31:30	PWM3NIOEN	RW	0	CT16Bn_PWM3N/GPIO selection bit 0: CT16Bn_PWM3N pin is act as GPIO 1: CT16Bn_PWM3N pin outputs the inverse signal with dead-band of CT16Bn_PWM3, but same High signal during dead-band period 2: CT16Bn_PWM3N pin outputs the inverse signal with dead-band of CT16Bn_PWM3, but same Low signal during dead-band period 3: CT16Bn_PWM3N pin outputs the same signal with dead-band of CT16Bn_PWM3
29:28	PWM2NIOEN	RW	0	CT16Bn_PWM0N/GPIO selection bit 0: CT16Bn_PWM2N pin is act as GPIO 1: CT16Bn_PWM2N pin outputs the inverse signal with dead-band of CT16Bn_PWM0, but same High signal during dead-band period 2: CT16Bn_PWM2N pin outputs the inverse signal with dead-band of CT16Bn_PWM0, but same Low signal during dead-band period 3: CT16Bn_PWM2N pin outputs the same signal with dead-band of CT16Bn_PWM0
27:26	PWM1NIOEN	RW	0	CT16Bn_PWM0N/GPIO selection bit 0: CT16Bn_PWM1N pin is act as GPIO 1: CT16Bn_PWM1N pin outputs the inverse signal with dead-band of

Bit	Field	Access	Initial	Description
				CT16Bn_PWM0, but same High signal during dead-band period 2: CT16Bn_PWM1N pin outputs the inverse signal with dead-band of CT16Bn_PWM0, but same Low signal during dead-band period 3: CT16Bn_PWM1N pin outputs the same signal with dead-band of CT16Bn_PWM0
25:24	PWM0NIOEN	RW	0	CT16Bn_PWM0N/GPIO selection bit 0: CT16Bn_PWM0N pin is act as GPIO 1: CT16Bn_PWM0N pin outputs the inverse signal with dead-band of CT16Bn_PWM0, but same High signal during dead-band period 2: CT16Bn_PWM0N pin outputs the inverse signal with dead-band of CT16Bn_PWM0, but same Low signal during dead-band period 3: CT16Bn_PWM0N pin outputs the same signal with dead-band of CT16Bn_PWM0
23	PWM3IOEN	RW	0	CT16Bn_PWM3/GPIO selection 0: CT16Bn_PWM3 pin is act as GPIO 1: CT16Bn_PWM3 pin act as match output, and output depends on PWM3EN bit
22	PWM2IOEN	RW	0	CT16Bn_PWM2/GPIO selection 0: CT16Bn_PWM2 pin is act as GPIO 1: CT16Bn_PWM2 pin act as match output, and output depends on PWM2EN bit
21	PWM1IOEN	RW	0	CT16Bn_PWM1/GPIO selection 0: CT16Bn_PWM1 pin is act as GPIO 1: CT16Bn_PWM1 pin act as match output, and output depends on PWM1EN bit
20	PWM0IOEN	RW	0	CT16Bn_PWM0/GPIO selection 0: CT16Bn_PWM0 pin is act as GPIO 1: CT16Bn_PWM0 pin act as match output, and output depends on PWM0EN bit
19:12	–	–	0	Reserved
11:10	PWM3MODE	RW	0	PWM3 output mode 0: PWM mode 1 PWM3 is 0 when TC<MR3 during Up-counting period PWM3 is 0 when TC≤MR3 during Down-counting period 1: PWM mode 2 PWM3 is 1 when TC<MR3 during Up-counting period PWM3 is 1 when TC≤MR3 during Down-counting period 2: PWM3 is forced to 0 3: PWM3 is forced to 1
9:8	PWM2MODE	RW	0	PWM2 output mode 0: PWM mode 1 PWM2 is 0 when TC<MR2 during Up-counting period PWM2 is 0 when TC≤MR2 during Down-counting period 1: PWM mode 2 PWM2 is 1 when TC<MR2 during Up-counting period PWM2 is 2 when TC≤MR2 during Down-counting period 2: PWM2 is forced to 0 3: PWM2 is forced to 1
7:6	PWM1MODE	RW	0	PWM1 output mode 0: PWM mode 1 PWM1 is 0 when TC<MR1 during Up-counting period PWM1 is 0 when TC≤MR1 during Down-counting period 1: PWM mode 2 PWM1 is 1 when TC<MR1 during Up-counting period PWM1 is 1 when TC≤MR1 during Down-counting period 2: PWM1 is forced to 0 3: PWM1 is forced to 1
5:4	PWM0MODE	RW	0	PWM0 output mode 0: PWM mode 1 PWM0 is 0 when TC<MR0 during Up-counting period PWM0 is 0 when TC≤MR0 during Down-counting period 1: PWM mode 2 PWM0 is 1 when TC<MR0 during Up-counting period PWM0 is 1 when TC≤MR0 during Down-counting period 2: PWM0 is forced to 0

Bit	Field	Access	Initial	Description
				3: PWM0 is forced to 1
3	PWM3EN	RW	0	PWM2 enable 0: CT16Bn_PWM3 is controlled by EMC3 1: Enable PWM mode for CT16Bn_PWM3
2	PWM2EN	RW	0	PWM2 enable 0: CT16Bn_PWM2 is controlled by EMC2 1: Enable PWM mode for CT16Bn_PWM2
1	PWM1EN	RW	0	PWM1 enable 0: CT16Bn_PWM1 is controlled by EMC1 1: Enable PWM mode for CT16Bn_PWM1
0	PWM0EN	RW	0	PWM0 enable 0: CT16Bn_PWM0 is controlled by EMC0 1: Enable PWM mode for CT16Bn_PWM0

8.10.25 CT16Bn PWM Control register (CT16Bn_PWMCTRL) (n=1,2,5)

Address Offset: 0x34

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

For CT16B1/2/5, a maximum of 4 single edge controlled PWM outputs can be selected on the CT16Bn_PWMCTRL[3:0] outputs. 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	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23	PWM3IOEN	RW	0	CT16Bn_PWM3/GPIO selection 0: CT16Bn_PWM3 pin is act as GPIO 1: CT16Bn_PWM3 pin act as match output, and output depends on PWM3EN bit
22	PWM2IOEN	RW	0	CT16Bn_PWM2/GPIO selection 0: CT16Bn_PWM2 pin is act as GPIO 1: CT16Bn_PWM2 pin act as match output, and output depends on PWM2EN bit
21	PWM1IOEN	RW	0	CT16Bn_PWM1/GPIO selection 0: CT16Bn_PWM1 pin is act as GPIO 1: CT16Bn_PWM1 pin act as match output, and output depends on PWM1EN bit
20	PWM0IOEN	RW	0	CT16Bn_PWM0/GPIO selection 0: CT16Bn_PWM0 pin is act as GPIO 1: CT16Bn_PWM0 pin act as match output, and output depends on PWM0EN bit
19:12	–	–	0	Reserved
11:10	PWM3MODE	RW	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 PWM3 is 1 when TC<MR3 during Up-counting period PWM3 is 1 when TC≤MR3 during Down-counting period 2: PWM3 is forced to 0 3: PWM3 is forced to 1
9:8	PWM2MODE	RW	0	PWM2 output mode 00: PWM mode 1 PWM2 is 0 when TC<MR2 during Up-counting period PWM2 is 0 when TC≤MR2 during Down-counting period 01: PWM mode 2 PWM2 is 1 when TC<MR2 during Up-counting period PWM2 is 2 when TC≤MR2 during Down-counting period 2: PWM2 is forced to 0

Bit	Field	Access	Initial	Description
				3: PWM2 is forced to 1
7:6	PWM1MODE	RW	0	PWM1 output mode 00: PWM mode 1 PWM1 is 0 when TC<MR1 during Up-counting period PWM1 is 0 when TC≤MR1 during Down-counting period 01: PWM mode 2 PWM1 is 1 when TC<MR1 during Up-counting period PWM1 is 1 when TC≤MR1 during Down-counting period 2: PWM1 is forced to 0 3: PWM1 is forced to 1
5:4	PWM0MODE	RW	0	PWM0 output mode 00: PWM mode 1 PWM0 is 0 when TC<MR0 during Up-counting period PWM0 is 0 when TC≤MR0 during Down-counting period 01: PWM mode 2 PWM0 is 1 when TC<MR0 during Up-counting period PWM0 is 1 when TC≤MR0 during Down-counting period 2: PWM0 is forced to 0 3: PWM0 is forced to 1
3	PWM3EN	RW	0	PWM2 enable 0: CT16Bn_PWM3 is controlled by EMC3 1: Enable PWM mode for CT16Bn_PWM3
2	PWM2EN	RW	0	PWM2 enable 0: CT16Bn_PWM2 is controlled by EMC2 1: Enable PWM mode for CT16Bn_PWM2
1	PWM1EN	RW	0	PWM1 enable 0: CT16Bn_PWM1 is controlled by EMC1 1: Enable PWM mode for CT16Bn_PWM1
0	PWM0EN	RW	0	PWM0 enable 0: CT16Bn_PWM0 is controlled by EMC0 1: Enable PWM mode for CT16Bn_PWM0

8.10.26 CT16Bn PWM Control register (CT16Bn_PWMCTRL) (n=3,4)

Address Offset: 0x34

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

For CT16B3/4, a maximum of 2 single edge controlled PWM outputs can be selected on the CT16Bn_PWMCTRL[1:0] outputs. 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	Field	Access	Initial	Description
31:28	–	–	0	Reserved
27:26	PWM1NIOEN	RW	0	CT16Bn_PWM0N/GPIO selection bit 0: CT16Bn_PWM1N pin is act as GPIO 1: CT16Bn_PWM1N pin outputs the inverse signal with dead-band of CT16Bn_PWM0, but same High signal during dead-band period 2: CT16Bn_PWM1N pin outputs the inverse signal with dead-band of CT16Bn_PWM0, but same Low signal during dead-band period 3: CT16Bn_PWM1N pin outputs the same signal with dead-band of CT16Bn_PWM0
25:24	PWM0NIOEN	RW	0	CT16Bn_PWM0N/GPIO selection bit 0: CT16Bn_PWM0N pin is act as GPIO 1: CT16Bn_PWM0N pin outputs the inverse signal with dead-band of CT16Bn_PWM0, but same High signal during dead-band period 2: CT16Bn_PWM0N pin outputs the inverse signal with dead-band of CT16Bn_PWM0, but same Low signal during dead-band period 3: CT16Bn_PWM0N pin outputs the same signal with dead-band of CT16Bn_PWM0
23:22	–	–	0	Reserved

Bit	Field	Access	Initial	Description
21	PWM1IOEN	RW	0	CT16Bn_PWM1/GPIO selection 0: CT16Bn_PWM1 pin is act as GPIO 1: CT16Bn_PWM1 pin act as match output, and output depends on PWM1EN bit
20	PWM0IOEN	RW	0	CT16Bn_PWM0/GPIO selection 0: CT16Bn_PWM0 pin is act as GPIO 1: CT16Bn_PWM0 pin act as match output, and output depends on PWM0EN bit
19:8	–	–	0	Reserved
7:6	PWM1MODE	RW	0	PWM1 output mode 0: PWM mode 1 PWM1 is 0 when TC<MR1 during Up-counting period PWM1 is 0 when TC≤MR1 during Down-counting period 1: PWM mode 2 PWM1 is 1 when TC<MR1 during Up-counting period PWM1 is 1 when TC≤MR1 during Down-counting period 2: PWM1 is forced to 0 3: PWM1 is forced to 1
5:4	PWM0MODE	RW	0	PWM0 output mode 00: PWM mode 1 PWM0 is 0 when TC<MR0 during Up-counting period PWM0 is 0 when TC≤MR0 during Down-counting period 01: PWM mode 2 PWM0 is 1 when TC<MR0 during Up-counting period PWM0 is 1 when TC≤MR0 during Down-counting period 10: PWM0 is forced to 0 11: PWM0 is forced to 1
3:2	–	–	0	Reserved
1	PWM1EN	RW	0	PWM1 enable 0: CT16Bn_PWM1 is controlled by EMC1 1: Enable PWM mode for CT16Bn_PWM1
0	PWM0EN	RW	0	PWM0 enable 0: CT16Bn_PWM0 is controlled by EMC0 1: Enable PWM mode for CT16Bn_PWM0

8.10.27 CT16Bn PWM Control register (CT16Bn_PWMCTRL) (n=8)

Address Offset: 0x60

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

For CT16B8, a maximum of 12 single edge controlled PWM outputs can be selected on the CT16Bn_PWMCTRL[11:0] outputs. 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	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23:22	PWM11MODE	RW	0	PWM11 output 00: PWM mode 1 PWM11 is 0 when TC<MR11 during Up-counting period 01: PWM mode 2 PWM11 is 1 when TC<MR11 during Up-counting period 10: PWM11 is forced to 0. 11: PWM11 is forced to 1.
21:20	PWM10MODE	RW	0	PWM10 output 00: PWM mode 1 PWM10 is 0 when TC<MR10 during Up-counting period 01: PWM mode 2 PWM10 is 1 when TC<MR10 during Up-counting period 10: PWM10 is forced to 0. 11: PWM10 is forced to 1.

Bit	Field	Access	Initial	Description
19:18	PWM9MODE	RW	0	PWM9 output mode 00: PWM mode 1 PWM9 is 0 when TC<MR9 during Up-counting period PWM9 is 1 when TC<MR9 during Up-counting period 10: PWM9 is forced to 0. 11: PWM9 is forced to 1.
17:16	PWM8MODE	RW	0	PWM8 output mode 00: PWM mode 1 PWM8 is 0 when TC<MR8 during Up-counting period 01: PWM mode 2 PWM8 is 1 when TC<MR8 during Up-counting period 10: PWM8 is forced to 0. 11: PWM8 is forced to 1.
15:14	PWM7MODE	RW	0	PWM7 output mode 00: PWM mode 1 PWM7 is 0 when TC<MR7 during Up-counting period 01: PWM mode 2 PWM7 is 1 when TC<MR7 during Up-counting period 10: PWM7 is forced to 0. 11: PWM7 is forced to 1.
13:12	PWM6MODE	RW	0	PWM6 output mode 00: PWM mode 1 PWM6 is 0 when TC<MR6 during Up-counting period 01: PWM mode 2 PWM6 is 1 when TC<MR6 during Up-counting period 10: PWM6 is forced to 0. 11: PWM6 is forced to 1.
11:10	PWM5MODE	RW	0	PWM5 output 00: PWM mode 1 PWM5 is 0 when TC<MR5 during Up-counting period 01: PWM mode 2 PWM5 is 1 when TC<MR5 during Up-counting period 10: PWM5 is forced to 0. 11: PWM5 is forced to 1.
9:8	PWM4MODE	RW	0	PWM4 output 00: PWM mode 1 PWM4 is 0 when TC<MR4 during Up-counting period 01: PWM mode 2 PWM4 is 1 when TC<MR4 during Up-counting period 10: PWM4 is forced to 0. 11: PWM4 is forced to 1.
7:6	PWM3MODE	RW	0	PWM3 output mode 00: PWM mode 1 PWM3 is 0 when TC<MR3 during Up-counting period 01: PWM mode 2 PWM3 is 1 when TC<MR3 during Up-counting period 10: PWM3 is forced to 0. 11: PWM3 is forced to 1.
5:4	PWM2MODE	RW	0	PWM2 output mode 00: PWM mode 1 PWM2 is 0 when TC<MR2 during Up-counting period 01: PWM mode 2 PWM2 is 1 when TC<MR2 during Up-counting period 10: PWM2 is forced to 0. 11: PWM2 is forced to 1.
3:2	PWM1MODE	RW	0	PWM1 output mode 0: PWM mode 1 PWM1 is 0 when TC<MR1 during Up-counting period PWM1 is 0 when TC≤MR1 during Down-counting period 1: PWM mode 2 PWM1 is 1 when TC<MR1 during Up-counting period PWM1 is 1 when TC≤MR1 during Down-counting period 2: PWM1 is forced to 0 3: PWM1 is forced to 1
1:0	PWM0MODE	RW	0	PWM0 output mode

Bit	Field	Access	Initial	Description
				00: PWM mode 1 PWM0 is 0 when TC<MR0 during Up-counting period PWM0 is 0 when TC≤MR0 during Down-counting period 01: PWM mode 2 PWM0 is 1 when TC<MR0 during Up-counting period PWM0 is 1 when TC≤MR0 during Down-counting period 10: PWM0 is forced to 0 11: PWM0 is forced to 1

8.10.28 CT16Bn PWM Enable register (CT16Bn_PWMENB) (n=8)

Address Offset: 0x64

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

For CT16B8, a maximum of 12 single edge controlled PWM outputs can be selected on the CT16B8_PWMCTRL[11:0] outputs. 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	Field	Access	Initial	Description
31:12	–	–	0	Reserved
11	PWM11EN	RW	0	PWM11 enable 0: CT16Bn_PWM11 is controlled by EMC11 1: Enable PWM mode for CT16Bn_PWM11
10	PWM10EN	RW	0	PWM10 enable 0: CT16Bn_PWM10 is controlled by EMC10 1: Enable PWM mode for CT16Bn_PWM10
9	PWM9EN	RW	0	PWM9 enable 0: CT16Bn_PWM9 is controlled by EMC9 1: Enable PWM mode for CT16Bn_PWM9
8	PWM8EN	RW	0	PWM8 enable 0: CT16Bn_PWM8 is controlled by EMC8 1: Enable PWM mode for CT16Bn_PWM8
7	PWM7EN	RW	0	PWM7 enable 0: CT16Bn_PWM7 is controlled by EMC7 1: Enable PWM mode for CT16Bn_PWM7
6	PWM6EN	RW	0	PWM6 enable 0: CT16Bn_PWM6 is controlled by EMC6 1: Enable PWM mode for CT16Bn_PWM6
5	PWM5EN	RW	0	PWM5 enable 0: CT16Bn_PWM5 is controlled by EMC5 1: Enable PWM mode for CT16Bn_PWM5
4	PWM4EN	RW	0	PWM4 enable 0: CT16Bn_PWM4 is controlled by EMC4 1: Enable PWM mode for CT16Bn_PWM4
3	PWM3EN	RW	0	PWM3 enable 0: CT16Bn_PWM3 is controlled by EMC3 1: Enable PWM mode for CT16Bn_PWM3
2	PWM2EN	RW	0	PWM2 enable 0: CT16Bn_PWM2 is controlled by EMC2 1: Enable PWM mode for CT16Bn_PWM2
1	PWM1EN	RW	0	PWM1 enable 0: CT16Bn_PWM1 is controlled by EMC1 1: Enable PWM mode for CT16Bn_PWM1
0	PWM0EN	RW	0	PWM0 enable 0: CT16Bn_PWM0 is controlled by EMC0 1: Enable PWM mode for CT16Bn_PWM0

8.10.29 CT16Bn PWM IO Enable register (CT16Bn_PWMIOENB) (n=8)

Address Offset: 0x68

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

For CT16B8, a maximum of 12 single edge controlled PWM outputs can be selected on the CT16Bn_PWMCTRL[11:0] outputs. 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	Field	Access	Initial	Description
31:12	–	–	0	Reserved
11	PWM11IOEN	RW	0	CT16Bn_PWM11/GPIO selection 0: CT16Bn_PWM11 pin is act as GPIO 1: CT16Bn_PWM11 pin act as match output, and output depends on PWM11EN bit
10	PWM10IOEN	RW	0	CT16Bn_PWM10/GPIO selection 0: CT16Bn_PWM10 pin is act as GPIO 1: CT16Bn_PWM10 pin act as match output, and output depends on PWM10EN bit
9	PWM9IOEN	RW	0	CT16Bn_PWM9/GPIO selection 0: CT16Bn_PWM9 pin is act as GPIO 1: CT16Bn_PWM9 pin act as match output, and output depends on PWM9EN bit
8	PWM8IOEN	RW	0	CT16Bn_PWM8/GPIO selection 0: CT16Bn_PWM8 pin is act as GPIO 1: CT16Bn_PWM8 pin act as match output, and output depends on PWM8EN bit
7	PWM7IOEN	RW	0	CT16Bn_PWM7/GPIO selection 0: CT16Bn_PWM7 pin is act as GPIO 1: CT16Bn_PWM7 pin act as match output, and output depends on PWM7EN bit
6	PWM6IOEN	RW	0	CT16Bn_PWM6/GPIO selection 0: CT16Bn_PWM6 pin is act as GPIO 1: CT16Bn_PWM6 pin act as match output, and output depends on PWM6EN bit
5	PWM5IOEN	RW	0	CT16Bn_PWM5/GPIO selection 0: CT16Bn_PWM5 pin is act as GPIO 1: CT16Bn_PWM5 pin act as match output, and output depends on PWM5EN bit
4	PWM4IOEN	RW	0	CT16Bn_PWM4/GPIO selection 0: CT16Bn_PWM4 pin is act as GPIO 1: CT16Bn_PWM4 pin act as match output, and output depends on PWM4EN bit
3	PWM3IOEN	RW	0	CT16Bn_PWM3/GPIO selection 0: CT16Bn_PWM3 pin is act as GPIO 1: CT16Bn_PWM3 pin act as match output, and output depends on PWM3EN bit
2	PWM2IOEN	RW	0	CT16Bn_PWM2/GPIO selection 0: CT16Bn_PWM2 pin is act as GPIO 1: CT16Bn_PWM2 pin act as match output, and output depends on PWM2EN bit
1	PWM1IOEN	RW	0	CT16Bn_PWM1/GPIO selection 0: CT16Bn_PWM1 pin is act as GPIO 1: CT16Bn_PWM1 pin act as match output, and output depends on PWM1EN bit
0	PWM0IOEN	RW	0	CT16Bn_PWM0/GPIO selection 0: CT16Bn_PWM0 pin is act as GPIO 1: CT16Bn_PWM0 pin act as match output, and output depends on PWM0EN bit

8.10.30 CT16Bn Timer Raw Interrupt Status register (CT16Bn_RIS) (n=0)

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	Field	Access	Initial	Description
31	BRKIF	R	0	Break Interrupt flag 0: No break condition occurs 1: Break condition occurs
30:6	–	–	0	Reserved

Bit	Field	Access	Initial	Description
5	MR9IF	R	0	Match channel 9 interrupt flag 0: No interrupt on match channel 9 1: Interrupt requirements met on match channel 9
4	CAP0IF	R	0	Capture channel 0 interrupt flag 0: No interrupt on CAP0 1: Interrupt requirements met on CAP0
3	MR3IF	R	0	Match channel 3 interrupt flag 0: No interrupt on match channel 3 1: Interrupt requirements met on match channel 3
2	MR2IF	R	0	Match channel 2 interrupt flag 0: No interrupt on match channel 2 1: Interrupt requirements met on match channel 2
1	MR1IF	R	0	Match channel 1 interrupt flag 0: No interrupt on match channel 1 1: Interrupt requirements met on match channel 1
0	MR0IF	R	0	Match channel 0 interrupt flag 0: No interrupt on match channel 0 1: Interrupt requirements met on match channel 0

8.10.31 CT16Bn Timer Raw Interrupt Status register (CT16Bn_RIS) (n=1,2,5)

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	Field	Access	Initial	Description
31:6	–	–	0	Reserved
5	MR9IF	R	0	Match channel 9 interrupt flag 0: No interrupt on match channel 9 1: Interrupt requirements met on match channel 9
4	CAP0IF	R	0	Capture channel 0 interrupt flag 0: No interrupt on CAP0 1: Interrupt requirements met on CAP0
3	MR3IF	R	0	Match channel 3 interrupt flag 0: No interrupt on match channel 3 1: Interrupt requirements met on match channel 3
2	MR2IF	R	0	Match channel 2 interrupt flag 0: No interrupt on match channel 2 1: Interrupt requirements met on match channel 2
1	MR1IF	R	0	Match channel 1 interrupt flag 0: No interrupt on match channel 1 1: Interrupt requirements met on match channel 1
0	MR0IF	R	0	Match channel 0 interrupt flag 0: No interrupt on match channel 0 1: Interrupt requirements met on match channel 0

8.10.32 CT16Bn Timer Raw Interrupt Status register (CT16Bn_RIS) (n=3)

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	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3	MR9IF	R	0	Match channel 9 interrupt flag 0: No interrupt on match channel 9 1: Interrupt requirements met on match channel 9
2	–	–	0	Reserved
1	MR1IF	R	0	Match channel 1 interrupt flag 0: No interrupt on match channel 1 1: Interrupt requirements met on match channel 1
0	MR0IF	R	0	Match channel 0 interrupt flag 0: No interrupt on match channel 0 1: Interrupt requirements met on match channel 0

8.10.33 CT16Bn Timer Raw Interrupt Status register (CT16Bn_RIS) (n=4)

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	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3	MR9IF	R	0	Match channel 9 interrupt flag 0: No interrupt on match channel 9 1: Interrupt requirements met on match channel 9
2	CAP0IF	R	0	Capture channel 0 interrupt flag 0: No interrupt on CAP0 1: Interrupt requirements met on CAP0
1	MR1IF	R	0	Match channel 1 interrupt flag 0: No interrupt on match channel 1 1: Interrupt requirements met on match channel 1
0	MR0IF	R	0	Match channel 0 interrupt flag 0: No interrupt on match channel 0 1: Interrupt requirements met on match channel 0

8.10.34 CT16Bn Timer Raw Interrupt Status register (CT16Bn_RIS) (n=6,7)

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	Field	Access	Initial	Description
31:1	–	–	0	Reserved
0	MR0IF	R	0	Match channel 0 interrupt flag 0: No interrupt on match channel 0 1: Interrupt requirements met on match channel 0

8.10.35 CT16Bn Timer Raw Interrupt Status register (CT16Bn_RIS) (n=8)

Address Offset: 0x6C

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	Field	Access	Initial	Description
31:14	–	–	0	Reserved
13	MRPERIODIF	R	0	Match channel of Period interrupt flag 0: No interrupt 1: Interrupt requirements met
12	–	–	0	Reserved
11	MR11IF	R	0	Match channel 11 interrupt flag 0: No interrupt on match channel 11 1: Interrupt requirements met on match channel 11
10	MR10IF	R	0	Match channel 10 interrupt flag 0: No interrupt on match channel 10 1: Interrupt requirements met on match channel 10
9	MR9IF	R	0	Match channel 9 interrupt flag 0: No interrupt on match channel 9 1: Interrupt requirements met on match channel 9
8	MR8IF	R	0	Match channel 8 interrupt flag 0: No interrupt on match channel 8 1: Interrupt requirements met on match channel 8
7	MR7IF	R	0	Match channel 7 interrupt flag 0: No interrupt on match channel 7 1: Interrupt requirements met on match channel 7
6	MR6IF	R	0	Match channel 6 interrupt flag 0: No interrupt on match channel 6 1: Interrupt requirements met on match channel 6
5	MR5IF	R	0	Match channel 5 interrupt flag 0: No interrupt on match channel 5 1: Interrupt requirements met on match channel 5
4	MR4IF	R	0	Match channel 4 interrupt flag 0: No interrupt on match channel 4 1: Interrupt requirements met on match channel 4
3	MR3IF	R	0	Match channel 3 interrupt flag 0: No interrupt on match channel 3 1: Interrupt requirements met on match channel 3
2	MR2IF	R	0	Match channel 2 interrupt flag 0: No interrupt on match channel 2 1: Interrupt requirements met on match channel 2
1	MR1IF	R	0	Match channel 1 interrupt flag 0: No interrupt on match channel 1 1: Interrupt requirements met on match channel 1
0	MR0IF	R	0	Match channel 0 interrupt flag 0: No interrupt on match channel 0 1: Interrupt requirements met on match channel 0

8.10.36 CT16Bn Timer Interrupt Clear register (CT16Bn_IC) (n=0)

Address Offset: 0x3C

Bit	Field	Access	Initial	Description
31	BRKIC	W	0	BRKIF clear bit 0: No effect 1: Clear BRKIF
30:6	–	–	0	Reserved
5	MR9IC	W	0	MR9IF clear bit 0: No effect 1: Clear MR9IF
4	CAP0IC	W	0	CAP0IF clear bit 0: No effect 1: Clear CAP0IF
3	MR3IC	W	0	MR3IF clear bit 0: No effect 1: Clear MR3IF
2	MR2IC	W	0	MR2IF clear bit 0: No effect 1: Clear MR2IF
1	MR1IC	W	0	MR1IF clear bit 0: No effect 1: Clear MR1IF
0	MR0IC	W	0	MR0IF clear bit 0: No effect 1: Clear MR0IF

8.10.37 CT16Bn Timer Interrupt Clear register (CT16Bn_IC) (n=1,2,5)

Address Offset: 0x3C

Bit	Field	Access	Initial	Description
31:6	–	–	0	Reserved
5	MR9IC	W	0	MR9IF clear bit 0: No effect 1: Clear MR9IF
4	CAP0IC	W	0	CAP0IF clear bit 0: No effect 1: Clear CAP0IF
3	MR3IC	W	0	MR3IF clear bit 0: No effect 1: Clear MR3IF
2	MR2IC	W	0	MR2IF clear bit 0: No effect 1: Clear MR2IF
1	MR1IC	W	0	MR1IF clear bit 0: No effect 1: Clear MR1IF
0	MR0IC	W	0	MR0IF clear bit 0: No effect 1: Clear MR0IF

8.10.38 CT16Bn Timer Interrupt Clear register (CT16Bn_IC) (n=3)

Address Offset: 0x3C

Bit	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3	MR9IC	W	0	MR9IF clear bit 0: No effect 1: Clear MR9IF
2	–	–	0	Reserved
1	MR1IC	W	0	MR1IF clear bit 0: No effect 1: Clear MR1IF
0	MR0IC	W	0	MR0IF clear bit 0: No effect 1: Clear MR0IF

8.10.39 CT16Bn Timer Interrupt Clear register (CT16Bn_IC) (n=4)

Address Offset: 0x3C

Bit	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3	MR9IC	W	0	MR9IF clear bit 0: No effect 1: Clear MR9IF
2	CAP0IC	W	0	CAP0IF clear bit 0: No effect 1: Clear CAP0IF
1	MR1IC	W	0	MR1IF clear bit 0: No effect 1: Clear MR1IF
0	MR0IC	W	0	MR0IF clear bit 0: No effect 1: Clear MR0IF

8.10.40 CT16Bn Timer Interrupt Clear register (CT16Bn_IC) (n=6,7)

Address Offset: 0x3C

Bit	Field	Access	Initial	Description
31:1	–	–	0	Reserved
0	MR0IC	W	0	MR0IF clear bit 0: No effect 1: Clear MR0IF

8.10.41 CT16Bn Timer Interrupt Clear register (CT16Bn_IC) (n=8)

Address Offset: 0x70

Bit	Field	Access	Initial	Description
31:14	–	–	0	Reserved
13	MRPERIODIC	W	0	MRPERIODIF clear bit 0: No effect 1: Clear MR12IF
12	–	–	0	Reserved
11	MR11IC	W	0	MR11IF clear bit 0: No effect 1: Clear MR11IF
10	MR10IC	W	0	MR10IF clear bit 0: No effect 1: Clear MR10IF
9	MR9IC	W	0	MR9IF clear bit 0: No effect 1: Clear MR9IF
8	MR8IC	W	0	MR8IF clear bit 0: No effect 1: Clear MR8IF
7	MR7IC	W	0	MR7IF clear bit 0: No effect 1: Clear MR7IF
6	MR6IC	W	0	MR6IF clear bit 0: No effect 1: Clear MR6IF
5	MR5IC	W	0	MR5IF clear bit 0: No effect 1: Clear MR5IF
4	MR4IC	W	0	MR4IF clear bit 0: No effect 1: Clear MR4IF
3	MR3IC	W	0	MR3IF clear bit 0: No effect 1: Clear MR3IF
2	MR2IC	W	0	MR2IF clear bit 0: No effect 1: Clear MR2IF
1	MR1IC	W	0	MR1IF clear bit 0: No effect 1: Clear MR1IF
0	MR0IC	W	0	MR0IF clear bit 0: No effect 1: Clear MR0IF

8.10.42 CT16Bn PWMmN Dead-band Period register (CT16Bn_PWMmNDB) (n=0,3,4)

Address Offset:

n = 0 : 0x44 (m=0), 0x48 (m=1), 0x4C (m=2), 0x50 (m=3)
n = 3,4 : 0x44 (m=0), 0x48 (m=1)

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

PWMxNIOEN	PWMN	DB	PWMN dead-band period time
0	GPIO	No effect	X
>0	V	0	**0**
>0	V	1	1 * PCLK * (PR+1)
>0	V	2	2 * PCLK * (PR+1)

Bit	Field	Access	Initial	Description
31:10	–	–	0	Reserved
9:0	DB	RW	0	Count of PWMmN output dead-band period time PWM0N output dead-band period time=DB*CT16Bn_PCLK*(PR+1) cycle

8.10.43 CT16Bn PWM Load Mode Control register (CT16Bn_LOADCTRL) (n=0,1,2,5)

Address Offset: 0x74

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:6	LOAD_MR3	RW	01b	MR3 load mode selection in center-aligned mode 00b: No effect 01b: Load MR3 value at TC=0 and MR3 value is unchanged at TC=MR9 10b: Reserved 11b: Load MR3 value at TC=0 and TC=MR9
5:4	LOAD_MR2	RW	01b	MR2 load mode selection in center-aligned mode 00b: No effect 01b: Load MR2 value at TC=0 and MR2 value is unchanged at TC=MR9 10b: Reserved 11b: Load MR2 value at TC=0 and TC=MR9
3:2	LOAD_MR1	RW	01b	MR1 load mode selection in center-aligned mode 00b: No effect 01b: Load MR1 value at TC=0 and MR1 value is unchanged at TC=MR9 10b: Reserved 11b: Load MR1 value at TC=0 and TC=MR9
1:0	LOAD_MR0	RW	01b	MR0 load mode selection in center-aligned mode 00b: No effect 01b: Load MR0 value at TC=0 and MR0 value is unchanged at TC=MR9 10b: Reserved 11b: Load MR0 value at TC=0 and TC=MR9

8.10.44 CT16Bn DMA Mode register (CT16Bn_DMA) (n=0,1,2,5)

Address Offset: 0xD0

Bit	Field	Access	Initial	Description
31:6	–	–	0	Reserved
5	DMA_MR9	RW	0	MR9 DMA request active enable 0: MR9 DMA request cannot issue 1: MR9 DMA request can issue
4	DMA_CAP0	RW	0	CAP0 DMA request active enable 0: CAP0 DMA request cannot issue 1: CAP0 DMA request can issue
3	DMA_MR3	RW	0	MR3 DMA request active enable 0: MR3 DMA request cannot issue 1: MR3 DMA request can issue
2	DMA_MR2	RW	0	MR2 DMA request active enable 0: MR2 DMA request cannot issue 1: MR2 DMA request can issue
1	DMA_MR1	RW	0	MR1 DMA request active enable 0: MR1 DMA request cannot issue 1: MR1 DMA request can issue
0	DMA_MR0	RW	0	MR0 DMA request active enable 0: MR0 DMA request cannot issue 1: MR0 DMA request can issue

8.10.45 CT16Bn DMA Mode register (CT16Bn_DMA) (n=3)

Address Offset: 0xD0

Bit	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3	DMA_MR9	RW	0	MR9 DMA request active enable 0: MR9 DMA request cannot issue 1: MR9 DMA request can issue
2	–	–	0	Reserved
1	DMA_MR1	RW	0	MR1 DMA request active enable 0: MR1 DMA request cannot issue 1: MR1 DMA request can issue
0	DMA_MR0	RW	0	MR0 DMA request active enable 0: MR0 DMA request cannot issue 1: MR0 DMA request can issue

8.10.46 CT16Bn DMA Mode register (CT16Bn_DMA) (n=4)

Address Offset: 0xD0

Bit	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3	DMA_MR9	RW	0	MR9 DMA request active enable 0: MR9 DMA request cannot issue 1: MR9 DMA request can issue
2	DMA_CAP0	RW	0	CAP0 DMA request active enable 0: CAP0 DMA request cannot issue 1: CAP0 DMA request can issue
1	DMA_MR1	RW	0	MR1 DMA request active enable 0: MR1 DMA request cannot issue 1: MR1 DMA request can issue
0	DMA_MR0	RW	0	MR0 DMA request active enable 0: MR0 DMA request cannot issue 1: MR0 DMA request can issue

8.10.47 CT16Bn DMA Mode register (CT16Bn_DMA) (n=6,7)

Address Offset: 0xD0

Bit	Field	Access	Initial	Description
31:1	–	–	0	Reserved
0	DMA_MR0	RW	0	MR0 DMA request active enable 0: MR0 DMA request cannot issue 1: MR0 DMA request can issue

8.10.48 CT16Bn DMA Mode register (CT16Bn_DMA) (n=8)

Address Offset: 0xD0

Bit	Field	Access	Initial	Description
31:14	–	–	0	Reserved
13	DMA_MR_PERIOD	RW	0	MR_PERIOD DMA request active enable 0: MR_PERIOD DMA request cannot issue 1: MR_PERIOD DMA request can issue
12:0	–	–	0	Reserved

8.10.49 CT16Bn DMA MRm Alias register 1 (CT16Bn_ DMAMRA1) (n=0,1,2,3,4,5,8)

Address Offset: 0xD4

Bit	Field	Access	Initial	Description
31:16	MR1_ALIAS	W	0	MR1 alias for DMA access
15:0	MR0_ALIAS	W	0	MR0 alias for DMA access

8.10.50 CT16Bn DMA MRm Alias register 2 (CT16Bn_ DMAMRA2) (n=0,1,2,5,8)

Address Offset: 0xD8

Bit	Field	Access	Initial	Description
31:16	MR3_ALIAS	W	0	MR1 alias for DMA access
15:0	MR2_ALIAS	W	0	MR0 alias for DMA access

8.10.51 CT16Bn DMA MRm Alias register 3 (CT16Bn_ DMAMRA3) (n=8)

Address Offset: 0xDC

Bit	Field	Access	Initial	Description
31:16	MR5_ALIAS	W	0	MR5 alias for DMA access
15:0	MR4_ALIAS	W	0	MR4 alias for DMA access

8.10.52 CT16Bn DMA MRm Alias register 4 (CT16Bn_ DMAMRA4) (n=8)

Address Offset: 0xE0

Bit	Field	Access	Initial	Description
31:16	MR7_ALIAS	W	0	MR7 alias for DMA access
15:0	MR6_ALIAS	W	0	MR6 alias for DMA access

8.10.53 CT16Bn DMA MRm Alias register 5 (CT16Bn_ DMAMRA5) (n=8)

Address Offset: 0xE4

Bit	Field	Access	Initial	Description
31:16	MR9_ALIAS	W	0	MR9 alias for DMA access
15:0	MR8_ALIAS	W	0	MR8 alias for DMA access

8.10.54 CT16Bn DMA MRm Alias register 6 (CT16Bn_ DMAMRA5) (n=8)

Address Offset: 0xE8

Bit	Field	Access	Initial	Description
31:16	MR11_ALIAS	W	0	MR11 alias for DMA access
15:0	MR10_ALIAS	W	0	MR10 alias for DMA access

8.10.55 CT16Bn Break Function Control register (CT16Bn_ BRKCTRL) (n=0)

Address Offset: 0xE0

Bit	Field	Access	Initial	Description
31	–	–	0	Reserved
30	BRKIE	RW	0	Enable generating an interrupt when BRKIF = 1 0: Disable 1: Enable
29:7	–	–	0	Reserved
6:4	BRKDB	RW	0	Break pin (BRK) debounce time 0: Debounce time=1*CT16B0_PCLK 1: Debounce time=2*CT16B0_PCLK 2: Debounce time=4*CT16B0_PCLK 3: Debounce time=8*CT16B0_PCLK 4: Debounce time=16*CT16B0_PCLK 5: Debounce time=32*CT16B0_PCLK 6: Debounce time=64*CT16B0_PCLK 7: Debounce time=128*CT16B0_PCLK
3	BRKLEVEL	RW	0	The trigger level of PWM channels break function 0: Low level. 1: High level.
2:0	BRKSEL	RW	0	The trigger source selection of PWM channels break function 0: Disable 4: Break pin (BRK) Other: Reserved

9 WATCHDOG TIMER (WDT)

9.1 OVERVIEW

The purpose of the Watchdog is to prevent the MCU from entering an erroneous state. When enabled, the WDT will generate a system reset or interrupt if the user program fails to “feed” (or reload) the Watchdog within predetermined amount of time.

Under the normal operation, users will restart WDT at the regular intervals before counter counts down to 0. If the counter does reach 0, WDT will generate one or a combination of signals, system reset, and system interrupt to reset the system, or interrupt the system correspondingly.

When reset, the WDT registers can reset the values. After the programmer sets WDTEN bit in the WDT_CTRL register, the WDT counter starts to reduce the counting. If the WDT timer count reaches 0, the reset signal, or interrupt signal will be triggered. As long as the signal is set in the WDT_INTRLEN register, the assertion of this signal will depend on the RSTEN and IE bits in the WDT_CTRL register, the signal will keep asserted for a period of time. The WDT_STATUS register is used to check if the counter reaches 0 or not.

To prevent the unexpected reset, the programmer needs to write 0x5AB9 to the WDT_RESTART register as the password to activate the down counting function. The WDT_LOAD register will be loaded into the counter.

The watchdog timer block use EXTCLK. The PCLK is used for the APB0 accesses to the watchdog register. The EXTCLK is used for the watchdog timer counting. ILRC is the clock source for EXTCLK.

The Watchdog should be used in the following manner:
1. Disable the Watchdog timer.
2. Set Load register.
3. Write 0x5AB9 to the WDT_RESTART register.
4. Set reset (RSTEN) or interrupt (IE) function.
5. Enable the Watchdog timer.
6. The watchdog timer should be fed again by writing 0x5AB9 is written in WDT_RESTART register before the Watchdog counter underflows to prevent reset or interrupt.

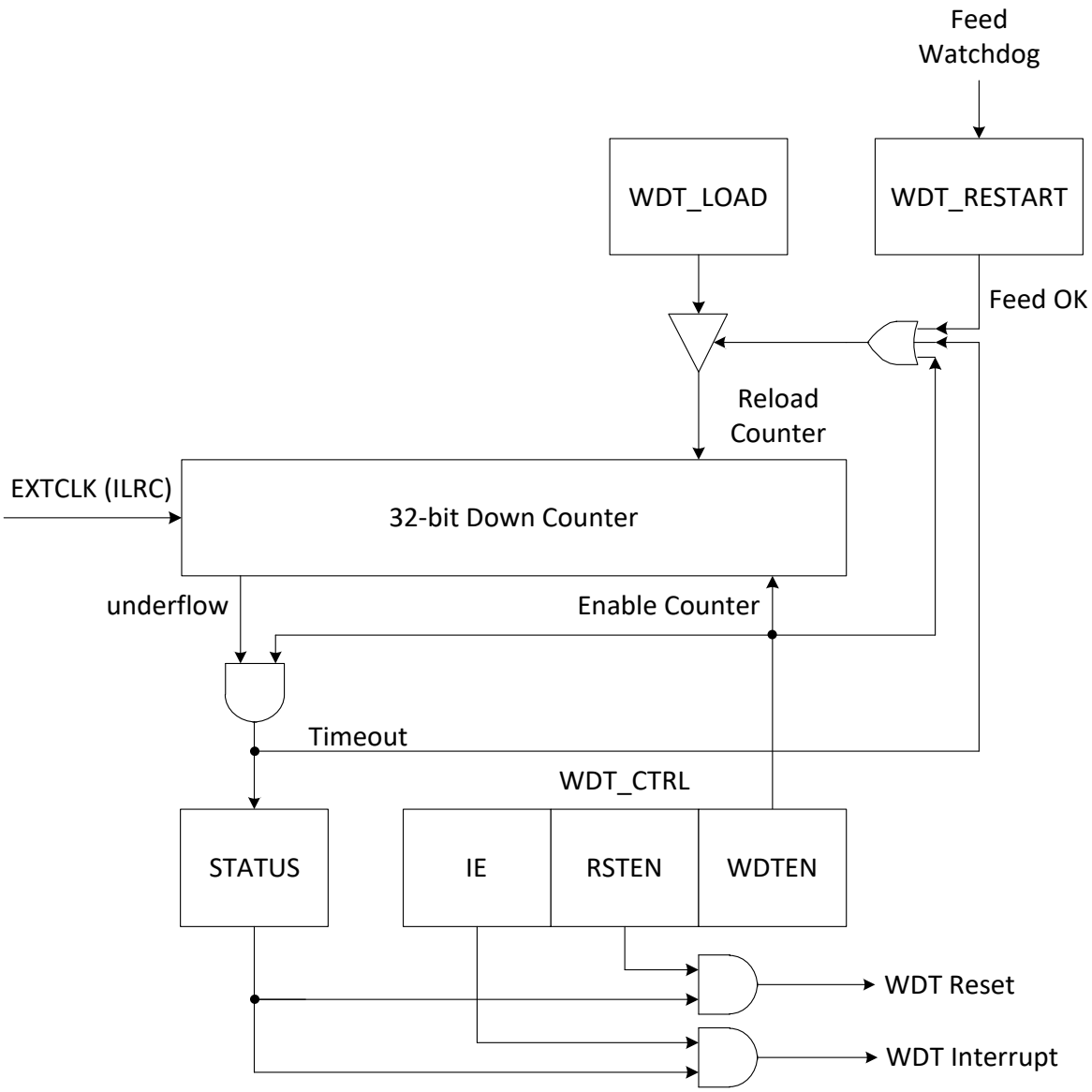
The watchdog timer clock is enabled by WDTCLKEN bit of SCU_APB0CLKG register. Enabled by WDTCLKEN bit of SCU_SLP_APB0CLKG in sleep mode.

✷ ***Note: there is a limitation between EXTCLK and PCLK, that is, the ratio of EXTCLK cycle time/PCLK cycle time ≧ 3.***

9.2 FEATURES

- 32-bit down counter
- External clock (ILRC 32k) source
- A variable timeout period
- During timeout, outputs system reset or system interrupt

9.3 BLOCK DIAGRAM



9.4 WDT REGISTERS

Base Address: 0x4000 B000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	WDT_COUNTER	WDT Counter register
0x0004	WDT_LOAD	WDT Load register
0x0008	WDT_RESTART	WDT Restart register
0x000C	WDT_CTRL	WDT Control register
0x0010	WDT_STATUS	WDT Status register
0x0014	WDT_CLEAR	WDT Clear register
0x0018	WDT_INTRLEN	WDT Duration register

9.4.1 Watchdog Counter register (WDT_COUNT)

Address Offset: 0x00

WDT_COUNT register contains the current counter value. When it is reset, the value will be set to 0x3EF1480. After writing 0x5AB9 to WDT_RESTART register, the value of WDT_LOAD register will be loaded into WDT_COUNT register. WDT_COUNT will start to decrease once the WDT_EN bit is set. If the Watchdog timer is disabled, WDT_COUNT will hold the value.

Bit	Field	Access	Initial	Description
31:0	COUNTER	R	0x3EF1480	Current counter value

9.4.2 Watchdog Load register (WDT_LOAD)

Address Offset: 0x04

Load contains the value which will be loaded into WDT_COUNT register. When it is reset or restarted, the value will be automatically loaded into the WDT_COUNT register. The reset value of WDT_LOAD is 0x3EF1480.

Bit	Field	Access	Initial	Description
31:0	LOAD	RW	0x3EF1480	The WDT reload value

9.4.3 Watchdog Restart register (WDT_RESTART)

Address Offset: 0x08

Bit	Field	Access	Initial	Description
31:0	RESTART	W	0	WDT restart 0: Automatically reset to 0 after finishing the restart cycle 0x5AB9: Restart WDT and the WDT counter will restart to decrease.

9.4.4 Watchdog Control register (WDT_CTRL)

Address Offset: 0x0C

Bit	Field	Access	Initial	Description
31:3	–	–	0	Reserved
2	IE	RW	0	WDT system interrupt enable bit 0: Disable 1: Enable
1	RSTEN	RW	0	WDT system reset enable bit 0: Disable 1: Enable
0	WDTEN	RW	0	WDT enable bit 0: Disable 1: Enable

9.4.5 Watchdog Status register (WDT_STATUS)

Address Offset: 0x10

Bit	Field	Access	Initial	Description
31:1	–	–	0	Reserved
0	STATUS	R	0	WDT counting status 0: WDT counter does not reach 0 yet 1: WDT counter reaches 0

9.4.6 Watchdog Clear register (WDT_CLEAR)

Address Offset: 0x14

Bit	Field	Access	Initial	Description
31:1	–	–	0	Reserved
0	CLEAR	W1C	0	WDT status clear bit Writing ‘1’ or ‘0’ to clear WDT_STATUS register.

9.4.7 Watchdog Clear register (WDT_INTRLEN)

Address Offset: 0x18

The duration of the assertion of WDT reset and interrupt. The default value is 0xFF, which means that the default assertion durations of WDT reset and interrupt are 256 clock cycles (PCLK).

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	LEN	RW	0xFF	The duration is LEN * WDT_PCLK cycle

10 WINDOW WATCHDOG TIMER (WWDT)

10.1 OVERVIEW

The purpose of the Watchdog is to prevent the MCU from entering an erroneous state. When enabled, the WWDT will generate a system reset or interrupt if the user program fails to “feed” (or reload) the Watchdog within predetermined amount of time.

Under the normal operation, users will restart WWDT at the regular intervals before counter counts down to 0. If the counter does reach 0, WWDT will generate one or a combination of signals, system reset, and system interrupt to reset the system, or interrupt the system correspondingly.

When reset, the WWDT registers can reset the values. When the window watchdog is started by writing the START_KEY code in the Key register (WWDT_KEY), the counter starts counting down from the value of the reload register.

When it reaches the end of count value (0x0000), a reset signal is generated reset (or interrupt). Whenever the KICKDOG_KEY or START_KEY is written in the WWDT_KEY register, the reload register (WWDT_RELOAD) value is reloaded in the counter, and the watchdog reset is prevented.

Write accesses to the prescaler register (WWDT_PRESCALER) and WWDT_RELOAD registers are protected. To modify them, users must first write the REPROG_KEY code in the WWDT_KEY register and please check lock status register (WWDT_LOCKST) to make sure that WWDT is at the unlock state before executing the next APB command. A write access to the WWDT_KEY register is register with a different value will break the sequence, and register access will be protected again. This implies that it is the case of the reload operation by writing KICKDOG_KEY. A lock status register is available to indicate that an update of the prescaler or the down-counter reload value is ongoing.

The window watchdog provides an underflow function. When the underflow function is enabled, it can only be reloaded when the delta count is greater than the underflow value, otherwise an underflow reset will be triggered. The underflow function can be used to prevent system errors that cause the WWDT to reload prematurely.

The watchdog timer block use EXTCLK. The PCLK is used for the APB0 accesses to the watchdog register. The EXTCLK is used for the watchdog timer counting. ILRC or APB0CLK is the clock source for EXTCLK. The EXTCLK is selected by WWDTCLKSEL bit of SCU_CLKSEL register.

The watchdog timer clock is enabled by WWDTCLKEN bit of SCU_APB0CLKG register. Enabled by WWDTCLKEN bit of SCU_SLP_APB0CLKG in sleep mode.

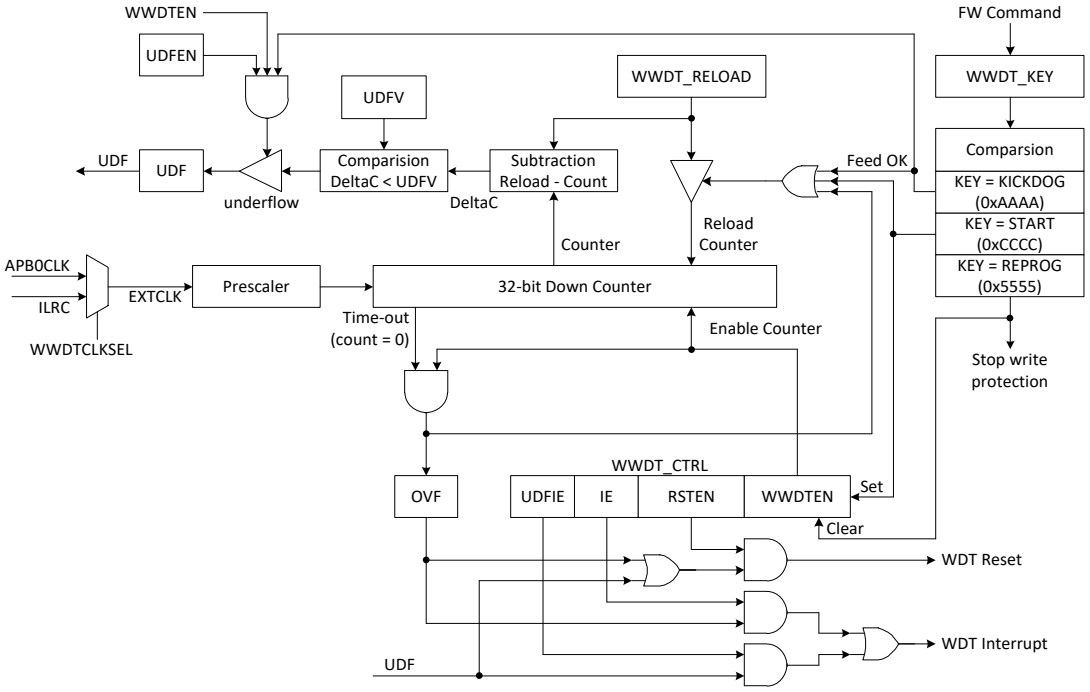
The window watchdog is an independent watchdog that cannot be easily turned off after it has been started. It prevents the watchdog is turned off during a system error. If you need to disable the watchdog, you can disable the watchdog by writing REPROG_KEY to WWDT_KEY register.

✷ ***Note: there is a limitation between EXTCLK and PCLK, that is, the ratio of EXTCLK cycle time/PCLK cycle time ≧ 3.***

10.2 FEATURES

- On timeout, watchdog outputs one or a combination of the following signals:
  - System reset
  - System interrupt
- 32-bit down counter
- External (APB0CLK or ILRC) clock source
- A variable time-out period of reset (if watchdog activated) when the down counter value of 0x0 is reached
- Underflow function for window-based watchdog
- Access protection

10.3 BLOCK DIAGRAM



10.4 PROGRAMMING SEQUENCE

The Watchdog should be used in the following manner:

1. Enable the Watchdog clock with WWDTCLKEN bit.
2. Reset the watchdog circuit with WWDTRST bit.
3. Select clock with WWDTCLKSEL bit.
4. Write REPROG_KEY to WWDT_KEY and check lock status register.
5. Set Prescaler and Reload register.
6. Set reset (RSTEN) or interrupt (IE) function.
7. (Option) Set underflow function with UDFEN and UDFIE. Set underflow value for underflow reset zone.
8. Writer START_KEY to WWDT_KEY register to enable the watchdog timer and reload counter.
9. Before the watchdog timer time-out or underflows, the watchdog timer should be fed by writing KICKDOG_KEY to the WWDT_KEY register to prevent reset or interrupt.

✸ ***Note: Before starting the WDT, there are 3 steps that must be followed, otherwise the WDT may operate abnormally.***

1. ***Set WWDTCLKEN bit of APB0CLKG register.***
2. ***Set WWDTRST bit of APB0RST register.***
3. ***Select clock source by WWDTCLKSEL bit of CLKSEL register.***

10.5 RESET ZONE

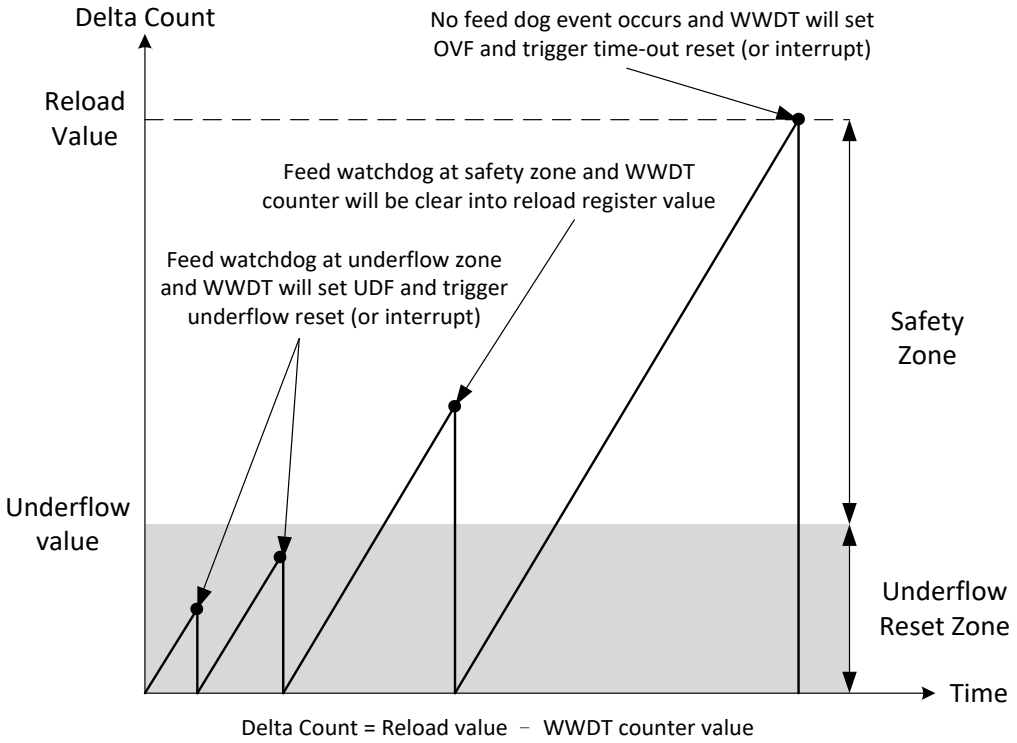
If the WWDT counter does reach 0, WWDT will set OVF and generate a system reset (or time-out interrupt)

When the underflow function is enabled, it can only be reloaded when the delta count is greater than the underflow value, otherwise an underflow reset will be triggered (or UDF interrupt).

Delta Count = Reload value – WWDT counter value

Delta Count	Event	UDFEN	UDFIE	IE	RSTEN	Result
Delta Count < Underflow Value	No Feed	x	x	x	x	1. No event occurs 2. WWDT keep run
	Feed	0	x	x	x	1. No event occurs 2. WWDT reload
		1	0	x	0	1. Set UDF 2. WWDT reload
		1	1	x	0	1. Set UDF and trigger underflow interrupt 2. WWDT reload
		1	x	x	1	1. Set UDF and trigger underflow reset 2. System reset
Underflow Value < Delta Count < Reload Value	No Feed	x	x	x	x	1. No event occurs 2. WWDT keep run
	Feed	x	x	x	x	1. No event occurs 2. WWDT reload
Delta Count = Reload Value (WWDT counter = 0)	Time-out	x	x	0	0	1. Set OVF 2. WWDT reload
		x	x	1	0	1. Set OVF and trigger timeout interrupt 2. WWDT reload
		x	x	x	1	1. Set OVF and trigger timeout reset 2. System reset

x: don't care



10.6 WWDT REGISTERS

Base Address: 0x4000 C000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	WWDT_KEY	WWDT Key register
0x0004	WWDT_PRESCALER	WWDT Prescaler register
0x0008	WWDT_RELOAD	WWDT Reload register
0x000C	WWDT_LOCKST	WWDT Lock Status register
0x0010	WWDT_RIS	WWDT Interrupt Status register
0x0014	WWDT_CTRL	WWDT Control register
0x0018	WWDT_INTL	WWDT Interrupter Length register
0x001C	WWDT_UDFV	WWDT Underflow Value register
0x0020 – 0x00FC	–	Reserved
0x0100	WWDT_STATUS	WWDT Status register

10.6.1 Window Watchdog Key register (WWDT_KEY)

Address Offset: 0x00

These bits must be written by software at regular intervals with the key value 0xAAAA (KICKDOG_KEY); otherwise, the WWDT generates a reset when the counter reaches 0.

Please note that after writing 0x5555 (REPROG_KEY), users must check whether the WWDT_LOCKST status is at the unlock state or not first. If WWDT is not at the prescaler lock or reload lock state, users can modify the WWDT_PRESCALER and WWDT_RELOAD value.

Write Protected Register

Protected Register	Protected Bit
WWDT_PRESCALER register	PR[2:0]
WWDT_RELOAD register	RL[31:0]
WWDT_CTRL register	UDFEN
	CLOCKSRC
	RSTEN
WWDT_INTL register	LEN[7:0]
WWDT_UDFV register	UDF[31:0]

✸ ***Note: Since REPROG_KEY will stop the down counter, write START_KEY after configuration to start the down counter.***

Bit	Field	Access	Initial	Description
31:16	–	–	0xFFFF	Reserved
15:0	KEY	W	0xFFFF	Write operation Key value 0x5555: REPROG_KEY. Stop down counter, disable the watchdog time and unlock the programming function for protected registers. (e.g. Prescaler and reload register)

Bit	Field	Access	Initial	Description
				0xAAAA: KICKDOG_KEY. Reload the RL value to the WWDT counter. After writing KICKDOG_KEY or START_KEY, WWDT will get into reload-lock status. 0xCCCC: START_KEY. Start the WWDT. Enable the watchdog timer and reload counter. Other value: After WWDT configuring, writing different value to disable access to the protected registers. In reload-lock status the RELOAD_LOCK bit will set, don't write KICKDOG_KEY or START_KEY before RELOAD_LOCK bit turn to 0. Read operation Current down counter value. If reading these bits, it represents the value of current down counter.

10.6.2 Window Watchdog Prescaler register (WWDT_PRESCALER)

Address Offset: 0x04

The PRESCALER_LOCK bit in WWDT_LOCKST register must be reset in order to be able to change the prescaler divider.

Bit	Field	Access	Initial	Description
31:3	–	–	0	Reserved
2:0	PR	RW	0	Prescaler divider 0: WWDT prescaler is divider/4 1: WWDT prescaler is divider/8 2: WWDT prescaler is divider/16 3: WWDT prescaler is divider/32 4: WWDT prescaler is divider/64 5: WWDT prescaler is divider/128 6: WWDT prescaler is divider/256 7: WWDT prescaler is divider/256 These bits are write access protected.

10.6.3 Window Watchdog Reload register (WWDT_RELOAD)

Address Offset: 0x08

These bits are written access protected. They are written by software to define the value to be loaded in the WWDT counter each time when the value, KICKDOG_KEY, is written in the WWDT_KEY register.

The WWDT counter counts down from this value. The timeout period is a function of this value and the clock prescaler.

The RELOAD_LOCK bit in WWDT_LOCKST register must be reset in order to be able to change the reload value.

Bit	Field	Access	Initial	Description
31:0	RL	RW	0xFFFFFFFF	WWDT counter reload value. These bits are written access protected.

10.6.4 Window Watchdog Lock Status register (WWDT_LOCKST)

Address Offset: 0x0C

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1	PRESCALER_LOCK	R	0	WWDT prescaler value update 0: No update of the prescaler value is ongoing 1: An update of the prescaler value is ongoing

Bit	Field	Access	Initial	Description
0	RELOAD_LOCK	R	0	WWDT counter reload value update 0: No update of the reload value is ongoing 1: An update of the reload value is ongoing

10.6.5 Window Watchdog Interrupt Status register (WWDT_RIS)

Address Offset: 0x10

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1	UDF	RW1C	0	Whether WWDT enters underflow state or not 0: Cleared 1: WWDT timer enters underflow state
0	OVF	RW1C	0	Whether WWDT counter reaches 0 or not 0: Cleared 1: WWDT counter reaches 0

10.6.6 Window Watchdog Control register (WWDT_CTRL)

Address Offset: 0x14

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	UDFEN	RW	0	WWDT underflow function enable bit 0: Disable 1: Enable The bit is written access protected.
6	UDFIE	RW	0	WWDT UDF interrupt enable bit 0: Disable 1: Enable
5:3	–	–	0	Reserved
2	RSTEN	RW	0	WWDT reset enable bit 0: Disable 1: Enable The bit is written access protected.
1	IE	RW	0	WWDT timeout (OVF) interrupt enable bit 0: Disable 1: Enable
0	WWDTEN	R	0	WWDT enable bit 0: Disable (After writing REPROG_KEY, WWDTEN will be cleared.) 1: Enable (After writing START_KEY, WWDTEN will be set.)

10.6.7 Window Watchdog Interrupt Length register (WWDT_INTL)

Address Offset: 0x18

The default value is 0xFF, which means that the default assertion durations of WWDT reset and WWDT interrupt are 256 clock cycles.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	LEN	RW	0xFF	The duration of the assertion of reset and interrupt. These bits are written access protected.

10.6.8 Window Watchdog Underflow Value register (WWDT_UDFV)

Address Offset: 0x1C

Bit	Field	Access	Initial	Description
31:0	UDF	RW	0	WWDT counter underflow value △Count = Reload value – counter value △Count > UDF value: WWDT keeps counting down △Count < UDF value: WWDT triggers underflow reset These bits are written access protected.

10.6.9 Window Watchdog Status register (WWDT_STATUS)

Address Offset: 0x100

Bit	Field	Access	Initial	Description
31:25	–	–	0	Reserved
24	OVFST	R	0	Overflow status 0: No overflow 1: Overflow occurs
23:17	–	–	0	Reserved
16	UDFST	R	0	Underflow status 0: No underflow 1: Underflow occurs
15:1	–	–	0	Reserved
0	IS	R	0	Interrupt status 0: No interrupt 1: Interrupt occurs

11 REAL-TIME CLOCK (RTC)

11.1 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.

RTC provides separate second, minute, hour, day, weekday, month, year and century counters. The second counter is toggled each second, the minute counter is toggled each minute, the hour counter is toggled each hour, the day counter is toggled each day, the weekday counter is toggled each day, the month counter is toggled each month, the year counter is toggled each year, and the century counter is toggled each century.

The RTC provides a programmable auto-alarm function. When the second auto-alarm function is turned on, the RTC will automatically trigger an interrupt each second. The automatic minute, hour, day and month alarms can be turned on as well. This function is useful for implementing a clock.

Real Time Clock (RTC) accepts two clock sources: ILRC clock (32kHz) and ELS clock (32.768 kHz clock). When the system enters the power down mode, the RTC will keep on counting. This mechanism promises the lowest power consumption when the system is asleep.

11.2 FEATURES

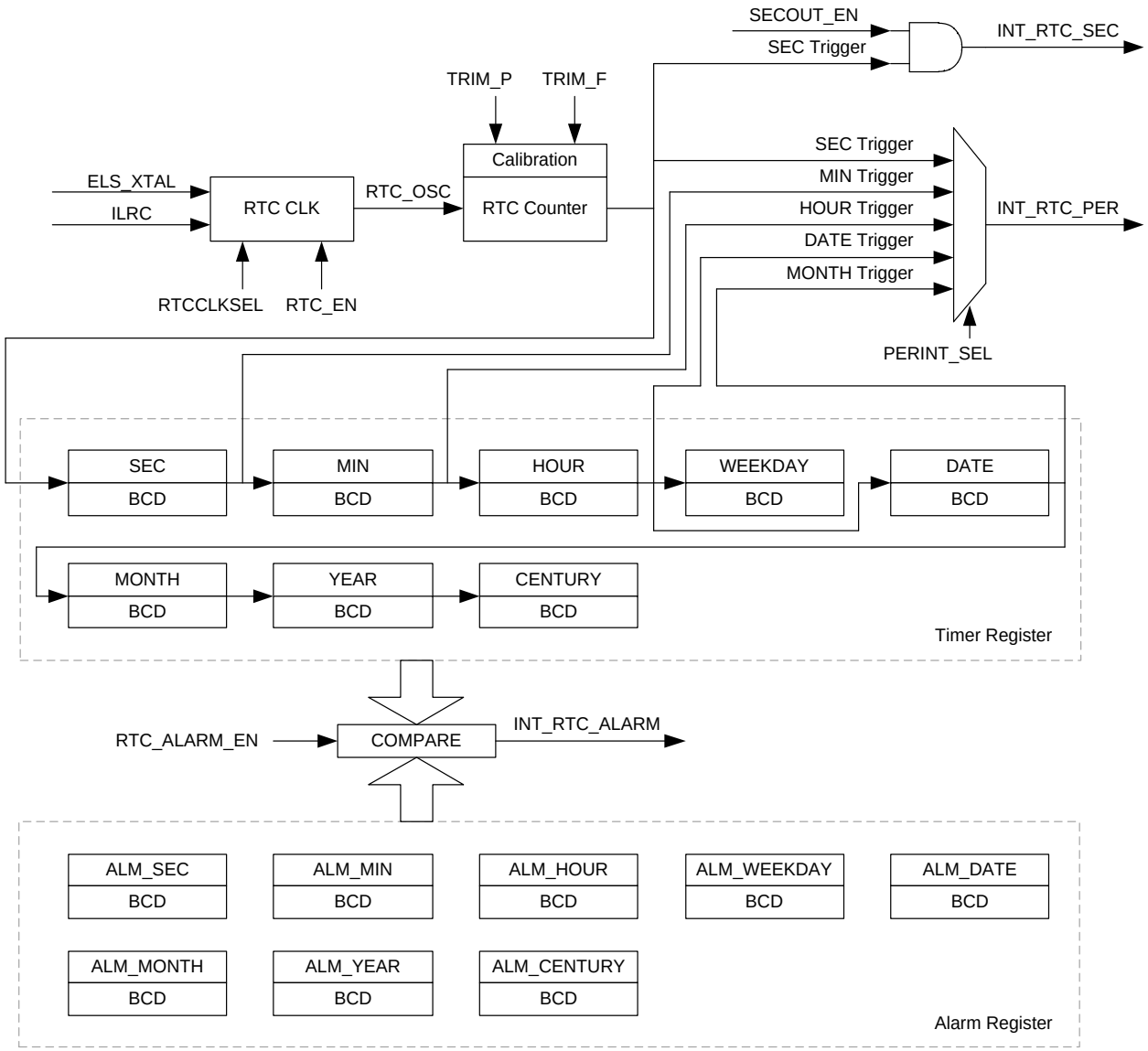
- Provides separate second, minute, hour, day, weekday, month, year and century counters.
- Support alarm time: second, minute, hour, day, weekday and month.
- Support periodic alarm: second, minute, hour, day, month alarms
- Support second interrupt
- Support tick trim for calibration
- Support BCD type counter and alarm
- Provides calendar function
- The RTC clock source could be any of the following:
  – ELS X'TAL
  – ILRC
- Reset sources of the RTC: POR

11.3 CLOCK

Setting			RTC Circuit	RTC clock source
RTCCLKEN[1]	**RTC_EN**[3]	**RTCCLKSEL**[2]		
0	X	X	OFF	-
1	0	X		-
	1	0	ON	ELS X'TAL
		1		ILRC

[1] RTC module clock enable by RTCCLKEN bit of ALWAYSON_POR_MISC register.
[2] RTC clock source select by RTCCLKSEL bit of ALWAYSON_POR_MISC register.
[3] RTC function enable by RTC_EN bit of SCU_RTC_CTRL register.

11.4 BLOCK DIAGRAM



✸ ***Note: The first RTC period is only half a second and is recommended to be ignored when using.***

11.5 FUNCTIONAL DESCRIPTION

11.5.1 Read Current Time

These time registers are updated every second. In the APB protocol, it only simultaneously issues a single register read and executes at an arbitrary, so it does not guarantee the accurate time reading, because the second register is possible to be changed after the first register reading. Three possible methods are recommended for reliable reading of the time registers:

1. Read after interrupt
   The INT_RTC_SEC interrupt indicates that the second counter has just been incremented, and the RTC registers will not change before the next second. A register read will immediately be executed after issuing an INT_RTC_SEC interrupt; therefore, it is considered as an accurate time reading.

2. Two consecutive reads
   If two consecutive reads within a short time (Less than 1 second apart) returns the same result, this is considered as the accurate reading. If the two results are different, the procedure should be repeated.

3. Using the time lock function
   Users can lock the current time before reading the register. To enable bit 2 of the RTC control register (RTC_CTRL), the current time should be locked for an accurate reading of the time1 register and can be released by one of the following three conditions:
   - Time2 register is read.
   - Time1 or Time2 register is written.
   - RTC is disabled.

11.5.2 Set New Time

If users want to set new RTC time, the current time register has to be immediately written and then the RTC_EN bit of the RTC control register should be set. When the setting values are reliably written to the RTC counter, the RTCEN_STS bit will be set and the RTC counter will start running. Before writing the RTC time register, RTC should be stopped to prevent glitches. The procedure is as follows:

1. Clear the RTC_EN bit of the RTC control register to stop the RTC counter
2. Check and read the RTCEN_STS bit of the RTC control register to confirm if RTC is disabled.
3. Set new time to registers Time1 and Time2
4. Set the RTC_EN bit and wait until the RTCEN_STS bit is changed to ‘1’ and the RTC counter will be re-enabled.

In the time setting procedure, the written values of the current time register will guarantee a valid time for each field by the software programming. For example, the content in second and minute must be between 0 and 59.

11.5.3 Set The Alarm

For setting the alarm, the procedure is similar to that of being used when setting the RTC current time. Users should disable the alarm, clear the ALARM_EN bit of the RTC control register, and prevent the accidental alarm event from being triggered:

1. Clear the ALARM_EN bit of the RTC control register to disable the RTC alarm
2. Check and read the RTC_ALM_STS bit of the RTC control register to confirm if RTC alarm is disabled.
3. Set new time to registers, Alarm Time1 and Alarm Time2
4. Set the ALARM_EN bit and wait until the RTC_ALM_STS bit is changed to ‘1’ and the setting of the RTC alarm is complete.

Users can individually disable the time match checking in each field. For example, ALM_WEEKDAY is set to ‘7’, and the alarm will not check if the field matches the current time.

11.5.4 RTC Configuration Procedure

11.5.4.1 RTC Enable Steps:

Step1. Check Always-on domain registers enable
- Check ALWAYSONREGEN = 1 (SCU_APB0CLKG bit 10)

Step2. Set RTC clock source
- If ELS 32K is enabled (ELSEN = 1), check ELSRDY = 1 (ALWAYSON_POR_MISC bit4)
- Select clock source by RTCCLKSEL (ALWAYSON_POR_MISC bit1)
- Enable RTC clock by RTCCLKEN = 1 (ALWAYSON_POR_MISC bit 0)
- Wait for RTC registers ready by RTC_REGRDY = 1 (SCU_RTC_CTRL bit15)

Step3. Set RTC Timer and execute RTC
- Set SCU_RTC_TIME1/ SCU_RTC_TIME2/ SCU_RTC_ALM1/ SCU_RTC_ALM2
- Enable RTC alarm by RTC_ALARM_EN = 1 (SCU_RTC_CTRL bit1)
- Wait for RTC alarm enable status by RTC_ALMEN_STS = 1 (SCU_RTC_CTRL bit9)
- Enable RTC by RTC_EN = 1 (SCU_RTC_CTRL bit0)
- Wait for RTC enable status by RTCEN_STS = 1 (SCU_RTC_CTRL bit 8)
- Wait for RTC alarm interrupt status by INT_RTC_ALARM= 1 (SCU_RIS bit16)

11.5.4.2 RTC Disable Steps:

Step1. Disable RTC & RTC alarm
- Disable RTC alarm by RTC_ALARM_EN = 0 (SCU_RTC_CTRL bit1)
- Wait RTC alarm enable status by RTC_ALMEN_STS = 0 (SCU_RTC_CTRL bit9)
- Disable RTC by RTC_EN = 0 (SCU_RTC_CTRL bit0)
- Wait for RTC enable status by RTCEN_STS = 0 (SCU_RTC_CTRL bit 8)

Step2. Disable RTC clock
- Select clock source to ILRC by RTCCLKSEL = 1 (ALWAYSON_POR_MISC bit1)
- Disable RTC clock by RTCCLKEN = 0 (ALWAYSON_POR_MISC bit 0)

11.5.5 Clock Calibration

RTC includes a tick generation block to provide precise tick for counting the time in each second. The tick can be trimmed by the RTC tick trim control register to adjust the counting value of the tick generation block and compensate the frequency drift caused by the crystal inaccuracies. The tick trim is fractional divider, the TRIM_P field is the fixed point part, and the TRIM_F field is the floating point part. The tick frequency equation is shown below.

Tick = T_{RTC_OSC}/(P + (F/1000)) KHz where, P is TRIM_P and F is TRIM_F.

T_{RTC_OSC} is the period of RTC clock of RTC, and it is usually from an external clock source of 32.768-KHz crystal. The tick should be tuned to 100 Hz.

The counting value TRIM_P and TRIM_F, the TRIM_P field is the fixed point part, and the TRIM_F field is the floating point part.

The tick frequency equation is shown below. For example, external crystal oscillator is 32.768 KHz, then, TRIM_P is 327 and TRIM_F is 680.

Tick = 32.768KHz/(P+(F/1000)) Hz where, P is TRIM_P and F is TRIM_F.

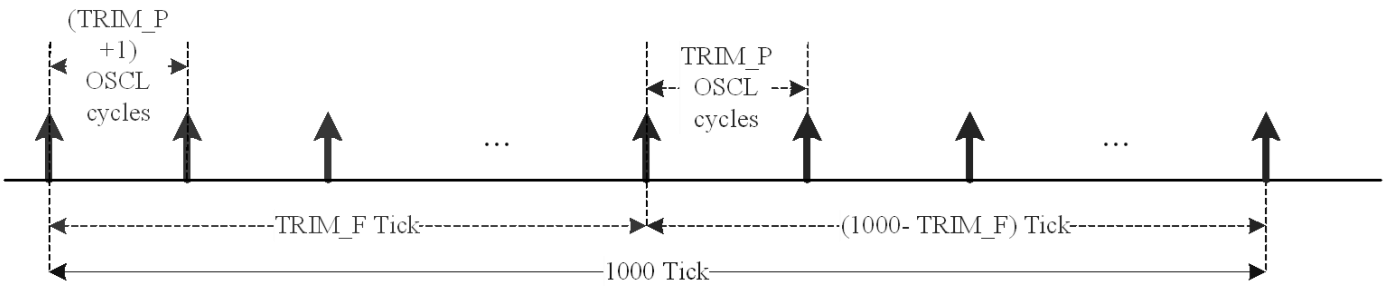
Tick fractional divider count 1000 ticks to achieve trimming function. There are two type tick period.

- Trimmed tick period: counting TRIM_P+1 external crystal oscillator cycles
- Un-trimmed tick period: TRIM _P external crystal oscillator cycles

Tick fractional divider will count F trimmed tick period and (1000- TRIM_F) un-trimmed tick period.

So, total cycles is TRIM_F x (TRIM_P+1) + (1000- TRIM_F) x TRIM_P =1000 x TRIM_P+ TRIM_F

If external crystal oscillator frequency is 32.768 KHz, the total cycle is 327680 cycles and trimming period is 10sec.



11.5.5.1 Calibration flow

1. Set P to 327 and F to 680.
2. Measure 1000 Tick time. Measure the time between the first RTC second interrupt and the 11th second interrupt.
3. Calculate the total count number of TRIM F ticks. (P+1)*F = (327+1)*680 = 223040
4. Calculate the total count number of 1000 - TRIM F ticks. P*(1000-F) = 327*(1000-680) = 104640
5. Sum two count values. 223040 + 104640 = 327680
6. Divide the total count value by 1000 tick time to get the calibration value.
7. The calibration value = New Trim P*100 + New Trim F/10.
8. Update P and F values. Recheck whether the 1000 Tick time meets 10 seconds.

11.5.5.2 Example

- If ELS clock is 32765Hz

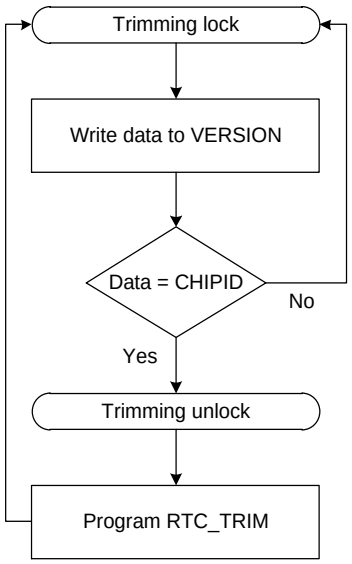
Action	ELS(Hz)	P	F	(P+1)*F	P*(1000-F)	1000 Tick (S)	Cal. Value
Calibration	32765	327	680	223040	104640	10.00092	32764.985
Update trim value	32765	327	650	213200	114450	10	-

- If ELS clock is 32771Hz

Action	ELS(Hz)	P	F	(P+1)*F	P*(1000-F)	1000 Tick (S)	Cal. Value
Calibration	32771	327	680	223040	104640	9.99908	32771.014
Update trim value	32771	327	710	232880	94830	10	-

11.5.6 Trimming Register Protection

The RTC trimming register does not allow to arbitrarily changing, it is protected to avoid the unexpected programming. The un-lock procedure is shown in the figure below. The unlock RTC ID is the same SCU_CHIPID, and writes the value to the RTC version Register (SCU_VERSION), it will unlock the write protected and allow once write operation to the RTC clock trimming Register (SCU_RTC_TRIM), the register will resume the lock state after the write operation. When SCU_CHIPID is programmed, the unlock key will also be updated to new CHIPID. Users should use new CHIPID to unlock the protected register in the next write procedure.



11.6 RTC INTERRUPT REGISTERS

Base Address: 0x4001 F000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0024	SCU_RIS	SCU Raw Interrupt Status register
0x0028	SCU_IE	SCU Interrupt Enable register

11.6.1 SCU Raw Interrupt Status register (SCU_RIS)

Address Offset: 0x24

Bit	Field	Access	Initial	Description
31:19	–	–	0	Reserved
18	INT_RTC_SEC	RW1C	0	RTC second out interrupt status 0: No RTC second out occurred 1: RTC second out occurred
17	INT_RTC_PER	RW1C	0	RTC periodic interrupt status 0: No RTC periodic occurred 1: RTC periodic occurred
16	INT_RTC_ALARM	RW1C	0	RTC alarm interrupt status 0: No RTC alarm occurred 1: RTC alarm occurred
15:12	–	–	0x0	Reserved, don't modify it
11	INT_REMAPCHG	RW1C	0	Remap is changed status after reboot command 0: Not change 1: REMAP flag is changed
10:7	–	–	0	Reserved, don't modify it
6	INT_FCS	RW1C	0	FCS command finish interrupt status 0: Not finish 1: FCS command finished
5:4	–	–	0	Reserved
3	INT_WAKEUP	RW1C	0	Sleep mode wakeup interrupt status 0: No wakeup from sleep mode 1: Wakeup from sleep mode
2	INT_DS_WAKEUP	RW1C	0	Deep sleep mode wakeup interrupt status 0: No wakeup from deep sleep mode 1: Wakeup from deep sleep mode
1:0	–	–	0	Reserved

11.6.2 SCU Interrupt Enable register (SCU_IE)

Address Offset: 0x28

Bit	Field	Access	Initial	Description
31:19	–	–	0	Reserved
18	RTC_SEC_EINT	RW	0	RTC second interrupt enable bit 0: Disable 1: Enable
17	RTC_PER_EINT	RW	0	RTC periodic interrupt enable bit 0: Disable 1: Enable
16	RTC_ALARM_EINT	RW	0	RTC alarm interrupt enable bit 0: Disable 1: Enable
15:12	–	–	0	Reserved

Bit	Field	Access	Initial	Description
11	REMAPCHG_EINT	RW	0	REMAP change interrupt enable 0: Disable REMAP change interrupt 1: Enable REMAP change interrupt
10:7	–	–	0x0	Reserved
6	FCS_EINT	RW	0	FCS command finish interrupt enable bit 0: Disable 1: Enable
5:4	–	–	0x0	Reserved
3	WAKEUP_EINT	RW	0	Sleep mode Wake-up event interrupt enable bit 0: Disable 1: Enable
2	DS_WAKEUP_EINT	RW	0	Deep sleep mode Wake-up event interrupt enable bit 0: Disable 1: Enable
1:0	–	–	0x0	Reserved

11.7 RTC CLOCK REGISTER

Base Address: 0x4001 B000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0110	ALWAYSON_POR_MISC	POR Miscellaneous Control register

11.7.1 POR Miscellaneous Control register (ALWAYSON_POR_MISC)

Address Offset: 0x110

Bit	Field	Access	Initial	Description
31:5	–	–	0	Reserved
16	PORRSTF	RW1C	1	POR reset flag 0: No POR occurred 1: POR occurred
15:5	–	–	0	Reserved
4	ELSRDY	R	0	ELS XTAL ready flag 0: ELS Xtal is Not Ready 1: ELS Xtal is Ready
3	ELSFLOFF	RW	1	External low-speed oscillator clock filter control 0: Enable clock filter. ELS with clock filter 1: Disable clock filter. ELS without clock filter ELSFLOFF can only be switched when ELSEN = 0, and cannot be switched at the same time as ELSEN.
2	ELSEN	RW	0	External low-speed oscillator enable 0: Disable External 32.768 KHz oscillator 1: Enable External 32.768 KHz oscillator
1	RTCCLKSEL	RW	0	RTC counter clock source 0: RTCCLK=ELS 1: RTCCLK=ILRC 32kHz
0	RTCCLKEN	RW	0	Enable clock for RTC 0: Disable 1: Enable

✸ ***Note: SW shall disable RTC (RTC_EN=0) when changing the value of RTCCLKSEL.***

11.8 RTC FUNCTION REGISTERS

Base Address: 0x4001 F000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0200	SCU_RTC_TIME1	SCU RTC Timer register 1
0x0204	SCU_RTC_TIME2	SCU RTC Timer register 2
0x0208	SCU_RTC_ALM1	SCU RTC Alarm Time register 1
0x020C	SCU_RTC_ALM2	SCU RTC Alarm Time register 2
0x0210	SCU_RTC_CTRL	SCU RTC Control register
0x0214	SCU_RTC_TRIM	SCU RTC Tick Trim Control register
0x0218 – 0x021C	–	Reserved
0x0220	SCU_BCD_RTC_TIME1	SCU RTC Timer for BCD register 1
0x0224	SCU_BCD_RTC_TIME2	SCU RTC Timer for BCD register 2
0x0228	SCU_BCD_RTC_ALM1	SCU RTC Alarm Time for BCD register 1
0x022C	SCU_BCD_RTC_ALM2	SCU RTC Alarm Time for BCD register 2

11.8.1 SCU RTC Timer register 1 (SCU_RTC_TIME1)

Address Offset: 0x200

✸ ***Note: If the time parameter is written to a value that exceeds the range, the time parameter will overflow.***

Bit	Field	Access	Initial	Description
31:27	–	–	0	Reserved
26:24	WEEKDAY	RW	0	Current weekday 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday 7: Reserved
23:21	–	–	0	Reserved
20:16	HOUR	RW	0	Current hour Valid value is ranging from 0 to 23
15:14	–	–	0	Reserved
13:8	MIN	RW	0	Current minute Valid value is ranging from 0 to 59
7:6	–	–	0	Reserved
5:0	SEC	RW	0	Current second Valid value is ranging from 0 to 59

11.8.2 SCU RTC Timer register 2 (SCU_RTC_TIME2)

Address Offset: 0x204

✷ ***Note: If the time parameter is written to a value that exceeds the range, the time parameter will overflow.***

Bit	Field	Access	Initial	Description
31	–	–	0	Reserved
30:24	CENTURY	RW	0x14	Current century Valid value is ranging from 0 to 99
23	–	–	0	Reserved
22:16	YEAR	RW	0xA	Current year Valid value is ranging from 0 to 99
15:12	–	–	0	Reserved
11:8	MONTH	RW	1	Current month 1: January 2: February 3: March 4: April 5: May 6: June 7: July 8: August 9: September 10: October 11: November 12: December Other: Reserved
7:5	–	–	0	Reserved
4:0	DATE	RW	1	Current date Valid value is ranging from 1 to 31

11.8.3 SCU RTC Alarm Timer register 1 (SCU_RTC_ALM1)

Address Offset: 0x208

Bit	Field	Access	Initial	Description
31:27	–	–	0	Reserved
26:24	WEEKDAY	RW	0x7	Alarm weekday 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday 7: Don't care
23:21	–	–	0	Reserved
20:16	HOUR	RW	0x1F	Alarm hour Valid value is ranging from 0 to 23 0x1F: Don't care
15:14	–	–	0	Reserved
13:8	MIN	RW	0x3F	Alarm minute Valid value is ranging from 0 to 59 0x3F: Don't care
7:6	–	–	0	Reserved
5:0	SEC	RW	0x3F	Alarm second

Bit	Field	Access	Initial	Description
				Valid value is ranging from 0 to 59 0x3F: Don't care

11.8.4 SCU RTC Alarm Timer register 2 (SCU_RTC_ALM2)

Address Offset: 0x20C

Bit	Field	Access	Initial	Description
31:12	–	–	0	Reserved
11:8	MONTH	RW	0	Alarm month 1: January 2: February 3: March 4: April 5: May 6: June 7: July 8: August 9: September 10: October 11: November 12: December Other: Reserved
7:5	–	–	0	Reserved
4:0	DATE	RW	0	Alarm date Valid value is ranging from 1 to 31

11.8.5 SCU RTC Control register (SCU_RTC_CTRL)

Address Offset: 0x210

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15	RTC_REGRDY	R	0	RTC registers ready flag. After RTC clock is enabled, user has to wait this flag. When flag is high, RTC registers can be programmed. 0: RTC registers are Not Ready 1: RTC registers are Ready The RTC reset should be confirmed through monitoring this bit, RTC register can be programed after this bit was set.
14:12	–	–	0	Reserved
11	PWUTEN_STS	R	0	Periodic wakeup timer enable status 0: Periodic wakeup timer is disabled 1: Periodic wakeup timer is enabled
10	–	–	0	Reserved
9	RTC_ALMEN_STS	R	0	RTC alarm enable status 0: RTC alarm function is disabled 1: RTC alarm function is enabled
8	RTCEN_STS	R	0	RTC enable status 0: RTC is disabled 1: RTC is enabled
7	SECOUT_EN	RW	0	Enable an event in each second 0: Disable 1: Enable
6:4	PERINT_SEL	RW	0	Periodic interrupt output signal select 0: Disable 3: Each month and triggered at 0 second of every month 4: Each day and triggered at 0 second of every day 5: Each hour and triggered at 0 second of every hour 6: Each minute and triggered at 0 second of every minute 7: Each second Other: Reserved

Bit	Field	Access	Initial	Description
3	–	–	0	Reserved
2	LOCK_EN	RW	0	Lock the Time2 register after reading the Time1 register The lock will be released by the following conditions: 1) Time2 is read. 2) Time1 or Time2 is written. 3) RTC is disabled 0: Disable 1: Enable
1	RTC_ALARM_EN	RW	0	RTC alarm enable bit 0: Disable 1: Enable
0	RTC_EN	RW	0	RTC enable bit 0: Disable 1: Enable

✷ ***Note: RTC_EN bit shall be set at last!***

11.8.6 SCU RTC Tick Trim Control register (SCU_RTC_TRIM)

Address Offset: 0x214

RTC includes a tick generation block to provide precise tick for counting the time in each second. The tick can be trimmed by the RTC tick trim Control Register to adjust the counting value of the tick generation block and compensate the frequency drift caused by the crystal inaccuracies. The tick trim is fractional divider, the TRIM_P field is the fixed point part, and the TRIM_F field is the floating point part.

The RTC trimming register does not allow to arbitrarily changing, it is protected to avoid the unexpected programming. The unlock ID is the same SCU_CHIPID, and writes the value to the version tag Register (SCU_VERSION), it will unlock the write protected and allow once write operation to the RTC clock trimming Register (SCU_RTC_TRIM), the register will resume the lock state after the write operation. When SCU_CHPID is programmed, the unlock key will also be updated to new SCU_CHIPID. Users should use new SCU_CHIPID to unlock the protected register in the next write procedure.

Bit	Field	Access	Initial	Description
31:16	TRIM_P	R/PW	0x147	RTC tick dividing factor
15:0	TRIM_F	R/PW	0x2A8	RTC tick fractional factor Valid between 0 and 999

11.8.7 SCU RTC Timer for BCD register 1 (SCU_BCD_RTC_TIME1)

Address Offset: 0x220

✷ ***Note:***
1. ***If the time parameter is written to a value that exceeds the range, the time parameter will overflow.***
2. ***It is recommended to write the units digit and tens digit at the same time.***
3. ***If you want to write the units digit and tens digit separately, please reset the units digit and tens digit to zero first, and then write a new value to avoid overflow.***

Bit	Field	Access	Initial	Description
31:27	–	–	0	Reserved
26:24	WEEKDAY	RW	0	Current weekday 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday 7: Reserved
23:20	HOUR_DEC	RW	0	Current hour in decimal values Ranging from 0 to 2 The legal programmed value is ranging from 0x0 to 0x23.
19:16	HOUR	RW	0	Current hour in unit values Ranging from 0 to 9 The legal programmed value is ranging from 0x0 to 0x23.
15:12	MIN_DEC	RW	0	Current minute in decimal values Ranging from 0 to 5 The legal programmed value is ranging from 0x0 to 0x59.
11:8	MIN	RW	0	Current minute in unit values Ranging from 0 to 9 The legal programmed value is ranging from 0x0 to 0x59.
7:4	SEC_DEC	RW	0	Current second in decimal values Ranging from 0 to 5 The legal programmed value is ranging from 0x0 to 0x59.
3:0	SEC	RW	0	Current second in unit values Ranging from 0 to 9 The legal programmed value is ranging from 0x0 to 0x59.

11.8.8 SCU RTC Timer for BCD register 2 (SCU_BCD_RTC_TIME2)

Address Offset: 0x224

✷ ***Note:***
1. ***If the time parameter is written to a value that exceeds the range, the time parameter will overflow.***
2. ***It is recommended to write the units digit and tens digit at the same time.***
3. ***If you want to write the units digit and tens digit separately, please reset the units digit and tens digit to zero first, and then write a new value to avoid overflow.***

Bit	Field	Access	Initial	Description
31:28	CENTURY_DEC	RW	0x2	Current century in decimal values Ranging from 0 to 9
27:24	CENTURY	RW	0	Current century in unit values Ranging from 0 to 9
23:20	YEAR_DEC	RW	1	Current year in decimal values Ranging from 0 to 9
19:16	YEAR	RW	0	Current year in unit values Ranging from 0 to 9
15:12	MONTH_DEC	RW	0	Current month in decimal values Ranging from 0 to 1
11:8	MONTH	RW	1	Current month in unit values Ranging from 1 to 9
7:4	DATE_DEC	RW	0	Current day in decimal values Ranging from 0 to 3
3:0	DATE	RW	1	Current day in unit values Ranging from 1 to 9

11.8.9 SCU RTC Alarm Time for BCD register 1 (SCU_BCD_RTC_ALM1)

Address Offset: 0x228

Bit	Field	Access	Initial	Description
31:27	–	–	0	Reserved
26:24	WEEKDAY	RW	0x7	Alarm weekday setting by BCD 0: Sunday 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday 7: Disable the weekday setting for the RTC alarm function
23:20	HOUR_DEC	RW	0x3	Alarm hour setting by BCD in decimal values Ranging from 0 to 2 The legal programmed value is up to 0x23, and set to 0x31 to disable the hour setting for the RTC alarm.
19:16	HOUR	RW	1	Alarm hour setting by BCD in unit values Ranging from 0 to 9 The legal programmed value is up to 0x23, and set to 0x31 to disable the hour setting for the RTC alarm.
15:12	MIN_DEC	RW	0x6	Alarm minute setting by BCD in decimal values Ranging from 0 to 5 The legal programmed value is up to 0x59, and set to 0x63 to disable the hour setting for the RTC alarm.
11:8	MIN	RW	0x3	Alarm minute setting by BCD in unit values Ranging from 0 to 9 The legal programmed value is up to 0x59, and set to 0x63 to disable the

Bit	Field	Access	Initial	Description
				hour setting for the RTC alarm.
7:4	SEC_DEC	RW	0x6	Alarm second setting by BCD in decimal values Ranging from 0 to 5 The legal programmed value is up to 0x59, and set to 0x63 to disable the hour setting for the RTC alarm.
3:0	SEC	RW	0x3	Alarm second setting by BCD in unit values Ranging from 0 to 9 The legal programmed value is up to 0x59, and set to 0x63 to disable the hour setting for the RTC alarm.

11.8.10 SCU RTC Alarm Time for BCD register 2 (SCU_BCD_RTC_ALM2)

Address Offset: 0x22C

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:12	MONTH_DEC	RW	0	Alarm month setting by BCD in decimal values The legal programmed value is ranging from 0x1 to 0x12, and set to 0x0 to disable the hour setting for the RTC alarm.
11:8	MONTH	RW	0	Alarm month setting by BCD in unit values The legal programmed value is ranging from 0x1 to 0x12, and set to 0x0 to disable the hour setting for the RTC alarm.
7:4	DATE_DEC	RW	0	Alarm date setting by BCD in decimal values The legal programmed value is ranging from 0x1 to 0x31, and set to 0x0 to disable the hour setting for the RTC alarm.
3:0	DATE	RW	0	Alarm date setting by BCD in unit values The legal programmed value is ranging from 0x1 to 0x31, and set to 0x0 to disable the hour setting for the RTC alarm.

11.9 RTC TRIMMING PROTECTION REGISTERS

Base Address: 0x4001 F000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0010	SCU_CHIPID	SCU RTC Chip ID register
0x0014	SCU_VERSION	SCU RTC Version register

11.9.1 SCU RTC Chip ID register (SCU_CHIPID)

Address Offset: 0x10

The CHIDID register is protected to avoid unexpected programming. The unlock RTC ID is the same CHIP_ID, and writes the value to the RTC version register. It will unlock the write protected and allow one write operation to the CHIPID register, and the register will resume the lock state after the write operation. When CHIP_ID programmed, the unlock key will also be updated to new CHIP_ID. User should use new CHIP_ID to unlock the protected register in the next write procedure.

Bit	Field	Access	Initial	Description
31:0	CHIP_ID	RPW	0x4281	RTC Chip ID

11.9.2 SCU RTC Version register (SCU_VERSION)

Address Offset: 0x14

Bit	Field	Access	Initial	Description
31:0	VERSION	R	0x30100	RTC IP version

12 DMA

12.1 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.

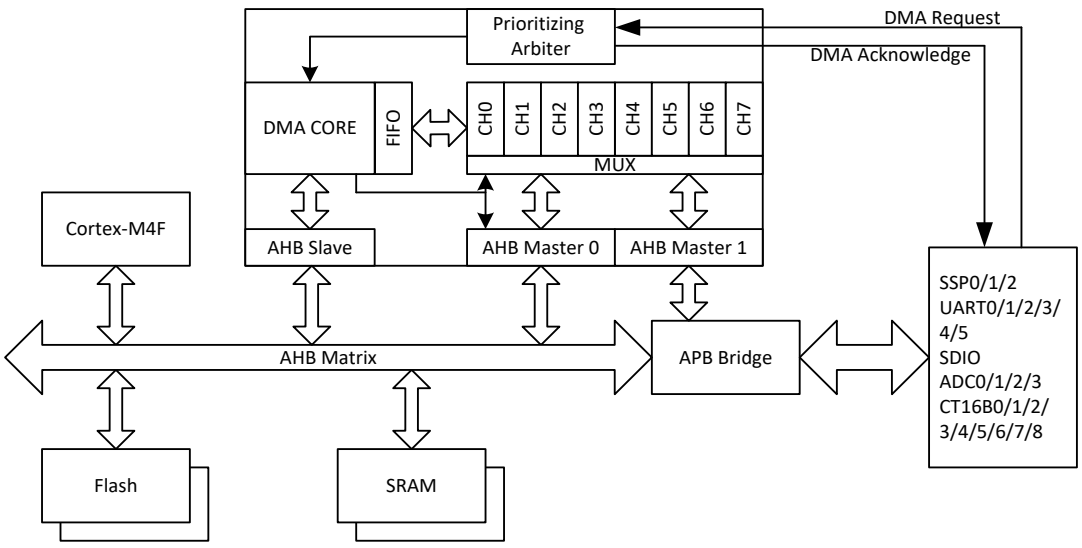
The DMA controller has up to 8 channels, each dedicated to managing memory access requests from one or more peripherals. It has an arbiter for handling the priority between DMA requests.

The DMAn clock is enabled by the DMAnCLKEN bit of SCU_AHBCLKG registers. Enabled by DMAnCLKEN bit of SCU_SLP_AHBCLKG in sleep mode.

12.2 FEATURES

- Up to 8 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
- 3 event flags (DMA Transfer count, DMA Transfer Abort and DMA Transfer Error) 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
- 2 AHB master interface for data transfer
- 64 DMA requests/acknowledges

12.3 BLOCK DIAGRAM



The DMA controller performs direct memory transfer by sharing the system bus with the CPU. 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.

12.4 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:

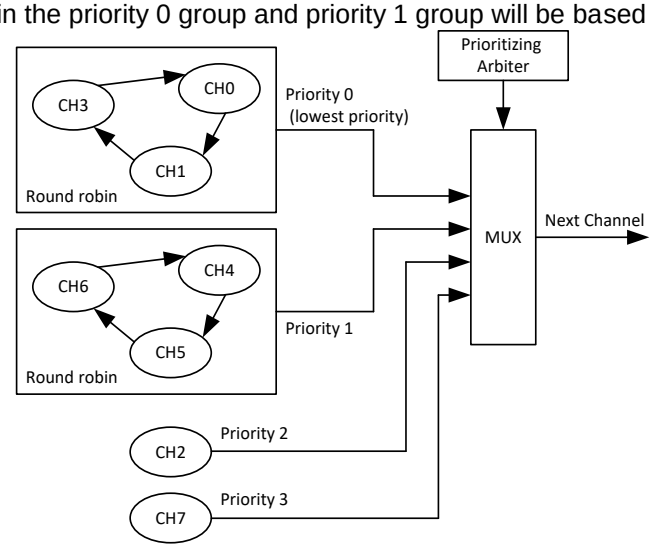
- The loading of data from the peripheral data register or a location in memory addressed through an internal current peripheral/memory address register. The start address used for the first transfer is the base peripheral/memory address programmed in the Cn_SRCADDR register.
- The storage of the data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

12.5 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.

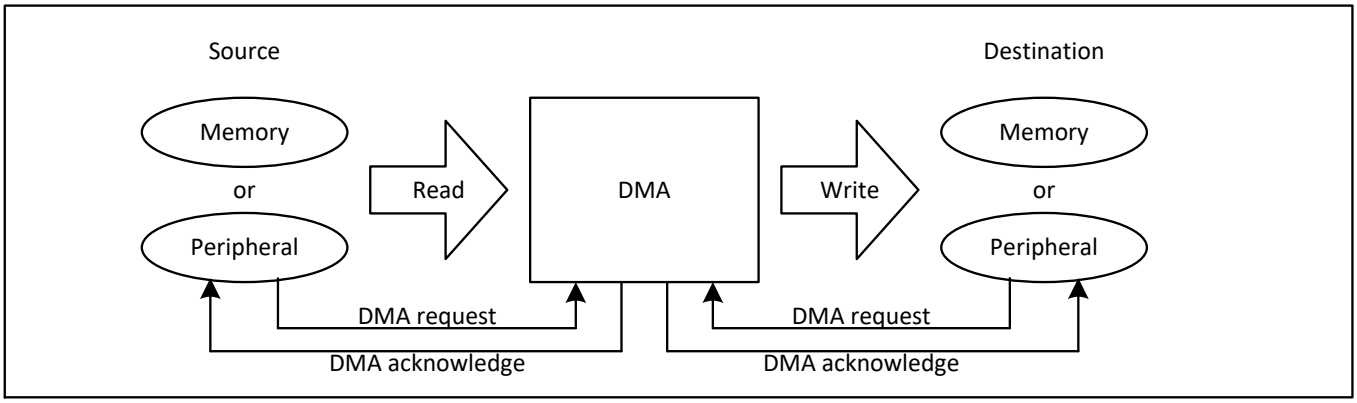
For example, channel 0, 1 and 3 are priority 0 level (lowest priority). Channel 4, 5 and 6 are priority 1 level (medium priority). Channel 2 is priority 2 level (high priority). Channel 7 is priority 3 level (very high priority). Priority level: 3 > 2 > 1 > 0. Therefore, channel 7 is the first operating 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.

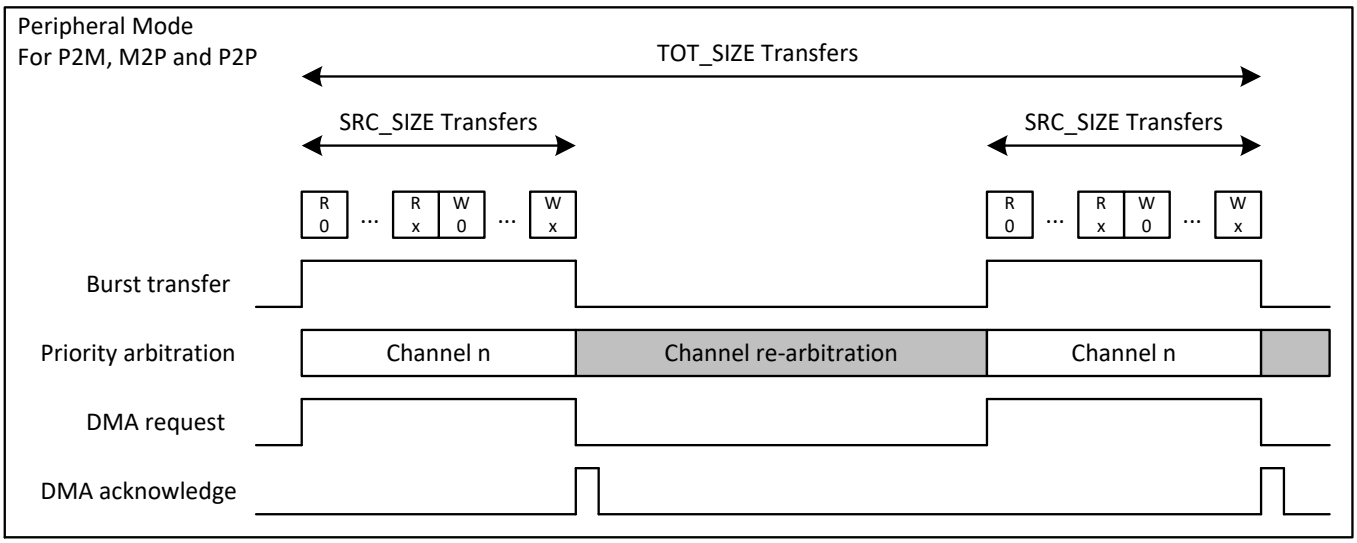


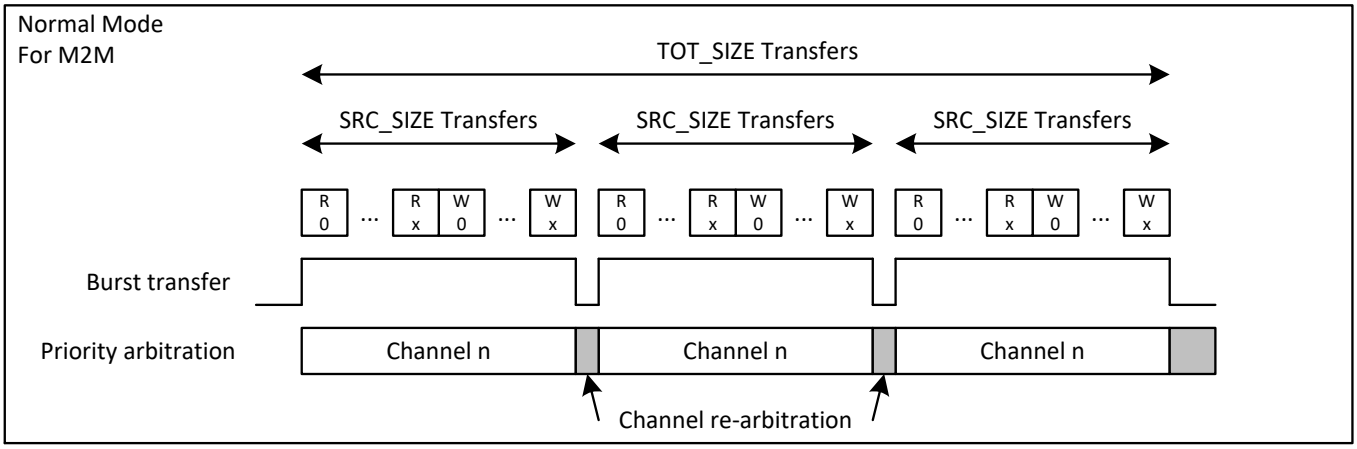
12.6 PERIPHERAL MODE

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.







12.7 CHANNEL CONFIGURATION

12.7.1 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.

12.7.2 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 cannot be larger than the FIFO size of the peripheral.***

12.7.3 Pointer Increment/Decrement

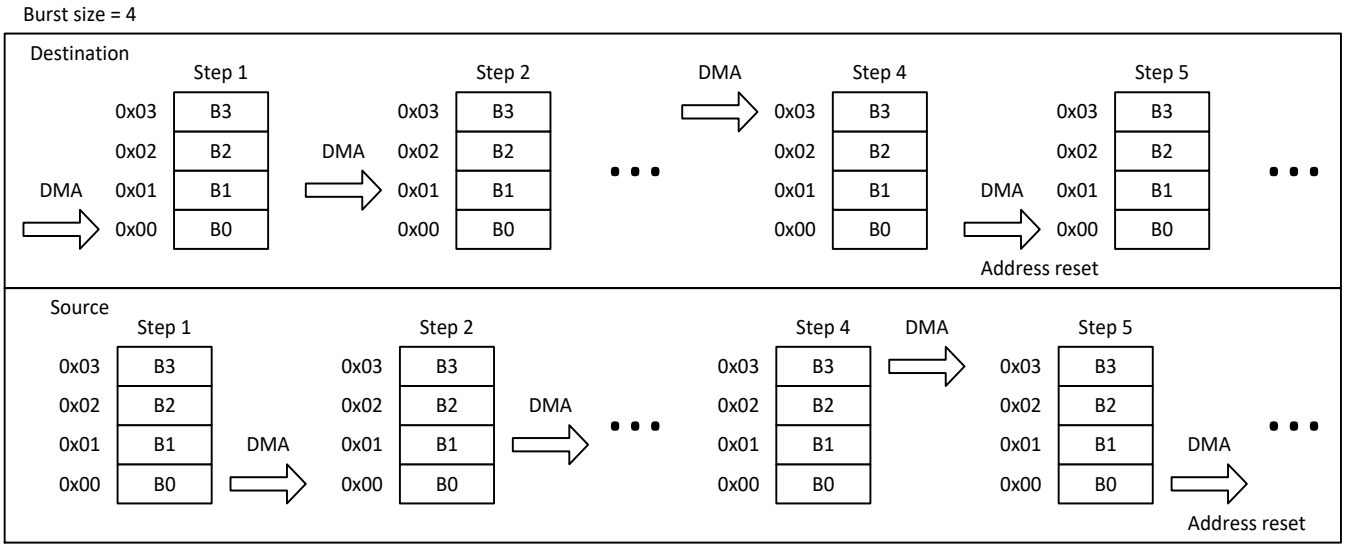
Peripheral and memory pointers can optionally be automatically post-incremented/decremented after each transaction depending on the SRCAD_CTL and DSTAD_CTL bits in the Cn_CSR register. If incremented mode is enabled, the address of the next transfer will be the address of the previous one incremented 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.

12.7.4 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. Usage is similar to incremental mode, except that the address is automatically reset. 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

✷ ***Note: When setting DSTAD_CTL to Increment cyclic mode, the SRC_WIDTH and DST_WIDTH must be set to the same width, otherwise DMA will behave incorrectly.***



12.7.5 Memory-to-Memory Mode

The DMA channels can also work without being triggered by a request from a peripheral. This mode is called Memory to Memory mode (M2M).

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.

12.7.6 Memory-to-Peripheral Mode

The DMA channels works with being triggered by a request from the destination peripheral. This mode is called Memory to Peripheral mode (M2P).

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.

12.7.7 Peripheral-to-Memory Mode

The DMA channels works with being triggered by a request from the source peripheral. This mode is called Peripheral to Memory mode (P2M).

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.

12.7.8 Peripheral-to-Peripheral Mode

The DMA channels works with being triggered by two request from the source peripheral and destination peripheral. This mode is called Peripheral to Peripheral mode (P2P).

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.

12.7.9 Configuration Procedure

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 Terminal Count Flag (TC) and Terminal Count interrupt Flag (INT_TC) are set and an interrupt is generated if the Terminal Count Interrupt Mask bit (INT_TC_MSK) is cleared.

12.8 TRANSFER MAPPING RULE

When SRC_WIDTH and DST_WIDTH are not equal, the DMA performs some data alignments as described in Burst Data is Enough for Packet Operation.

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.

Limitation: Do not set SRCAD_CTL = 01 (decrement source address) when the pack function works; otherwise DMA will have a wrong action.
Limitation: Do not set SRC_SIZE = 0 (burst size = 1) when the pack function works; otherwise DMA will have a wrong action.

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.

When source or destination is set big-endian, the DMA transfer results as described in the table: 12.8.4 Little-endian and Big-endian behavior for increment type

12.8.1 Burst Data is Enough for Packet Operation

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

Source			Destination		
Width	**Source content Data @Address N**		**Width**	**Destination content Data @Address M**	
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 @M+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

12.8.2 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.

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

Source			Destination		
Width	**Source content Data @Address N**		**Width**	**Destination content Data @Address M**	
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 @M+6 0xRR56 @M+4 0xRR34 @M+2 0xRR12 @M	0xRR12 @M 0xRR34 @M-2 0xRR56 @M-4 0xRR78 @M-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.***

12.8.3 Pack Function is Not Allowed for 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

12.8.4 Little-endian and Big-endian behavior for increment type

The conditions in the table below are when burst data is enough for packet operations.

Source			Destination		
Address Control	**Increment**		**Address Control**	**Increment**	
Width	**Source content Data @Address**		**Width**	**Destination content Data @Address**	
Endian Configuration	Little-endian	Big-endian	Endian Configuration	Little-endian	Big-endian
8	0x78 @N+3 0x56 @N+2 0x34 @N+1 0x12 @N	0x12 @N+3 0x34 @N+2 0x56 @N+1 0x78 @N	8	0x78 @M+3 0x56 @M+2 0x34 @M+1 0x12 @M	0x78 @M+3 0x56 @M+2 0x34 @M+1 0x12 @M
8	0x78 @N+3 0x56 @N+2 0x34 @N+1 0x12 @N	0x12 @N+3 0x34 @N+2 0x56 @N+1 0x78 @N	16	0x7856 @M+2 0x3412 @M	0x5678 @M+2 0x1234 @M
8	0x78 @N+3 0x56 @N+2 0x34 @N+1 0x12 @N	0x12 @N+3 0x34 @N+2 0x56 @N+1 0x78 @N	32	0x78563412 @M	0x12345678 @M
16	0x7856 @N+2 0x3412 @N	0x1234 @N+2 0x5678 @N	8	0x78 @M+3 0x56 @M+2 0x34 @M+1 0x12 @M	0x78 @M+3 0x56 @M+2 0x34 @M+1 0x12 @M
16	0x7856 @N+2 0x3412 @N	0x1234 @N+2 0x5678 @N	16	0x7856 @M+2 0x3412 @M	0x5678 @M+2 0x1234 @M
16	0x7856 @N+2 0x3412 @N	0x1234 @N+2 0x5678 @N	32	0x78563412 @M	0x12345678 @M
32	0x78563412 @N	0x12345678 @N	8	0x78 @M+3 0x56 @M+2 0x34 @M+1 0x12 @M	0x78 @M+3 0x56 @M+2 0x34 @M+1 0x12 @M
32	0x78563412 @N	0x12345678 @N	16	0x7856 @M+2 0x3412 @M	0x5678 @M+2 0x1234 @M
32	0x78563412 @N	0x12345678 @N	32	0x78563412 @M	0x12345678 @M

12.9 ABORT

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.

12.10 ERROE

DMA transfer errors are caused by AHB response errors during DMA read/write accesses. When a DMA transfer error occurs during a DMA read or a write access, the faulty channel is automatically disabled through a hardware clear of its CH_EN bit in the corresponding Channel control register (Cn_CSR). During the transfer, if the source or destination slave returns an ERROR response, the DMA will set the ERR bit (bits 7:0 of Error/Abort Status register (ERR_ABT) and terminate the DMA transfer at once. Then, if INT_ERR_MSK (bit 1 of Cn_CFG register) is not set, the DMA controller asserts INT_ERR. If you want to restart a new transmission event, first turn off the peripheral function of the source or destination. Make sure to cancel the last transfer request to avoid erroneous transfers.

IP	Memory address	DMA Read	DMA Write
Reserved	0x4008 0000 ~ 0xFFFF FFFF	Error	Error
AHB1 [1]	0x4005 0000 ~ 0x4007 FFFF	Follow AHB MATRIX	
Reserved	0x4004 0000 ~ 0x4004 FFFF	Follow AHB MATRIX	
APB1 [1]	0x4002 0000 ~ 0x4003 FFFF	Follow AHB MATRIX	
APB0 [1]	0x4000 0000 ~ 0x4001 FFFF	Follow AHB MATRIX	
Reserved	0x2002 8000 ~ 0x3FFF FFFF	Error	Error
SRAM 1~3	0x2000 0000 ~ 0x2002 7FFF	Follow AHB MATRIX	
Reserved	0x1002 8000 ~ 0x1FFF FFFF	Error	Error
Aliasing SRAM1~3	0x1000 0000 ~ 0x1002 7FFF	Error	Error
Reserved	0x0028 0000 ~ 0x0FFF FFFF	X [2]	X [2]
User ROM	0x0020 2000 ~ 0x0027 FFFF	Follow AHB MATRIX	Error
Boot ROM	0x0020 0000 ~ 0x0020 1FFF	Follow AHB MATRIX	Error
Shadow Memory (Reserved)	0x0007 D000 ~ 0x001F FFFF	X [2]	Error
Shadow Memory (Aliasing User ROM)	0x0000 0000 ~ 0x0007 CFFF	Follow AHB MATRIX	Error

[1] Reserved momery address in bus will respond errors during DMA read/write accesses , e.g., memory address 0x4005 6000 ~ 0x4005 6FFF in AHB1.
[2] Some memory address will respond error and some will not, but DMA Write will not succeed.

12.11 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. Each channel has its own interrupt vector, corresponding to the INT[7:0] bits

12.11.1 Request Selection

The hardware requests from the peripherals. Request source selected by SRC_RS bits and DST_RS bits. 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. But 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.

DMA0 SRC_RS/DST_RS Table

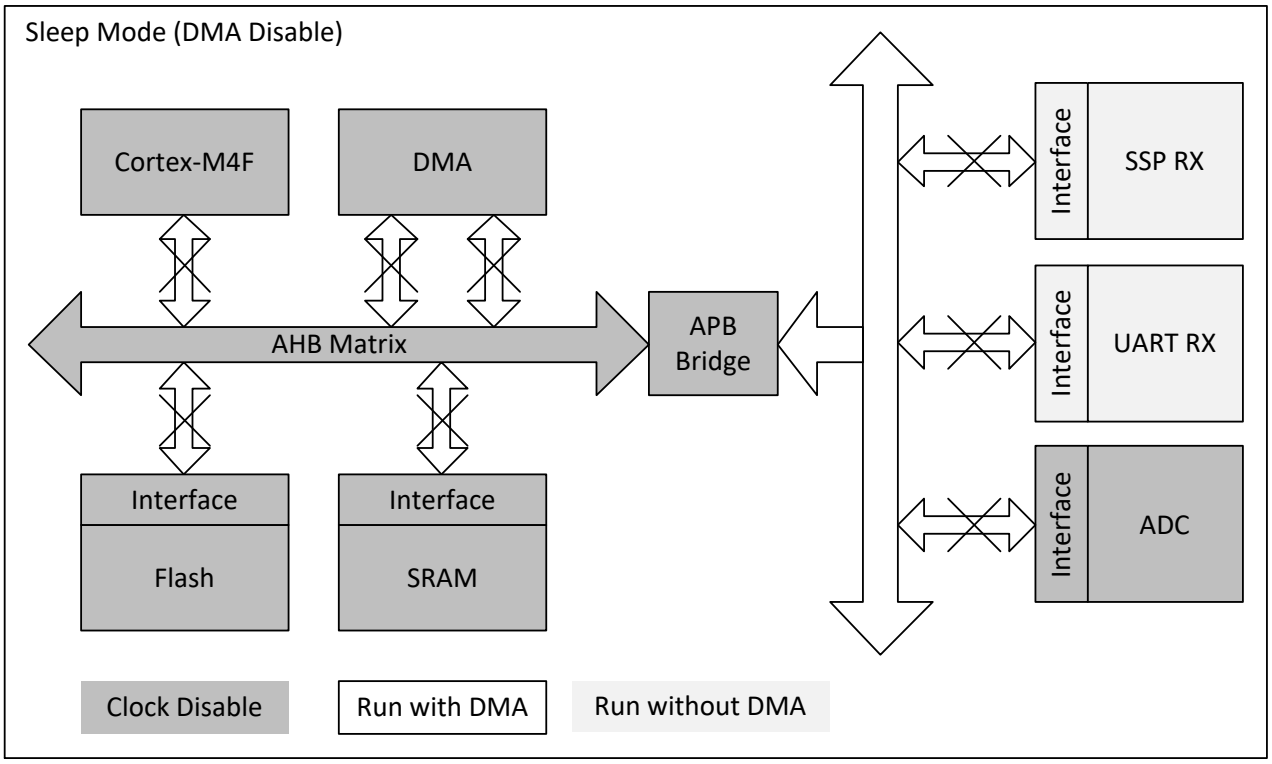
SRC_RS[5:0]/DST_RS[5:0]	Peripheral function	DMA Request
0	UART1 TX	Destination
1	UART1 RX	Source
2	UART2 TX	Destination
3	UART2 RX	Source
4	UART3 TX	Destination
5	UART3 RX	Source
6	UART4 TX	Destination
7	UART4 RX	Source
8	UART1 FIR	Source/Destination
9	UART2 FIR	Source/Destination
10	UART3 FIR	Source/Destination
11	UART4 FIR	Source/Destination
12	Reserve	
13	Reserve	
14	Reserve	
15	Reserve	
16	CT16B6 MR0	Source/Destination
17	CT16B7 MR0	Source/Destination
18~63	Reserve	

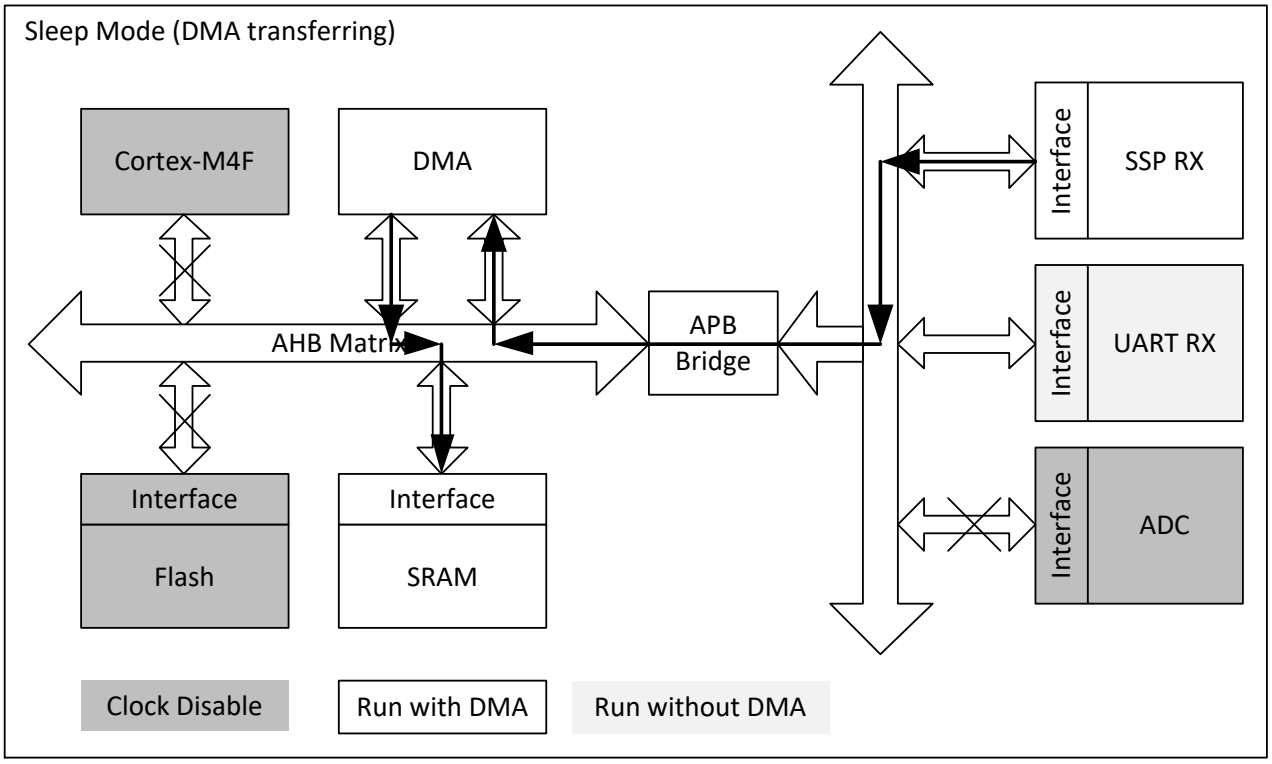
DMA1 SRC_RS/DST_RS Table

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	SSP0 TX	Destination	29	CT16B1 MR1	Source/Destination
1	SSP0 RX	Source	30	CT16B1 MR2	Source/Destination
2	SSP1 TX	Destination	31	CT16B1 MR3	Source/Destination
3	SSP1 RX	Source	32	CT16B2 CAP0	Source
4	SSP2 TX	Destination	33	CT16B2 MR9	Source/Destination
5	SSP2 RX	Source	34	CT16B2 MR0	Source/Destination
6	UART0 TX	Destination	35	CT16B2 MR1	Source/Destination
7	UART0 RX	Source	36	CT16B2 MR2	Source/Destination
8	UART5 TX	Destination	37	CT16B2 MR3	Source/Destination
9	UART5 RX	Source	38	Reserve	
10	ADC0_0	Source	39	CT16B3 MR9	Source/Destination
11	ADC0_1	Source	40	CT16B3 MR0	Source/Destination
12	ADC0_2	Source	41	CT16B3 MR1	Source/Destination
13	ADC0_3	Source	42	CT16B4 CAP0	Source
14	SDIO	Source/Destination	43	CT16B4 MR9	Source/Destination
15	UART0 FIR	Source/Destination	44	CT16B4 MR0	Source/Destination
16	UART5 FIR	Source/Destination	45	CT16B4 MR1	Source/Destination
17	LCM	Destination	46	CT16B5 CAP0	Source
18	Reserve		47	CT16B5 MR9	Source/Destination
19	Reserve		48	CT16B5 MR0	Source/Destination
20	CT16B0 CAP0	Source	49	CT16B5 MR1	Source/Destination
21	CT16B0 MR9	Source/Destination	50	CT16B5 MR2	Source/Destination
22	CT16B0 MR0	Source/Destination	51	CT16B5 MR3	Source/Destination
23	CT16B0 MR1	Source/Destination	52	Reserve	
24	CT16B0 MR2	Source/Destination	53	CT16B8 MR_PERIOD	Source/Destination
25	CT16B0 MR3	Source/Destination	54~63	Reserve	
26	CT16B1 CAP0	Source			
27	CT16B1 MR9	Source/Destination			
28	CT16B1 MR0	Source/Destination			

12.12 SLEEP MODE

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.





12.12.1 Sleep mode clock control register Table

DMA Source/Destination	Clock enable during sleep mode	
	Register	Control bit
Flash 512KB (Memory)	SCU_PWRMODE (0x4001F020) SCU_SLP_AHBCLKG (0x4001F058)	EFC_STB_OFF (bit 31) FLASHEN (bit 5)
SRAM1 32KB (Memory)	SCU_SLP_AHBCLKG (0x4001F058)	SRAM1EN (bit 2)
SRAM2 96KB (Memory)	SCU_SLP_AHBCLKG (0x4001F058)	SRAM2EN (bit 3)
SRAM3 32KB (Memory)	SCU_SLP_AHBCLKG (0x4001F058)	SRAM3EN (bit 4)
Backup SRAM	SCU_SLP_AHBCLKG (0x4001F058)	BKPSRAMEN (bit 10)
Always-On domain registers	SCU_SLP_APB0CLKG (0x4001F068)	ALWAYSONREGEN (bit 10)
ADCn (n = 0,1,2,3)	SCU_SLP_APB1CLKG (0x4001F06C)	ADC0CLKEN (bit 0)
SSPn RX/TX (n = 0,1,2)	SCU_SLP_APB1CLKG (0x4001F06C)	SSPnCLKEN (bit 1~3)
UARTn RX/TX (n = 0,1,2,3,4,5)	SCU_SLP_APB0CLKG (0x4001F068) SCU_SLP_APB1CLKG (0x4001F06C)	UARTnCLKEN, n=1~4 (bit 4~7) UARTnCLKEN, n=0,5 (bit 4~5)
SDIO	SCU_SLP_AHBCLKG (0x4001F058)	SDIOCLKEN (bit 6)
CT16Bn CAP0 (n = 0,1,2,3,4,5,8) CT16Bn MRm (n = 0,1,2,3,4,5,6,7,8)	SCU_SLP_APB0CLKG (0x4001F068) SCU_SLP_APB1CLKG (0x4001F06C)	CT16BnCLKEN, n=6,7 (bit 8~9) CT16BnCLKEN, n=0~5,8 (bit 10~15)

✸ ***Note:***
1. ***The DMAn can run in sleep mode by DMAnCLKEN bit of SCU_SLP_AHBCLKG register.***
2. ***The AHB Bus Matrix and APBn bridge can run in sleep mode by AHBMCLKEN bit and H2PnHCLKEN bit of SCU_SLP_AHBCLKG register.***

12.13 DMA REGISTERS

Base Address: 0x4005 3000 (DMA0)
0x4005 4000 (DMA1)

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	DMAn_INT	DMA n Interrupt Status register
0x0004	DMAn_INT_TC	DMA n Terminal Count Interrupt Status register
0x0008	DMAn_INT_TC_CLR	DMA n Terminal Count Interrupt Status Clear register
0x000C	DMAn_INT_ERR_ABT	DMA n Error/Abort Interrupt Status register
0x0010	DMAn_INT_ERR_ABT_CLR	DMA n Error/Abort Interrupt Status Clear register
0x0014	DMAn_TC	DMA n Terminal Count Status register
0x0018	DMAn_ERR_ABT	DMA n Error/Abort Status register
0x001C	DMAn_CH_EN	DMA n Channel Enable Status register
0x0020	DMAn_CH_BUSY	DMA n Channel Busy Status register
0x0024	DMAn_MCSR	DMA n Main Configuration Status register
0x0028	DMAn_SYNC	DMA n Synchronization register
0x002C – 0x00FC	–	Reserved
0x0100	DMAn_C0_CSR	DMA n Channel 0 control register
0x0104	DMAn_C0_CFG	DMA n Channel 0 Configuration register
0x0108	DMAn_C0_SRCADDR	DMA Channel 0 Source Address register
0x010C	DMAn_C0_DSTADDR	DMA n Channel 0 Destination Address register
0x0110	–	Reserved
0x0114	DMAn_C0_SIZE	DMA n Channel 0 Transfer Size register
0x0118 – 0x011C	–	Reserved
0x0120	DMAn_C1_CSR	DMA n Channel 1 control register
0x0124	DMAn_C1_CFG	DMA n Channel 1 Configuration register
0x0128	DMAn_C1_SRCADDR	DMA n Channel 1 Source Address register
0x012C	DMAn_C1_DSTADDR	DMA n Channel 1 Destination Address register
0x0130	–	Reserved
0x0134	DMAn_C1_SIZE	DMA n Channel 1 Transfer Size register
0x0138 – 0x013C	–	Reserved
0x0140	DMAn_C2_CSR	DMA n Channel 2 control register
0x0144	DMAn_C2_CFG	DMA n Channel 2 Configuration register
0x0148	DMAn_C2_SRCADDR	DMA n Channel 2 Source Address register
0x014C	DMAn_C2_DSTADDR	DMA n Channel 2 Destination Address register
0x0150	–	Reserved
0x0154	DMAn_C2_SIZE	DMA n Channel 2 Transfer Size register
0x0158 – 0x015C	–	Reserved
0x0160	DMAn_C3_CSR	DMA n Channel 3 control register
0x0164	DMAn_C3_CFG	DMA n Channel 3 Configuration register
0x0168	DMAn_C3_SRCADDR	DMA n Channel 3 Source Address register
0x016C	DMAn_C3_DSTADDR	DMA n Channel 3 Destination Address register
0x0170	–	Reserved
0x0174	DMAn_C3_SIZE	DMA n Channel 3 Transfer Size register
0x0178 – 0x017C	–	Reserved
0x0180	DMAn_C4_CSR	DMA n Channel 4 control register
0x0184	DMAn_C4_CFG	DMA n Channel 4 Configuration register
0x0188	DMAn_C4_SRCADDR	DMA n Channel 4 Source Address register
0x018C	DMAn_C4_DSTADDR	DMA n Channel 4 Destination Address register
0x0190	–	Reserved
0x0194	DMAn_C4_SIZE	DMA n Channel 4 Transfer Size register
0x0198 – 0x019C	–	Reserved
0x01A0	DMAn_C5_CSR	DMA n Channel 5 control register
0x01A4	DMAn_C5_CFG	DMA n Channel 5 Configuration register
0x01A8	DMAn_C5_SRCADDR	DMA n Channel 5 Source Address register
0x01AC	DMAn_C5_DSTADDR	DMA n Channel 5 Destination Address register
0x01B0	–	Reserved

0x01B4	DMAn_C5_SIZE	DMA n Channel 5 Transfer Size register
0x01B8 – 0x01BC	–	Reserved
0x01C0	DMAn_C6_CSR	DMA n Channel 6 control register
0x01C4	DMAn_C6_CFG	DMA n Channel 6 Configuration register
0x01C8	DMAn_C6_SRCADDR	DMA n Channel 6 Source Address register
0x01CC	DMA1_C6_DSTADDR	DMA n Channel 6 Destination Address register
0x01D0	–	Reserved
0x01D4	DMAn_C6_SIZE	DMA n Channel 6 Transfer Size register
0x01D8 – 0x01DC	–	Reserved
0x01E0	DMAn_C7_CSR	DMA n Channel 7 control register
0x01E4	DMAn_C7_CFG	DMA n Channel 7 Configuration register
0x01E8	DMAn_C7_SRCADDR	DMA n Channel 7 Source Address register
0x01EC	DMAn_C7_DSTADDR	DMA n Channel 7 Destination Address register
0x01F0	–	Reserved
0x01F4	DMAn_C7_SIZE	DMA n Channel 7 Transfer Size register

12.13.1 DMA n Interrupt Status register (DMAn_INT) (n=0,1)

Address Offset : 0x00

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

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	INT7	R	0	The result of INT_ABT7 | INT_ERR7 | INT_TC7 0: Channel 7 has no pending interrupt 1: Channel 7 has a pending interrupt
6	INT6	R	0	The result of INT_ABT6 | INT_ERR6 | INT_TC6 0: Channel 6 has no pending interrupt 1: Channel 6 has a pending interrupt
5	INT5	R	0	The result of INT_ABT5 | INT_ERR5 | INT_TC5 0: Channel 5 has no pending interrupt 1: Channel 5 has a pending interrupt
4	INT4	R	0	The result of INT_ABT4 | INT_ERR4 | INT_TC4 0: Channel 4 has no pending interrupt 1: Channel 4 has a pending interrupt
3	INT3	R	0	The result of INT_ABT3 | INT_ERR3 | INT_TC3 0: Channel 3 has no pending interrupt 1: Channel 3 has a pending interrupt
2	INT2	R	0	The result of INT_ABT2 | INT_ERR2 | INT_TC2 0: Channel 2 has no pending interrupt 1: Channel 2 has a pending interrupt
1	INT1	R	0	The result of INT_ABT1 | INT_ERR1 | INT_TC1 0: Channel 1 has no pending interrupt 1: Channel 1 has a pending interrupt
0	INT0	R	0	The result of INT_ABT0 | INT_ERR0 | INT_TC0 0: Channel 0 has no pending interrupt 1: Channel 0 has a pending interrupt

12.13.2 DMA n Terminal Count Interrupt Status register (DMAn_INT_TC) (n=0,1)

Address Offset:0x04

This register shows the status of the DMA terminal count 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 terminal count interrupt or not.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	INT_TC7	R	0	Status of the DMA terminal count interrupts after masking 0: Channel 7 has no pending interrupt 1: Channel 7 has a pending interrupt
6	INT_TC6	R	0	Status of the DMA terminal count interrupts after masking 0: Channel 6 has no pending interrupt 1: Channel 6 has a pending interrupt
5	INT_TC5	R	0	Status of the DMA terminal count interrupts after masking 0: Channel 5 has no pending interrupt 1: Channel 5 has a pending interrupt
4	INT_TC4	R	0	Status of the DMA terminal count interrupts after masking 0: Channel 4 has no pending interrupt 1: Channel 4 has a pending interrupt
3	INT_TC3	R	0	Status of the DMA terminal count interrupts after masking 0: Channel 3 has no pending interrupt 1: Channel 3 has a pending interrupt
2	INT_TC2	R	0	Status of the DMA terminal count interrupts after masking 0: Channel 2 has no pending interrupt 1: Channel 2 has a pending interrupt
1	INT_TC1	R	0	Status of the DMA terminal count interrupts after masking 0: Channel 1 has no pending interrupt 1: Channel 1 has a pending interrupt
0	INT_TC0	R	0	Status of the DMA terminal count interrupts after masking 0: Channel 0 has no pending interrupt 1: Channel 0 has a pending interrupt

12.13.3 DMA n Terminal Count Interrupt Status Clear register (DMAn_INT_TC_CLR) (n=0,1)

Address Offset:0x08

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	INT_TC_CLR7	W	0	Write 1 to clear the INT_TC7 and TC7 status
6	INT_TC_CLR6	W	0	Write 1 to clear the INT_TC6 and TC6 status
5	INT_TC_CLR5	W	0	Write 1 to clear the INT_TC5 and TC5 status
4	INT_TC_CLR4	W	0	Write 1 to clear the INT_TC4 and TC4 status
3	INT_TC_CLR3	W	0	Write 1 to clear the INT_TC3 and TC3 status
2	INT_TC_CLR2	W	0	Write 1 to clear the INT_TC2 and TC2 status
1	INT_TC_CLR1	W	0	Write 1 to clear the INT_TC1 and TC1 status
0	INT_TC_CLR0	W	0	Write 1 to clear the INT_TC0 and TC0 status

12.13.4 DMA n Error/Abort Interrupt Status register (DMAn_INT_ERR_ABT) (n=0,1)

Address Offset:0x0C

INT_ERR is the status of the DMA error interrupts after masking. The mask bit of these interrupts is bit 1 (INT_ERR_MSK) of channel configuration register (Cn_CFG). If this mask bit is set, the content of INT_ERR[n] of this register is always 0 no matter there exists a pending DMA error interrupt or not.

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	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23	INT_ABT7	R	0	Status of the DMA abort interrupts after masking 0: Channel 7 has no pending interrupt 1: Channel 7 has a pending interrupt
22	INT_ABT6	R	0	Status of the DMA abort interrupts after masking 0: Channel 6 has no pending interrupt 1: Channel 6 has a pending interrupt
21	INT_ABT5	R	0	Status of the DMA abort interrupts after masking 0: Channel 5 has no pending interrupt 1: Channel 5 has a pending interrupt
20	INT_ABT4	R	0	Status of the DMA abort interrupts after masking 0: Channel 4 has no pending interrupt 1: Channel 4 has a pending interrupt
19	INT_ABT3	R	0	Status of the DMA abort interrupts after masking 0: Channel 3 has no pending interrupt 1: Channel 3 has a pending interrupt
18	INT_ABT2	R	0	Status of the DMA abort interrupts after masking 0: Channel 2 has no pending interrupt 1: Channel 2 has a pending interrupt
17	INT_ABT1	R	0	Status of the DMA abort interrupts after masking 0: Channel 1 has no pending interrupt 1: Channel 1 has a pending interrupt
16	INT_ABT0	R	0	Status of the DMA abort interrupts after masking 0: Channel 0 has no pending interrupt 1: Channel 0 has a pending interrupt
15:8	–	–	0	Reserved
7	INT_ERR7	R	0	Status of the DMA error interrupts after masking 0: Channel 7 has no pending interrupt 1: Channel 7 has a pending interrupt
6	INT_ERR6	R	0	Status of the DMA error interrupts after masking 0: Channel 6 has no pending interrupt 1: Channel 6 has a pending interrupt
5	INT_ERR5	R	0	Status of the DMA error interrupts after masking 0: Channel 5 has no pending interrupt 1: Channel 5 has a pending interrupt
4	INT_ERR4	R	0	Status of the DMA error interrupts after masking 0: Channel 4 has no pending interrupt 1: Channel 4 has a pending interrupt
3	INT_ERR3	R	0	Status of the DMA error interrupts after masking 0: Channel 3 has no pending interrupt 1: Channel 3 has a pending interrupt
2	INT_ERR2	R	0	Status of the DMA error interrupts after masking 0: Channel 2 has no pending interrupt 1: Channel 2 has a pending interrupt
1	INT_ERR1	R	0	Status of the DMA error interrupts after masking 0: Channel 1 has no pending interrupt 1: Channel 1 has a pending interrupt
0	INT_ERR0	R	0	Status of the DMA error interrupts after masking 0: Channel 0 has no pending interrupt 1: Channel 0 has a pending interrupt

12.13.5 DMA n Error/Abort Interrupt Status Clear register (DMAn_INT_ERR_ABT_CLR) (n=0,1)

Address Offset:0x10

Bit	Field	Access	Initial	Description
31:24	–	–	0	Reserved

Bit	Field	Access	Initial	Description
23	INT_ABT_CLR7	W	0	Write 1 to clear the INT_ABT7 and ABT7 status
22	INT_ABT_CLR6	W	0	Write 1 to clear the INT_ABT6 and ABT6 status
21	INT_ABT_CLR5	W	0	Write 1 to clear the INT_ABT5 and ABT5 status
20	INT_ABT_CLR4	W	0	Write 1 to clear the INT_ABT4 and ABT4 status
19	INT_ABT_CLR3	W	0	Write 1 to clear the INT_ABT3 and ABT3 status
18	INT_ABT_CLR2	W	0	Write 1 to clear the INT_ABT2 and ABT2 status
17	INT_ABT_CLR1	W	0	Write 1 to clear the INT_ABT1 and ABT1 status
16	INT_ABT_CLR0	W	0	Write 1 to clear the INT_ABT0 and ABT0 status
15:8	–	–	0	Reserved
7	INT_ERR_CLR7	W	0	Write 1 to clear the INT_ERR7 and ERR7 status
6	INT_ERR_CLR6	W	0	Write 1 to clear the INT_ERR6 and ERR6 status
5	INT_ERR_CLR5	W	0	Write 1 to clear the INT_ERR5 and ERR5 status
4	INT_ERR_CLR4	W	0	Write 1 to clear the INT_ERR4 and ERR4 status
3	INT_ERR_CLR3	W	0	Write 1 to clear the INT_ERR3 and ERR3 status
2	INT_ERR_CLR2	W	0	Write 1 to clear the INT_ERR2 and ERR2 status
1	INT_ERR_CLR1	W	0	Write 1 to clear the INT_ERR1 and ERR1 status
0	INT_ERR_CLR0	W	0	Write 1 to clear the INT_ERR0 and ERR0 status

12.13.6 DMA n Terminal Count Status register (DMAn_TC) (n=0,1)

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	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	TC7	R	0	Status of the DMA terminal count 0: Channel 7 has no terminal count status 1: Channel 7 has a terminal count status
6	TC6	R	0	Status of the DMA terminal count 0: Channel 6 has no terminal count status 1: Channel 6 has a terminal count status
5	TC5	R	0	Status of the DMA terminal count 0: Channel 5 has no terminal count status 1: Channel 5 has a terminal count status
4	TC4	R	0	Status of the DMA terminal count 0: Channel 4 has no terminal count status 1: Channel 4 has a terminal count status
3	TC3	R	0	Status of the DMA terminal count 0: Channel 3 has no terminal count status 1: Channel 3 has a terminal count status
2	TC2	R	0	Status of the DMA terminal count 0: Channel 2 has no terminal count status 1: Channel 2 has a terminal count status
1	TC1	R	0	Status of the DMA terminal count 0: Channel 1 has no terminal count status 1: Channel 1 has a terminal count status
0	TC0	R	0	Status of the DMA terminal count 0: Channel 0 has no terminal count status 1: Channel 0 has a terminal count status

12.13.7 DMA n Error/Abort Status register (DMAn_ERR_ABT) (n=0,1)

Address Offset:0x18

ERR is the status of the DMA error. If an AHB ERROR response happens during DMA transfer, the DMA controller will stop the current DMA transfer and set ERR[n] to 1. Then, if INT_ERR_MSK is not set, the DMA controller will set INT_ERR[n] to 1 and assert interrupt.

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	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23	ABT7	R	0	Status of the DMA abort 0: Channel 7 has no abort status 1: Channel 7 has an abort status
22	ABT6	R	0	Status of the DMA abort 0: Channel 6 has no abort status 1: Channel 6 has an abort status
21	ABT5	R	0	Status of the DMA abort 0: Channel 5 has no abort status 1: Channel 5 has an abort status
20	ABT4	R	0	Status of the DMA abort 0: Channel 4 has no abort status 1: Channel 4 has an abort status
19	ABT3	R	0	Status of the DMA abort 0: Channel 3 has no abort status 1: Channel 3 has an abort status
18	ABT2	R	0	Status of the DMA abort 0: Channel 2 has no abort status 1: Channel 2 has an abort status
17	ABT1	R	0	Status of the DMA abort 0: Channel 1 has no abort status 1: Channel 1 has an abort status
16	ABT0	R	0	Status of the DMA abort 0: Channel 0 has no abort status 1: Channel 0 has an abort status
15:8	–	–	0	Reserved
7	ERR7	R	0	Status of the DMA error 0: Channel 7 has no error status 1: Channel 7 has an error status
6	ERR6	R	0	Status of the DMA error 0: Channel 6 has no error status 1: Channel 6 has an error status
5	ERR5	R	0	Status of the DMA error 0: Channel 5 has no error status 1: Channel 5 has an error status
4	ERR4	R	0	Status of the DMA error 0: Channel 4 has no error status 1: Channel 4 has an error status
3	ERR3	R	0	Status of the DMA error 0: Channel 3 has no error status 1: Channel 3 has an error status
2	ERR2	R	0	Status of the DMA error 0: Channel 2 has no error status 1: Channel 2 has an error status
1	ERR1	R	0	Status of the DMA error 0: Channel 1 has no error status 1: Channel 1 has an error status
0	ERR0	R	0	Status of the DMA error 0: Channel 0 has no error status 1: Channel 0 has an error status

12.13.8 DMA n Channel Enable Status register (DMAn_CH_EN) (n=0,1)

Address Offset:0x1C

This register shows the DMA channel enable status.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	CH_EN7	R	0	Status of the channel 7 CH_EN bit of C7_CSR register 0: CH_EN = 0 1: CH_EN = 1
6	CH_EN6	R	0	Status of the channel 6 CH_EN bit of C6_CSR register 0: CH_EN = 0 1: CH_EN = 1
5	CH_EN5	R	0	Status of the channel 5 CH_EN bit of C5_CSR register 0: CH_EN = 0 1: CH_EN = 1
4	CH_EN4	R	0	Status of the channel 4 CH_EN bit of C4_CSR register 0: CH_EN = 0 1: CH_EN = 1
3	CH_EN3	R	0	Status of the channel 3 CH_EN bit of C3_CSR register 0: CH_EN = 0 1: CH_EN = 1
2	CH_EN2	R	0	Status of the channel 2 CH_EN bit of C2_CSR register 0: CH_EN = 0 1: CH_EN = 1
1	CH_EN1	R	0	Status of the channel 1 CH_EN bit of C1_CSR register 0: CH_EN = 0 1: CH_EN = 1
0	CH_EN0	R	0	Status of the channel 0 CH_EN bit of C0_CSR register 0: CH_EN = 0 1: CH_EN = 1

12.13.9 DMA n Channel Busy Status register (DMAn_CH_BUSY) (n=0,1)

Address Offset:0x20

This register shows the DMA channel busy status.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	CH_BUSY7	R	0	Status of the channel 7 BUSY bit of C7_CFG register 0: BUSY = 0 1: BUSY = 1
6	CH_BUSY6	R	0	Status of the channel 6 BUSY bit of C6_CFG register 0: BUSY = 0 1: BUSY = 1
5	CH_BUSY5	R	0	Status of the channel 5 BUSY bit of C5_CFG register 0: BUSY = 0 1: BUSY = 1
4	CH_BUSY4	R	0	Status of the channel 4 BUSY bit of C4_CFG register 0: BUSY = 0 1: BUSY = 1
3	CH_BUSY3	R	0	Status of the channel 3 BUSY bit of C3_CFG register 0: BUSY = 0 1: BUSY = 1
2	CH_BUSY2	R	0	Status of the channel 2 BUSY bit of C2_CFG register 0: BUSY = 0 1: BUSY = 1
1	CH_BUSY1	R	0	Status of the channel 1 BUSY bit of C1_CFG register 0: BUSY = 0 1: BUSY = 1
0	CH_BUSY0	R	0	Status of the channel 0 BUSY bit of C0_CFG register 0: BUSY = 0

Bit	Field	Access	Initial	Description
				1: BUSY = 1

12.13.10 DMA n Main Configuration Status register (DMAn_MCSR) (n=0,1)

Address Offset:0x24

Bit	Field	Access	Initial	Description
31:3	–	–	0	Reserved
2	M1ENDIAN	RW	0	AHB Master 1 endian configuration 0: Little-endian 1: Big-endian
1	M0ENDIAN	RW	0	AHB Master 0 endian configuration 0: Little-endian 1: Big-endian
0	DMACEN	RW	0	DMA controller enable 0: Disable 1: Enable

12.13.11 DMA n Synchronization register (DMAn_SYNC) (n=0,1)

Address Offset:0x28

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	SYNC7	RW	0	DMA synchronization logic enable for channel 7 request 0: Disable 1: Enable
6	SYNC6	RW	0	DMA synchronization logic enable for channel 6 request 0: Disable 1: Enable
5	SYNC5	RW	0	DMA synchronization logic enable for channel 5 request 0: Disable 1: Enable
4	SYNC4	RW	0	DMA synchronization logic enable for channel 4 request 0: Disable 1: Enable
3	SYNC3	RW	0	DMA synchronization logic enable for channel 3 request 0: Disable 1: Enable
2	SYNC2	RW	0	DMA synchronization logic enable for channel 2 request 0: Disable 1: Enable
1	SYNC1	RW	0	DMA synchronization logic enable for channel 1 request 0: Disable 1: Enable
0	SYNC0	RW	0	DMA synchronization logic enable for channel 0 request 0: Disable 1: Enable

12.13.12 DMA n Channel m Control register (DMAn_Cm_CSR) (n=0,1/m=0,1,2,3,4,5,6,7)

Address Offset:0x100, 0x120, 0x140, 0x160, 0x180, 0x1A0, 0x1C0, 0x1E0

Bit	Field	Access	Initial	Description
31	TC_MSK	RW	0	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
30:27	–	–	0	Reserved
26:24	DMA_FF_TH	RW	0	DMA FIFO threshold value. FIFO depth 8x32 0: Threshold value = 1 1: Threshold value = 2 2: Threshold value = 4 3: Reserved 4: Reserved Other: Reserved When DMA FIFO space ≥ DMA_FF_TH, then DMA controller will start to transfer the data from the source to FIFO. When the number of valid data in the DMA FIFO is greater than DMA_FF_TH, then the DMA controller will start to pop out data from FIFO to the destination. Note that DMA_FF_TH cannot be larger than 1/2 DMA FIFO size.
23:22	CHPRI	RW	0	Channel priority level 0: Lowest priority 1: 3rd high priority 2: 2nd high priority 3: Highest priority
21:19	–	–	0	Reserved
18:16	SRC_SIZE	RW	0	Source burst size selection 0: Burst size = 1 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 Note: 1. Source burst size is not related to the HBURST (AHB signals). It 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). 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
15	ABT	W	0	Transaction abort Writing 1 to this bit will cause the DMA to stop the current transfer, then set the ABT[n] bit of Error/Abort Status register and assert dmaint interrupt if INT_ABT_MST = 0. Note: No matter what value you write, if you write 1 to bit-15 (ABT), all other bits of this register will remain the same. That is, you can't program bit[15:1] and the other bits of this register simultaneously.
14:13	–	–	0	Reserved
12:11	SRC_WIDTH	RW	0x2	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

Bit	Field	Access	Initial	Description
				2: Transfer width is 32 bits Other: Reserved Note: 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. Limitation: Do not set SRCAD_CTL = 01 (decrement source address) when the pack function works; otherwise DMA will have a wrong action. 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.
10	–	–	0	Reserved
9:8	DST_WIDTH	RW	0x2	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 Note: 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. Limitation: Do not set SRCAD_CTL = 01 (decrement source address) when the pack function works; otherwise DMA will have a wrong action. 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.
7	MODE	RW	0	Support Hardware Handshake Mode 0: Normal mode 1: Peripheral mode
6:5	SRCAD_CTL	RW	0	Source Address Control 0: Increment source address 1: Decrement source address 2: Fixed source address 3: Increment cyclic mode Note: 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. Limitation: Do not set SRCAD_CTL = 01 (decrement source address) when the pack function works; otherwise DMA will have a wrong action. 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.
4:3	DSTAD_CTL	RW	0	Destination Address Control 0: Increment destination address 1: Decrement destination address 2: Fixed destination address 3: Increment cyclic mode Note: 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. Limitation: Do not set SRCAD_CTL = 01 (decrement source address) when the pack function works; otherwise DMA will have a wrong action. 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.

Bit	Field	Access	Initial	Description
				Limitation: When setting DSTAD_CTL to Increment cyclic mode, the SRC_WIDTH and DST_WIDTH must be set to the same width, otherwise DMA will behave incorrectly.
2	SRC_SEL	RW	0	Choose AHB Master for Source 0: AHB Master 0 is the source 1: AHB Master 1 is the source Please refer to AHB BUS Matrix to access the slave device.
1	DST_SEL	RW	0	Choose AHB Master for Destination 0: AHB Master 0 is the destination 1: AHB Master 1 is the destination Please refer to AHB BUS Matrix to access the slave device.
0	CH_EN	RW	0	Channel m Enable 0: Disable 1: Enable

12.13.13 DMA 0 Channel m Configuration register (DMA0_Cm_CFG) (m=0,1,2,3,4,5,6,7)

Address Offset:0x104, 0x124, 0x144, 0x164, 0x184, 0x1A4, 0x1C4, 0x1E4

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:24	DST_RS	RW	0	Destination DMA request select 0: UART1 TX 1: UART1 RX 2: UART2 TX 3: UART2 RX 4: UART3 TX 5: UART3 RX 6: UART4 TX 7: UART4 RX 8: UART1 FIR 9: UART2 FIR 10: UART3 FIR 11: UART4 FIR 16: CT16B6 MR0 17: CT16B7 MR0 Other: Reserved
23:22	–	–	0	Reserved
21:16	SRC_RS	RW	0	Source DMA request select 0: UART1 TX 1: UART1 RX 2: UART2 TX 3: UART2 RX 4: UART3 TX 5: UART3 RX 6: UART4 TX 7: UART4 RX 8: UART1 FIR 9: UART2 FIR 10: UART3 FIR 11: UART4 FIR 16: CT16B6 MR0 17: CT16B7 MR0 Other: Reserved
15:14	–	–	0	Reserved
13	DST_HE	RW	1	Destination Peripheral Mode enable 0: Disable 1: Enable When you disable the destination hardware handshake (peripheral mode), DMA will start transfer data without waiting the destination request. This bit is only valid when DMA is in the Peripheral Mode.
12:9	–	–	0	Reserved

Bit	Field	Access	Initial	Description
8	BUSY	R	0	The DMA channel is busy 0: The DMA channel is not busy 1: The DMA channel is busy
7	SRC_HE	RW	1	Source Peripheral Mode enable 0: Disable 1: Enable When you disable the source hardware handshake (peripheral mode), DMA will start transfer data without waiting the source request. This bit is only valid when DMA is in the Peripheral Mode.
6:3	–	–	0	Reserved
2	INT_ABT_MSK	RW	1	Channel abort interrupt mask 0: No mask interrupt 1: Mask interrupt
1	INT_ERR_MSK	RW	1	Channel error interrupt mask 0: No mask interrupt 1: Mask interrupt
0	INT_TC_MSK	RW	1	Channel terminal count interrupt mask 0: No mask interrupt 1: Mask interrupt

12.13.14 DMA 1 Channel m Configuration register (DMA1_Cm_CFG) (m=0,1,2,3,4,5,6,7)

Address Offset:0x104, 0x124, 0x144, 0x164, 0x184, 0x1A4, 0x1C4, 0x1E4

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:24	DST_RS	RW	0	Destination DMA request select 0: SSP0 TX 1: SSP0 RX 2: SSP1 TX 3: SSP1 RX 4: SSP2 TX 5: SSP2 RX 6: UART0 TX 7: UART0 RX 8: UART5 TX 9: UART5 RX 10: ADC0_0 11: ADC0_1 12: ADC0_2 13: ADC0_3 14: SDIO 15: UART0 FIR 16: UART5 FIR 17: LCM 20: CT16B0 CAP0 21: CT16B0 MR9 22: CT16B0 MR0 23: CT16B0 MR1 24: CT16B0 MR2 25: CT16B0 MR3 26: CT16B1 CAP0 27: CT16B1 MR9 28: CT16B1 MR0 29: CT16B1 MR1 30: CT16B1 MR2 31: CT16B1 MR3 32: CT16B2 CAP0 33: CT16B2 MR9 34: CT16B2 MR0 35: CT16B2 MR1 36: CT16B2 MR2 37: CT16B2 MR3

Bit	Field	Access	Initial	Description
				38: Reserved 39: CT16B3 MR9 40: CT16B3 MR0 41: CT16B3 MR1 42: CT16B4 CAP0 43: CT16B4 MR9 44: CT16B4 MR0 45: CT16B4 MR1 46: CT16B5 CAP0 47: CT16B5 MR9 48: CT16B5 MR0 49: CT16B5 MR1 50: CT16B5 MR2 51: CT16B5 MR3 52: Reserved 53: CT16B8 MR_PERIOD Other: Reserved
23:22	–	–	0	Reserved
21:16	SRC_RS	RW	0	Source DMA request select 0: SSP0 TX 1: SSP0 RX 2: SSP1 TX 3: SSP1 RX 4: SSP2 TX 5: SSP2 RX 6: UART0 TX 7: UART0 RX 8: UART5 TX 9: UART5 RX 10: ADC0_0 11: ADC0_1 12: ADC0_2 13: ADC0_3 14: SDIO 15: UART0 FIR 16: UART5 FIR 17: LCM 20: CT16B0 CAP0 21: CT16B0 MR9 22: CT16B0 MR0 23: CT16B0 MR1 24: CT16B0 MR2 25: CT16B0 MR3 26: CT16B1 CAP0 27: CT16B1 MR9 28: CT16B1 MR0 29: CT16B1 MR1 30: CT16B1 MR2 31: CT16B1 MR3 32: CT16B2 CAP0 33: CT16B2 MR9 34: CT16B2 MR0 35: CT16B2 MR1 36: CT16B2 MR2 37: CT16B2 MR3 38: Reserved 39: CT16B3 MR9 40: CT16B3 MR0 41: CT16B3 MR1 42: CT16B4 CAP0 43: CT16B4 MR9 44: CT16B4 MR0 45: CT16B4 MR1 46: CT16B5 CAP0 47: CT16B5 MR9 48: CT16B5 MR0

Bit	Field	Access	Initial	Description
				49: CT16B5 MR1 50: CT16B5 MR2 51: CT16B5 MR3 52: Reserved 53: CT16B8 MR_PERIOD Other: Reserved
15:14	–	–	0	Reserved
13	DST_HE	RW	0	Destination Peripheral Mode enable 0: Disable 1: Enable When you disable the destination hardware handshake (peripheral mode), DMA will start transfer data without waiting the destination request. This bit is only valid when DMA is in the Peripheral Mode.
12:9	–	–	0	Reserved
8	BUSY	R	0	The DMA channel is busy 0: The DMA channel is not busy 1: The DMA channel is busy
7	SRC_HE	RW	0	Source Peripheral Mode enable 0: Disable 1: Enable When you disable the source hardware handshake (peripheral mode), DMA will start transfer data without waiting the source request. This bit is only valid when DMA is in the Peripheral Mode.
6:3	–	–	0	Reserved
2	INT_ABT_MSK	RW	1	Channel abort interrupt mask 0: No mask interrupt 1: Mask interrupt
1	INT_ERR_MSK	RW	1	Channel error interrupt mask 0: No mask interrupt 1: Mask interrupt
0	INT_TC_MSK	RW	1	Channel terminal count interrupt mask 0: No mask interrupt 1: Mask interrupt

12.13.15 DMA n Channel m Source Address register (DMAn_Cm_SRCADDR) (n=0,1/m=0,1,2,3,4,5,6,7)

Address Offset:0x108, 0x128, 0x148, 0x168, 0x188, 0x1A8, 0x1C8, 0x1E8

Bit	Field	Access	Initial	Description
31:0	SRCADDR	RW	0	Source starting address Note: When the DMA transaction is done, its value changes to the DMA source ending address.

12.13.16 DMA n Channel m Destination Address register (DMA_Cn_DSTADDR) (n=0,1/m=0,1,2,3,4,5,6,7)

Address Offset:0x10C, 0x12C, 0x14C, 0x16C, 0x18C, 0x1AC, 0x1CC, 0x1EC

Bit	Field	Access	Initial	Description
31:0	DSTADDR	RW	0	Destination starting address Note: When the DMA transaction is done, its value changes to the DMA destination ending address.

12.13.17 DMA n Channel m Transfer Size register (DMAn_Cm_SIZE) (n=0,1/m=0,1,2,3,4,5,6,7)

Address Offset:0x114, 0x134, 0x154, 0x174, 0x194, 0x1B4, 0x1D4, 0x1F4

Bit	Field	Access	Initial	Description
31:22	–	–	0	Reserved
21:0	TOT_SIZE	RW	0	Total transfer size for source 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 Note: When the DMA transaction is done, its value changes to 0.

13 SYNCHRONOUS SERIAL PORT (SSP)

13.1 OVERVIEW

SSP is a synchronous serial port interface that allows the host processor to serve as a master or a slave. Various devices can be connected to this controller by using the serial protocol. SSP supports the Serial Peripheral Interface (SPI) from Motorola, MICROWIRE from National Semiconductor and I2S from Philips formats. The serial data formats may range from 4 bits to 128 bits in length. The SSP controller can use the on-chip DMA to directly transfer data between the external serial device and the system memory without the intervention from the processor.

The SPI mode 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.

The I2S bus specification defines a 4-wire serial bus, having data in, data out, BCLK and word select signal. The basic I2S connection has one master, which is always the master, and one slave.

✷ ***Note: The synchronizer from SSPCLK to PCLK needs three PCLK cycles. To ensure that the data are stable, the clock period of SCLK (Tsclk) and the clock period of PCLK (Tpclk) should meet the following formula: (3.5 x Tpclk) < (Tsclk x SDL_length)***

13.2 FEATURES

- Supports Motorola SPI, National Semiconductor MICROWIRE, and Philips I2S frame
- Supports master mode or slave mode
- SPI mode: 4-bit to 16-bit frame.
- I2S mode: Capable of handling 8/16/24/32-bit data length and 16/24/32-bit channel length.
- Maximum SPI0 speed of 48 Mbps (master) or 32 Mbps (slave)
- Maximum SPI1/2 speed of 32 Mbps (master) or 32 Mbps (slave)
- Programmable serial data transfer format (MSB or LSB first)
- The start phase of data sampling location selection is 1st-phase or 2nd-phase controlled register.
- Supports internally or externally controlled serial bit clock
- Supports internally or externally controlled SEL/WS format
- Programmable SEL/WS polarity
- Programmable serial bit clock polarity, phase, and frequency
- Programmable I2S format including zero bit padding and right/left justification
- Programmable threshold interrupt of transmit/receive FIFO
- 8-level FIFOs for both transmitter and receiver
- Supply DMA transfer

13.3 PIN DESCRIPTION

SPI Mode

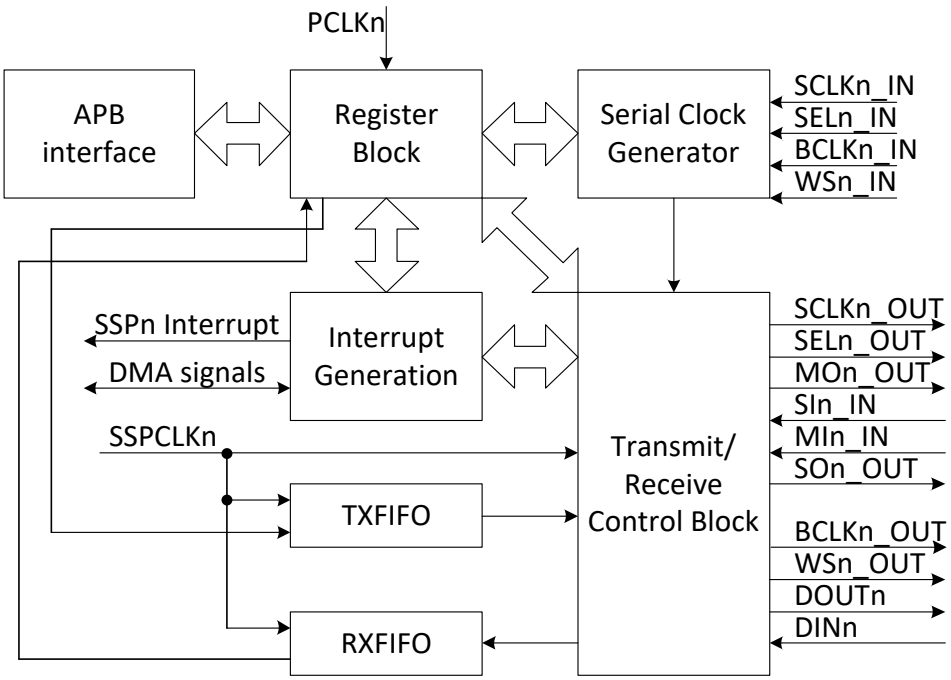
Pin Name	Type	Description	GPIO Configuration
SCKn	O	SPI Serial clock (Master)	Depends on AFIO register
	I	SPI Serial clock (Slave)	Depends on AFIO register

SELn	O	SPI Slave Select (Master)	Depends on AFIO register
	I	SPI Slave Select (Slave)	Depends on AFIO register
MIn	I	Master In (Master)	Depends on AFIO register
SOn	O	Slave Out (Slave)	Depends on AFIO register
MOn	O	Master Out (Master)	Depends on AFIO register
SIn	I	Slave In (Slave)	Depends on AFIO register

I2S Mode

Pin Name	Type	Description	GPIO Configuration
BCLKn	O	I2S Bit clock (Master)	Depends on AFIO register
	I	I2S Bit clock (Slave)	Depends on AFIO register
WSn	O	I2S Word Select (Master)	Depends on AFIO register
	I	I2S Word Select (Slave)	Depends on AFIO register
DINn	I	I2S Received Serial data	Depends on AFIO register
DOUTn	O	I2S Transmitted Serial data	Depends on AFIO register

13.3.1 Block Diagram



13.3.1.1 Interrupt Generation Control

The interrupt generation control block collects information (FIFO underrun/overrun, FIFO threshold) from the transmit/receive control block and provides the value to the register block. If the interrupt condition matches, the SSPn Interrupt signal will be asserted.

This block also generates the DMA request signals, if necessary.

13.3.1.2 Serial Clock Generator

This block generates a serial clock for communication. The format of the serial clock is defined through the control register 0. Generally, the clock will not start if the SSP controller is not enabled. Once the SSP controller is enabled and the clock running condition is matched (For example, when SPI is specified and the transmit FIFO is not empty, or when I2S is specified), the serial clock will start running. The operating frequency of the serial clock depends on the setting of the SCLKDIV register.

This block also sends the serial clock and SEL/WS. information to the data transmit/receive control block.

13.3.1.3 Transmit/Receive Control

The main function of this block is to perform the parallel-to-serial transmission or handle the serial-to-parallel reception from the external devices.

If the master mode is specified and the transmit FIFO is not empty, the data in the transmit FIFO will be read and shifted out via the data output pin. If a whole word is completely shifted out and the transmit FIFO still contains valid data, the next data will be read and shifted out. The received data can be shifted in via the data input pin. Once the reception is complete, the received word will be written into the receive FIFO, and the receive control logic will continue to receive the next word. If the half-duplex protocol is specified, the receive control logic will not start until the transmission is completed.

If the slave mode is specified, the SSP controller will start to transmit/receive data when SEL/WS is activated. The transmission or reception will continue until the SSP controller is disabled or SEL/WS.

is deactivated. If the half-duplex protocol is specified, the transmit control logic will not start until the reception has been completed. The transmission and reception can be activated simultaneously when the full-duplex protocol is specified.

13.4 INTERFACE DESCRIPTION

13.4.1 Motorola 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 SCLKPO and SCLKPH bits in SPIn_CTRL0 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.

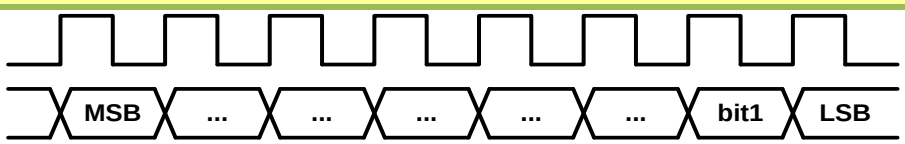
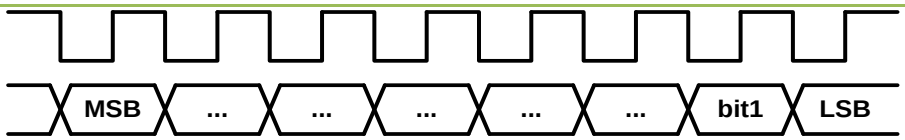
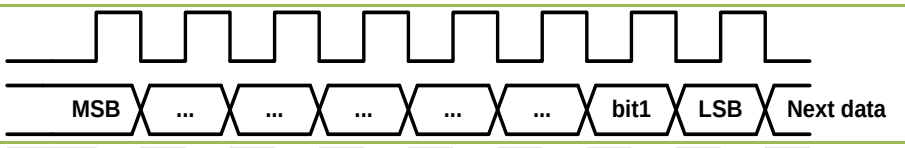
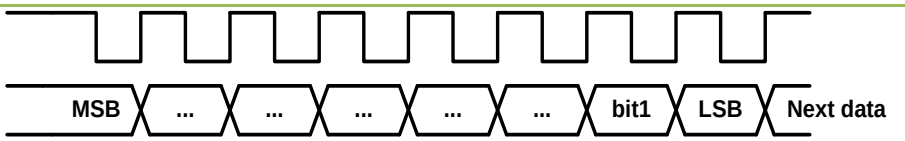
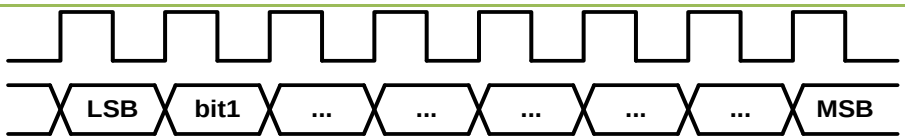
If SCLKPO and SCLKPH have the same value, the transmit logic will transmit data at the falling edge of SCLK, and the receive logic will latch data at the rising edge of SCLK. If SCLKPO and SCLKPH have the opposite values, the transmit logic will transmit data at the rising edge of SCLK, and the receive logic will receive data at the falling edge of SCLK.

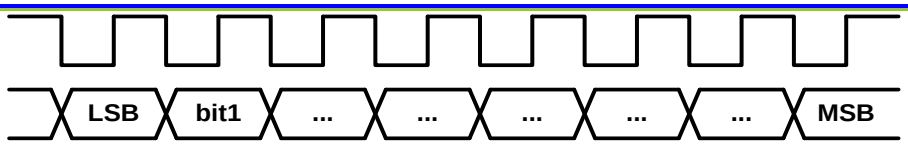
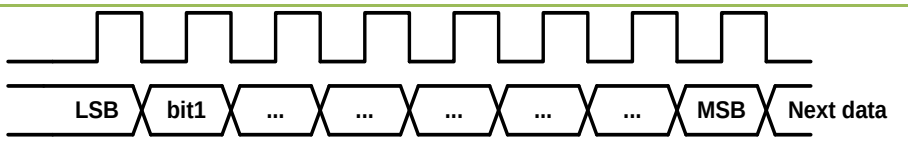
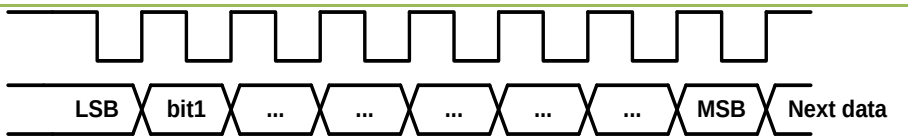
In the master mode, if the transmit FIFO contains data and SPI is enabled, the transmit logic will read data in the transmit FIFO and shift it out at each transmit edge (Depending on SCLKPO and SCLKPH). The receive logic will receive data at the same time. After LSB is transmitted or received, SEL will hold low for half or one SCLK cycle, depending on SCLKPH, and then pull high if there is no data in the transmit FIFO. The received data will be written to the receive FIFO. If data are still available in the transmit FIFO, the transmit logic will restart to transmit data.

In the slave mode, if SEL is activated and SCLK starts running, the transmit logic will try to shift the data from the transmit FIFO even if the transmit FIFO is empty. The transmit FIFO underrun interrupt will be issued if the transmit FIFO is enabled. The receive logic will start receiving data at the same time, and the received data will be written to the receive FIFO.

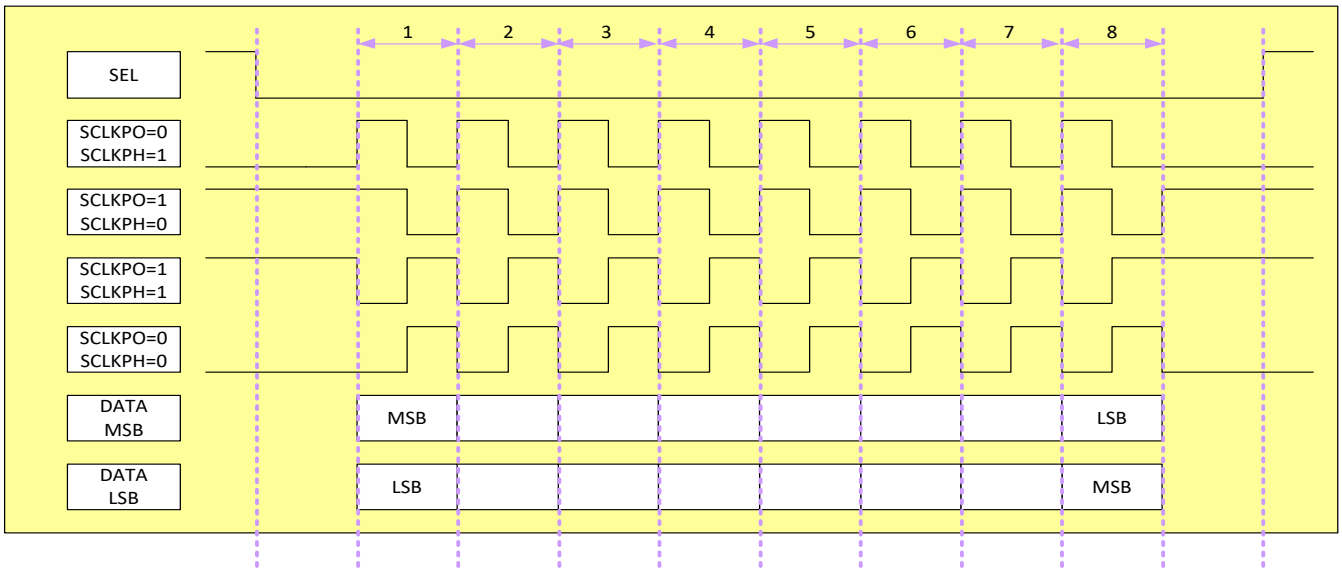
The bit-shifting sequence depending on the LSB setting in control register 0 is from MSB or LSB regardless of the controller is in the master or slave mode. SPIFSPO in control register 0 decides the polarity of the SEL signal.

The SPI data transfer timing as following figure:

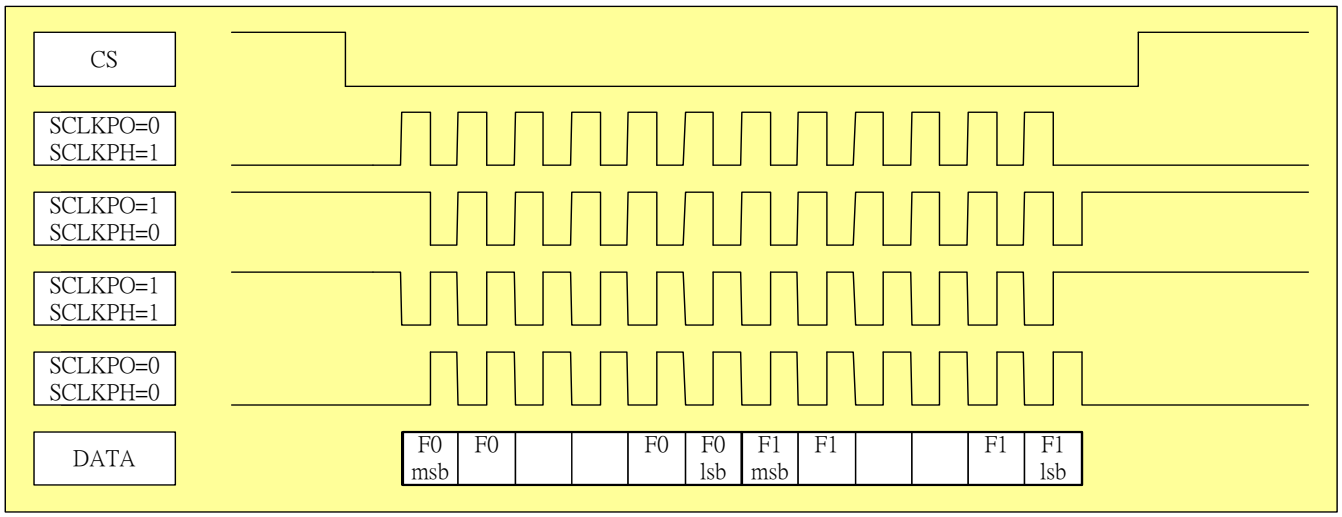
LSB	SCLKPO	SCLKPH	SCK Idle Status	Diagrams
0	0	1	Low	
0	1	1	High	
0	0	0	Low	
0	1	0	High	
1	0	1	Low	

1	1	1	High	
1	0	0	Low	
1	1	0	High	

13.4.1.1 SINGLE-FRAME



13.4.1.2 MULTI-FRAME



13.4.1.3 Continuous Transfer

For a continuous transfer, SCLKPH decides SEL between each successive transfer. When SSPCR0 SPICONTX bit is specified as logic 0 and SCLKPH is specified as logic 0, SEL must be de-asserted and reasserted between each successive transfer as shown in Figure 13-1. When SCLKPH is specified as logic 1, SEL may remain active low between each successive transfer as shown in Figure 13-2. When the SSPCR0 SPICONTX bit is specified as logic 1 and SCLKPH is specified as logic 0, SEL may remain active low between each successive transfer as shown in Figure 13-3.

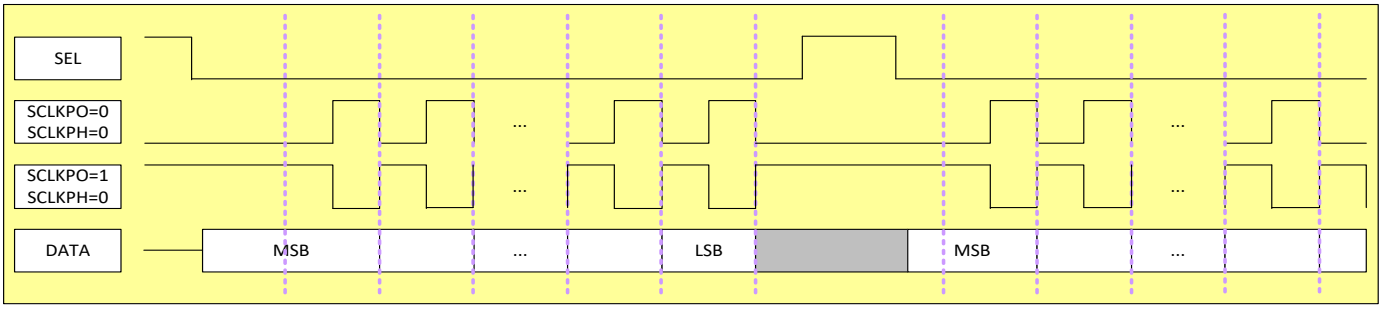


Figure 13-1. Continuous Transfer Using Motorola SPI Frame Format (SCLKPH = 0, SPICONTX = 0)

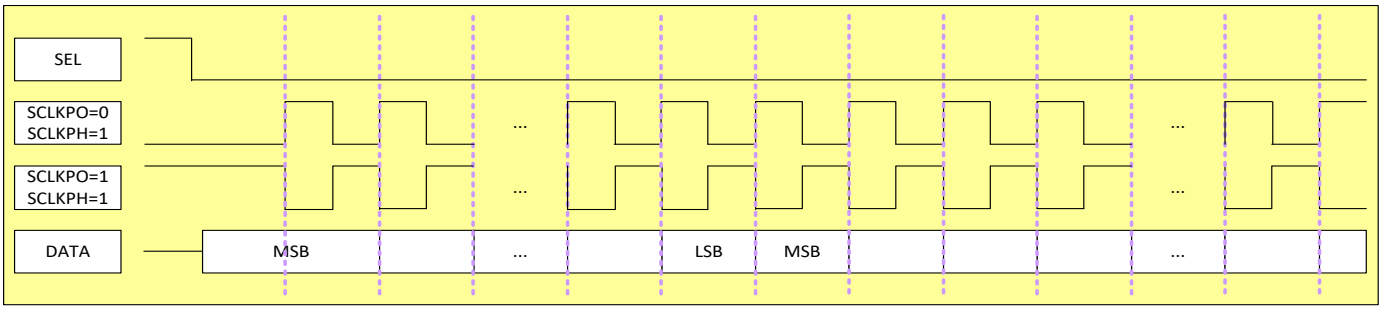


Figure 13-2. Continuous Transfer Using Motorola SPI Frame Format (SCLKPH = 1)

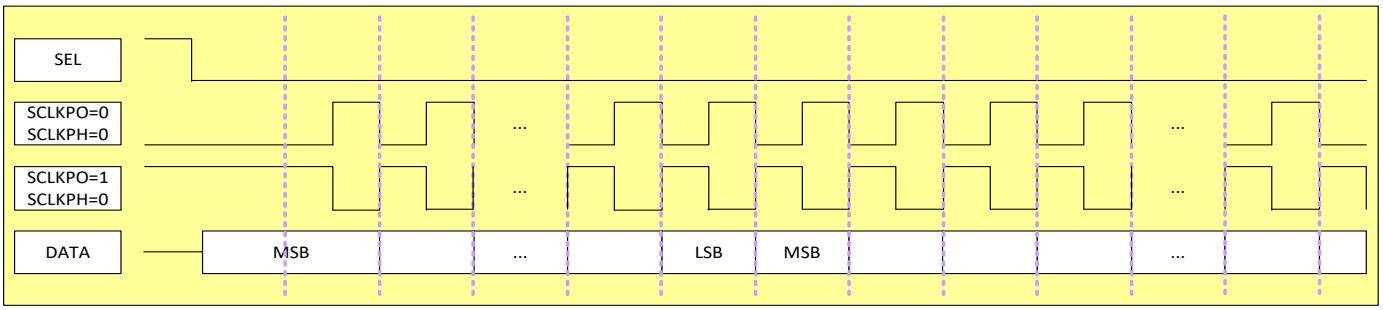


Figure 13-3. Continuous Transfer Using Motorola SPI Frame Format (SCLKPH = 0, SPICONTX = 1)

In the master mode, PCL of SSP control register 3 controls the slave select de-assertion time between transactions as shown in Figure 13-4. The de-assertion time is equal to (PCL+1)*SCLK.
For PCL = 0 and SCLKPH = 1, a continuous transaction will occur if TXFIFO is filled prior to enabling TXEN.

In the SPI frame format, SCLK will stop toggling when it is idle, and SEL is often used as a chip-select signal.

In the SPI master mode and RX (MI pin) only mode, the controller will reserve one Serial data length (SDL) size to avoid RX data lost. The controller will stop receiving RX data when receive FIFO empty entry is less than/equal to round up (SDL/32). Then, the SEL will not be enabled. For example, if SDL is 32 bits, and FIFO depth is 32, the controller will stop receive RX data when receive FIFO empty entry equals to 1. Then, the SEL will not be enabled.

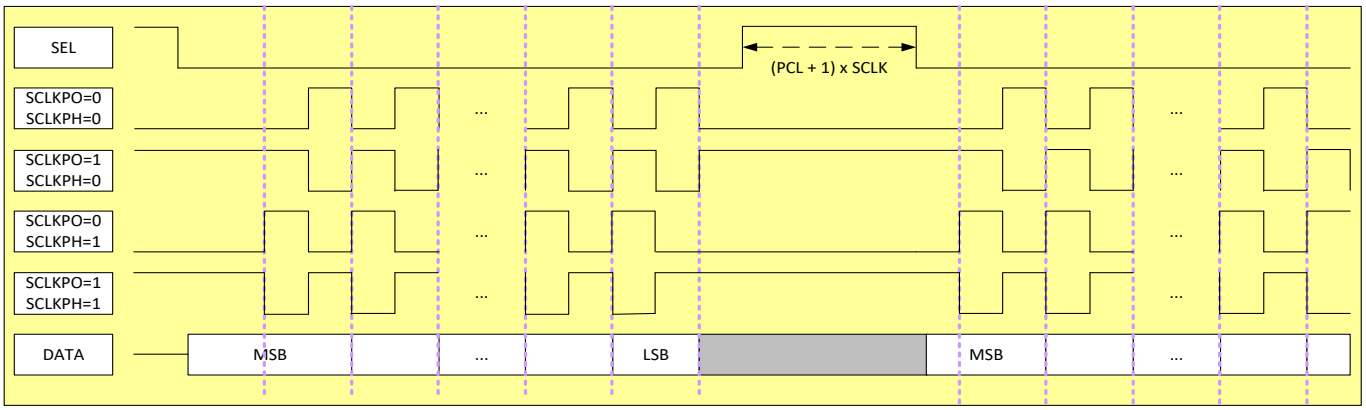


Figure 13-4. Continuous Transfer with Padding Cycle Length

13.4.2 National Semiconductor MICROWIRE

The MICROWIRE frame format of National Semiconductor is similar to the SPI frame format of Motorola, except for the half-duplex transfer. During a transmission, the receive logic will be halted until a complete word is transmitted. When the transmission and reception hands over, an extra cycle is inserted (Please refer to Figure 13-5). Data will be transmitted at the falling edge of SCLK and the received data will be latched at the rising edge of SCLK.

Since the transmitting and receiving data lengths may be different, SSP controller provides independent data length programming for transmission and reception.

The SSP controller, when in the frame format of National Semiconductor MICROWIRE, defines two phases. In the master mode, the first phase is the transmitting phase. If SSP is enabled and the transmit FIFO contains data, the transmit logic will shift the data out from the transmit FIFO to the slave devices depending on the Padding Data Length (PDL) specified in control register 1. The receive logic will be halted first. After the transmission is complete, the SSP controller will check the Serial Data Length (SDL) specified in control register 1. If SDL is specified as a non-zero value, the SSP controller will enter the second phase to receive data and halt the transmit logic. When the reception is complete, the SSP controller returns to the first phase to transmit data if there are existing data in the transmit FIFO.

In the slave mode, the first phase is the receiving phase. If SSP is enabled and SCLK starts running while the SEL is activated, the receive logic starts to receive data according to the Padding Data Length (PDL) specified in control register 1. After the reception is complete, the SSP controller will check the Serial Data Length (SDL) specified in control register 1. If SDL is specified as a non-zero value, the SSP controller will enter the second phase to transmit data and halt the receive logic. When the transmission is complete, the SSP controller will return to the first phase to receive data if SCLK and SEL are active.

Figure 13-5 shows the transfer format of National Semiconductor MICROWIRE. FSPO of control register 0 is set to ‘1’.

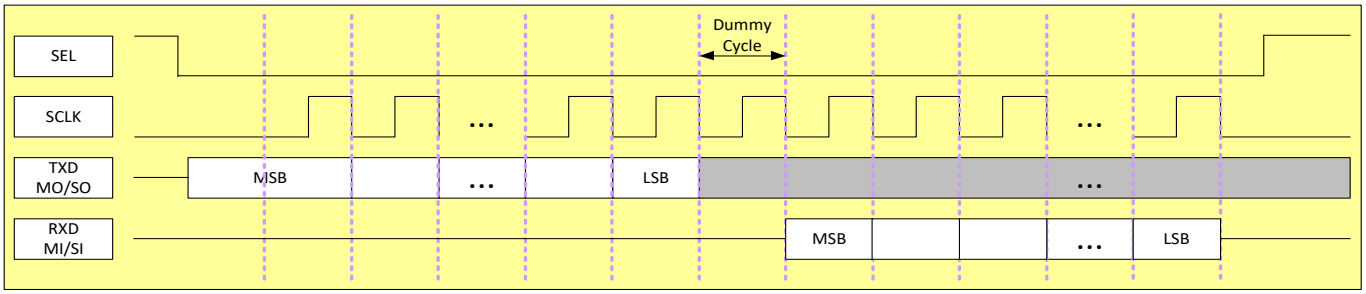


Figure 13-5. Single Transfer Using National Semiconductor MICROWIRE Frame Format

If there are still data in the transmit FIFO, the transmit logic will continue to transmit data to TXD and receive data from RXD. Figure 13-6 shows the frame format of a continuous transfer.

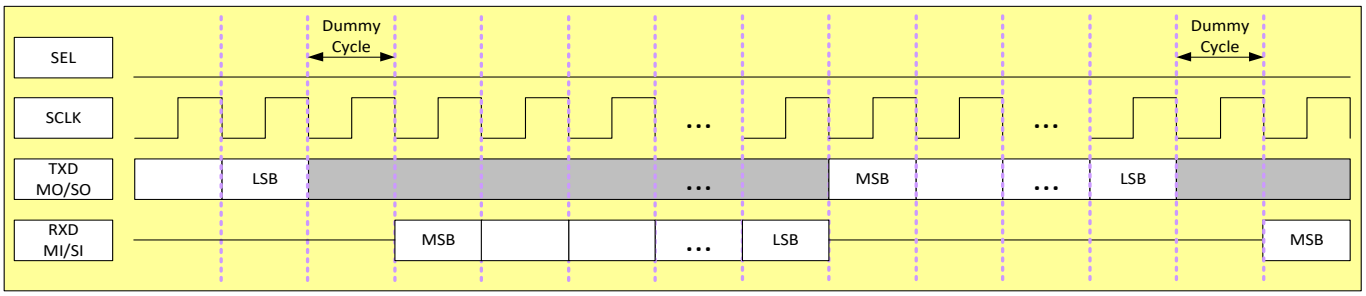


Figure 13-6. Continuous Transfer Using National Semiconductor MICROWIRE Frame Format

If the National Semiconductor MICROWIRE frame format is specified, the bit shifting sequence is defined from MSB to LSB. The LSB setting in control register 0 will be ignored. SCLKPH and SCLKPO will be ignored as well. The polarity of SEL is user-specified.

13.4.3 Philips I2S

The frame format for Inter-IC Sound (I2S) of Philips is used for the serial audio applications. The WS signal is used for channel selection. The SSP controller in I2S frame format provides either the stereo or mono mode. If the stereo mode is specified, WS will be changed when a complete word is transmitted or received. FSPO in control register 0 specifies the first channel. If FSPO is specified as logic 1, the first data word will be transmitted or received when WS is set to low. If FSPO is specified as logic 0, the first data word will be transmitted or received when WS is set to high. If the mono mode is specified, WS will also be changed when a complete word is transmitted as well. However, data will be retrieved or stored only when WS is inverted to FSPO, as specified in control register 0.

The SSP controller in the I2S frame format can act either in the receiving mode, or in the recording mode, or both modes. If Transmit Data Output Enable (TXDOE) in control register 2 is set to '1', SSP can simultaneously transmit data in the transmit FIFO via the DOUT pin and receive data to be latched into the receive FIFO via the DIN pin. If TXDOE is set to logic 0, SSP will be in the recording mode only; that is, no data in the transmit FIFO will be read and shifted out when SSP is enabled.

- As shown in Figure 13-7, FSDIST in control register 0 is set to '1'. The first bit of data will be transmitted/ received after one serial clock cycle when the channel-select is changed.

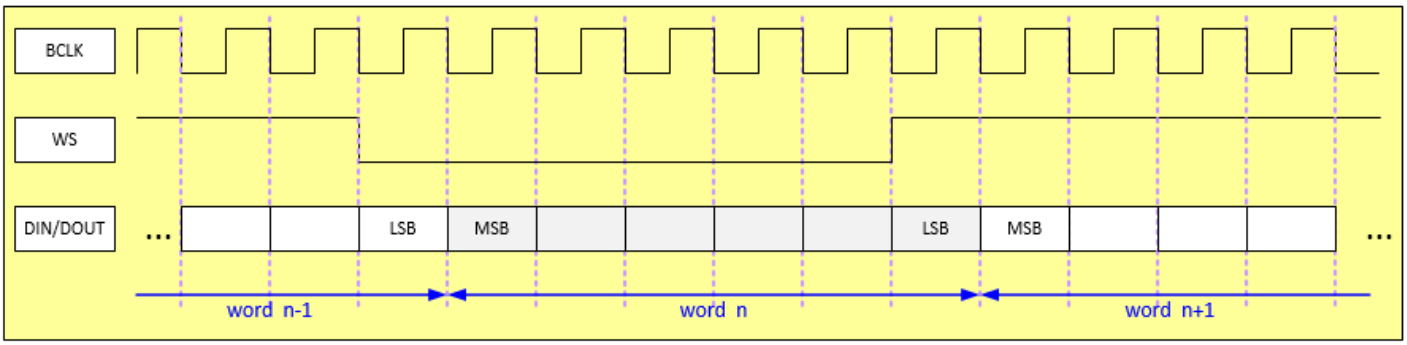


Figure 13-7. Philips I2S Basic Frame Format

- Figure 13-8 shows one variation of the I2S frame format. In this case, the width of the WS. is larger than the serial data bits. The valid data bit is right-justified to the whole frame.
  The extra padding data bits are inserted before the valid data bits. The number of the padding data bits is specified in PDL of control register 1. In this case, FSDIST is set to '0'.

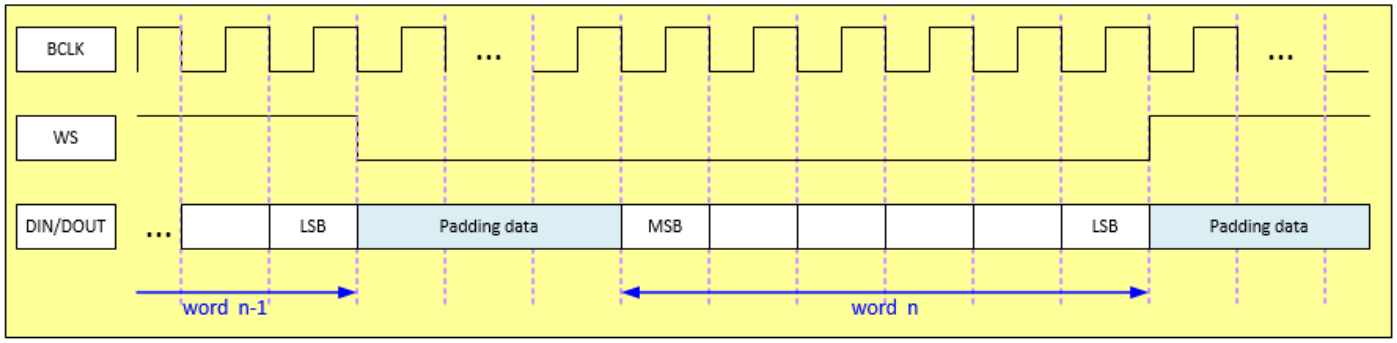


Figure 13-8. Possible Variation of I2S Frame Format (Right-justified)

- Figure 13-9 shows another variation of the I2S frame format. In this case, the width of the WS. is also larger than the serial data bits; however, the valid data bit is left-justified to the whole frame. The extra padding data bits are inserted after the valid data bits. The number of the padding data bits is specified in PDL in control register 1, and the first bit of the serial data is valid at the first serial clock after the change of WS. (FSDIST is set to '0').

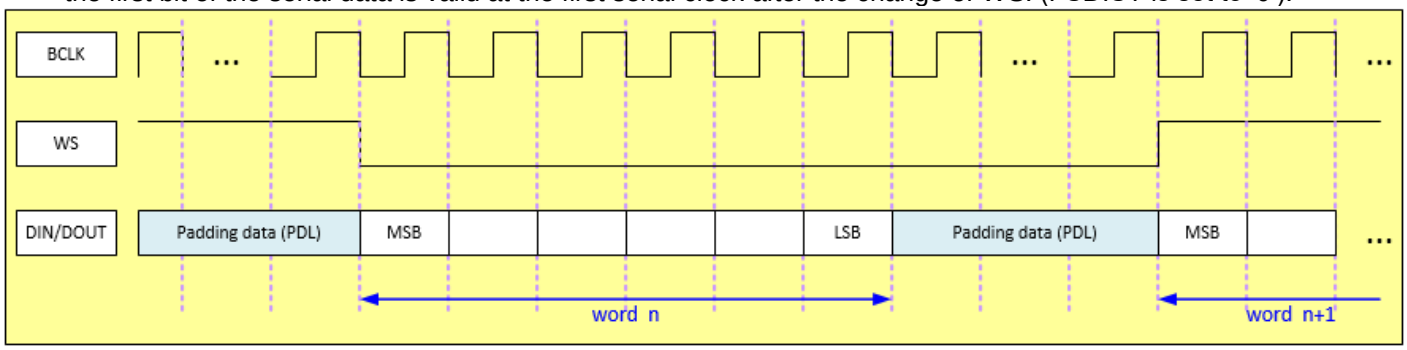


Figure 13-9. Possible Variation of I2S Frame Format (Left-justified)

Figure 13-10 and Figure 13-11 show the different padding data length on the right/left channel. The I2S frame format supports the different padding data length when DPDLEN in control register 3 is set to '1'.

- In Figure 13-10, the width of WS. is larger than the serial data bits. The valid data bit is right-justified to the whole frame. The extra padding data bits are inserted before the valid data bits. The number of the padding data bits is specified in DPDL in control register 3 when WS. equals to zero. The number of the padding data bits is specified in PDL in control register 1 when WS. equals to one.

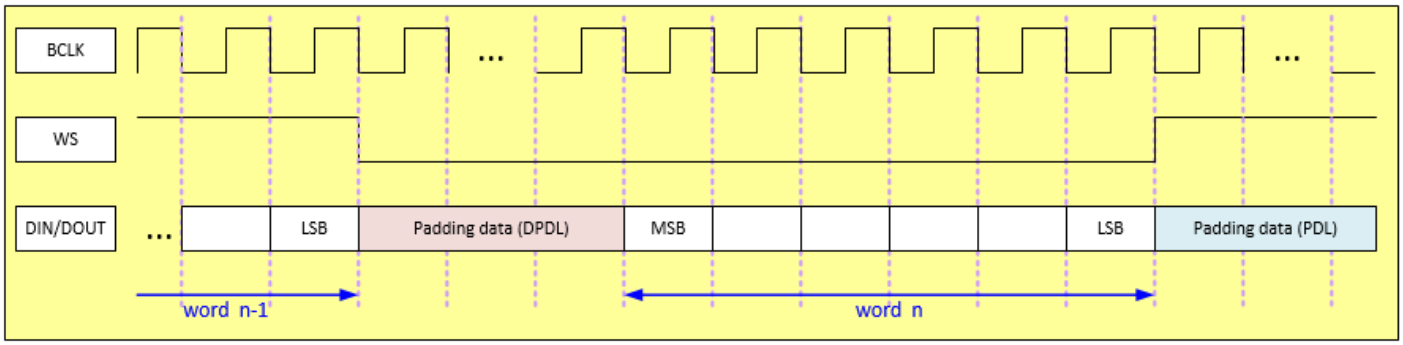


Figure 13-10. Possible Variation with DPDL of I2S Frame Format (Right-justified)

- In Figure 13-11, the width of WS. is larger than the serial data bits. The valid data bit is left-justified to the whole frame. The extra padding data bits are inserted after the valid data bits. The number of the padding data bits is specified in DPDL in control register 3 when WS. equals to zero. The number of the padding data bits is specified in PDL in control register 1 when WS. equals to one.

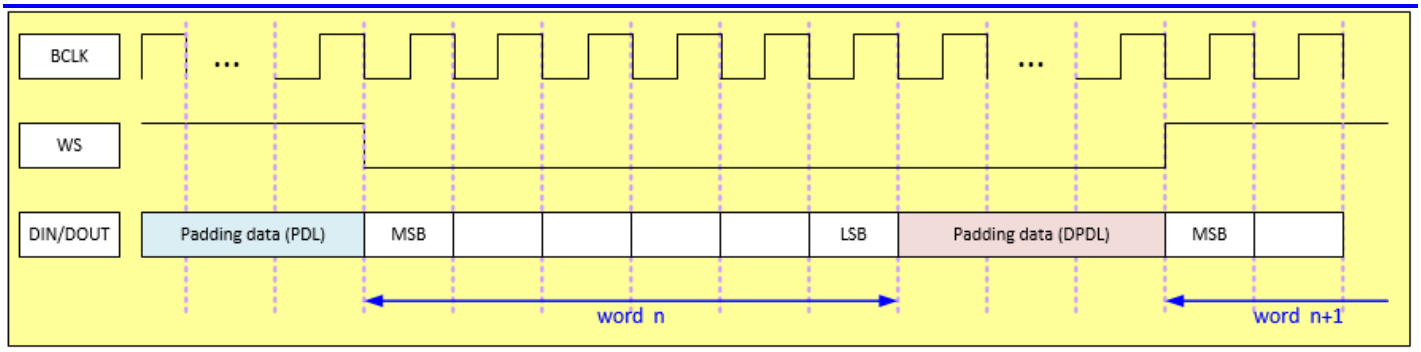


Figure 13-11. Possible Variation with DPDL of I2S Frame Format (Left-justified)

13.5 SSP Serial Clock

The frequency of the internally-generated serial clock is controlled by SCLKDIV in the control register 1. The frequency of the internally-generated serial clock is determined by the following formula:

$$F_{SCLK} = \frac{F_{SSPCLK}}{2 \times (SCLKDIV + 1)}$$

When data are transmitted or received according to the SCLK frequency, the bit rate of SSP will be equal to the frequency of SCLK. If the externally-generated serial clock is specified, SCLKDIV will be ignored and all the transfers will depend on the externally-supplied SCLK. Because SSP has to synchronize the externally-generated SCLK with SSPCLK, the frequency of the externally-generated SCLK is expected to be slower than SSPCLK by at least six times. SSPCLK needs 50%±5 duty cycle to synchronize the externally-generated SCLK. The table below shows the settings of the SCLKDIV signal and the ratios of SSPCLK and SCLK in the master mode and slave mode.

Mode	Minimum ratio of SSPCLK/SCLK	SCKDIV setting
Master mode	4 (SPI/ MICROWIRE) 6 (I2S)	By register setting
Slave mode	6	Don't care

✷ ***Note: The minimum ratio of SPI1 and SPI2 is 4, but the master is limited to a maximum speed of 36Mbps.***

Please note that the synchronizer from SSPCLK to PCLK needs three PCLK cycles. To ensure that the data are stable, the clock period of SCLK (Tsclk) and the clock period of PCLK (Tpclk) should meet the following formula:

$$(3.5 \times T_{PCLK}) < (T_{SCLK} \times SDL\_Length)$$

13.6 FIFO READ/WRITE

The transmit FIFO and receive FIFO occupy the same address space. When the data register is read, the data from the receive FIFO will be retrieved. When the data register is written, the written data will be stored in the transmit FIFO.

If the I2S protocol is specified, a FIFO overrun/underrun condition may take place because the FIFO cannot be filled in time. If this exception occurs, there might be an unexpected result. If I2S is specified, the channel data sent to the codec may interchange from left to right or from right to left. Therefore, users have to reset the codec and restart the audio data transmission.

13.7 SSP Registers

Base Address: 0x4002 A000 (SPI0)
0x4002 B000 (SPI1)
0x4003 1000 (SPI2)

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	SPIn_CTRL0	SPI Control register 0
0x0004	SPIn_CTRL1	SPI Control register 1
0x0008	SPIn_CTRL2	SPI Control register 2
0x000C	SPIn_STATUS	SPI Status register
0x0010	SPIn_ICTRL	SPI Interrupt Control register
0x0014	SPIn_RIS	SPI Raw Interrupt Status register
0x0018	SPIn_DATA	SPI Transmit/Receive Data register
0x001C	SPIn_CTRL3	SPI Control register 3

13.7.1 SPI n Control register 0 (SPIn_CTRL0) (n=0,1,2)

Address Offset:0x00

✷ ***Note: 1. If FFMT=0 (invalid function), continuous SCK will be generated, please select standard settings (1: Motorola SPI; 2: NI MICROWIRE; 3: Philips I2S) before SSP is enabled.***
2. If user want to set a single bit, please refer to the steps below:
Step1. First, Enable SN_SCU->APB1CLKG_b.SSPxCLKEN.
Step2. Select SN_SPIx->CTRL0_b.FFMT.
Step3. User can setting others bit for SPI register.

Bit	Field	Access	Initial	Description
31:20	–	–	0	Reserved
19	SPICONTX	RW	0	SPI Continuous Transfer control 0: SEL may assert when the Transmit FIFO is not empty 1: Continuously transfer without asserting SEL when the Transmit FIFO is not empty
18	FLASHTX	RW	0	FLASH mode transmit control 0: SCLK will still trigger when no transmit data in FLASH mode until the RX FIFO is full 1: SCLK will stop when no transmit data in FLASH mode
17	FSFDBK	RW	0	Select the mode of SEL/WS 0: Slave mode 1: Master mode
16	SCLKFDBK	RW	0	Select the mode of SCLK/BCLK 0: Slave mode 1: Master mode
15	SPIFSPO	RW	0	SEL polarity for SPI mode 0: Active low 1: Active high
14:12	FFMT	RW	0	Frame format 1: Motorola SPI 2: NI MICROWIRE 3: Philips I2S Other: Reserved
11	FLASH	RW	0	SPI Flash mode (Base on Motorola SPI format) When RXEN is enabled and FLASHTX is set to '0', the controller will continuous receive data until the RX FIFO is full. When TXEN and RXEN are enabled, and FLASHTX is set to '1', the controller will continuous transmit/receive data until TX

Bit	Field	Access	Initial	Description
				FIFO is empty or RX FIFO is full. 0: Disable SPI Flash mode 1: Enable SPI Flash mode. The SEL will control by FS bit of CTRL2 register
10	–	–	0	Reserved
9:8	FSDIST	RW	1	WS and data distance for I2S mode These bits define the relationship between the WS. and the data bits. If these bits are set to zero, the first bit of serial data will be valid in the first cycle of WS. If non-zero is specified, FSDIST will define the number of SCLK cycles between the first bit of the valid received data and the frame start. For standard I2S frame format, these bits must be set to '1'. 0: The first bit of serial data will be valid in the first cycle of WS 1: 1 SCLK cycle between the first bit of the valid received data and the frame start 2: 2 SCLK cycles between the first bit of the valid received data and the frame start 3: 3 SCLK cycles between the first bit of the valid received data and the frame start
7	LBM	RW	0	Loopback mode 0: Operate in the normal mode and the transmitted/received data will be independent 1: The transmitted data will be connected to the received data internally
6	LSB	RW	0	Bit sequence indicator 0: MSB of the data will be transmitted or received first 1: LSB of the data will be transmitted or received first Please note that when SSP is in I2S master mode or SPI master mode, user wants to transmit the least significant bit (LSB) of a data word first, register 0x08 TXDOE must set to 1.
5	FSPO	RW	0	SEL polarity for MICROWIRE mode or WS polarity for I2S mode 0: Active high 1: Active low
4	FSJSTFY	RW	0	Data justify for I2S mode 0: The number of zeros specified in PDL will be appended in the back of the serial data 1: The padding data will be in front of the serial data
3:2	OPM	RW	0x3	Operation mode 0: Slave mode for SPI/MICROWIRE mode, slave mono mode for I2S mode 1: Slave mode for SPI/MICROWIRE mode, slave stereo mode for I2S mode 2: Master mode for SPI/MICROWIRE mode, master mono mode for I2S mode 3: Master mode for SPI/MICROWIRE mode, master stereo mode for I2S mode
1	SCLKPO	RW	0	SCLK polarity for SPI mode 0: SCLK will remain Low when SPI is idle 1: SCLK will remain High when SPI is idle
0	SCLKPH	RW	0	SCLK phase for SPI mode 0: SCLK will start toggling after one SCLK cycle time when SEL is activated 1: SCLK will start running after half an SCLK cycle time when SEL is activated

13.7.2 SPI n Control register 1 (SPIn_CTRL1) (n=0,1,2)

Address Offset: 0x04

If I2S mode is specified, PDL will be used to define the bit length of the padding bits in front/back of a data word. If PDL is set to '0', no padding bits will be inserted. If a non-zero value is specified, the actual number of zeroes will be inserted/appended in front/back of the data word.

If MICROWIRE mode is specified, PDL will define the bit length of the first phase (Transmit in the master mode and receive in the slave mode), and the actual data length will be equal to this register plus one. The maximum value of this register should not exceed 31. The minimum value of this register should be more than one.

Bit	Field	Access	Initial	Description
31:24	PDL	RW	0	Padding data length for I2S/MICROWIRE mode In the Philips I2S frame format, the actual data length is equal to these bits. In MICROWIRE frame format, the actual data length equals to these bits plus one. The padding data bits can be 0 or 1. Please just ignore it.
23	–	–	0	Reserved

Bit	Field	Access	Initial	Description
22:16	SDL	RW	0x7	Serial data length These bits define the bit length of a transmitted/received data word. The actual data length equals to these bits plus one. The minimum value should not be fewer than four. If MICROWIRE mode is specified, these bits will define the bit length of the second phase (Receive in the master mode or transmit in the slave mode).
15:0	SCLKDIV	RW	0x8000	SCLK divider

13.7.3 SPI n Control register 2 (SPIn_CTRL2) (n=0,1,2)

Address Offset: 0x08

✸ ***Note: To stop SPI SCK communication, please set SPIEN = 0 first, and then set TX/RXEN = 0.***

Bit	Field	Access	Initial	Description
31:10	–	–	0	Reserved
9	FS	RW	0	SEL output level for SPI Flash mode 0: SEL output will be LOW 1: SEL output will be HIGH
8	TXEN	RW	0	Transmit function enable bit 0: Disable 1: Enable
7	RXEN	RW	0	Receive function enable bit 0: Disable 1: Enable
6	RESET	RW	0	The software reset of the state machine For the slave mode of the SPI controller, the serial clock will be controlled by the external master and will be stopped under certain circumstances, and will cause the SPI state machine to halt. Writing 1 to this bit puts the SPI controller back to the idle state when carrying out the following operations. 0: No effect 1: Reset the state machine even if the normal transmission or reception is in progress
5:4	–	–	0	Reserved
3	TXFCLR	RW1C	0	TX FIFO clear bit 0: No effect 1: Clear TX FIFO
2	RXFCLR	RW1C	0	RX FIFO clear bit 0: No effect 1: Clear RX FIFO
1	TXDOE	RW	1	Transmit data output enable bit In the multi-slave system, it is possible for an SPI master to broadcast a message to all slaves within the system while ensuring that only one slave drives data onto its serial output line. In such a system, the rxd line from multiple slaves can be tied together. To operate in such a system, TXDOE may be cleared if the SPI slave is not supposed to drive the txd line. (1) When TXDOE is set, SPI will drive data on the transmit data line. When TXDOE is not set, SPI will not drive data on the transmit data line. (2) If the I2S frame format is specified, this bit will define either the receiving (Recording) mode (0) or the simultaneous transmitting/receiving (Playing/Recording) mode (1). (3)If the SPI frame format and slave mode are specified, TXDOE should also be set to high. (4)When LSB is programmed to 1, TXDOE must 1;otherwise, LSB will be invalid 0: Disable 1: Enable
0	SPIEN	RW	0	SPI/I2S enable bit 0: Disable

Bit	Field	Access	Initial	Description
				1: Enable

13.7.4 SPI n Status register (SPIn_STATUS) (n=0,1,2)

Address Offset: 0x0C

This register shows the raw status before interrupt is enabled.

Bit	Field	Access	Initial	Description
31:18	–	–	0	Reserved
17:12	TFVE	R	0	The number of entries in the transmit FIFO waiting to be transmitted
11:10	–	–	0	Reserved
9:4	RFVE	R	0	The number of entries in the receive FIFO waiting for DMA or CM4 processor to read them
3	–	–	0	Reserved
2	BUSY	R	0	Busy Indicator 0: SPI is idle 1: SPI is transmitting and/or receiving data
1	TFNF	R	1	Transmit FIFO Not Full 0: FIFO is completely full 1: The transmit FIFO is available for DMA or CM4 processor to write
0	RFF	R	0	Receive FIFO Full 0: The receive FIFO is not full 1: The receive FIFO is full

13.7.5 SPI n Interrupt Control register (SPIn_ICTRL) (n=0,1,2)

Address Offset: 0x10

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

Bit	Field	Access	Initial	Description
31:19	–	–	0	Reserved
18	TXCIEN	RW	0	Transmit Data Complete Interrupt Enable 0: The interrupt will be disabled 1: The interrupt will take place when transmit is done and TX valid entry is empty
17	RFTHOD_UNIT	RW	0	Receive FIFO Threshold Unit 0: RX FIFO threshold = RFTHOD 1: RX FIFO threshold = RFTHOD + 1
16:15	–	–	0	Reserved
14:12	TFTHOD	RW	0x2	Transmit FIFO Threshold. If the valid data in the transmit FIFO is equal to or less than the actual threshold, the DMA request and/or the interrupt will be asserted. If this bit is set to ‘0’, the interrupt will be disabled.
11:10	–	–	0	Reserved
9:7	RFTHOD	RW	0x2	Receive FIFO Threshold. If the valid data in the receive FIFO is equal to or greater than the actual threshold, the DMA request and/or the interrupt will be asserted. If this bit is set to ‘0’, the interrupt will be disabled.
6	–	–	0	Reserved
5	TFDMAEN	RW	0	Transmit DMA Request Enable 0: No DMA request will be issued 1: The DMA request will be issued when the transmit FIFO threshold is hit
4	RFDMAEN	RW	0	Receive DMA Request Enable 0: no DMA request will be issued 1: The DMA request will be issued when the receive FIFO threshold is hit
3	TFTHIEN	RW	0	Transmit FIFO Threshold Interrupt Enable 0: Disable this interrupt 1: The interrupt will be issued when the valid entries in the transmit FIFO are less than or equal to the threshold value

Bit	Field	Access	Initial	Description
2	RFTHIEN	RW	0	Receive FIFO Threshold Interrupt Enable 0: Disable this interrupt 1: The interrupt will be issued when the valid entries in the receive FIFO are greater than or equal to threshold value
1	TFURIEN	RW	0	Transmit FIFO Underrun Interrupt Enable 0: The interrupt will be masked even when the transmit FIFO underrun happens 1: The transmit FIFO underrun will assert interrupt
0	RFORIEN	RW	0	Receive FIFO Overrun Interrupt Enable 0: The interrupt will be masked even when the receive FIFO overrun occurs 1: The receive FIFO overrun will assert interrupt

13.7.6 SPI n Raw Interrupt Status register (SPIn_RIS) (n=0,1,2)

Address Offset: 0x14

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

Bit	Field	Access	Initial	Description
31:6	–	–	0	Reserved
5	TXCI	RC	0	Transmit Data Complete Interrupt 0: Cleared after reading 1: When transmit is done and TX valid entry is empty
4	–	–	0	Reserved
3	TFTHI	R	1	Transmit FIFO Threshold Interrupt 0: The valid entries in the transmit FIFO are larger than TFTHOD 1: If the TX FIFO is equal to or less than the threshold
2	RFTHI	R	0	RX FIFO Threshold interrupt 0: Cleared when the condition above is removed 1: The RX FIFO is equal to or greater than the threshold
1	TFURI	RC	0	TX FIFO Underrun Interrupt 0: Cleared until users read this bit 1: If the transmit logic tries to retrieve data from the empty transmit FIFO
0	RFORI	RC	0	RX FIFO Overrun Interrupt 0: Cleared until users read this bit 1: If the receive logic tries to receive data when the receive FIFO is full

13.7.7 SPI n DATA register (SPIn_DATA) (n=0,1,2)

Address Offset: 0x18

The transmitting and receiving FIFOs which are 32-bit wide and can be configured separately. The transmitted and received data occupy the same address space. The write operation writes data to the transmit FIFO and the read operation reads data from the receive FIFO. If the size of a serial data length is less than 32 bits, data will be automatically right-justified during reception or transmission. If the size of a serial data length is larger than 32 bits, the data will be processed from LSB to MSB based on the 32-bit data width during receiving or transmitting data.

Bit	Field	Access	Initial	Description
31:0	DATA	RW	0	SPI Transmit/Receive data

13.7.8 SPI n Control register 3 (SPIn_CTRL3) (n=0,1,2)

Address Offset: 0x1C

This register defines the SPI padding cycle between transactions.

- For PCL > 0, all SPI modes of operation will generate discrete transactions with the (PCL+1) SPI clocks between transactions.
- For PCL = 0 and SCLKPH = 1 (SCLKPH is the bit [0] of SPI control register 0), a continuous transaction will occur if TXFIFO is filled prior to enabling the TXEN bit in CTRL2 register.
- For PCL = 0 and SCLKPH = 0, the multiple discrete transactions will occur.

Bit	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23:16	DPDL	RW	0	Different Padding Data Length for I2S mode only. In the I2S stereo mode, DPDL is used when WS equals to zero. The padding data bits can be 0 or 1. Please just ignore it.
15:13	–	–	0	Reserved
12	DPDLEN	RW	0	Different Padding Data Length enable bit for I2S mode only 0: Disable different padding data length 1: Enable different padding data length
11:10	–	–	0	Reserved
9:0	PCL	RW	0	Padding Cycle Length for SPI mode only. In the master mode, SPI will wait (PCL + 1) SCLK cycles between each successive transfer. PCL must be zero in SPI Flash mode

14 I2C

14.1 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

14.2 FEATURES

The I2C interface complies with the entire I2C specification, supporting the ability to turn power off to the ARM Cortex-M4 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:
  - 1 Programmable I2C Address
  - 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.
- Generation and detection of 7-bit/10-bit addressing and General Call.
- Supports glitch suppression.

14.3 PIN DESCRIPTION

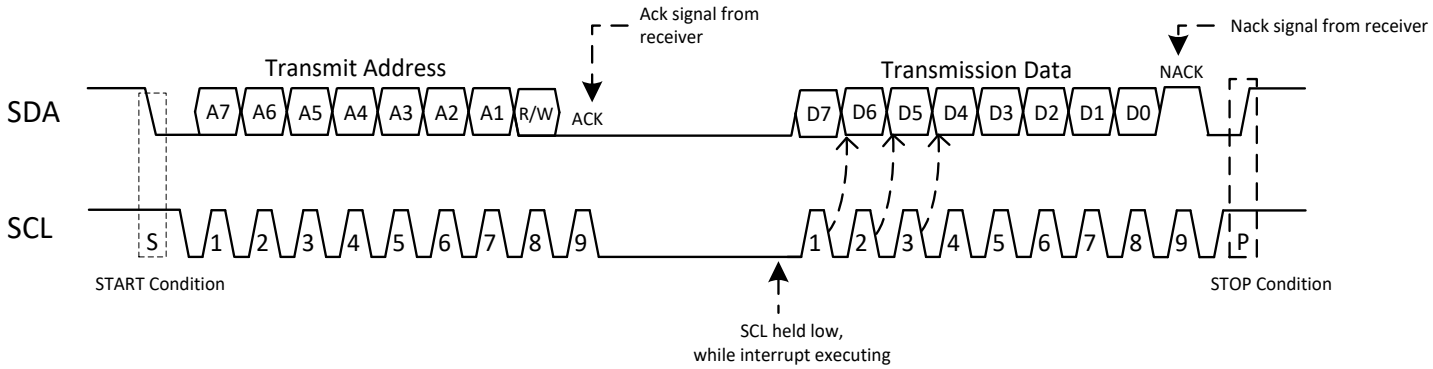
Pin Name	Type	Description	GPIO Configuration
SCLn	I/O	I2C Serial clock	Depends on AFIO register
SDAn	I/O	I2C Serial data	Depends on AFIO register

14.4 I2C PROPOCOL

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.

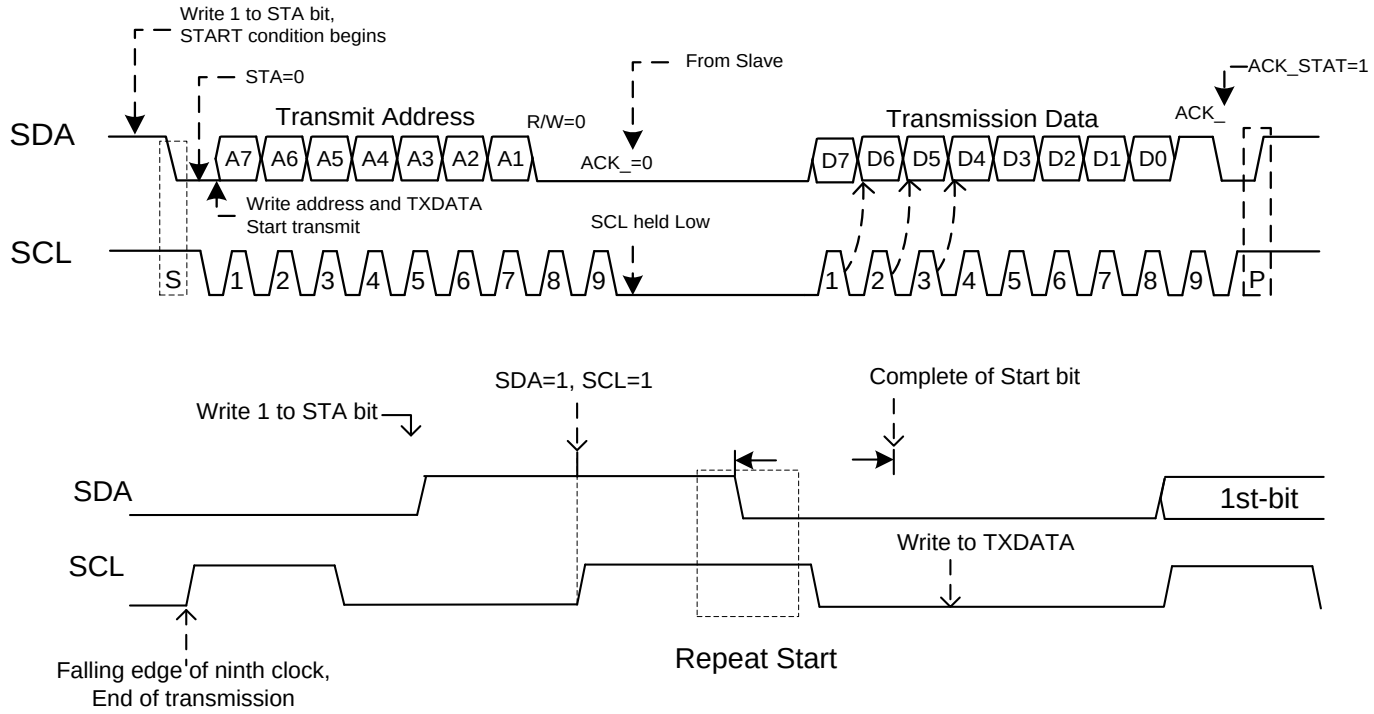
Data is transmitted with the Most Significant Bit (MSB) first. In address byte, the higher 7-bit is address bit and the lowest bit is data direction (R/W) bit. When R/W=0, it assigns a “WRITE” operation. When R/W=1, it assigns a “READ” operation.

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.

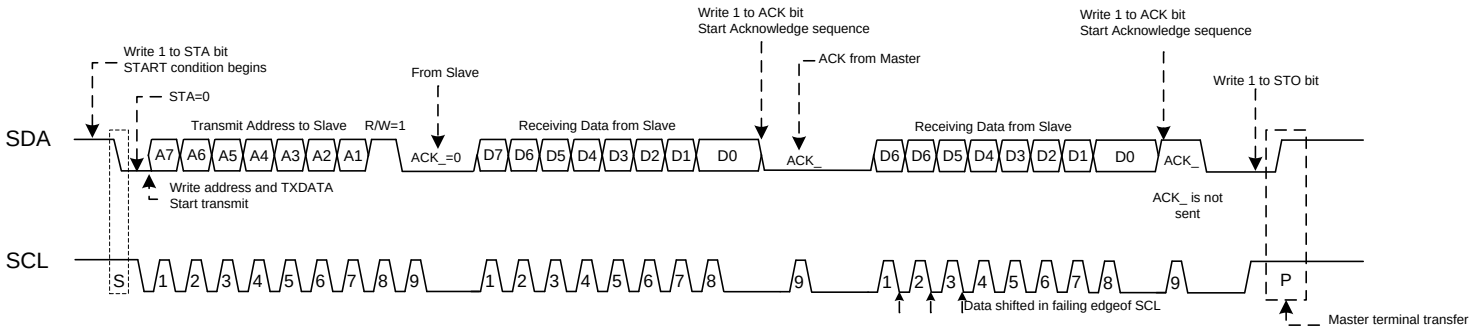


14.4.1 7-BIT ADDRESSING MODES

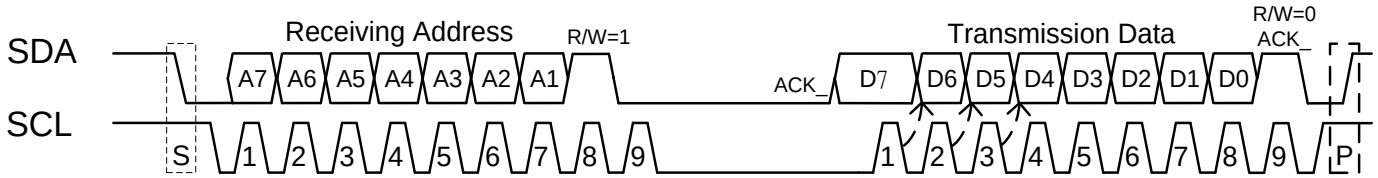
- **MASTER TRANSMITTER MODE**



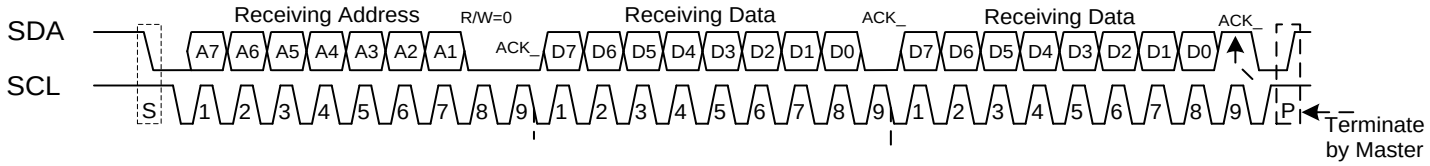
- **MASTER RECEIVER MODE**



- **SLAVE TRANSMITTER MODE**

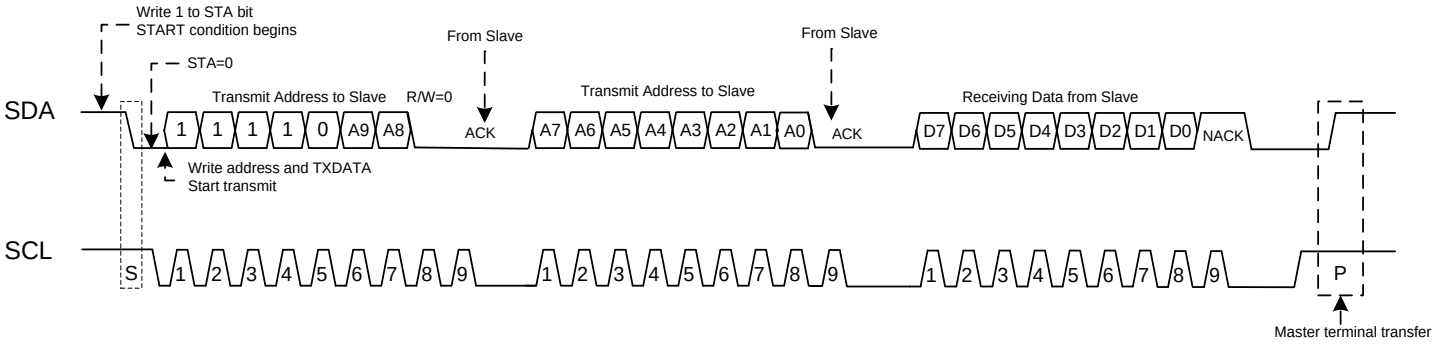


- **SLAVE RECEIVER MODE**

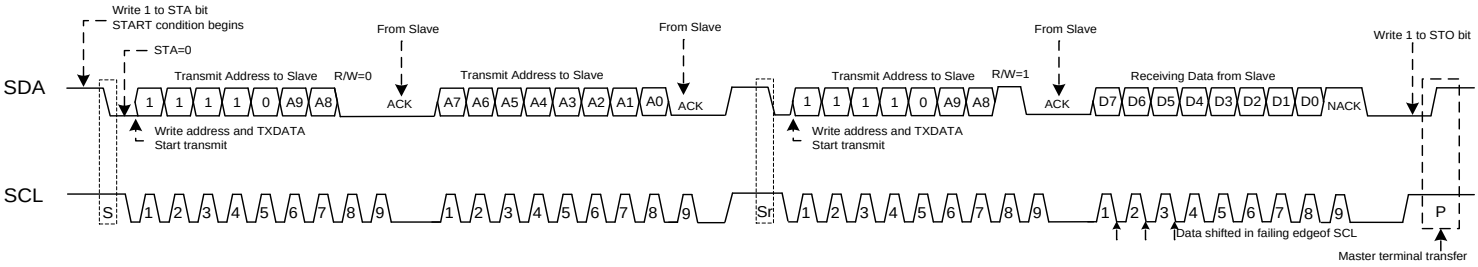


14.4.2 10-BIT ADDRESSING MODES

- **MASTER TRANSMITTER MODE**



- **MASTER RECEIVER MODE**



14.5 ARBITRATION

In the master transmitter mode, the arbitration logic checks that every transmitted logic 1 actually appears as logic 1 on

the I2C-bus. If another device on the bus overrules a logic 1 and pulls the SDA line low, arbitration is lost, and the I2C block immediately changes from master transmitter to slave receiver. The I2C block will continue to output clock pulses (on SCL) until transmission of the current serial byte is complete.

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.

14.6 Glitch Suppression Circuit

Glitch Suppression circuit is used to eliminate the glitch on the I2C bus. Glitches are suppressed according to the GSR internal bus clock period.

14.7 Programming Sequence

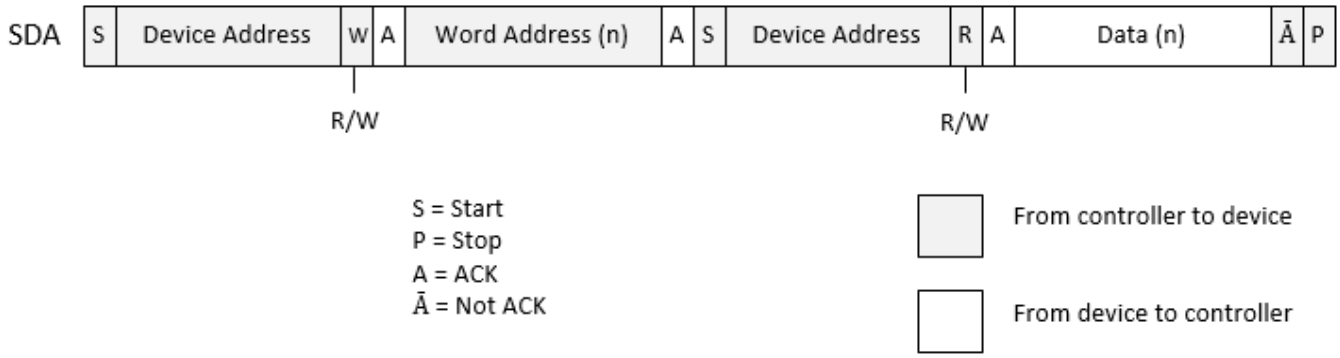
14.7.1 TX/RX Slave Mode

1. Write I2Cn_ADDR: Set the slave address.
2. Write I2Cn_CTRL: Enable all interrupts and I2C enable, and disable MSTEN.
3. Wait for the data receive interrupt - Read I2Cn_STATUS:
   - If RW=1, TD=1, HIT=1, I2CB=1, and NACK=0, then write 1 to clear the corresponding I2Cn_STATUS bits, and go to Step 4.
   - If RW=0, TD=1, HIT=1, I2CB=1, and NACK=0, then write 1 to clear the corresponding I2Cn_STATUS bits, and go to Step 8.
4. Write I2Cn_DATA: Load the data byte to I2Cn_DATA for transfer.
5. Write I2Cn_CTRL: Set the TBEN bit.
6. Wait for the data transmit interrupt - Read I2Cn_STATUS:
   TD=1 or STOP=1, then write 1 to clear the corresponding I2Cn_STATUS bits.
7. If STOP is set, go to the initial step. If STOP is not set, go to Step 4 to load the second data byte.
8. Write I2Cn_CTRL: Set the TBEN bit and set NACK if required.
9. Wait for the data receive interrupt - Read I2Cn_STATUS:
   - If TD=1 and STOP=0: Write 1 to clear the corresponding I2Cn_STATUS bits, then go to Step 10.
   - If TD=0 and STOP=1: Write 1 to clear the corresponding I2Cn_STATUS bits, then go to Step 11.
10. Read I2Cn_DATA: To get data.
11. If STOP is set, go to the initial step. If STOP is not set, go to Step 8 to get the next data.

✷ ***Note: If the STOP interrupt occurs, go back to the initial step.***

14.7.2 RX Slave Mode with Repeat-Start

This section is a special case that slave needs a repeat-start to switch write to read like the EEPROM behavior below.



1. Write I2Cn_ADDR: Set the slave address.
2. Write I2Cn_CTRL: Enable all interrupts and I2C enable, and disable MSTEN.
3. Wait for the data receive interrupt - Read I2Cn_STATUS:
   TD=1, HIT=1, I2CB=1, and NACK=0, then write 1 to clear the corresponding I2Cn_STATUS bits.
4. Write I2Cn_CTRL: Set the TBEN bit and set NACK if required.
5. Wait for the data receive interrupt - Read I2Cn_STATUS:
   - If TD=1 and START=0: Write 1 to clear corresponding I2Cn_STATUS bits, then go to Step 6.
   - If TD=0 and START=1: Write 1 to clear corresponding I2Cn_STATUS bits, then go to Step 7.
   - If TD=1, START=1, and HIT=1: Write 1 to clear corresponding I2Cn_STATUS bits, then go to Step 4 listed in “TX/RX Slave Mode” section above.
   - - If TD=1, START=0, and HIT=1: Write 1 to clear corresponding I2Cn_STATUS bits, then go to Step 4 listed in “TX/RX Slave Mode” section above.
6. Read I2Cn_DATA: To get data.
7. Go to Step 4.

✸ ***Note: If the STOP interrupt occurs, go back to the initial step.***

14.7.3 TX in Master Mode

1. Write I2Cn_CLKDIV: Set the clock count.
2. Write I2Cn_DATA: Target slave address and RW bit for slave.
3. Write I2Cn_CTRL: Enable all interrupts, set MSTEN, set the I2C enable, set the START bit, clear the STOP bit, and set the TBEN bit.
4. Wait for the data transmit interrupt - Read I2Cn_STATUS:
   (a) No arbitration lose: Read I2Cn_STATUS. TD=1, NACK=0, RW=0, AL=0 and BB=0, then write 1 to clear the corresponding I2Cn_STATUS bits. Then, go to Step 5.
   (b) Arbitration lose: Read I2Cn_STATUS. TD=0, NACK=0, RW=0, AL=1 and BB=1, then write 1 to clear the corresponding I2Cn_STATUS bits. Read I2Cn_STATUS until STOP bit turns to 1, then write 1 to clear corresponding STOP bit. After that, go to Step 2.
5.
   (a) If the master wants to stop after one byte of data has been transferred:
   - Write I2Cn_DATA: First data will be sent.
   - Write I2Cn_CTRL: Clear the START bit, set the STOP bit, and set the TBEN bit.
   - Wait for the data transmit interrupt - Read I2Cn_STATUS: TD=1, and AL=0, then write 1 to clear the corresponding I2Cn_STATUS bits.
   - Go to the initial step.

   (b) If the master wants to the send the second data byte transfer after the first data byte transfer:
   - Write I2Cn_DATA: First data will be sent.
   - Write I2Cn_CTRL: Clear the START bit, clear the STOP bit, and set the TBEN bit.
   - Wait for the data transmit interrupt - Read I2Cn_STATUS: TD=1, NACK=0, and AL=0, then write 1 to clear

the corresponding I2Cn_STATUS bits.
- Go to Step 5 or Step 6.

6. If the master wants to send data byte transfer on the same slave or other slaves (i.e. repeat-start): Go to Step 2.

14.7.4 RX in Master Mode

1. Write I2Cn_CLKDIV: Set the clock count.
2. Write I2Cn_DATA: Target slave address and RW bit for slave.
3. Write I2Cn_CTRL: Enable all interrupts, set MSTEN, set the I2C enable, set the START bit, clear the STOP bit, and set the TBEN bit.
4. Wait for the data transmit interrupt - Read I2Cn_STATUS:
   (a) No arbitration lose: Read I2Cn_STATUS. TD=1, NACK=0, RW=1, AL=0 and BB=0, then write 1 to clear the corresponding I2Cn_STATUS bits. Then, go to Step 5.
   (b) Arbitration lose: Read I2Cn_STATUS. TD=0, NACK=0, RW=1, AL=1 and BB=1, then write 1 to clear the corresponding I2Cn_STATUS bits. Read I2Cn_STATUS until STOP bit turns to 1, then write 1 to clear corresponding STOP bit. After that, go to Step 2.
5.
   (a) If the master wants to stop after one byte of data has been received:
   - Writ I2Cn_CTRL: Clear the START bit, set the STOP bit, set the NACK bit, and set the TBEN bit.
   - Wait for the data receive interrupt - Read I2Cn_STATUS: TD=1, NACK=1, then write 1 to clear the corresponding I2Cn_STATUS bits.
   - Read I2Cn_DATA: To get data.
   - Go to the initial step.

   (b) If the master wants to receive the second data byte transfer after the first data byte transfer:
   - Write I2Cn_CTRL: Clear the START bit, clear the STOP bit, and set the TBEN bit.
   - Wait for the data receive interrupt - Read I2Cn_STATUS: TD=1, NACK=0, then write 1 to clear the corresponding I2Cn_STATUS bits.
   - Read I2Cn_DATA: To get data.
   - Go to Step 5 or Step 6.
6. If the master wants to receive the data byte transfer from the same slave or other slaves (i.e. repeat-start): Go to Step 2.

14.7.5 TX in Master Mode with mode 2 (or START byte)

1. Write I2Cn_CLKDIV: Set COUNT for mode 1 and set COUNTH for mode 2.
2. Write I2Cn_DATA: Reserved address and RW bit (START byte: 0x01 or mode 2 master code: 8'000001XXX) for slave.
3. Write I2Cn_CTRL: Enable all interrupts, set MSTEN, set the I2C enable, set the START bit, clear the STOP bit, and set the TBEN bit.
4. Wait for the data transmit interrupt - Read I2Cn_STATUS:
   TD=1, NACK=1, and AL=0, then write 1 to clear the corresponding I2Cn_STATUS bits.
5. Write I2Cn_DATA: Target slave address and RW bit for slave.
6. Write I2Cn_CTRL: Set HSMODE bit to 1 to transfer data in mode 2 by using DUTY_OFFSET and COUNTH, enable all interrupts, set MSTEN, set the I2C enable, set the START bit, clear the STOP bit, and set the TBEN bit.
7. Wait for the data transmit interrupt - Read I2Cn_STATUS:
   TD=1, NACK=0, RW=0, and AL=0, then write 1 to clear the corresponding I2Cn_STATUS bits.
8.
   (a) If the master wants to stop after one byte of data has been transferred:
   - Write I2Cn_DATA: First data will be sent.
   - Write I2Cn_CTRL: Set HSMODE bit, clear the START bit, set the STOP bit, and set the TBEN bit.
   - Wait for the data transmit interrupt - Read I2Cn_STATUS: TD=1, and AL=0, then write 1 to clear the corresponding I2Cn_STATUS bits.
   - Go to the initial step.

   (b) If the master wants to the send the second data byte transfer after the first data byte transfer:
   - Write I2Cn_DATA: First data will be sent.
   - Write I2Cn_CTRL: Set HSMODE bit, clear the START bit, clear the STOP bit, and set the TBEN bit.
   - Wait for the data transmit interrupt - Read I2Cn_STATUS: TD=1, NACK=0, and AL=0, then write 1 to clear the corresponding I2Cn_STATUS bits.
   - Go to Step 8 or Step 9.
9. If the master wants to send data byte transfer on the same slave or other slaves (i.e. repeat-start): Go to Step 5.

✷ ***Note: As long as the STOP interrupt is issued by master, I2C bus will exit the mode 2 and recover to mode 1.***

14.7.6 TX in Master Mode with mode 2 (or START byte)

1. Write I2Cn_CLKDIV: Set COUNT for mode 1 and set COUNTH for mode 2.
2. Write I2Cn_DATA: Reserved address and RW bit (START byte: 0x01 or mode 2 master code: 8'000001XXX) for slave.
3. Write I2Cn_CTRL: Enable all interrupts, set MSTEN, set the I2C enable, set the START bit, clear the STOP bit, and set the TBEN bit.
4. Wait for the data transmit interrupt - Read I2Cn_STATUS:
   TD=1, NACK=1, and AL=0, then write 1 to clear the corresponding I2Cn_STATUS bits.
5. Write I2Cn_DATA: Target slave address and RW bit for slave.
6. Write I2Cn_CTRL: Set HSMODE bit to 1 to transfer data in mode 2 by using DUTY_OFFSET and COUNTH, enable all interrupts, set MSTEN, set the I2C enable, set the START bit, clear the STOP bit, and set the TBEN bit.
7. Wait for the data transmit interrupt - Read I2Cn_STATUS:
   TD=1, NACK=0, RW=1, and AL=0, then write 1 to clear the corresponding I2Cn_STATUS bits.
8.
   (a) If the master wants to stop after one byte of data has been received:
   - Writ I2Cn_CTRL: Set HSMODE bit, clear the START bit, set the STOP bit, set the NACK bit, and set the TBEN bit.
   - Wait for the data receive interrupt - Read I2Cn_STATUS: TD=1, NACK=1, then write 1 to clear the corresponding I2Cn_STATUS bits.
   - Read I2Cn_DATA: To get data.
   - Go to the initial step.

   (b) If the master wants to receive the second data byte transfer after the first data byte transfer:
   - Write I2Cn_CTRL: Set HSMODE bit, clear the START bit, clear the STOP bit, and set the TBEN bit.
   - Wait for the data receive interrupt - Read I2Cn_STATUS: TD=1, NACK=0, then write 1 to clear the corresponding I2Cn_STATUS bits.
   - Read I2Cn_DATA: To get data.
   - Go to Step 8 or Step 9.
9. If the master wants to receive the data byte transfer from the same slave or other slave (i.e. repeat-start): Go to Step 5.

✷ ***Note: As long as the STOP interrupt is issued by master, I2C bus will exit the mode 2 and recover to mode 1.***

14.7.7 TX/RX in Master Mode with mode 2 (or START byte)

1. Write I2Cn_ADDR: Set the slave address.
2. Write I2Cn_CTRL: Enable all interrupts and I2C enable, and disable MSTEN.
3. Wait for the data receive interrupt - Read I2Cn_STATUS:
   HSS=1 or SBS=1, then write 1 to clear the corresponding I2Cn_STATUS bits.
4. Wait for the data receive interrupt - Read I2Cn_STATUS:
   - If RW=1, TD=1, HIT=1, I2CB=1, and NACK=0, then write 1 to clear the corresponding I2Cn_STATUS bits, go to Step 4.
   - If RW=0, TD=1, HIT=1, I2CB=1, and NACK=0, then write 1 to clear the corresponding I2Cn_STATUS bits, go to Step 9.
5. Write I2Cn_DATA: Load the data byte to I2Cn_DATA for transfer.
6. Write I2Cn_CTRL: Set the TBEN bit.
7. Wait for the data transmit interrupt - Read I2Cn_STATUS:
   TD=1 or STOP=1, then write 1 to clear the corresponding I2Cn_STATUS bits.
8. If STOP is set, go to the initial step. If STOP is not set, go to Step 5 to load the second data byte.
9. Write I2Cn_CTRL: Set the TBEN bit and set NACK if required.

10. Wait for the data receive interrupt - Read I2Cn_STATUS:
    - If TD=1 and STOP=0: Write 1 to clear the corresponding I2Cn_STATUS bits, then go to Step 10.
    - If TD=0 and STOP=1: Write 1 to clear the corresponding I2Cn_STATUS bits, then go to Step 11.
11. Read I2Cn_DATA: To get data.
12. If STOP is set, go to the initial step. If STOP is not set, go to Step 9 to get the next data.

14.7.8 Master TX Burst mode

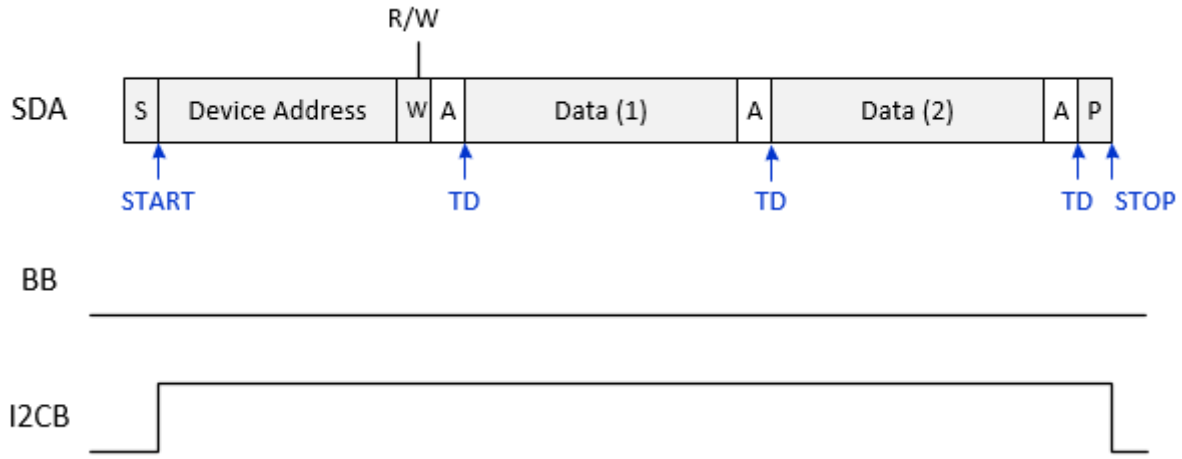
1. Write I2Cn_CLKDIV: Set COUNT and COUNTH.
2. Write I2Cn_ADDR: Set ADDR, MEM_IDX, and ADDR10EN (If users need to enable 10-bit addressing mode). Set M2BIDX_EN and MEM_IDX2 (If users need to transmit 2nd index).
3. Write BSTM: Set total burst data count (TDC).
4. Write data to I2Cn_DATA.
5. Write I2Cn_CTRL: Enable TD interrupts, set HSMODE (if needed), and set BURSTEN to 2'b01 for TX.
6. Wait for the data transmit interrupt - Read I2Cn_STATUS:
    - If NACK=1, TD=0, then write 1 to clear the corresponding I2Cn_STATUS bits, TX burst may fail during the data or address transfer, set RESET to 1.
    - If AL=1, then write 1 to clear the corresponding I2Cn_STATUS bits, set RESET to 1.
    - If NACK=0, TD=1, then write 1 to clear the corresponding I2Cn_STATUS bits, TX burst done.
    - If NACK=1, TD=1, then write 1 to clear the corresponding I2Cn_STATUS bits, TX burst may fail at the last data.

14.7.9 Master RX Burst Mode

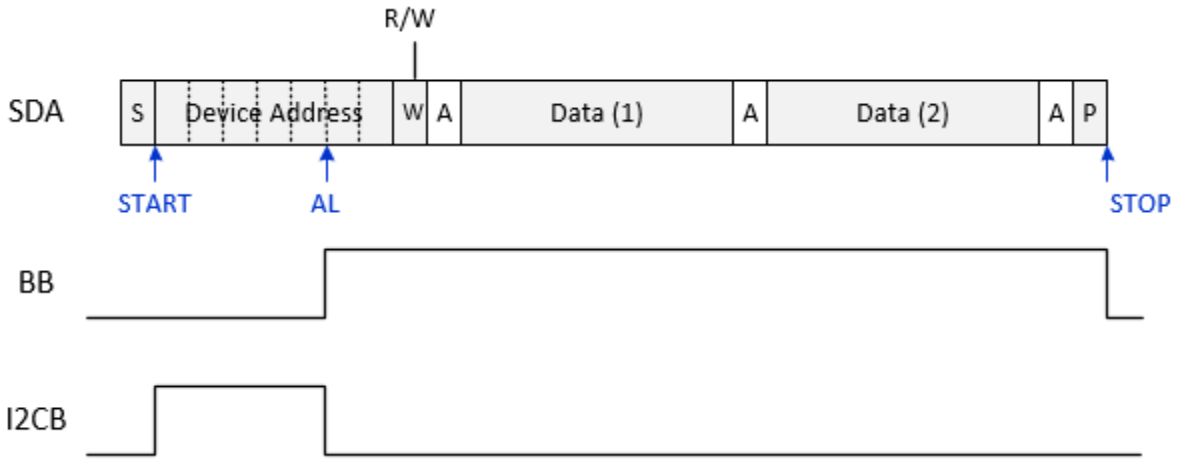
1. Write I2Cn_CLKDIV: Set COUNT and COUNTH.
2. Write I2Cn_ADDR: Set ADDR, MEM_IDX, and ADDR10EN (If users need to enable 10-bit addressing mode). Set M2BIDX_EN and MEM_IDX2 (If users need to transmit 2nd index). Please refer to the note above.
3. Write BSTM: Set total burst data count (TDC), and burst threshold (BSTTHOD).
4. Write I2Cn_CTRL: Enable TD and BSTTHOD interrupts, set HSMODE (if needed), and set BURSTEN to 2'b10 for RX.
5. Wait for the BSTTHOD or TD interrupt - Read I2Cn_STATUS:
    - If NACK=1, TD=0, then write 1 to clear the corresponding I2Cn_STATUS bits, TX burst may fail during the data or address transfer, set RESET to 1.
    - If AL=1, then write 1 to clear the corresponding I2Cn_STATUS bits, set RESET to 1.
    - If BSTTHODSR=1 and TD=0, then users (a) read number of BSTTHOD data, (b) write 1 to clear the corresponding I2Cn_STATUS bits, and (c) go back to Step 4. While users read the data, the controller is keeping receiving data at the same time.
    - If BSTTHODSR=1 and TD=1, users read the number of TDC data, and then write 1 to clear the corresponding I2Cn_STATUS bits.
    - If BSTTHODSR=0 and TD=1, users read the rest number (TDC - BSTTHOD) of data, and then write 1 to clear the corresponding I2Cn_STATUS bits.

14.7.10 Waveform with Status Example

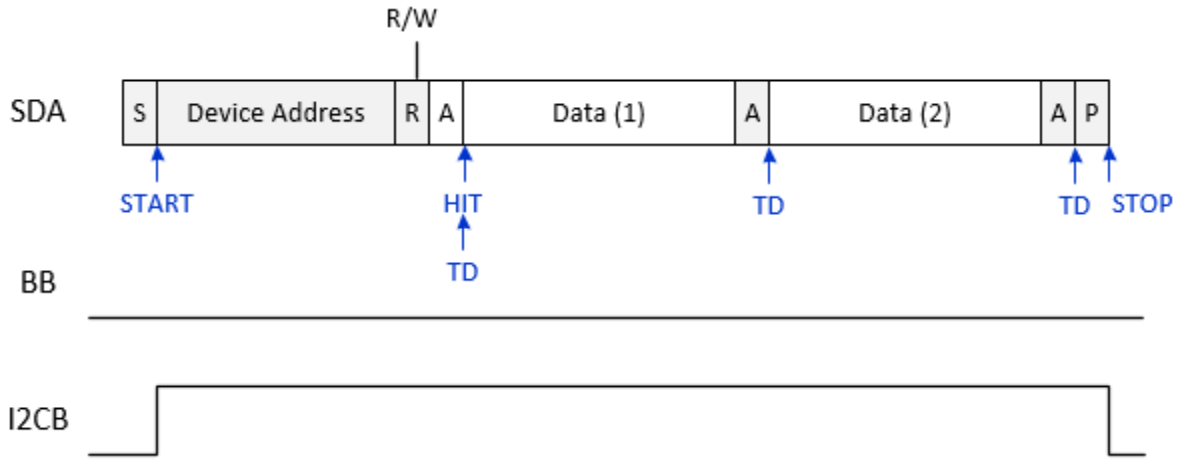
1. In Master TX Mode, no matter Device address is Start Byte or not, SBS will not be set.



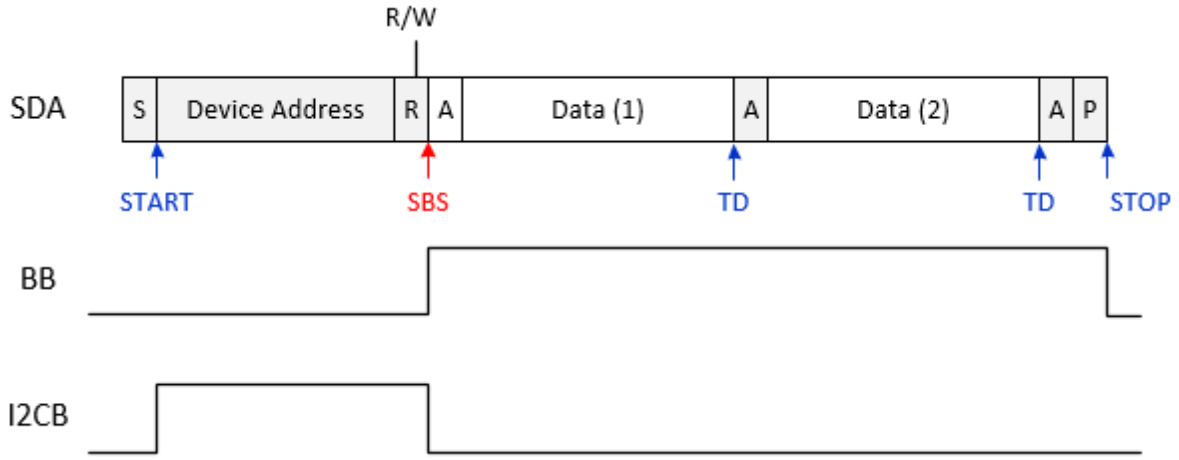
2. In Master TX Mode with arbitration lost, no matter Device address is Start Byte or not, SBS will not be set.



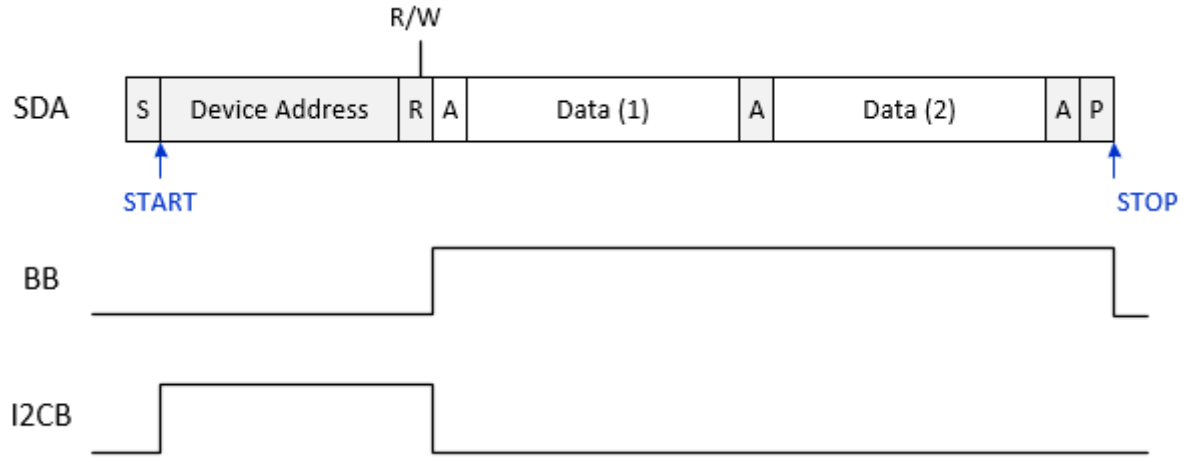
3. In Slave RX Mode with address match, if Device address is not Start Byte, SBS will not be set.



4. In Slave RX Mode with address match, if Device address is Start Byte, SBS will be set.



5. In Slave RX Mode with address mismatch, SBS will not be set.



14.8 I2C REGISTERS

Base Address: 0x4003 3000 (I2C0)
0x4003 4000 (I2C1)
0x4003 5000 (I2C2)

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	I2Cn_CTRL	I2C Control register
0x0004	I2Cn_STATUS	I2C Status register
0x0008	I2Cn_CLKDIV	I2C Clock Division register
0x000C	I2Cn_DATA	I2C Data register
0x0010	I2Cn_ADDR	I2C Address register
0x0014	I2Cn_TGS	I2C Set/Hold Time and Glitch Suppression Setting register
0x0018	I2Cn_BM	I2C Bus Monitor register
0x001C	I2Cn_BSTM	I2C Burst Mode register

14.8.1 I2C n Control register (I2Cn_CTRL) (n=0,1,2)

Address Offset: 0x00

Bit	Field	Access	Initial	Description
31:26	–	–	0	Reserved
25:24	BURSTEN	RW	0	I2C burst transaction control 0: I2C Master/Slave protocol 1: Master TX burst mode 2: Master RX burst mode Other: Reserved
23:22	–	–	0	Reserved
21	SBIEN	RW	0	START byte interrupt enable bit 0: Disable 1: Enable to interrupt the CM4 processor when the I2C controller detects a START byte in the slave mode.
20	HSIEN	RW	0	Mode 2 code interrupt enable bit in the slave mode 0: Disable 1: Enable to interrupt the CM4 processor when the I2C controller detects an mode 2 code in the slave mode.
19	HSMODE	RW	0	Switch between mode 2 and mode 1 0: Generate mode 1 SCL refer to COUNT 1: Generate mode 2 SCL refer to COUNTH
18	ARBOFF	RW	1	Ignore the arbitration lose detection in the single I2C master environment 0: The AL and interrupt will assert (if ALIEN is 1) to notify CM4 processor that the I2C bus has detected the arbitration lose, but I2C controller will still finish the entire I2C protocol. 1: Ignore the arbitration lose detection in the single I2C master environment.
17	–	–	0	Reserved
16	SDALOW	RW	0	SDA tied to LOW 0: For I2C normal case 1: SDA out will be tied to LOW
15	SCLLOW	RW	0	SCL tied to LOW 0: For I2C normal case 1: SCL out will be tied to Low
14	STARTIEN	RW	0	Start condition interrupt enable bit 0: Disable 1: Enable to interrupt the CM4 processor when the I2C controller detects a start condition
13	ALIEN	RW	0	Lose arbitration interrupt enable bit 0: Disable 1: Enable to interrupt the CM4 processor when the I2C controller loses the

Bit	Field	Access	Initial	Description
				arbitration
12	SAMIEN	RW	0	Slave address hit interrupt enable bit 0: Disable 1: Enable to interrupt the CM4 processor when the I2C controller detects a slave address that matches the SAR register or a general call address.
11	STOPIEN	RW	0	Stop condition interrupt enable bit 0: Disable 1: Enable to interrupt the CM4 processor when the I2C controller detects a stop condition.
10	NACKIEN	RW	0	NACK interrupt enable bit 0: Disable 1: Enable to interrupt the CM4 processor when the I2C controller detects the NACK response.
9	TDIEN	RW	0	Data interrupt enable bit 0: Disable 1: Enable to interrupt the CM4 processor when the I2C controller I2Cn_DATA register has transmitted/received one data byte from the I2C bus, or has transmitted/received the total burst data count (TBC bits in BSTM register) in the master RX/TX burst mode.
8	BSTTHODIEN	RW	0	Burst threshold interrupt enable bit 0: Disable 1: Enable to interrupt the CM4 processor when the I2C controller read data reaches the burst threshold size threshold (BSTTHOD bits in BSTM register) in the master RX burst mode.
7	TBEN	RW	0	Transfer Byte enable bit 0: Not ready to receive or transmit one byte. The I2C controller will insert the wait state by pulling SCL Low. 1: Ready to receive or transmit one byte, and will be automatically cleared when the device receive or transmit one byte or detect the stop/start condition.
6	ACKNACK	RW	0	ACK/NACK signal control bit If I2C controller detects stop condition on bus, this bit will auto clear to 0. 0: ACK signal. 1: NACK signal.
5	STOP	RW	0	Stop condition initial bit 0: No action. 1: Initiate a stop condition after transferring the next data byte on bus when I2C is in the master mode.
4	START	RW	0	Start condition initial bit 0: No action. 1: Initiate a start condition when the I2C bus is idle or initiates a repeated start condition after transferring the next data byte on bus in the master mode.
3	GCEN	RW	0	General call response enable bit 0: Disable 1: Enable
2	MSTEN	RW	0	Master mode enable bit 0: Slave mode 1: Master mode
1	I2CEN	RW	0	I2C enable controller 0: Disable 1: Enable
0	RESET	RW	0	I2C reset bit 0: Idle status 1: Reset I2C. CTRL / STATUS / DATA registers. This bit will be automatically cleared after 2 * PCLK clocks.

14.8.2 I2C n Status register (I2Cn_STATUS) (n=0,1,2)

Address Offset: 0x04

When the I2C controller interrupt is asserted, software reads the I2Cn_STATUS bits to check the status of the I2C controller and write 1 to clear I2Cn_STATUS.

I2Cn_STATUS is used to clear the following interrupts by reading the register status:
- Data transfer (TX/RX) is completed.
- Slave address is detected.
- Bus NACK is detected.
- Start condition is detected.
- Stop condition is detected.
- Arbitration loss is detected.
- General Call is detected.
- START byte is detected.

Bit	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23	SBS	RW1C	0	START byte status 0: No START byte detected 1: Detect a START byte in slave mode
22	HSS	RW1C	0	Mode 2 code status 0: No mode 2 code detected 1: Detect a mode 2 code
21:12	–	–	0	Reserved
11	START	RW1C	0	Start condition status 0: No start condition detected 1: Detect a start condition
10	AL	RW1C	0	Arbitration lost status 0: Not lose arbitration 1: Lose arbitration in master mode
9	GC	RW1C	0	General-call status 0: No GC address match 1: Match GC address in slave mode
8	HIT	RW1C	0	Address hit status 0: No address hit 1: Receive slave address that hits the address in SLVADDR register.
7	STOP	RW1C	0	Stop condition status 0: No stop condition detected 1: Detect a stop condition
6	NACK	RW1C	0	NACK status 0: No NACK response detected 1: Detect an NACK response
5	TD	RW1C	0	Data transferring status 0: Data is transferring 1: Finish transferring a byte, or the total data count (TDC bits in BSTM register) is finished in the master RX/TX burst mode.
4	BSTTHODSR	RW1C	0	Data reaches the threshold in the master RX burst mode 0: Not reach the threshold 1: Read data reaches the threshold (BSTTHOD bits in BSTM register) in the master RX burst mode.
3	BB	R	0	Bus busy status 0: Bus is idle 1: I2C bus is busy, but the I2C controller is not involved in the transaction.
2	I2CB	R	0	Busy status 0: Idle state 1: During the time period between START and STOP.
1	–	–	0	Reserved
0	RW	R	0	Act status 0: Master in TX mode/Slave in RX mode 1: Master in RX mode/Slave in TX mode

14.8.3 I2C n Clock Division register (I2Cn_CLKDIV) (n=0,1,2)

Address Offset: 0x08

Define the divided values that are used to generate the I2C SCL clock. This register can be configured to select the transfer speed needed in the I2C bus.

Bit	Field	Access	Initial	Description
31:28	DUTY_OFFSET	RW	0	I2C clock duty High period = ((COUNT) - DUTY_OFFSET) + GSR + 3) * PCLK period Low period = ((COUNT) + DUTY_OFFSET) + 1) * PCLK period
27:20	COUNTH	RW	0	Counter value for mode 2 to generate an I2C clock from the PCLK. SCL(Hz) = PCLK (Hz)/(2 * COUNTH + GSR + 4) For example: If PCLK is 12MHz, COUNTH is 12, and GSR is 2, then SCL is 400kHz.
19:0	COUNT	RW	0x8	Counter value for mode 1 to generate an I2C clock from the PCLK. SCL(Hz) = PCLK (Hz)/(2* COUNT + GSR + 4) For example: If PCLK is 12MHz, COUNT is 57, GSR is 2, then SCL is 100kHz.

14.8.4 I2C n Data register (I2Cn_DATA) (n=0,1,2)

Address Offset: 0x0C

This register is used to transmit and receive data from the I2C bus, and is accessed by either the CM4 processor or the I2C controller. Data coming from the I2C bus interface are received by the DATA register after a full data byte has been received. Data going out of the I2C bus interface are written into the DATA register by the CM4 processor and sent to the serial bus.

Data are moved from the DATA register when the TBEN bit in CTRL register is set. The data transmit interrupt is signaled when one byte of data has been transferred on the I2C bus. If DATA register is not written by the CM4 processor before the next byte package, the I2C controller will insert a wait state until the CM4 processor writes to DATA register and sets the TBEN bit.

When the I2C controller has received one new data byte, it will automatically clear TBEN bit and issue ACK/NACK on the I2C bus. After issuing ACK/NACK, the I 2C controller will insert the TD interrupt to processor. Users must set the TBEN bit again for the next byte transfer on the I2C bus.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	DATA	RW	0	Data buffer for the I2C transmission and reception.

14.8.5 I2C n Address register (I2Cn_ADDR) (n=0,1,2)

Address Offset: 0x10

The I2C address register defines the 10-bit I2C controller that transmits slave ID, 1/2 indexes, 10-bit addressing mode in the master RX/TX burst mode, defines slave address which the processor responds to when the I2C controller operates in the slave mode, or defines the slave address transmitted in the master burst mode. The CM4 processor writes this register before enabling the I2C operation. The register is fully programmable to be set to a value rather than fix to the slave peripheral address preexisting in the system.

Bit	Field	Access	Initial	Description
31:24	MEM_IDX2	RW	0	2nd transmitted index for the master RX/TX burst mode.
23:16	MEM_IDX	RW	0	1st transmitted index for the master RX/TX burst mode.
15:14	–	–	0	Reserved
13	M2BIDX_EN	RW	0	2nd index bytes transfer enable bit 0: Do not transfer 2nd index bytes. 1: Transfer 2nd index bytes.
12	ADDR10EN	RW	0	10-bit addressing mode enable bit 0: 7-bit addressing mode

Bit	Field	Access	Initial	Description
				1: 10-bit addressing mode
11:10	–	–	0	Reserved
9:0	ADDR	RW	0	The 7-bit/10-bit address in the master burst mode or in the slave mode.

14.8.6 I2C n Set/Hold Time and Glitch Suppression register (I2Cn_TGS) (n=0,1,2)

Address Offset: 0x14

This register defines the values of the PCLK clock period when the I2C bus interface has a built in glitch suppression logic. Glitches are suppressed according to GSR * PCLK clock period.

For example,
with a 66-MHz (15-ns period) PCLK clock, and GSR = 4'b100, glitches of 60 ns or less are suppressed.
with a 40-MHz (25-ns period) PCLK clock, and GSR = 4'b10, glitches of 50 ns or less are suppressed.
This is within the 50-ns glitch suppression specification.
The only limitation is COUNT or COUNTH >= 4 + GSR + TSR.

Please note SCL will be released after the wait state and then count TSR+1 cycles in slave mode.

Bit	Field	Access	Initial	Description
31:14	–	–	0	Reserved
13:10	GSR	RW	1	The period to suppress glitch is GSR * PCLK clock cycles.
9:0	TSR	RW	1	The delay values of the PCLK clock cycles between SCL and SDA while data or ACK is driven. The actual delay value is GSR + TSR + 4.

14.8.7 I2C n Bus Monitor register (I2Cn_BM) (n=0,1,2)

Address Offset: 0x18

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1	SCLIN	R	1	The value of the SCL pin 0: SCL is Low 1: SCL is High
0	SDAIN	R	1	The value of the SDA pin 0: SDA is Low 1: SDA is High

14.8.8 I2C n Burst Mode register (I2Cn_BSTM) (n=0,1,2)

Address Offset: 0x1C

Bit	Field	Access	Initial	Description
31:19	–	–	0	Reserved
18:16	BUFSZ	R	5	Buffer size 1: 2 bytes 2: 4 bytes 3: 8 bytes 4: 16 bytes 5: 32 bytes Other: Reserved
15:14	–	–	0	Reserved
13:8	TDC	RW	1	Total burst data count, which must be less than or equal to BUFSZ. Please do not set to 0. *In burst TX mode, if number of data stored in I2Cn_DATA is less than TDC, the I2C controller will stop transmitting until writing the new data to I2Cn_DATA.
7:6	–	–	0	Reserved

Bit	Field	Access	Initial	Description
5:0	BSTTHOD	RW	1	Burst threshold size. These bits must be less than or equal to TDC. Please note that BSTTHOD is only valid in the master RX burst mode. Please do not set them to 0.

15 UNIVERSAL ASYNCHRONOUS RECEIVER AND TRANSMITTER (UART)

15.1 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. It also supports IrDA (infrared data association) SIR ENDEC specifications, and hardware flow control function with CTS and RTS pins.

15.2 FEATURES

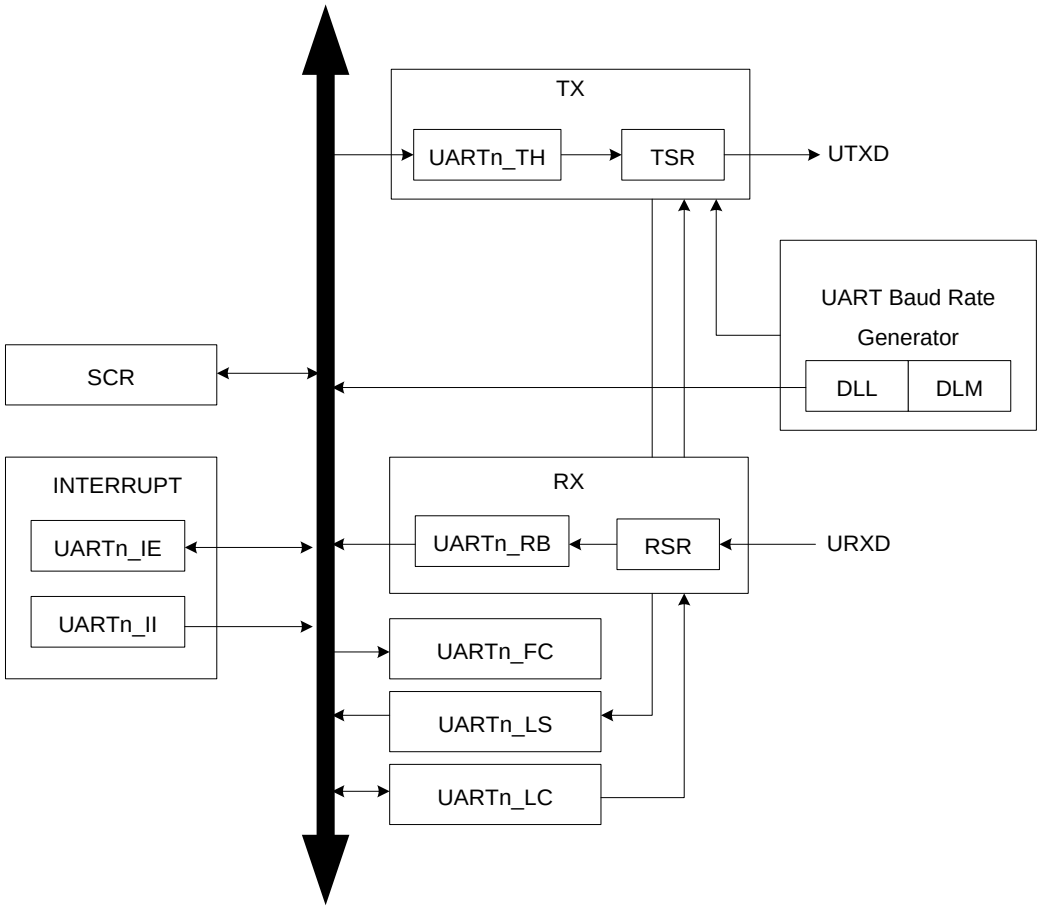
- Full-duplex, 2-wire asynchronous data transfer.
- 16-byte receive and transmit FIFOs
- Register locations conform to 16550 industry standard.
- Receiver FIFO trigger points at 1, 4, 8, and 14 bytes.
- Built-in baud rate generator. Up to 3M baud UART or 1 M baud Smartcard.
- Support hardware flow control by RTS/CTS.
- IrDA 1.3 SIR with Programmable SIR pulse width as 1.6 μs or 3/16 of baud-rate pulse width.
- Supply DMA transfer
- Supports hardware flow control by RTS/CTS pins

15.3 PIN DESCRIPTION

Pin Name	Type	Description	GPIO Configuration
UTXDn	O	Serial Transmit data.	Depends on AFIO register
URXDn	I	Serial Receive data.	Depends on AFIO register
UCTSn[1]	I	CTC Flow Control	Depends on AFIO register
URTSn[1]	O	RTC Flow Control	Depends on AFIO register

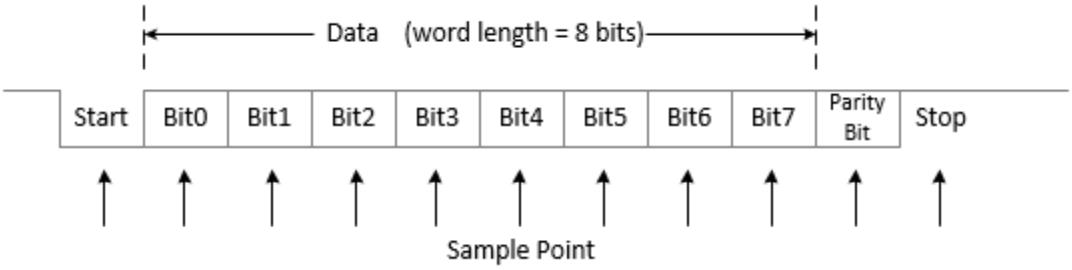
[1] UART4 and UART5 do not have CTS/RTS pin.

15.4 Block Diagram



15.5 UART Mode

The UART mode uses a wired interface for the serial communication with a remote device or a modem. Both data transmission and data reception can occur simultaneously, which is known as a full-duplex operation.
A data character contains 5 to 8 data bits. It is preceded by a start bit and is followed by an optional parity bit and a stop bit. Data is transferred in little-endian order (Least significant bit first).



15.6 Baud Rate Calculation

The clock for both transmit and receive channels is provided by an internal baud generator that divides the pre-scaled clock by any divisor value from 1 to 216 - 1. The output clock frequency of the baud generator must be programmed to be sixteen times the baud rate value. The baud generator input clock is derived from UARTnUCLK clock through a programmable prescaler. The UARTn_PRE register determines the prescaler value. Its default value is 0x01.

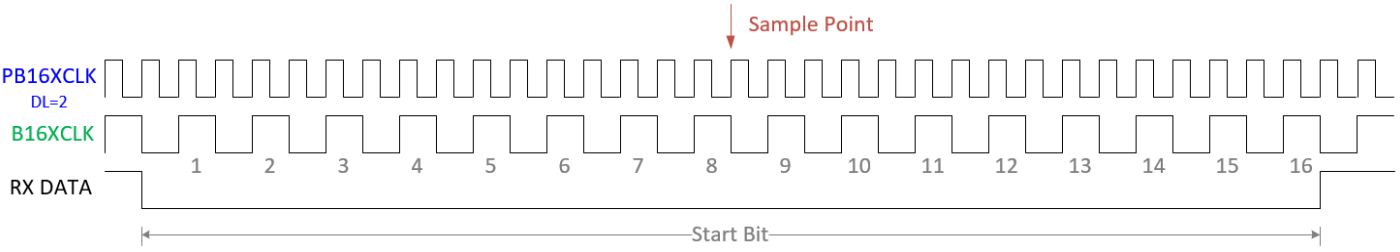
Both the communications format and baud rate must be properly programmed before operation. The communications format is programmed by setting the UARTn_LC register, while the baud rate is selected by programming the baud generator divisor registers (DLL and DLM). The software can read the status of the device at any time during operation. The status information includes state for FIFO, and any other condition detected on the received data stream, like parity error, framing error, data overrun, or break event.

The prescaler divides UARTnUCLK by using the divisors of 1 to 31 and produces a pre-divided clock, PB16XCLK, to the baud rate generator.
The baud rate generator is capable of dividing the pre-divided clock, PB16XCLK, by using the divisors of 1 to 65535 and producing a 16 times baud clock, B16XCLK.

$$\text{Baud Rate} = \frac{UARTnUCLK}{PRE \times DL \times 16}$$

$$DL = 256 \times DLM + DLL = 1 \sim 65535$$

The following figure shows the signal relationship among for UARTnUCLK, PB16XCLK, and B16XCLK, when the UARTn_PRE register is set to ‘1’ and DL is set to ‘2’.
The reference clock for data transfer is B16XCLK at the 8-th rising edge if B16XCLK is the data sample point.



The frequency of B16XCLK must be programmed to be no more than 56 times faster than the frequency of PCLK, and the frequency constraint of PCLK and B16XCLK is shown below:

$$56 \times F_{PCLK} > F_{B16XCLK}$$

- Baud rate limitation:

In some applications, the ideal baud rate is not equal to the practical baud rate, like the following application.
The valid deviation between the ideal and practical condition is within 3.125%. It is calculated by the following formula:
Deviation = [(Practical baud rate/Ideal baud rate) - 1] * 100%
Deviation < | 3.125% | is a valid operation. If the deviation > 3.125%, it will cause an error data transfer.

UARTnUCLK (MHz)	Target BR (bps)	PRE = 1~31	DL = 1~65535	Deviation (%)
192	4800	10	250	0
	9600	5	250	0
	19200	5	125	0
	57600	2	104	0.16
	115200	2	52	0.16
	460800	2	13	0.16
	921600	1	13	0.16

15.7 TX/RX FIFO

Both TX FIFO and RX FIFO are 8-bit wide, configurable location deep. Both TX FIFO and RX FIFO can be disabled to act as a one-byte holding register.

FIFOs are enabled or disabled by using the UARTn_FIFOCTRL register. When FIFOs are disabled, the behavior is the same as if they had not been implemented. The UART working without FIFOs is commonly called the 16450 mode, due to the name of the industry standard UART without FIFOs.

15.8 IrDA 1.3 SIR Mode

IrDA 1.3 SIR mode is the first operational mode that has been defined by the IrDA committee. This mode supports the bidirectional data communication with a remote device by using the infrared radiation as the transmission medium. IrDA 1.3 SIR mode allows the serial communication at the baud rates of up to 115.2 kbps. The format of the serial data is similar to the UART data format. Each data word is serially sent with a zero value start bit at the beginning, followed by the 8-data bits, and ending with one stop bit with a binary value of one. Sending a single infrared pulse signals a zero. A one is signaled by not sending any pulse. The width of each pulse can be either 1.6 μs or 3/16 of a single bit time (1.6 μs is equal to 3/16 of a bit time at 115.2 kbps) by programming the SIRPW bit in UARTn_AUX register.

The device operation in the IrDA SIR mode is similar to the operation in the UART mode. The main difference is that those data transfer operations are normally performed in the half-duplex fashion.

Select the IrDA SIR mode is controlled by the MODE bits in the UARTn_MD register. Each data byte starts with a start bit 0, 1 byte of data, and then ends with at least a stop bit 1. Each serial data bit is encoded before the transmission and decoded after reception. A ‘1’ is decoded with no IR pulse and a ‘0’ is decoded by sending 3/16 of one bit time IR pulse. Similarly, the received serial pulse is decoded as a ‘0’ and the absence of an IR pulse is decoded as a ‘1’.

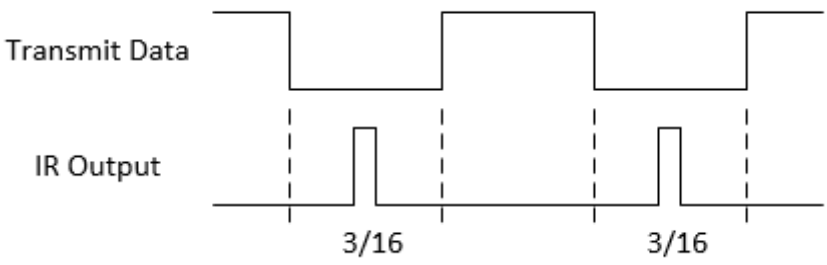


Figure. SIR Encoding

15.8.1 SIR Data Transmission Mode

The following programming steps of the register are required for data transmission in the SIR mode:

1. Program the UARTn_FIFOCTRL register:

- Reset TX FIFO and Rx FIFO
- Enable FIFO and set the trigger level for TX FIFO and RX FIFO

2. Program the UARTn_IE register to enable only the data transmission interrupt:

- Enable the THRE interrupt
- Disable the RLS interrupt
- Disable the RDA interrupt

3. Program the SIRPW bit in UARTn_AUX register to select the fixed 1.6 μs or 3/16 of the pulse width and set TXEN bit to 1 to enable the data transmission

15.8.2 SIR Data Receive Mode

1. Program the FCR register:

- Reset TX FIFO and Rx FIFO
- Enable FIFO and set the trigger level for TX FIFO and RX FIFO

2. Program the IER register to enable the data reception related interrupts:

- Enable the RDA interrupt
- Disable the THR Empty

3. Set RXEN bit in UARTn_AUX register to 1 to enable the data reception.

15.9 UART Hardware Flow Control

The feature of the hardware flow control is fully selectable; enable users to control the serial data flow by using the CTSn and RTSn signal.

- When the RTS flow control is enabled, the RTSn signal is asserted until the receive FIFO is filled up to the programmed trigger level.
- When the CTS flow control is enabled, the transmitter can only transmit data when the CTSn signal is asserted.

The hardware flow control is selectable by using the RTSEN and CTSEN bits in the UART_IE Register.

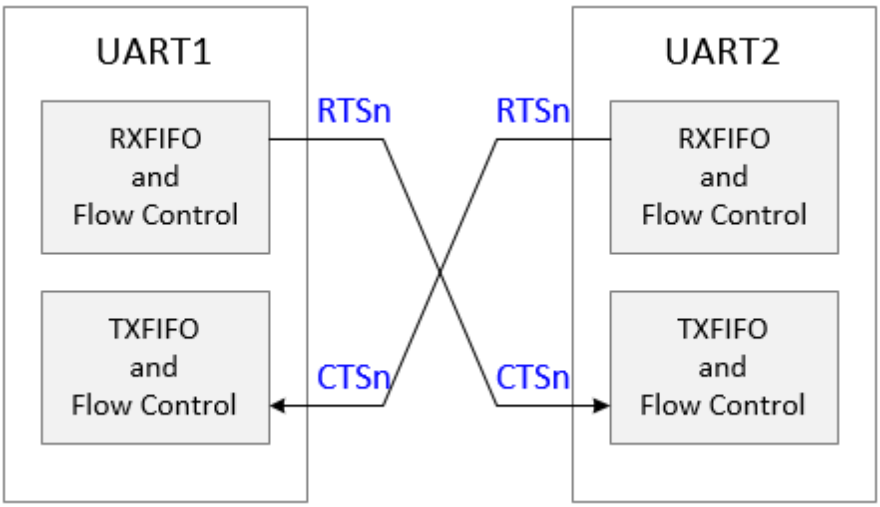


Figure. Hardware flow control between two similar device

15.9.1 RTS Flow Control

The RTS flow control logic is linked to the programmable receive FIFO trigger levels in UARTn_FIFOCTRL register.

When the RTS flow control is enabled, the RTSn signal is asserted until the receive FIFO is filled up to the trigger level. When the receive FIFO trigger level is reached, the RTSn signal will be de-asserted. This indicates that there will be no space to receive more data. The transmission of data will be stopped after the current character is transmitted. The RTSn signal will be re-asserted when data are read out from the receive FIFO; therefore, it is filled to less than the trigger level.

15.9.2 CTS Flow Control

If the CTS flow control is enabled, the transmitter will check the CTSn signal before transmitting the next byte. If the CTSn signal is asserted, the byte will be transmitted; otherwise, the transmission will not occur. The data continues transmitting while the CTSn signal is asserted and the transmit FIFO is not empty. If the transmit FIFO is empty and the CTSn signal is asserted, no data will be transmitted. If the CTSn signal is de-asserted and the CTS flow control is enabled, the current character transmission will be completed before stopping.

15.10 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.
Please refer to the table below for register settings:

- UART TX DMA

UART registers	DMA registers			
FIFOEN	**SRC_WIDTH**	**SRC_SIZE**	**DST_WIDTH**	**TOT_SIZE**
Disable	0x0 (byte)	0x0 (burst=1)	0x0 (byte)	N (byte)
Enable	0x0 (byte)	0x0 (burst=1)	0x0 (byte)	N (byte)

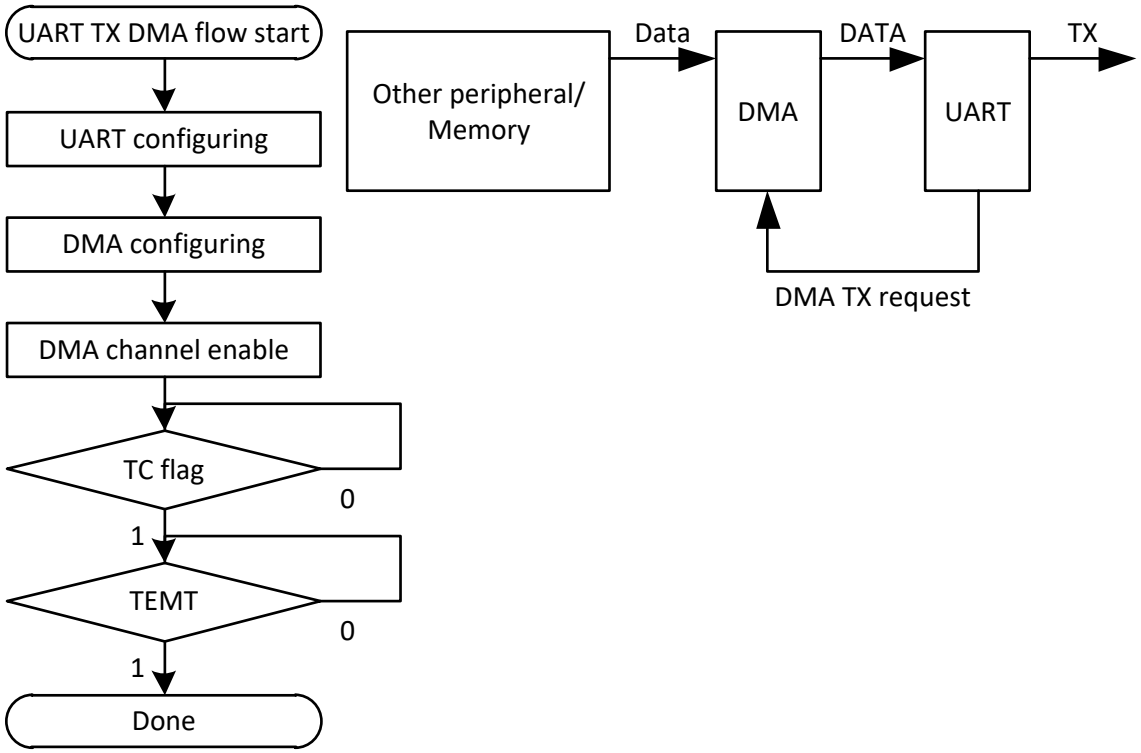
- UART RX DMA

UART registers		DMA registers			
FIFOEN	**RXTL**	**SRC_WIDTH**	**SRC_SIZE**	**DST_WIDTH**	**TOT_SIZE**
Disable	X	0x0 (byte)	0x0 (burst=1)	0x0 (byte)	N (byte)
Enable	X	0x0 (byte)	0x0 (burst=1)	0x0 (byte)	N (byte)

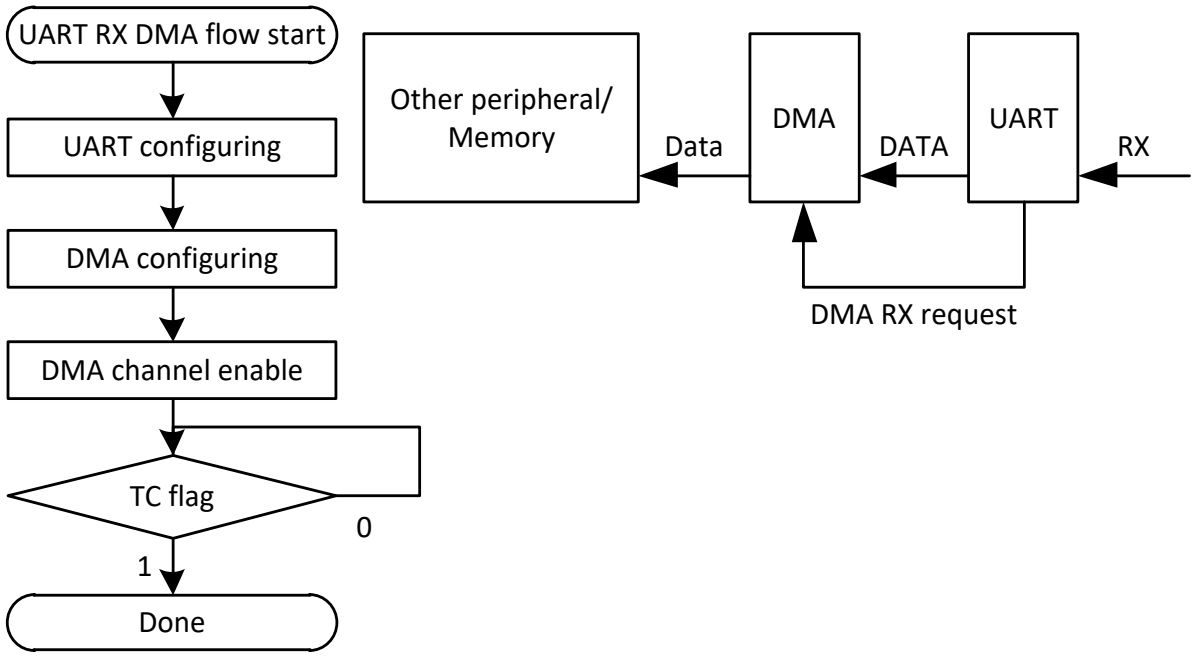
X = Don’t care

15.10.1 DMA Flow Control

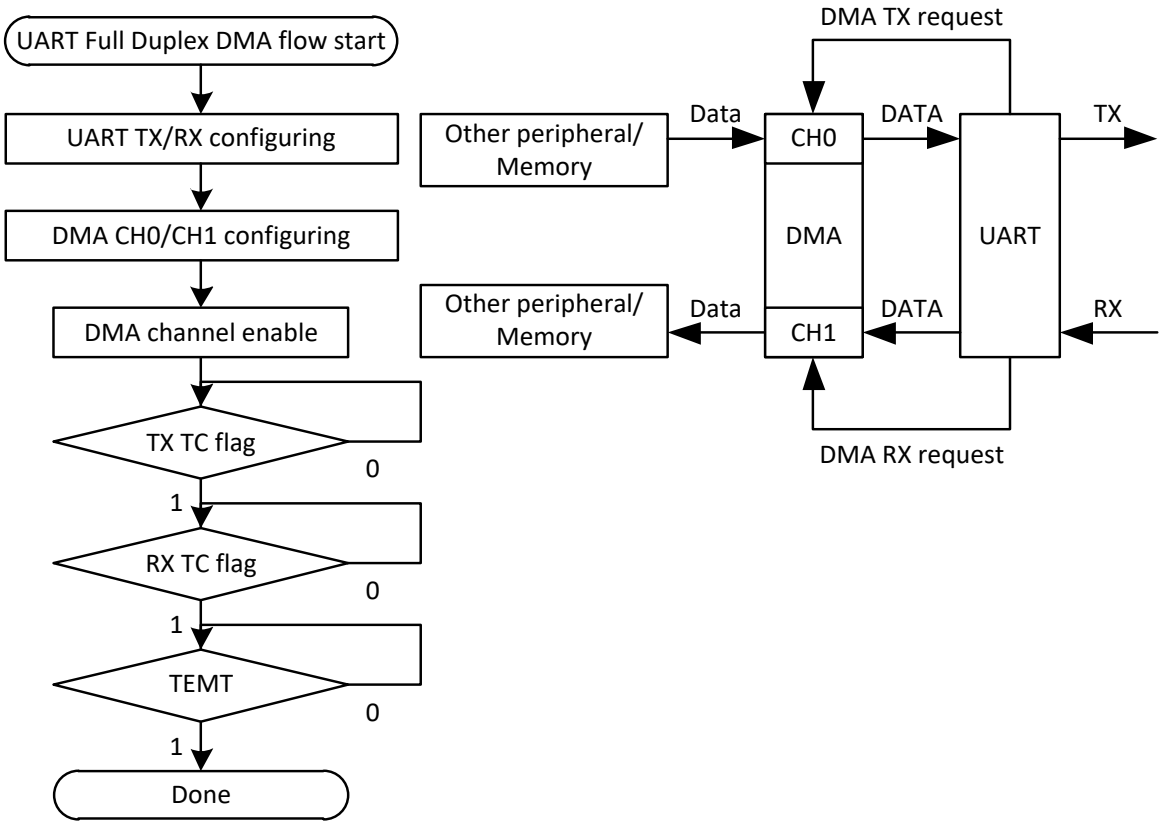
- UART TX DMA



- UART RX DMA



- UART Duplex DMA



15.11 UART REGISTERS

Base Address: 0x4002 4000 (UART0)
0x4001 1000 (UART1)
0x4001 2000 (UART2)
0x4001 3000 (UART3)
0x4001 4000 (UART3)
0x4002 5000 (UART3)

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	UARTn_RB	UARTn Receiver Buffer register
0x0000	UARTn_TH	UARTn Transmit Holding register
0x0000	UARTn_DLL	UARTn Divisor Latch LSB register
0x0004	UARTn_DLM	UARTn Divisor Latch MSB register
0x0004	UARTn_IE	UARTn Interrupt Enable register
0x0008	UARTn_II	UARTn Interrupt Identification register
0x0008	UARTn_PRE	UARTn Prescaler register
0x0008	UARTn_FIFOCTRL	UARTn FIFO Control register
0x000C	UARTn_LC	UARTn Line Control register
0x0010	UARTn_MC	UARTn Modem Control register
0x0014	UARTn_LS	UARTn Line Status register
0x0018	UARTn_MS	UARTn Modem Status register
0x001C	UARTn_SP	UARTn Scratch Pad register
0x0020	UARTn_MD	UARTn Mode Definition register
0x0024	UARTn_AUX	UARTn Auxiliary register
0x0028 – 0x0058	–	Reserved
0x005C	UARTn_RXFFCNT	UARTn RX FIFO Count register

15.11.1 UART n Receiver Buffer register (UARTn_RB) (n=0,1,2,3,4,5)

Address Offset: 0x00

This register is the top byte of the UART RX FIFO, and contains the oldest 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 the UARTn_RB register.

The user can get the data by reading this read-only location. It is the data read port of RX FIFO.
- If FIFOs are enabled: Once a character has been assembled at the receive shift register, the contents will be written to RXFIFO. This RBR refers to RXFIFO location which the current read pointer indicates.
- If FIFOs are not enabled: Once a character has been assembled at the receive shift register, the contents will be written to RXFIFO. This RBR refers to RXFIFO location which the read pointer “0” indicates.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	RB	R	0	The received byte in UART RX FIFO

15.11.2 UART n Transmitter Holding register (UARTn_TH) (n=0,1,2,3,4,5)

Address Offset: 0x00

This register is the top byte of the UART TX FIFO. The top byte is the newest 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.

Users can write data to the transmitter holding register. It is the data write port of the TX FIFO.

- If FIFOs are enabled: Once the transmit shift register is empty, the contents of THR will be written to the transmit shift register for the transmitted data shift out. This THR refers to TXFIFO location which the current read pointer indicates.
- If FIFOs are not enabled: Once the transmit shift register is empty, the contents of THR will be written to the transmit shift register for the transmitted data shift out. This THR refers to TXFIFO location which the read pointer "0" indicates.

If the transmitted character width is less than eight bits, it must be right-justified. Left bits (i.e. MSB) are "don't care" bits. For example, with a word length of five bits, writing 0xd3 or 0xf3 will result in the transmission of a 13h character. Before writing this register, the user must ensure that UART is ready to accept data for transmission, for example, checking if the THRE flag is set in UARTn_LS register .

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	TH	W	0	The byte to be transmitted in UART TX FIFO when transmitter is available

15.11.3 UART n Divisor Latch LSB registers (UARTn_DLL) (n=0,1,2,3,4,5)

Address Offset: 0x00

The DLL 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 16X of the desired baud rate.

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.

These two registers, together with the UARTn_PRE register select the speed at which the communication will occur. This is the baud rate at which characters will be transmitted and the expected baud rate for the characters that will be received. Only one baud rate is defined for both transmission and reception.

DL = 256 * DLM+DLL = 1~65535

DLL contains the lower 8 bits of the divisor and DLM contains the higher 8 bits. There is no clock output when DLM and DLL are 0.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	DLL	RW	1	LSB of UART baud rate divisor, along with the DLM register, determines the baud rate of the UART

15.11.4 UART n Divisor Latch MSB register (UARTn_DLM) (n=0,1,2,3,4,5)

Address Offset: 0x04

The Divisor Latch Access Bit (DLAB) in the UARTn_LC register must be one in order to access this register.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	DLM	RW	0	MSB of UART baud rate divisor, along with the DLL register, determines the baud rate of the UART

15.11.5 UART n Interrupt Enable register (UARTn_IE) (n=0,1,2,3)

Address Offset: 0x04

The DLAB bit in UARTn_LC register must be zero in order to access this register. This register individually enables each of the possible interrupt sources.

Bit	Field	Access	Initial	Description
31:6	–	–	0	Reserved
5	CTSEN	RW	0	CTS flow control enable bit When CTSEN is set to 1,the loopback mode must not be set to 1 and FIFO must be enabled. 0: Disable 1: Enable
4	RTSEN	RW	0	RTS flow control enable bit When RTSEN is set to 1,the loopback mode must not be set to 1 and FIFO must be enabled 0: Disable 1: Enable
3	MSIE	RW	0	Modem status interrupt enable 0: Disable 1: Enable
2	RLSIE	RW	0	RLS interrupt enable 0: Disable 1: Enable
1	THREIE	RW	0	THRE interrupt enable 0: Disable 1: Enable
0	RDAIE	RW	0	RDA interrupt enable 0: Disable 1: Enable

15.11.6 UART n Interrupt Enable register (UARTn_IE) (n=4,5)

Address Offset: 0x04

The DLAB bit in UARTn_LC register must be zero in order to access this register. This register individually enables each of the possible interrupt sources.

Bit	Field	Access	Initial	Description
31:4	–	–	0	Reserved
2	RLSIE	RW	0	RLS interrupt enable 0: Disable 1: Enable
1	THREIE	RW	0	THRE interrupt enable 0: Disable 1: Enable
0	RDAIE	RW	0	RDA interrupt enable 0: Disable 1: Enable

15.11.7 UART n Interrupt Identification register (UARTn_II) (n=0,1,2,3,4,5)

Address Offset: 0x08

The Divisor Latch Access Bit (DLAB) in UARTn_LC register must be zero in order to access this register.

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.

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.

The main purpose of this register is to identify the interrupt with the highest priority that is currently pending. There is a four-level priority encoder from the highest priority to the lowest priority.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:6	FIFOEN	R	0	Equal to FIFOEN bit in UARTn_FIFOCTRL register
5	–	–	0	Reserved
4	TXFIFOFULL	R	0	TX FIFO full 0: TX FIFO is not Full 1: TX FIFO is Full
3:1	INTID	R	0	Interrupt identification which identifies an interrupt corresponding to the UARTn RX FIFO. 0: 4-Modem status 1: 3-Transmitter Holding Register Empty (THRE) 2: 2a-Receive Data Available (RDA) 3: 1-Receive Line Status(RLS) 6: 2b-Character Time-out Indicator (CTI) Other: Reserved
0	INTSTATUS	R	1	Interrupt status 0: As least 1 interrupt is pending 1: No interrupt is pending

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 Level	RX data in FIFO reached trigger level (FCR0=1)	Read UARTn_RB register or UART FIFO drops below trigger level
CTI	1100	2nd Level	Minimum of one character in the RX FIFO and no character input or removed during a time period depending on how many characters are in FIFO and what the trigger level is set at 3.5 to 4.5 character times.	Read UARTn_RB register
THRE	0010	3rd Level	THRE	Read UARTn_II register (if source of interrupt) or Write THR register
MS	0000	Lowest	CTS, DSR, RI, or DCD.	MSR Read

15.11.8 UART n FIFO Control register (UARTn_FIFOCTRL) (n=0,1,2,3,4,5)

Address Offset: 0x08

The Divisor Latch Access Bit (DLAB) in UARTn_LC register must be zero in order to access this register.

This is a write-only register at the same location as UARTn_II register which is read-only. This register is used to enable and clear FIFOs, and set the TX and RX FIFO trigger level. If users reset FIFO, the status of FIFO will not clear. Users must clear these registers by reading the corresponding registers.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:6	RXTL	W	0	Trigger level of RX FIFO interrupt 00: 1 character 01: 4 characters 10: 8 characters 11: 14 characters
5:3	–	–	0	Reserved
2	TXFIFORST	W	0	TX FIFO reset 0: No effect 1: Clear all bytes in TX FIFO and resets the counter to 0, and will return to zero automatically. The shift register is not cleared, so any reception active will continue.
1	RXFIFORST	W	0	RX FIFO reset 0: No effect 1: Clear all bytes in RX FIFO and resets the counter to 0, and will return to zero automatically. The shift register is not cleared, so any reception active will continue.
0	FIFOEN	W	1	FIFO enable 0: Disable 1: Enable Change this bit will automatically reset both TX and RX FIFO.

Note:
1. ***This is a write-only register at the same location as UARTn_II register which is read-only.***
2. ***When using the value read back from (UART II) to set the TX or RX FIFO reset in UARTn_FIFOCTRL. It may cause the FIFO to be disabled or RXTL to be changed causing malfunction.***

15.11.9 UART n FIFO Prescaler register (UARTn_PRE) (n=0,1,2,3,4,5)

Address Offset: 0x08

The Divisor Latch Access Bit (DLAB) in UARTn_LC register must be one in order to access this register.

This 5-bit register adds a second programmable division factor to obtain the desired baud rate. The division factor is the value hold in this register, so the maximum factor is 31 and the minimum is 0. When PRE is 0, there is no input clock to divisor latch unit. Therefore, programming DLL and DLM is useless when PRE is 0.

Bit	Field	Access	Initial	Description
31:5	–	–	0	Reserved
4:0	PRE	RW	1	Prescale value

15.11.10 UART n Line Control register (UARTn_LC) (n=0,1,2,3,4,5)

Address Offset: 0x0C

This register controls the way in which the transmitted characters are serialized and received characters are assembled and checked.

The UARTn_LC register determines the format of the data character that is to be transmitted or received.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	DLAB	RW	0	Divisor Latch access 0: Disable access to Divisor Latch 1: Enable access to Divisor Latch
6	BC	RW	0	Break control 0: Disable 1: Enable. UTXD is forced to logic 0. If several characters are stored in the transmit FIFO, they will be removed from this FIFO and passed sequentially to the Transmitter Shift register which serializes them, even if BC bit is set. This fact can be useful to establish the break time making use of the THRE and TEMT flags in the UARTn_LS register. Firmware can follow the sequence below to assure no erroneous or extraneous characters will be transmitted because of the break: - Set break when transmitter is idle - Write a character with any value to TH. - Wait for the transmitter to become idle - Clear break when normal transmission has to be restored.
5:4	PS	RW	0	Parity selection 0: Odd parity. Number of 1s in the transmitted character and the attached parity bit will be odd. 1: Even parity. Number of 1s in the transmitted character and the attached parity bit will be even. 2: Forced 1 sticky parity 3: Forced 0 sticky parity
3	PE	RW	0	Parity enable 0: Disable parity generation and checking 1: Enable parity generation and checking
2	SBS	RW	0	Stop bit selection 0: 1 stop bit 1: 2 stop bit (1.5 stop bit if WLS=0)
1:0	WLS	RW	0	Word length selection 0: 5-bit character 1: 6-bit character 2: 7-bit character 3: 8-bit character

15.11.11 UART n Modem Control register (UARTn_MC) (n=0,1,2,3)

Address Offset: 0x10

Bit	Field	Access	Initial	Description
31:5	–	–	0	Reserved
4	LMS	RW	0	Loopback mode enable When UART is set in UART loopback mode, UTXDn/URXDn/UCTSn/URTSn pins are not used and UTXDn pin is set to inactive state. When UART is set in SIR loopback mode, IRDA_RXDL pin is not used but IRDA_TXDn pin is still in active state. 0: Disable 1: Enable
3:2	–	–	0	Reserved
1	RTSCTRL	RW	0	Source from modem output (RTSn) pin. RTSn pin is always forced to inactive state (high) in modem loopback mode. 0: RTSn pin output High (inactive)

Bit	Field	Access	Initial	Description
				1: RTSn pin output Low (active)
0	–	–	0	Reserved

15.11.12 UART n Modem Control register (UARTn_MC) (n=4,5)

Address Offset: 0x10

Bit	Field	Access	Initial	Description
31:5	–	–	0	Reserved
4	LMS	RW	0	Modem loopback mode enable 0: Disable 1: Enable
3:0	–	–	0	Reserved

15.11.13 UART n Line Status register (UARTn_LS) (n=0,1,2,3,4,5)

Address Offset: 0x14

This register informs users of the status of the transmitter and the receiver. In order to acquire the information about a received character, LSR must be read before reading the received character from RB.

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	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	RXFE	R	0	Receiver FIFO error flag 0: UARTn_RB contains no UART RX errors or FIFOEN=0 1: UARTn_RB contains at least 1 UART RX error. The 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.
6	TEMT	R	1	Transmitter Empty flag 0: THR and/or TSR contains valid data 1: THR and TSR are both empty
5	THRE	R	1	THR 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
4	BI	R	0	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. **Note:** The break interrupt is associated with the character at the top of the UARTn_RB FIFO. 0: No break interrupt 1: Break interrupt status is active
3	FE	R	0	Framing error flag

Bit	Field	Access	Initial	Description
				When the stop bit of a received character is 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. **Note:** A framing error is associated with the character at the top of the UARTn_RB FIFO. 0: No framing error 1: Framing error status is active
2	PE	R	0	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. **Note:** A parity error is associated with the character at the top of the UARTn_RB FIFO. 0: No parity error 1: Parity error status is active
1	OE	R	0	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: No overrun error 1: Overrun error status is active
0	RDR	R	0	Receiver data ready flag 0: UARTn_RB FIFO is empty 1: UARTn_RB FIFO contains valid data

15.11.14 UART n Modem Status register (UARTn_MS) (n=0,1,2,3)

Address Offset: 0x18

This register provides information about the status of the four modem control input pins.

The four least significant bits can generate an interrupt (Modem Status interrupt) if enabled by the corresponding bit in UARTn_IE register. The interrupt will be generated as soon as any of them is '1'. They are reset to logic 0 whenever this register is read.

Bit	Field	Access	Initial	Description
31:5	–	–	0	Reserved
4	CTS	R	0	Complement of CTS pin input signal
3:1	–	–	0	Reserved
0	DCTS	R	0	Delta CTS 0: No change detected on modem input CTS pin 1: State changes detected on modem input CTS pin

15.11.15 UART n Scratch Pad register (UARTn_SP) (n=0,1,2,3,4,5)

Address Offset: 0x1C

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	PAD	RW	0	Pad information

15.11.16 UART n Mode Definition register (UARTn_MD) (n=0,1,2,3,4,5)

Address Offset: 0x20

Bit	Field	Access	Initial	Description
31:7	–	–	0	Reserved
6	IRDAINVTX	RW	0	IRDA SIR pulse invert 0: IRDA SIR pulse is not inverted 1: IRDA SIR pulse is inverted
5:2	–	–	0	Reserved
1:0	MODE	RW	0	UART mode 0: UART mode 1: IRDA SIR mode Other: Reserved

15.11.17 UART n Auxiliary register (UARTn_AUX) (n=0,1,2,3,4,5)

Address Offset: 0x24

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	SIRPW	RW	0	IRDA SIR pulse width 0: Pulse width = 3/16 1: Pulse width = 3/ PB16XCLK (When PB16XCLK = 1.8432MHz, pulse width = 1.6us)
6:2	–	–	0	Reserved
1	RXEN	RW	0	IrDA RX enable 0: Disable 1: Enable
0	TXEN	RW	0	IrDA TX enable 0: Disable 1: Enable

15.11.18 UART n RX FIFO Count register (UARTn_RXFFCNT) (n=0,1,2,3,4,5)

Address Offset: 0x5C

The RX FIFO Count register represents the number of data bytes in RX FIFO. This register is useful when the number of remaining bytes of a frame received in RX FIFO is below the set trigger level. When FIFO is full, RXFF_CNTR is 0x10 for FIFO with the size of 16-byte. The 4-byte CRC32 appended to the frame is also received in RX FIFO. The bit-width of this register depends on the FIFO depth configuration.

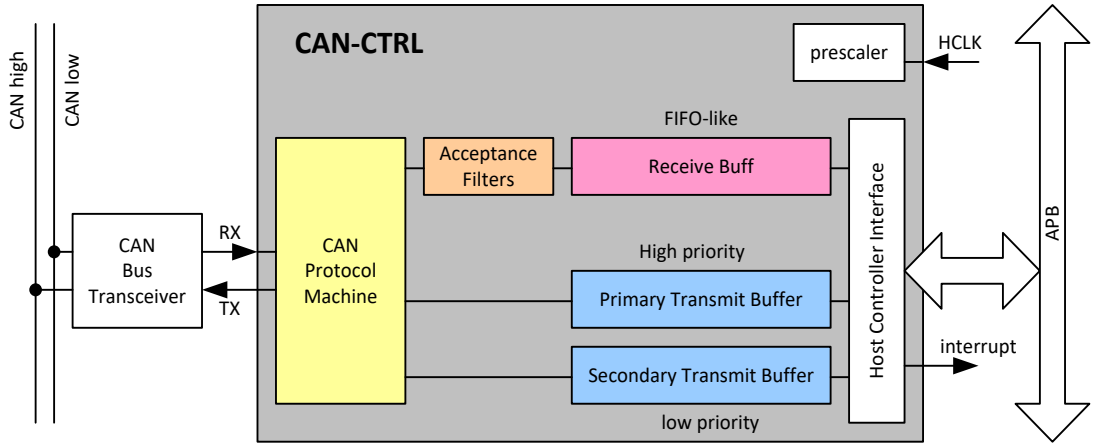
This register is for the FIFO mode only. When the FIFO mode is disabled, the returned value is unexpected.

Bit	Field	Access	Initial	Description
31:5	–	–	0	Reserved
4:0	COUNT	R	0	The number of data bytes in RX FIFO

16 CAN BUS (CAN)

16.1 OVERVIEW

The CAN bus controller is a serial communication controller that performs serial communication according to the CAN protocol. This CAN bus interface meets all constraints of the CAN-specification 2.0A, B active.
The CAN protocol uses a multi-master bus configuration for the transfer of frames between nodes of the network and manages the error handling without any burden on the CPU.
CAN communication is organized in frames. Two types of frames exist : standard and extended frames.
For CAN 2.0 the maximum data payload is up to 8 bytes.



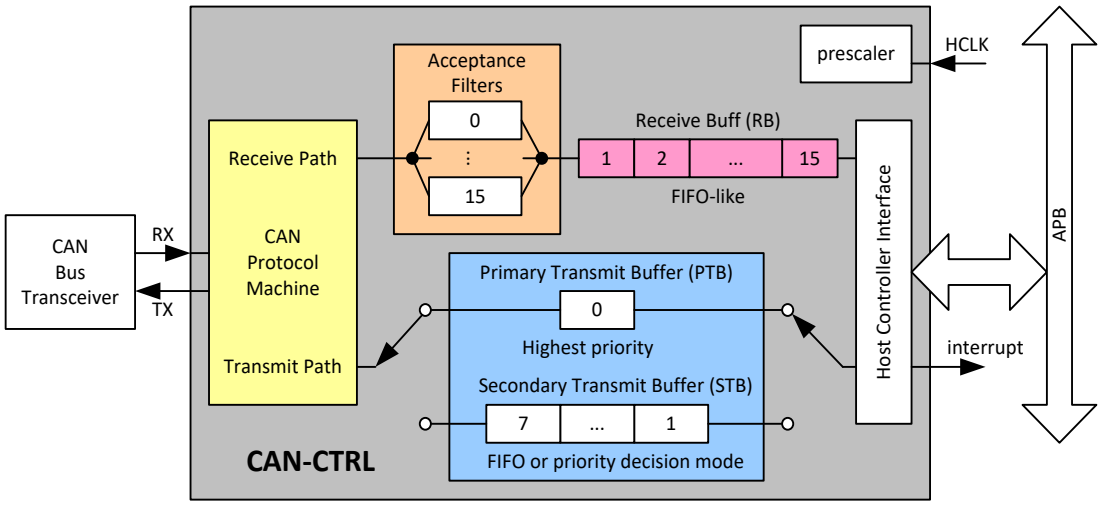
16.2 FEATURES

- Supports CAN 2.0A,B
- Data rates up to 1Mbit/s
- 15 receive buffers (RB)
  - Operation in FIFO
  - Received frames which are “not accepted” or “incorrect” don’t overwrite already stored frames.
- Two transmit buffers
  - 1 frame slots of primary transmit buffer (PTB)
  - 7 frame slots of secondary transmit buffer (STB)
  - Operation in FIFO or priority decision mode
- 16 sets of internal acceptance filter
- Extended features
  - Single shot transmission mode
  - Listen only mode
  - Loop back mode (internal and external)
  - Transceiver standby mode
- Maskable interrupt
- Extended status and error report
  - Capturing of last occurred kind of error and of arbitration lost position
  - Programmable error warning limit
- Time-stamping
  - Supports ISO 11898-4 Time-Triggered CAN with partial hardware

16.3 PIN DESCRIPTION

Pin Name	Type	Description	GPIO Configuration
CAN_TXn	O	CAN Transmit data.	
CAN_RXn	I	CAN Receive data.	
CAN_STBYn	O	CAN Transceiver standby control	

16.4 BLOCK DIAGRAM



16.5 FUNCTIONAL DESCRIPTION

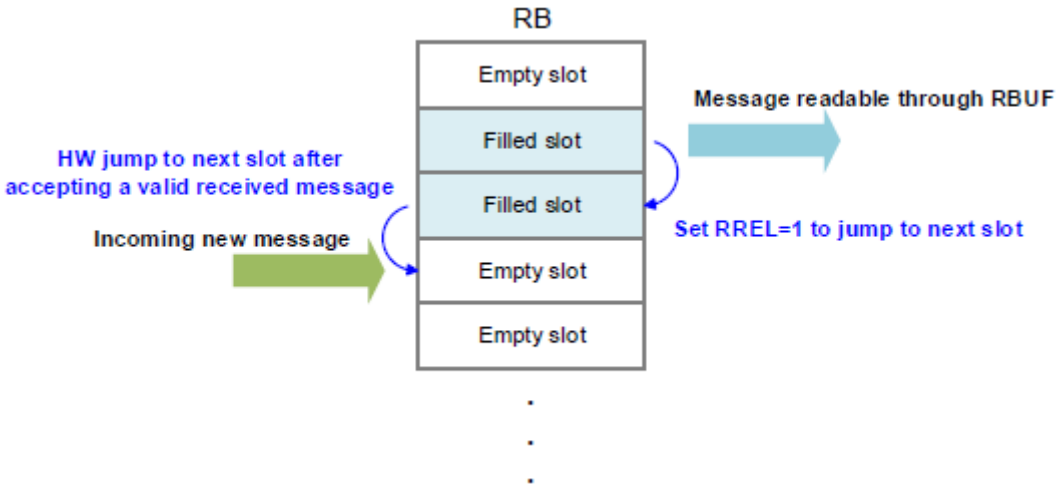
16.5.1 Receive Buffer

Address	Bit position								Register
(offset)	7	6	5	4	3	2	1	0	
0x00	ID[7:0]								RBID
0x01	ID[15:8]								
0x02	ID[23:16]								
0x03	-			ID[28:24]					
0x04	IDE	RTR	-			DLC[3:0]			RBSTA
0x05	KOER[2:0]			TX	-				
0x06	CYCLE_TIME[7:0]								
0x07	CYCLE_TIME[15:8]								
0x08	D1[7:0]								RBDATA1
0x09	D2[7:0]								
0x0A	D3[7:0]								
0x0B	D4[7:0]								
0x0C	D5[7:0]								RBDATA2
0x0D	D6[7:0]								
0x0E	D7[7:0]								
0x0F	D8[7:0]								

16.5.1.1 Access to RBUF

The RBUF registers (0x00 to 0xF) point the message slot with the oldest received message in the RB as can be seen in the following figure.

- KOER in RBUF has the same meaning as the bits KOER in register ERRSTA.
  KOER in RBUF becomes meaningful if RBALL=1.
- Status bit TX in RBUF is set to 1 if the loop back mode is activated and it has received its own transmitted frame.
  This can be useful if LBME=1 and other nodes in the network do also transmissions.
- The time-stamp CYCLE_TIME will be stored in RBUF only in TTCAN mode.
  This is the cycle time at the SOF of this frame. The cycle time of a reference message is always 0.



※ RBUF is composed of SRAM, so the initial value cannot be given through reset.
※ When the DATA received by RBUF is less than 8 bytes, the remaining space will be filled with random data.

16.5.1.2 Message Reception

The received data will be stored in the RB.

- Every received frame that is valid and accepted sets RIF=1 if RIE is enabled. RSTAT is set depending of the fill state.
- When the number of filled buffers is equal to the programmable value AFWL then RAFIF is set if RAFIE is enabled.
- When all buffers are full, the RFIF is set if RFIE is enabled.

The RB always maps the frame slot containing the oldest frame to the RBUF registers.
The maximum payload length for CAN 2.0 messages is 8 bytes. The individual length of each message is defined by the DLC.
Because of this the RB provides slots for each message and the host controller is required to set RREL to jump to the next RB slot. All RBUF bytes of the actual slot can be read in any order.

When the RB is full:

- If ROM=0, the oldest message will be overwritten by the newest.
- If ROM=1, the newest message will be discarded.

In both cases ROIF is set if ROIE is enabled.
If the host controller reads the oldest message and sets RREL before a new incoming message becomes valid then no message will be lost.

16.5.1.3 Handling Frame Receptions

Reading the RB shall be done as follows:

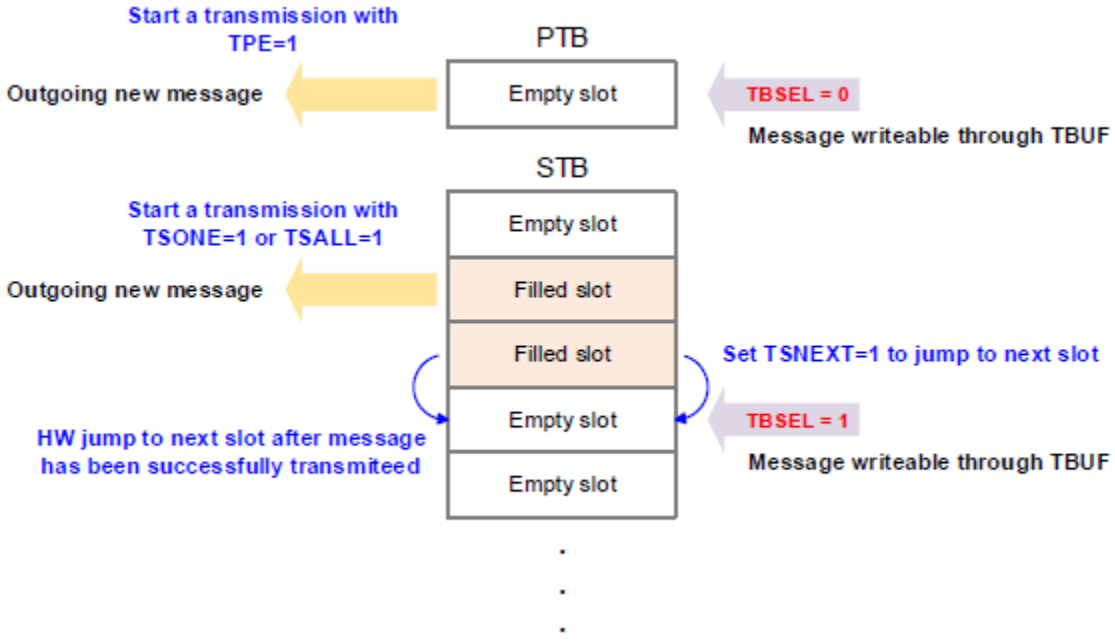
1. Read the oldest message from the RB FIFO using the RBUF registers.
2. Release the RB slot with RREL=1. This selects the next message (the next FIFO slot). RBUF will be updated automatically.
3. Repeat these actions until RSTAT signals an empty RB.

16.5.2 Transmit Buffer

Address	Bit position								Register
(offset)	7	6	5	4	3	2	1	0	
0x50	ID[7:0]								TBID
0x51	ID[15:8]								
0x52	ID[23:16]								
0x53	-			ID[28:24]					
0x54	IDE	RTR	-			DLC[3:0]			TBSTA
0x55	-								
0x56	-								
0x57	-								
0x58	D1[7:0]								TBDATA1
0x59	D2[7:0]								
0x5A	D3[7:0]								
0x5B	D4[7:0]								
0x5C	D5[7:0]								TBDATA2
0x5D	D6[7:0]								
0x5E	D7[7:0]								
0x5F	D8[7:0]								

16.5.2.1 Access to TBUF

The TBUF registers (0x50 to 0x5F) point the next empty message slot in the STB if TBSEL=1 or to the PTB otherwise. All TBUF registers can be written in any order. For the STB it is necessary to set TSNEXT to mark a slot filled and to jump to the next message slot.



※ TBUF is composed of SRAM, so the initial value cannot be given through reset. The reserved position can be read and written, but it does not affect the transmission behavior.

16.5.2.2 Message Transmission

Before starting any transmission, at least one of the TBUF (PTB or STB) has to be loaded with a message.
Below is the recommended programming flow:

1. Set TBSEL to the desired value to select either the PTB or the STB.
2. Write the frame to the TBUF registers.
3. For the STB set TSNEXT=1 to finish loading of this STB slot.

The maximum payload length for CAN 2.0 messages is 8 bytes. The length of each message is defined by the DLC. The DLC becomes meaningless of remote frames, because remote frames always have a data length of 0 bytes.

The host controller is required to set TSNEXT to jump to the next STB slot. All TBUF bytes can be written in any order. Setting TSNEXT=1 is meaningless if TBSEL=0 selects the PTB. In this case TSNEXT is automatically cleared.

- PTB: TPE should be set to start a transmission.
- STB: TSONE should be set to start a transmission of a single message or TSALL to transmit all messages.

The PTB has always a higher priority than the STB. If both transmit buffers have the order to transmit, the PTB message will be always sent first regardless of the frame identifiers.
If a transmission from the STB is already active, it will be completed before the message from the PTB is sent at the next possible transmit position (the next interframe slot).
After the PTB transmission is completed or aborted, the HW returns to process other pending messages from the STB.

When the transmission is completed, the following transmission interrupts are set:

- For the PTB, TPIF is set if TPIE is enabled
- For the STB using TSONE, TSIF is set if one frame has been completed and TSIE is enabled.
- For the STB using TSALL, TSIF is set if all messages have been completed and if TSIE is enabled.

Therefore, if the host controller writes an additional message to the STB after a TSALL transmission has been started

then the additional message will be also transmitted before TSIF will be set.

It is meaningless to set TSONE or TSALL while the STB is empty, TSONE or TSALL will be reset automatically.

16.5.2.3 Message transmission abort

If the situation arises, where a frame in a transmit buffer cannot be sent due to its low priority, this would block the buffer for a long time. In order to avoid this, the host controller can withdraw the transmission request by setting TPA or TSA respectively, if the transmission has not yet been started.

Both TPA and TSA source a single interrupt flag: AIF. The CAN protocol machine executes an abort only if it does not transmit anything to the CAN bus. (No active transmission will be interrupted.) Therefore the following rules are valid:

- There is no abort during bus arbitration.
  - If the node loses arbitration, the abort will be executed afterwards.
  - If the node wins arbitration, the frame will be transmitted.
- There is no abort while a frame is transmitted.
  - If a frame is transmitted successfully then a successful transmission is signaled to the host controller. In this case no abort is signaled.
  - After an unsuccessful transmission after any kind of error, the error counter is incremented and the abort will be executed.
  - If there is at least one frame left in the STB, while the host has commanded all frames to be transmitted (TSALL=1), then both the completed frame as well as the abort is signaled to the host.

Because of these facts aborting a transmission may take some time depending on the CAN communication speed and frame length. If an abort is executed, this results in the following actions:

- TPA releases the PTB which results in TPE=0. The frame data is still stored in the PTB after releasing the PTB.
- TSA releases one single frame slot or all frame slots of the STB. This depends on whether TSONE or TSALL was used to start the transmission. TSSTAT will be updated accordingly.
  Releasing a frame in the STB results in discarding the frame data because the host cannot access it.

Setting both TPA and TSA simultaneously is not recommended.

To clear the entire STB both TSALL and TSA need to be set. In order to detect if a frame cannot be sent for a long time because it loses arbitration the host may use the ALIF / ALIE.

16.5.2.4 A Full STB

After writing a frame to the STB, TSNEXT=1 marks a buffer slot filled and jumps to the next free frame slot.
TSNEXT is automatically reset to 0 by CAN bus controller after this operation.

If the last frame slot has been filled and therefore all frame slots are occupied, then TSNEXT stays set until a new frame slot becomes free. While TSNEXT=1, then writing to TBUF is blocked by CAN bus controller.

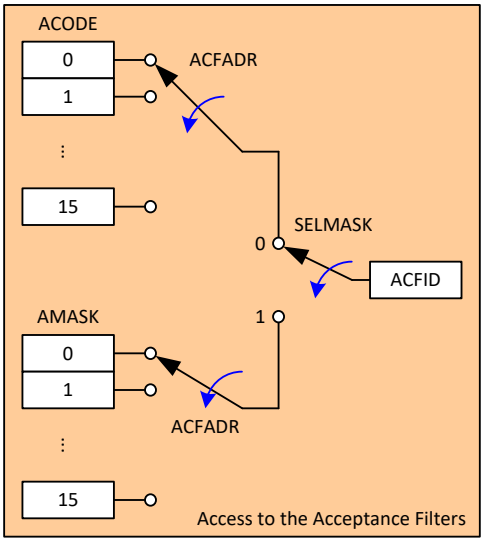
When a slot becomes free, then CAN bus controller automatically resets TSNEXT to 0.
A slot becomes free if a frame form the STB is transmitted successfully or if the host requests an abort (TSA=1).
If a TSALL transmission is aborted, then TSNEXT is also reset, but additionally the complete STB is marked as empty.

16.5.3 Acceptance Filters

16.5.3.1 Access to Acceptance Filter

The acceptance filter registers ACFID provide access to the acceptance filter codes ACODE_x and acceptance filter masks AMASK_x depending on the setting of SELMASK .
Write access to acceptance filter x is only possible if RESET=1.

Only acceptance filter 0 is affected by the reset and it is configured to accept both frame types after power-up.
This means all bits of AMASK_0 are read as 1 and all bits of ACODE_0 are read as 0 after reset.



※ ACODE/AMASK is composed of SRAM, so the initial value cannot be given through reset. The reserved position can be read and written, but it does not affect the filter behavior.

16.5.3.2 Operation

To reduce the load of received frames for the host controller, the CAN bus controller checks the message identifier during acceptance filtering. Therefore, the length of each acceptance filter is 29 bits.

- If a message passes one of the filters, then it will be accepted. If accepted, the message will be stored into the RB and finally RIF is set if RIE is enabled.
- If the message is not accepted, RIF is not set and the RB FIFO pointer is not increased. Messages that are not accepted will be discarded and overwritten by the next message.

Independently of the result of acceptance filtering, the CAN bus controller checks every message on the bus and sends an acknowledge or an error frame to the bus.

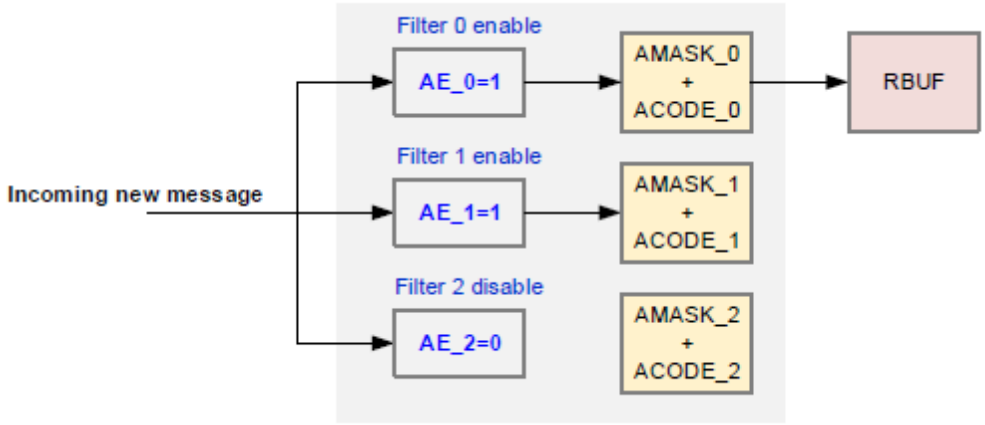
The acceptance mask defines which bits shall be compared while the acceptance code defines the appropriate values.

- Setting mask bits to 0 enables the comparison of the selected acceptance code bits with the corresponding message identifier bits.
- Setting mask bits to 1 are disabled for the acceptance check and this results in accepting the message.

The identifier bits will be compared with the corresponding acceptance code bits ACODE as follows:

- Standard: ID(10:0) with ACODE(10:0)
- Extended: ID(28:0) with ACODE(28:0)

Example:
If AMASK_x(0)=0 and all other AMASK_x bits are 1 then the value of the last ID bit has to be equal to ACODE(0) for an accepted message. All other ID bits are ignored by the filter.



The above figure gives an example of acceptance filtering using several filters.
The filter 0 and 1 are enabled by the appropriate AE_x bits in the ACFCTRL registers. The filter 2 is disabled and therefore do not accept any message.
For the enabled filters the combination of AMASX_x and ACODE_x defines if a message is accepted (like in the example for filter 0) or not accepted (like in the example for filter 1).

✷ ***Note: 1. Disabling a filter by setting AE_x=0 blocks messages.***
2. Disabling a mask bit in AMASK_x disables the check for this bit in accepting messages.

The definitions of AMASK and ACODE alone do not distinguish between standard or extended frames. If bit AIDEE=1 then the value of AIDE defines with frame type is accepted. Otherwise if AIDE=0 both types are accepted.

After reset the CAN bus controller is configured to accept all messages.

16.5.4 Interrupt Flags

CAN bus controller provides several interrupt flags. Interrupt flags are set if the appropriate event occurs and the related interrupt-enable bit is active (if a related interrupt-enable bit exists).
To reset an interrupt flag, the host controller needs to write a 1 to the flag.

- IRQ CANx_TX : TSIF,TPIF.
- IRQ CANx_RX : RAFIF, RFIF, ROIF, RIF.
- IRQ CANx_TT : TTIF, TEIF, WTIF.
- IRQ CANx_EW : AIF, EIF, BEIF, ALIF, EPIF.

16.5.5 Error Handling

CAN bus controller may optionally give detailed information about errors and signal every error to the host by interrupt. This enables to run an application like a CAN bus monitor at the host.

Every CAN node has 3 states of error handling:
1. **Error Active**: The node automatically transmits active error frames upon detection of an error.
2. **Error Passive**: The node transmits a passive error frame upon detection of an error.
   (It does not transmit a dominant value to the bus, but expects an error frame transmitted by other CAN nodes.)
3. **Bus Off**: After too many errors a node goes “bus off” where it stops touching the bus.

To handle the 3 error states every CAN node has two error counters: the transmit and the receive error counter.
Both are incremented and decremented as defined by the CAN specification and the node enters the appropriate error state upon reaching a counter level defined by the CAN specification.

The error counters are incremented if there are errors.

Dangerous errors that are probably caused by this node will result in incrementing the counter by 8, errors that are probably caused by other nodes will result in incrementing the counter by 1.
Valid frame transmission or valid receptions result in decrementing the counters.

An error frame is different to a data frame. It is a dominant pulse for at least 6 consecutive dominant bits, which is a stuff bit violation for other nodes.

- If CAN bus controller is commanded to transmit a frame, then it automatically tries retransmissions as fast as possible until the frame gets transmitted without error or the node goes bus off.
- If a frame is received and CAN bus controller detects an error, then the received data is discarded.

Because of the automatically transmitted error frame the sender of the frame will retransmit the frame.
Frames in RBUF are never overwritten by frames with errors. Only valid frames may result in a RBUF overflow.

16.5.6 The Bus Off State

The “bus off” state is signaled using the status bit BUSOFF in register SYSCTRL.
A CAN node enters the “bus off” state automatically if its transmit error counter becomes >255. Then it will not take part in further communications until it returns into the error active state again.
Setting BUSOFF to 1 also sets the EIF interrupt if EIE is enabled.

A CAN node returns to error active state if it is reset by a reset or if it receives 128 sequences of 11 recessive bits (recovery sequences).

In the “bus off” state, RECNT and TECNT stay unchanged. Therefore, it is recommended to use TECNT before the node enters bus off state and status bit BUSOFF afterwards.

If the node recovers from “bus off” state, then RECNT and TECNT are automatically reset to 0.
If a frame is pending for transmission but the CAN node has entered bus off state, then this frame is kept pending.
If a node returns to error active state after bus off, then the transmission of the pending frame will be tried.
If this is not desired, then the frame should be aborted by the host controller.

※ CAN bus controller can be reset via CANnRST to return the “bus off” state to the error-active state.
But CANnRST will reset the whole CAN bus controller, so the configuration needs to be set again.

16.6 Extended Status and Error Report

During CAN bus communication errors may occur. The following features support detection and analysis of them.
This can be used for extended bus monitoring.

16.6.1 Programmable Error Warning Limit

Errors during reception / transmission are counted by RECNT and TECNT. A programmable error warning limit EWL, located in register INTCFG, can be used by the host controller for flexible reaction on those events.
The limit values can be chosen in steps of 8 errors from 8 to 128: Limit of count of errors = (EWL+1)*8

The interrupt EIF will be set if enabled by EIE under the following conditions:
- The border of the error warning limit has been crossed in either direction by RECNT or TECNT.
- The BUSOFF bit has been changed in either direction.

16.6.2 Arbitration Lost Capture (ALC)

The core is able to detect the exact bit position in the Arbitration Field where the arbitration has been lost.
This event can be signaled by the ALIF interrupt if it is enabled. The value of ALC stays unchanged if the node is able to win the arbitration, then ALC holds the old value of the last loss of arbitration.

The value of ALC is defined as follows: A frame starts with the SOF bit and then the first bit of the ID is transmitted. This first ID bit has ALC value 0, the second ID bit ALC value 1 and so on. Arbitration is only allowed in the arbitration field. Therefore, the maximum value of ALC is 31 which is the RTR bit in extended frames.

✷ ***Note: If a standard remote frame arbitrates with an extended frame, then the extended frame loses arbitration at the IDE bit and ALC will be 12.***
The node transmitting the standard remote frame will not notice that an arbitration has been taken place, because this node has won.

16.6.3 Kind Of Error (KOER)

The core recognizes errors on the CAN bus and stores the last error event in the KOER bits.
A CAN bus error can be signaled by the BEIF interrupt if it is enabled.
Every new error event overwrites the previous stored value of KOER. Therefore, the host controller has to react quickly on the error event.
KOER is updated with each new error. Therefore it stays untouched when frames are successfully transmitted or received. This opens the opportunity for a delayed error investigation.

Error Code	Description
000	No error
001	Bit error
010	Form error
011	Stuff error
100	Acknowledgement error
101	CRC error
110	Other error (dominant bits after own error flag, received active Error Flag too long, dominant bit during Passive-Error-Flag after ACK error)
111	Not Used

16.6.4 Reception of All Data Frames (RBALL)

If RBALL=1, then all received data frames are stored even those with error. This holds also for loop back mode.
Only data frames are stored in RBUF. Error frames or overload frames are not stored.
Most of the possible errors can only occur if the node is the transmitter of the frame. In this case a frame would only be stored in RBUF if a loop back mode is activated.

KOER	Condition	Description
No error	All	Successful reception.
Bit error	Receiver	Can only occur in the ACK slot.
	Transmitter	In the arbitration phase detection of a wrong bit is part of the arbitration and therefore not a BIT ERROR. But if a stuff bit in the arbitration phase is detected wrong by the transmitter of the frame, then this is a BIT ERROR and in this case the header bits are invalid.
Form error	All	Only FORM ERRORs in a data frame are covered. This includes errors in the CRC delimiter, the ACK delimiter or the EOF bits.
Stuff error	Receiver	The position of a STUFF ERROR is unknown.
	Transmitter	Can only happen during arbitration.
ACK error	Receiver	Can only occur if the node is in LOM.
	Transmitter	Can only occur in loop back mode without self-ACK.
CRC error	Receiver	Can only occur if the received CRC value is different from the transmitter.

16.7 Extended Features

16.7.1 Single Shot Transmission

Sometimes an automatic re-transmission is not desired. Therefore the order to transmit a message only once can be set separately for the transmit buffers PTB by the bit TPSS and for the transmit buffer STB by the bit TSSS.

No re-transmission will be performed in the event of an error or arbitration lost if the selected transmission is active.
In the case of an immediate successful transmission, there is no difference to normal transmission.
But in the case of an unsuccessful transmission the following will happen:

- TPIF gets set if enabled , the appropriate transmit buffer slot gets cleared
- In case of an error, KOER and the error counters get updated. BEIF gets set if BEIE is enabled.
- In case of a lost arbitration, ALIF gets set if ALIE is enable.

Therefore for single shot transmission, TPIF alone does not indicate if the frame has been successfully transmitted. Therefore single shot transmission should only be used together with BEIF and ALIF.

If single shot transmission is used with TSALL and there is more than one frame in the STB, then for each frame a single shot transmission is done.
Regardless if any of the frames is not successfully transmitted, the CAN bus controller advances to the next frame and stops if the STB is empty. Therefore in this situation only the error counters indicate what has happened.

This can be quite complex to evaluate because if one of two frames got errors the host cannot detect which one was the successful one.

16.7.2 Listen Only Mode (LOM)

LOM provides the ability of monitoring the CAN bus without any influence to the bus.
Another application is the bit rate detection where the host controller tries different timing settings until no errors occur. Errors will be monitored (KOER, BEIF) in LOM.
In LOM, CAN bus controller is not able to write dominant bits onto the bus (no active error flags or overload flags, no acknowledge).

This is done using the following rules:

- In LOM, the protocol machine acts as if in error passive mode where only recessive error flags are generated. (Only the protocol machine acts as if in error passive mode. All other components including the status registers are not touched.)
- In LOM, the protocol machine does not generate a dominant ACK.
- The error counters stay unchanged regardless of any error condition.

Important facts regarding ACKs for LOM:

- If a frame is transmitted by a node then an ACK visible at the bus will be only generated if at least one additional node is attached to the bus that is not in LOM.
  Then there will be no error and all nodes (also those in LOM) will receive the frame.
- If there is an active or passive error flag after an ACK error, the node in LOM is able to detect this as ACK error.

The loop back mode (external) LBME plays an important role for the behavior of LOM.

- If LBME is disabled, then LOM acts as described above and the node cannot write any dominant bit to the bus.
- If LBME is enabled, then the node is allowed to transmit a frame.

LBME allows only the transmission of an frame including the optional self ACK (SACK), but the node will not respond with an ACK to frames from other nodes and will not generate error or overload frames.

In Summary the combination of LOM and LBME is “a silent receiver, that is able to transmit if necessary”.

16.7.3 Bus Connection Test

To check if a node is connected to the bus the following steps shall be done:

- Transmit a frame. If the node is connected to the bus then its TX bits are visible on its RX input.
- If there are other nodes connected to the CAN bus, then a successful transmission (including an acknowledge from the other nodes) is expected. No error will be signaled.
- If the node is the only node that is connected to the CAN bus (but the connection between the bus, the transceiver and the CAN bus controller is fine) then the first regular error situation occurs in the ACK slot because of no acknowledge from other nodes.
  Then a BEIF error interrupt will be generated if enabled and KOER="100" (ACK error)
- If the connection to the transceiver or the bus is broken then immediately after the SOF bit.
  The BEIF error interrupt will be set and KOER =="001" (BIT error)

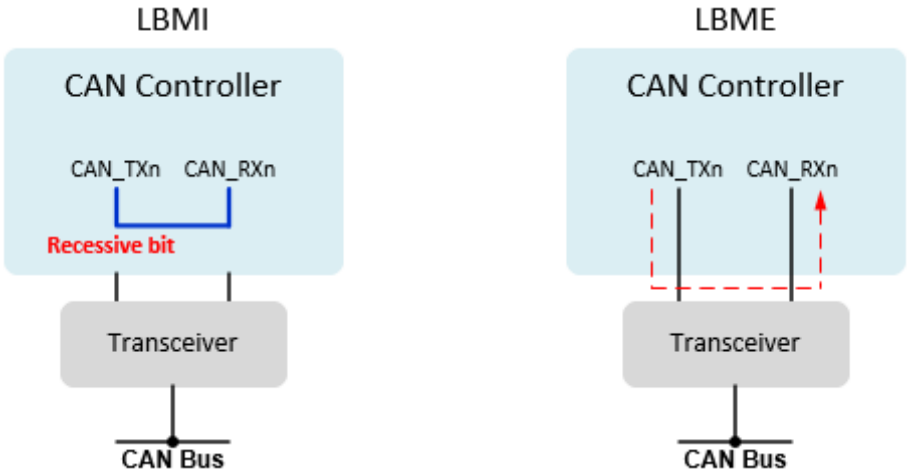
16.7.4 Loop Back Mode (LBMI and LBME)

CAN bus controller supports two Loop Back Modes: internal (LBMI) and external (LBME). Both modes result in reception of the own transmitted frame which can be useful for self tests.

- In LBMI, CAN bus controller is disconnected from the CAN bus and the CAN_TXn output is set to recessive. The output data is internally feedback to the input. In LBMI, the node generates a self ACK to avoid an ACK error.
- In LBME, CAN bus controller stays connected to the transceiver. A transmitted frame will be visible on the bus. With the help of the transceiver CAN bus controller receives its own frame.

In LBME, the node does not generate a self ACK when SACK=0, but generates a self ACK when SACK=1.
Therefore in LBME with SACK=0 there are two possible results upon a frame transmission:

1. Another node receives the frame too and generates an ACK. This will result in a successful transmission and reception.
2. No other node is connected to the bus and this results in an ACK error.
   To avoid retransmissions and incrementing the error counters it is recommended to use TPSS or TSSS if it is unknown if other nodes are connected.



In Loop Back Mode the core receives its own message, stores it in the RBUF and sets the appropriate receive and transmit interrupt flags if enabled.

- LBMI can be useful for chip-internal and software tests.
- LBME can test the transceiver and the connections to it.

> ✷ ***Note: Activation of LBMI or LBME should not be done while a transmission is active.***

If the node is connected to a CAN bus then switching back from LBMI to normal operation must not be done by simply setting LBMI to 0, because then it may be that this occurs just at the moment while another CAN node is transmitting.

In this case switching back to normal operation shall be done by setting bit RESET to 1.
This automatically clears LBMI to 0. Finally RESET can be disabled and the core returns back to normal operation.

In contrast to this LBME can be disabled every time.
LBME can be used in combination with LOM.

16.7.5 Transceiver Standby Mode

Using the bit STBY, the output signal CAN_STBYn is driven.
It can be used to activate a standby mode for a transceiver.
Once standby is activated no transmission is possible and therefore TPE, TSONE and TSALL cannot be set.
On the other hand CAN bus controller does not allow STBY to be set if a transmission is already active (TPE, TSONE or TSALL is set).

- If STBY is set the transceiver enters a low-power mode. In this mode it is unable to receive a frame at full speed but monitors the CAN bus for a dominant state.
- If the dominant state is active for a time which is defined in the transceivers data sheet the transceiver will pull the CAN_RXn signal low.
- If CAN_RXn is low CAN bus controller automatically clears STBY to 0 which disables standby mode for the transceiver. This is done silently without an interrupt to the host controller.

One node transmits a frame for wakeup. If all others nodes are in standby mode, then the transmitter gets an ACK error and will automatically repeat the frame. During the repetition the nodes are back in active mode and will respond with an ACK.

STBY is not affected by bit RESET.

16.7.6 Error Counter Reset

According to the CAN 2.0 standard RECNT counts receive errors and TECNT counts transmit errors. After too many transmit errors a CAN node has to go to bus off state.
Bit RESET does not modify the errors counters or the bus off state.

✸ ***Note: Writing a 1 to the bit BUSOFF resets the error counters and therefore forces the node to leave the bus off state. This is done without setting EIF.***
But this should be used with extra care and is suggested to use only for debugging purposes.

16.8 Software Reset

16.8.1 Register Reset

If bit RESET in SYSCTRL(offset: 0xA0) is set to 1 then the software reset is active.
Several components are forced to a reset state if RESET=1 but not all components are touched by RESET.
Some components are only sensitive to a hardware reset. The reset values of all bits are always the same for software and hardware reset.

Register	Bits	RESET	Comment
RBID	ID[28:0]	(Yes)	All RB slots are marked as empty. RBUF contains unknown data.
RBSTA	CYCLE_TIME[15:0]	(Yes)	All RB slots are marked as empty. RBUF contains unknown data.
	KOER[2:0]	(Yes)	All RB slots are marked as empty. RBUF contains unknown data.
	TX	(Yes)	All RB slots are marked as empty. RBUF contains unknown data.
	IDE	(Yes)	All RB slots are marked as empty. RBUF contains unknown data.
	RTR	(Yes)	All RB slots are marked as empty. RBUF contains unknown data.
	DLC[3:0]	(Yes)	All RB slots are marked as empty. RBUF contains unknown data.
RBDATA1 RBDATA2	D1~D8	(Yes)	All RB slots are marked as empty. RBUF contains unknown data.
TBID	ID[28:0]	(Yes)	All STB slots are marked as empty. Because of TBSEL/TBUF points to the PTB.
TBSTA	IDE	(Yes)	All STB slots are marked as empty. Because of TBSEL/TBUF points to the PTB.
	RTR	(Yes)	All STB slots are marked as empty. Because of TBSEL/TBUF points to the PTB.
	DLC[3:0]	(Yes)	All STB slots are marked as empty. Because of TBSEL/TBUF points to the PTB.
TBDATA1 TBDATA2	D1~D8	(Yes)	All STB slots are marked as empty. Because of TBSEL/TBUF points to the PTB.
SYSCTRL	SACK	Yes	-
	ROM	No	-
	ROV	Yes	All RB slots are marked as empty.
	RREL	Yes	-
	RBALL	Yes	-
	RSTAT[1:0]	Yes	All RB slots are marked as empty.
	TSNEXT	Yes	-
	TSMODE	No	-
	TTTBM	No	-
	TSTAT[1:0]	Yes	All STB slots are marked as empty.
	TBSEL	Yes	TBUF fixed to point to PTB.

	LOM	No	-
	STBY	No	-
	TPE	Yes	-
	TPA	Yes	-
	TSONE	Yes	-
	TSALL	Yes	-
	TSA	Yes	-
	LBME	Yes	-
	LBMI	Yes	-
	TPSS	Yes	-
	TSSS	Yes	-
	RACTIVE	Yes	Reception is immediately cancelled even if a reception is active.
	TACTIVE	Yes	All transmissions are immediately terminated with RESET.
	BUSOFF	No	An error counter reset by setting BUSOFF=1 also resets BUSOFF.
INTCFG	AFWL[3:0]	No	-
	EWL[3:0]	Yes	-
	EWARN	No	-
	EPASS	No	-
	EPIE	No	-
	EPIF	Yes	-
	ALIE	No	-
	ALIF	Yes	-
	BEIE	No	-
	BEIF	Yes	-
	RIF	Yes	-
	ROIF	Yes	-
	RFIF	Yes	-
	RAFIF	Yes	-
	TPIF	Yes	-
	TSIF	Yes	-
	EIF	No	-
	AIF	Yes	-

	RIE	No	-
	ROIE	No	-
	RFIE	No	-
	RAFIE	No	-
	TPIE	No	-
	TSIE	No	-
	EIE	No	-
	TSFF	Yes	All STB slots are marked as empty.
SBITCFG	S_PRESC[7:0]	No	Register is writeable if RESET=1 and write-locked if 0.
	S_SJW[6:0]	No	Register is writeable if RESET=1 and write-locked if 0.
	S_SEG2[6:0]	No	Register is writeable if RESET=1 and write-locked if 0.
	S_SEG1[7:0]	No	Register is writeable if RESET=1 and write-locked if 0.
ERRSTA	TECNT[7:0]	No	An error counter reset is possible by setting BUSOFF=1.
	RECNT[7:0]	No	An error counter reset is possible by setting BUSOFF=1.
	KOER[2:0]	Yes	-
	ALC[4:0]	Yes	-
ACFCTRL	AE_[15:0]	No	-
	SELMASK	No	-
	ACFADR[3:0]	No	-
ACFID	AIDEE	No	Register is writeable if RESET=1 and write-locked if 0.
	AIDE	No	Register is writeable if RESET=1 and write-locked if 0.
	ACFBUF[28:0]	No	Register is writeable if RESET=1 and write-locked if 0.
TTCTRL	WTIE	No	-
	WTIF	Yes	-
	TEIF	Yes	-
	TTIE	No	-
	TTIF	Yes	-
	T_PRESC[1:0]	No	-
	TTEN	Yes	-
	TBE	Yes	-
	TBF	Yes	-
	TBPTR[5:0]	No	-

	VERSION[15:0]		
REFID	REF_IDE	No	-
	REF_ID[28:0]	No	-
TRIGCFG	TT_TRIG[15:0]	No	-
	TEW[3:0]	No	-
	TTYPE[2:0]	No	-
	TTPTR[5:0]	No	-
TRIGWTR	TT_WTRIG[15:0]	No	-
PINCTRL	STBYEN	No	-
	RXEN	No	-
	TXEN	No	-

16.8.2 RESET modes

After a hardware reset, CAN bus controller is in RESET mode. When RESET bit is cleared, CAN bus controller is in normal mode (bus communication).

Before entering normal mode, CAN bus controller always has to synchronize on the CAN bus.
To synchronize, CAN bus controller waits until the CAN bus is idle, this means 11 consecutive recessive bits have been monitored on RX.

In RESET mode, Reception is immediately cancelled even if a reception is active. No ACK will be generated.
All transmissions are immediately terminated with RESET. If a transmission is active this will result in an invalid frame. Other nodes will generate error frames.

16.9 Time-Triggered CAN

Time-Triggered CAN (TTCAN) is an operation mode according to ISO 11898-4 where frames will be transmitted only at predefined time windows.
There are three time window types:

- Exclusive time window (only one node is allowed to transmit one frame with a defined ID)
- Free time window (unused time window for further extension of the network)
- Arbitrating time window (several nodes may transmit a frame and arbitration takes place)

The timing is defined by a TTCAN system administrator and organized in a system matrix as shown in the below figure. A row is called a basic cycle and it starts with a reference message. The reference message is transmitted by the time master. Other messages can be transmitted by any node including the time master.

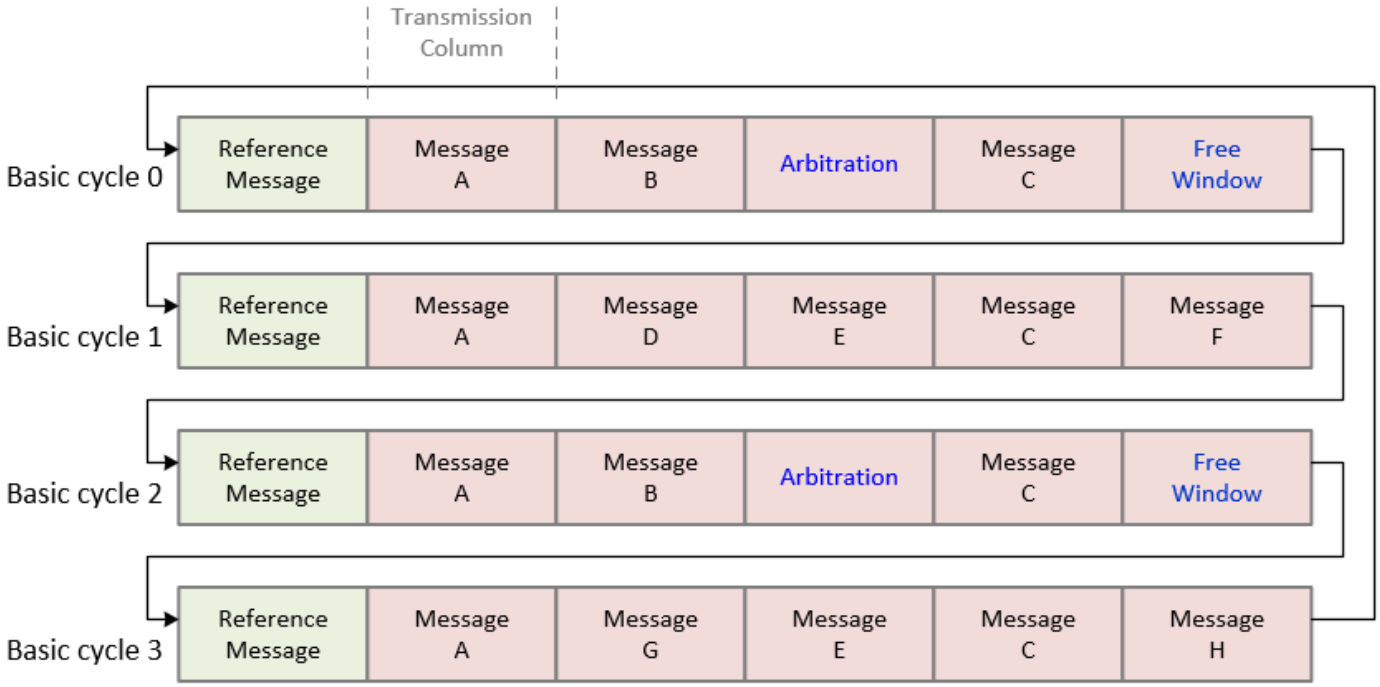


Figure. Example System Matrix

The time of the SOF of a message is a Sync_Mark. The Sync_Mark of a reference message is the Ref_Mark. Timing is done by a free-running 16 bit timer. The difference between the timer and the value of the Ref_Mark is the cycle time. In other words: each basic cycle starts the cycle time at the Ref_Mark.
The cycle time is stored in the RBUF as a time stamp.

The length of a time window is defined by the TTCAN administrator and it is long enough to carry one message. **Therefore messages must be transmitted in single shot mode**.
There is only one exception from this rule: if several arbitrating time window are merged then it is possible to enable retransmission. But this needs to be disabled early enough to not influence the following time window.

To transmit a frame in a time window, CAN bus controller offers a hardware trigger. This trigger needs to be configured by the host. The desired frame needs to be stored in a TBUF slot and the trigger needs to point to this slot. Then if the cycle time comes to the time that is defined for the trigger, CAN bus controller automatically transmits the frame in single shot mode.

If the trigger gets active this will be signaled to the host by an interrupt. Then the host needs to configure the next trigger for the next time window. This requires a real-time response by the host. The duration of one time window is the time that the host is allowed to use.

If CAN bus controller needs to operate as time master, then the reference message needs to be placed in a TBUF slot like all other messages and will be transmitted when the hardware trigger gets active.

Detection of a reference message is done automatically by all of the time slaves and the time master. This is done by setting the correct REF_ID and REF_IDE bits of the reference message in the REFID registers.
Upon detection CAN bus controller automatically updates the Ref_Mark which starts the cycle time.

Additionally to the trigger for messages, CAN bus controller includes a watch trigger. This should be used to check whether the time since the last reference message has been too long.
The host shall configure the watch trigger for the periodic case where each basic cycle is followed by the next basic cycle or for the event triggered case where there is a time gap between the basic cycles and the next cycle is started upon an event. If the watch trigger gets active, this results in the watch interrupt.

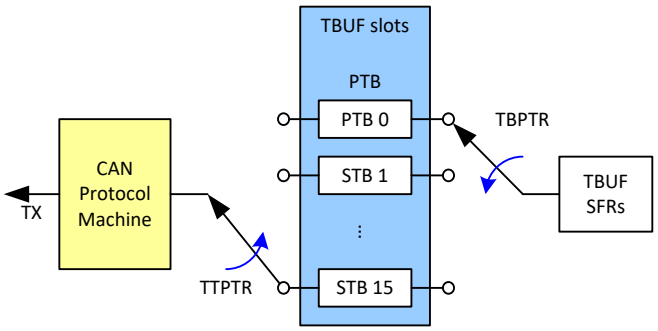
Beside support for ISO 11898-4, CAN bus controller offers a mixture of event driven CAN communication combined with time stamping for received messages. This mode is selected if register bit TTTBM =0 In this mode CAN bus controller acts similar to event driven CAN communication (TTEN=0), but reference messages can be detected and time stamps for received messages are provided. Additionally only time triggers and the watch trigger are supported in this mode.

16.9.1 The TBUF in TTCAN Mode

The behavior if the TBUF depends on the register bit TTTBM.

- If TTTBM=1, then each TBUF slot can be addressed by the controller which is required to use transmission trigger.
- If TTTBM=0, then only time stamping and time trigger are supported.

16.9.1.1 TBUF in TTCAN Mode if TTTBM=1



In TTCAN mode with TTTBM=1 the STB is used as an array of message slots. Each slot can be addressed by TBPTR.
The host can mark a slot as filled or as empty using TBF and TBE. Filled slots are write-locked.
TBSEL and TSNEXT have no meaning in TTCAN mode and are ignored.
The PTB can be addressed by TBPTR=0. This makes the PTB useable as any other STB slot.
This furthermore results in the fact that a successful transmission is always signaled using TSIF.
There is no FIFO operation and no priority decision for the TBUF in TTCAN mode. Furthermore only one frame can be selected for transmission.

A message trigger defines the time when a message needs to be transmitted (the beginning of a time window) and it selects the message using the pointer TTPTR. If the trigger event takes place, then selected message transmission is started . Finally the trigger interrupt is set to signal the host that the next action needs to be prepared.
All transmissions can only be started using a trigger. See chapter “TTCAN Trigger Types” for details.

TPE, TSONE, TSALL, TPSS and TPA are fixed to 0 and ignored in TTCAN mode.

16.9.1.2 TBUF in TTCAN Mode if TTTBM=0

TTTBM=0 offers the combination of event driven CAN communication and time stamping for received messages.
In this mode the PTB and the STB provide the same behavior as if TTEN=0.
Then the PTB always has higher priority than the STB and the STB can operated in FIFO mode (TSMODE=0) or in priority mode (TSMODE=1).

bit TTEN	bit TTTBM	bit TSMODE	TBUF status
0	Ignored	0	Separate PTB and STB PTB : The highest priority STB : FIFO mode
		1	Separate PTB and STB PTB : The highest priority STB : Priority decision mode
1	0	0	Separate PTB and STB PTB : The highest priority STB : FIFO mode
		1	Separate PTB and STB PTB : The highest priority STB : Priority decision mode
	1	Ignored	TTCAN support : Buffer slots selectable by TBPTR and TTPTR

16.9.2 TTCAN Operation

After power up a time master needs to do the initialization as defined in ISO 11898-4. There may be up to 8 potential time masters in a CAN network. Each one has its own reference message ID (the last 3 bits of the ID).
Potential time masters may try to transmit their reference message according to their priority. Lower-prioritized time masters shall try to transmit their reference messages later.

If TTEN is set, then the 16 bit timer is running. If a reference message is detected or if a time master successfully transmits it's reference message then CAN bus controller copies the Sync_Mark of this message to the Ref_Mark which sets the cycle time to 0. The reception of the reference message will set RIF respectively the successful transmission will set TPIF or TSIF. Then the host needs to prepare the trigger for the next action.

A trigger for an action can be a reception trigger. This just triggers the interrupt and it can be used to detect if an expected message is not received. Such a trigger can also be used for other actions, but this depends on the host application.

A different trigger is a transmission trigger. This starts the frame in the TBUF slot, where the trigger points to with TTPTR. If the TBUF slot is marked as empty, then no transmission is started, but the trigger interrupt is set.

It is possible that one host task updates a message slot with a new message if it is necessary by the host application. This could be e.g. a new sensor value. Later then if TTCAN requires this message to be transmitted, it will be activated by the trigger. In other words one host task needs to be responsible for TTCAN and a different host task may update the message slots. This requires that one message slot is exclusively used for one message (e.g. slot 1 for a temperature sensor). If not enough TBUF slots are available then slots need to be shared by different messages.

The TTCAN host application needs to keep track of the system matrix. If a trigger gets active, then the host application needs to prepare the next transmission column. The host needs to handle also the basic cycles. The reference message includes the cycle count.

Most operations in ISO 11989-4 require single shot transmission. See chapter "Single Shot Transmission" for further details about handling successful and unsuccessful transmissions.

16.9.3 TTCAN Timing

CAN bus controller supports ISO 11898-4 level 1. This includes a 16-bit timer running at the speed of a CAN bit time (defined by S_PRESC, S_SEG1, S_SEG2). An additional prescaler is defined by T_PRESC.
If TTEN=1 then the timer is counting continuously.

At the SOF of the message the timer value is the Sync_Mark. If the message was a reference message, then this value is copied to the Ref_Mark. The cycle time is the timer value minus the Ref_Mark.
It is the time that is used as time stamp for received messages or as trigger time for messages, that need to be transmitted. An overflow protection is included and therefore the cycle time is always strictly monotonic inside a basic

cycle.

Interrupts because of trigger events, reception or transmission require clock domain crossing and therefore include some clocks delay.
This only becomes relevant if after a trigger the host application decides to start a transmission. (Therefore it is recommended to use a transmission trigger)
In all other cases the host application has enough time to prepare all actions for the next trigger event (which is the next time).
Almost all hosts are fast enough to do a lot of actions during the duration of a CAN frame (which is the duration of a time window).

Timing according ISO 11898-4 level 1 is not perfect. The cycle time counts not synchronously with the CAN bits, because a CAN bit can be shortened or lengthened because of CAN synchronization. Therefore the cycle time of one node may differ compared to the cycle time of other nodes.
In many cases this will be +/- 1 tick but it is not limited to this. The begin of a new basic cycle with the transmission of a reference message will resynchronize the nodes.

If a transmission trigger becomes active, then the appropriate transmission can only be started with the next CAN bit. Therefore the earliest point of transmission of the SOF of the frame will be at TT_TRIG+1.

16.9.4 TTCAN Trigger Types

The trigger type is defined by TTYPE. TTPTR is a pointer to a TB message slot and TT_TRIG defines the cycle time of the trigger.
Triggers and the related actions shall finish before the maximum cycle time 0xFFFF is reached as this is the maximum length of a basic cycle.

Except for the immediate trigger, all triggers set TTIF if TTIE is enabled If TTTBM=1, only time triggers are supported. Using other triggers in this mode will result in setting TEIF.
If a trigger is activated after a write access to TRIGCFG[31:24] bits, then write access to TRIGCFG[31:0] bits are locked until the trigger time is reached (TTIF is set if TTIE is set) or an error is detected (TEIF is set). Therefore no new trigger can override an active trigger. The write lock is also removed if TTEN=0.

If TTPTR of an immediate trigger points to an empty slot, then TEIF is set.

16.9.4.1 Immediate Trigger

An immediate trigger starts the immediate transmission of a frame that is pointed to by TTPTR. TTIF is not set. To start a trigger, TRIGCFG[31:16] bits need to be written. The value that is written, does not care for an immediate trigger.

In TTCAN mode, TPE, TSONE, TSALL cannot be used. The immediate trigger is the replacement. Only the transmission of one frame can be started with an immediate trigger. The host must not command a second immediate trigger before the first transmission has been completed successfully or unsuccessfully.

For an immediate trigger the single shot mode can be selected by TSSS(TSIF is set if TSIE is set) . A transmission can be aborted using TSA (TPSS and TPA have no meaning.)

If TTPTR of an immediate trigger points to an empty slot, then TEIF is set.

✸ ***Note: In immediate trigger mode, TRIGCFG bits need to be written before setting TTPTR bits.***

16.9.4.2 Time Trigger

A time trigger just generates an event by setting TTIF and therefore generates an interrupt. No other actions are done.

The time trigger can be used as a reception trigger. If the node expects a message to be received in a time window, then if the message is missing and RIF is not set, a reception trigger can be used to signal this.
A reception trigger shall be set after the latest moment when the message is expected to be successfully received.

If TT_TRIG is lower than the actual cycle time, then TEIF is set.
A time trigger can be used if TTTBM=1. This is the only trigger type that is available in this mode.

16.9.4.3 Single Shot Transmit Trigger

A single shot transmit trigger is intended to be used for exclusive time windows, where a message needs to be transmitted in single shot mode. The selected message is defined by TTPTR.

Single shot mode is automatically used regardless of the stat e of TSSS. The register bit TSSS is ignored and stays unchanged.

Single shot transmit triggers are intended to be used for exclusive time windows. For this ISO 11898-4 defines a transmit enable window of up to 16 ticks of the cycle time. The register bits TEW[3:0]+1 define the number of ticks. A frame cannot be started if the bus is occupied by another frame.
This should not happen in an exclusive time window, but the transmit enable window ensures that there is no delayed start which would result in a violation of the next time window.

If the transmit enable window closes and the frame could not be started, then it is aborted. As a result the TB slot of the frame will be marked as empty and AIF will be set if AIE is set. The data of the frame in the TB slot will not be touched and therefore the slot only needs to be marked as filled again if the same data shall be transmitted in a next attempt.

If TT_TRIG is lower than the actual cycle time, then TEIF is set and no action is done.

16.9.4.4 Transmit Start Trigger

A transmit start trigger is intended to be used for merged arbitrating time windows, where several nodes may transmit messages and CAN arbitration takes place. The selected message is defined by TTPTR.

TSSS defines if retransmission or single shot mode is used.

If the selected frame cannot be transmitted (arbitration loss, several transmissions after an error), then the transmission can be stopped using the transmit stop trigger.

If TT_TRIG is lower than the actual cycle time, then TEIF is set and no action is done.

16.9.4.5 Transmit Stop Trigger

A transmit stop trigger is intended to be used to stop a transmission, that is started with the transmit start trigger.

If a transmission is stopped, then the frame is aborted. Therefore AIF is set if AIE is set and the TB slot of the frame is marked as empty The data of the frame in the TB slot will not be touched and therefore the slot only needs to marked as filled again if the same data shall be transmitted in a next attempt.

The behavior of a transmit stop trigger is similar to the end of the transmit enable window used by the single shot transmit trigger. If a transmit stop trigger points to an empty slot (which means, that the message has already been transmitted), then not action is done an TEIF is not set.

If TT_TRIG is lower than the actual cycle time, then TEIF is set but the stop is executed.

16.9.5 TTCAN Watch Trigger

In contrast to the generic trigger types explained in TTCAN Trigger Types chapter the watch trigger has a dedicated interrupt flag WTIF. If the cycle count equal to the value defined by TT_WTRIG, then WTIF is set if WTIE is set.

The watch trigger is intended to be used if the time since the last valid reference message has been too long. Reference messages can be received in a periodic cycle or after an event. The host application needs to take care of this and has to adjust the watch trigger accordingly.

The default value of WTIE is 1 and therefore the default watch trigger 0xFFFF is automatically valid. To turn off the watch trigger, WTIE needs to be set to 0.

If TT_WTRIG is updated and it is lower than the actual cycle time, then TEIF is set.

The watch trigger can be used if TTTBM=1.

16.10 CAN Bit Time

16.10.1 Data Bit Rates

CAN 2.0 defines data bit rates up to 1Mbit/s. The CAN bus controller core can be programmed to arbitrarily chosen data rates only limited by the range of the bit settings in the appropriate bit timing and prescaler registers.

16.10.2 Definitions

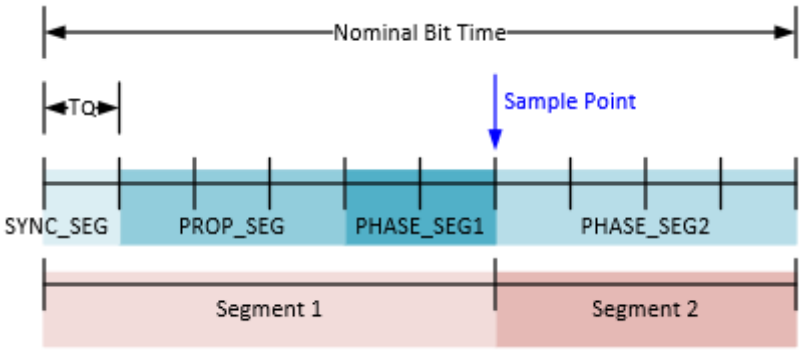


Figure 1. CAN Bit Timing Specifications

BT : The CAN bit time
BR : The CAN bus baud rate

The CAN bit time consists of several segments. Each segment consists of a number of time quanta units n_{TQ}.

The duration of a time quanta T_Q is:

$$T_Q = \frac{n_{prescaler}}{F_{CLOCK}}$$

The values of n_{TQ} and $n_{prescaler}$ have to be chosen depending on the system clock frequency F_{CLOCK} to match BT_{real} as close as possible to $BT_{ideal} = 1/BR$:

$$BT_{ideal} \approx BT_{real} = \frac{n_{prescaler} \times n_{TQ}}{F_{CLOCK}} = t_{seg\_1} + t_{seg\_2}$$

The CAN specification requires several relationships between the segment lengths which results in relationships between t_{seg_1}, t_{seg_2} and the maximum synchronization jump width t_{SJW}.
Please note that Table. CAN Timing Segments lists the minimum configuration ranges defined by the CAN specification.

Table 1. CAN Timing Segments (Minimum Configuration Ranges)

Segment	Description	Minimum Configuration Ranges
SYNC_SEG	Synchronization Segment	1 T_Q

PROP_SEG	Propagation Segment	1 ~ 8 T_Q
PHASE_SEG1	Phase Buffer Segment 1	1 ~ 8 T_Q
PHASE_SEG2	Phase Buffer Segment 2	2 ~ 8 T_Q
SJW	Synchronization Jump Width	1 ~ 4 T_Q
IPT	Information Processing Time	0 ~ 2 T_Q , PHASE_SEG2 ≥ IPT

The CAN bus controller collects SYNC_SEG, PROP_SEG and PHASE_SEG1 into one group and the length of the group is configurable with t_{seg_1}. Table 2 lists the available configuration ranges.

Please note that the CAN bus controller core does not check if all rules are met and offers a wider configuration range than defined by the CAN 2.0 specification.

Table 2. CAN bus controller Timing Settings (Available Configuration ranges)

Setting	Requirements
t_{seg_1}	2 ~ 65 T_Q
t_{seg_2}	1 ~ 8 T_Q , $t_{seg\_1} \geq t_{seg\_2} + 2$
t_{SJW}	1 ~ 16 T_Q , $t_{seg\_2} \geq t_{SJW}$

For bit rate the register settings S_SEG1, S_SEG2, S_SJW and S_PRESC define the appropriate segment lengths.

$$t_{seg\_1} = (\text{S\_SEG1} + 2) \times T_Q$$
$$t_{seg\_2} = (\text{S\_SEG2} + 1) \times T_Q$$
$$t_{SJW} = (\text{S\_SJW} + 1) \times T_Q$$
$$n_{prescaler} = \text{S\_PRESC} + 1$$

An illustration a suitable bit rate configuration is given in the following figure where $F_{tq\_clk} = \frac{F_{CLOCK}}{n_{prescaler}}$.

The bit time BT_{real}, the sample point and the synchronization jump width SJW will be derived from F_{tq_clk}.

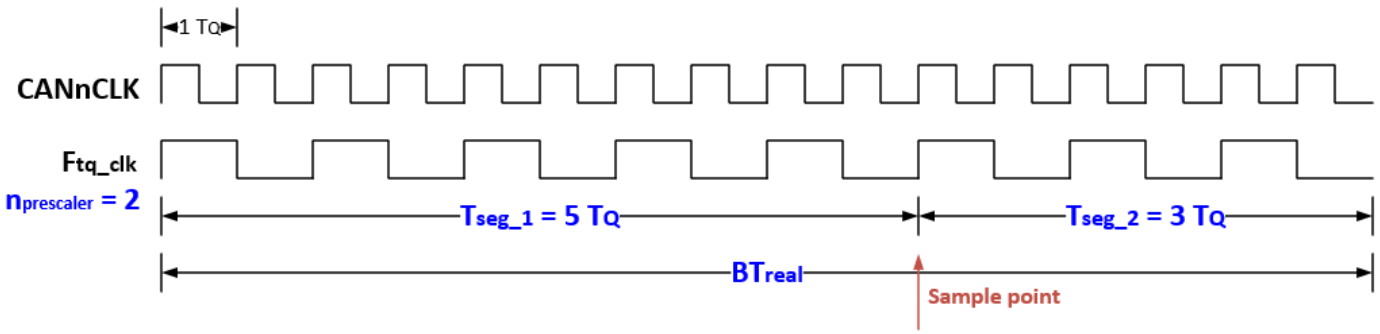


Figure 2. Clock Division Example for Bit Sampling

Having the requirements from Table 2 , the host controller has to define the length of segment 1, segment 2 and the SJW for the slow bit rate. Some suggestions below:

- Segment 1 must be slightly larger than segment 2. Then the sample point is in the a little bit later than in the middle of the bit time.

- The SJW must not be bigger than segment 2.
  - If SJW is too small, the CAN node may be too slow to resynchronize.
  - if SJW is too big, the CAN node may resynchronize too often.
  - SJW being half as long as segment 2 seems to be a suitable value.
- All CAN nodes connected to a CAN bus should choose similar settings if possible.

16.10.3 Example Configuration

This example refers to Figure 2. It is an example for CAN 2.0 slow bit rate configuration. The following steps need to carried for the configuration of the CAN bus controller:

1. Set bit RESET=1.
2. Set registers S_SEG1 and S_SEG2:
   In the example the data rate on the bus $F_{bus} = 1M\ Baud$ and the system clock is 16 MHz.
   The values of n_{TQ} and $n_{prescaler}$ have to be selected to fit real BT_{real} as close as possible to BT_{ideal} .
   In this example $n_{prescaler} = 2$ and $n_{TQ} = 8$ are chosen which results in a perfect match:
   $BT_{ideal} = BT_{real} = 8T_Q$
   With $BT_{real} = t_{seg\_1} + t_{seg\_2}$ and the time segment definitions given in Chapter Data Bit Rates $t_{seg\_1} = 5T_Q$ and $t_{seg\_2} = 3T_Q$ can be chosen as suitable values which finally results in the register settings S_SEG1 = 3 and S_SEG2 = 2.
3. Load the acceptance code and mask registers (optional).
4. Set register S_SJW:
   With $t_{seg\_2} \geq t_{SJW}$ one is free to chose $t_{SJW} = 3$ which finally results in S_SJW = 2.
5. Load the clock prescaler register S_PRESC:
   $n_{prescaler} = S\_PRESC + 1$ results in S_PRESC = 1.
6. Set bit RESET=0.
7. Continue with configuration of interrupts other configuration bits and execute commands.

✸ ***Note:***
The given order is not mandatory. It is merely necessary to set bit RESET=1 at the beginning, as it is otherwise not possible to load the bit timing, ACODE and AMASK registers. RESET=0 is required upon completion of the configuration.

16.11 CAN Registers

Base Address: 0x4001 9000 (CAN 0)
0x4001 A000 (CAN 1)

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	CANn_RBID	CANn Receive Buffer ID register
0x0004	CANn_RBSTA	CANn Receive Buffer Status register
0x0008	CANn_RBDATA1	CANn Receive Buffer Data 1 register
0x000C	CANn_RBDATA2	CANn Receive Buffer Data 2 register
0x0010 – 0x004C	–	Reserved
0x0050	CANn_TBID	CANn Transmit Buffer ID register
0x0054	CANn_TBSTA	CANn Transmit Buffer Status register
0x0058	CANn_TBDATA1	CANn Transmit Buffer Data 1 register
0x005C	CANn_TBDATA2	CANn Transmit Buffer Data 2 register
0x0060 – 0x009C	–	Reserved
0x00A0	CANn_SYSCTRL	CANn System Control register
0x00A4	CANn_INTCFG	CANn Interrupt Configuration register
0x00A8	CANn_SBITCFG	CANn Classic Bit Timing Configuration register
0x00AC	–	Reserved
0x00B0	CANn_ERRSTA	CANn Error Status register
0x00B4	CANn_ACFCTRL	CANn Acceptance Filter Control register
0x00B8	CANn_ACFID	CANn Acceptance Filter Identifier register
0x00BC	CANn_TTCTRL	CANn TTCAN Control register
0x00C0	CANn_REFID	CANn TTCAN Reference Message ID register
0x00C4	CANn_TRIGCFG	CAN0 TRIGCFG register
0x00C8	CANn_TRIGWTR	CANn TTCAN Watch Trigger Time register
0x00CC – 0x00D0	–	Reserved
0x00D4	CANn_PINCTRL	CANn Pin Control register

16.11.1 CAN n Receive Buffer ID register (CANn_RBID) (n=0,1)

Address Offset: 0x00

Bit	Field	Access	Initial	Description
31:29	–	–	–	Reserved
28:0	ID	R	–	Frame Identifier If IDE = 0 , ID(10:0) is valid (standard ID) If IDE = 1 , ID(28:0) is valid (extended ID)

✷ ***Note: RBUF is composed of SRAM, so the initial value cannot be given through reset.***

16.11.2 CAN n Receive Buffer Status register (CANn_RBSTA) (n=0,1)

Address Offset: 0x04

Bit	Field	Access	Initial	Description
31:16	CYCLE_TIME	R	–	Cycle time (time-stamp for TTCAN) The time-stamp CYCLE_TIME will be stored in RBUF only in TTCAN mode. This is the cycle time at the SOF of the frame. The cycle time of a reference message is always 0.
15:13	KOER	R	–	Kind of error

Bit	Field	Access	Initial	Description
				KOER for received frames has the same meaning as the bits KOER in register ERRSTA. KOER for received frames becomes meaningful if RBALL=1. 0: No error 1: Bit Error 2: FORM Error 3: STUFF Error 4: ACK Error 5: CRC Error 6: OTHER Error Other: Reserved
12	TX	R	–	Own transmitted frame Status bit TX in RBUF is set to 1 if the loop back mode is activated and the core has received its own transmitted frame. This can be useful if LBME=1 and other nodes in the network do also transmissions. 0: No received 1: Received the frame sent by myself
11:8	–	–	–	Reserved
7	IDE	R	–	IDentifier type 0: Standard Format, ID[10:0] 1: Extended Format, ID[28:0]
6	RTR	R	–	Remote transmission request 0: Data frame 1: Remote frame
5:4	–	–	–	Reserved
3:0	DLC	R	–	Data length code DLC in RBUF defines the length of the number of payload bytes in a frame. 0: The payload is 0 byte 1: The payload is 1 byte 2: The payload is 2 bytes 3: The payload is 3 bytes 4: The payload is 4 bytes 5: The payload is 5 bytes 6: The payload is 6 bytes 7: The payload is 7 bytes 8~15: The payload is 8 bytes

✸ ***Note: RBUF is composed of SRAM, so the initial value cannot be given through reset.***

16.11.3 CAN n Receive Buffer data 1 register (CANn_RBDATA1) (n=0,1)

Address Offset: 0x08

Bit	Field	Access	Initial	Description
31:24	D4	R	–	Payload data 4 of the frame
23:16	D3	R	–	Payload data 3 of the frame
15:8	D2	R	–	Payload data 2 of the frame
7:0	D1	R	–	Payload data 1 of the frame

✸ ***Note:***
1. ***RBUF is composed of SRAM, so the initial value cannot be given through reset.***
2. ***Unused payload data in RBUF do contain data which needs to be ignored.***

16.11.4 CAN n Receive Buffer Data 2 register (CANn_RBDATA2) (n=0,1)

Address Offset: 0x0C

Bit	Field	Access	Initial	Description
31:24	D8	R	–	Payload data 8 of the frame
23:16	D7	R	–	Payload data 7 of the frame
15:8	D6	R	–	Payload data 5 of the frame
7:0	D5	R	–	Payload data 6 of the frame

✸ ***Note:***
1. ***RBUF is composed of SRAM, so the initial value cannot be given through reset.***
2. ***Unused payload data in RBUF do contain data which needs to be ignored.***

16.11.5 CAN n Transmit Buffer ID register (CANn_TBID) (n=0,1)

Address Offset: 0x50

Bit	Field	Access	Initial	Description
31:29	–	–	–	Reserved
28:0	ID	RW	–	Frame Identifier If IDE = 0 , ID(10:0) is valid (standard ID) If IDE = 1 , ID(28:0) is valid (extended ID)

✸ ***Note: TBUF is composed of SRAM, so the initial value cannot be given through reset.***

16.11.6 CAN n Transmit Buffer Status register (CANn_TBSTA) (n=0,1)

Address Offset: 0x54

Bit	Field	Access	Initial	Description
31:8	–	–	–	Reserved
7	IDE	RW	–	IDentifier type 0: Standard Format, ID[10:0] 1: Extended Format, ID[28:0]
6	RTR	RW	–	Remote transmission request 0: Data frame 1: Remote frame
5	ITB	RW	–	Used for internal testing and the only value “0” is allowed.
4	–	–	–	Reserved
3:0	DLC	RW	–	Data length code DLC in TBUF defines the length of the number of payload bytes in a frame. 0: The payload is 0 byte 1: The payload is 1 byte 2: The payload is 2 bytes 3: The payload is 3 bytes 4: The payload is 4 bytes 5: The payload is 5 bytes 6: The payload is 6 bytes 7: The payload is 7 bytes 8~15: The payload is 8 bytes

✸ ***Note: TBUF is composed of SRAM, so the initial value cannot be given through reset.***

16.11.7 CAN n Transmit Buffer Data 1 register (CANn_TBDATA1) (n=0,1)

Address Offset: 0x58

Bit	Field	Access	Initial	Description
31:24	D4	RW	–	Payload data 4 of the frame
23:16	D3	RW	–	Payload data 3 of the frame
15:8	D2	RW	–	Payload data 2 of the frame
7:0	D1	RW	–	Payload data 1 of the frame

✸ ***Note:***
1. ***TBUF is composed of SRAM, so the initial value cannot be given through reset.***
2. ***Unused payload data in TBUF do contain data which needs to be ignored.***

16.11.8 CAN n Transmit Buffer Data 2 register (CANn_TBDATA2) (n=0,1)

Address Offset: 0x5C

Bit	Field	Access	Initial	Description
31:24	D8	RW	–	Payload data 8 of the frame
23:16	D7	RW	–	Payload data 7 of the frame
15:8	D6	RW	–	Payload data 5 of the frame
7:0	D5	RW	–	Payload data 6 of the frame

✸ ***Note:***
1. ***TBUF is composed of SRAM, so the initial value cannot be given through reset.***
2. ***Unused payload data in TBUF do contain data which needs to be ignored.***

16.11.9 CAN n System Control register (CANn_SYSCTRL) (n=0,1)

Address Offset: 0xA0

Setting both TSONE and TSALL is meaningless.

While TSALL is already set, it is impossible to set TSONE and vice versa.

If both TSONE and TSALL are set simultaneously then TSALL wins and TSONE is cleared by CAN bus controller.

Bit	Field	Access	Initial	Description
31	SACK	RW	0	Self-ACKnowledge (SACK can write 1 only when RESET is 0.) 0: No self-ACK 1: Self-ACK when LBME=1
30	ROM	RW	0	Receive buffer overflow mode 0: The oldest frame will be overwritten 1: The new frame will not be stored
29	ROV	R	0	Receive buffer overflow ROV is cleared by setting RREL=1. 0: No overflow 1: At least one frame is lost
28	RREL	RW	0	Receive buffer release The host controller has read the actual RB slot and releases it. Afterwards the CAN bus controller points to the next RB slot.

Bit	Field	Access	Initial	Description
				0: No release 1: The host had read the RB
27	RBALL	RW	0	Receive buffer stores all data frames 0: Normal operation 1: RB stores correct data frames as well as data frames with error
26	–	–	0	Reserved
25:24	RSTAT	R	0	Receive buffer status 0: Empty 1: Greater than empty and less than AFWL 2: Greater than or equal to AFWL, but not full and no overflow 3: Full (stays set in case of overflow – for overflow signaling see ROV)
23	–	–	0	Reserved
22	TSNEXT	RW	0	Transmit buffer secondary next 0: No action 1: STB slot filled, select next slot After all frame bytes are written to the TBUF registers, the host controller has to set TSNEXT to signal that this slot has been filled. Then CAN bus controller connects the TBUF registers to the next slot.
21	TSMODE	RW	0	Transmit buffer secondary operation mode 0: FIFO mode 1: Priority decision mode In FIFO mode frames are transmitted in the order in that they are written into the STB. In priority decision mode the frame with the highest priority in the STB is automatically transmitted first. The ID of a frame is used for the priority decision. A lower ID means a higher priority of a frame. A frame in the PTB has always the highest priority regardless of the ID. TSMODE shall be switched only if the STB if empty.
20	TTTBM	RW	1	TTCAN transmit buffer mode 0: Separate PTB and STB, behavior defined by TSMODE 1: Full TTCAN support: buffer slots selectable by TBPTR and TTPTR TTTBM shall be switched only if the TBUF is empty.
19:18	–	–	0	Reserved
17:16	TSTAT	R	0	Transmission secondary status bits 0: When TTEN = 1 and TTTBM = 1, PTB and STB are empty, otherwise STB is empty 1: When TTEN = 1 and TTTBM = 1, PTB and STB are not empty and not full, otherwise STB is less than or equal to half full 2: When TTEN = 0 or TTTBM = 0, STB is more than half full 3: When TTEN = 1 and TTTBM = 1, PTB and STB are full, otherwise STB is full The update of TSSTAT after setting TSNEXT=1 takes a one additional cycle. Therefore reading TSSTAT must not be done immediately after setting TSNEXT=1.
15	TBSEL	RW	0	Transmit buffer select Selects the transmit buffer to be loaded with a frame. 0: PTB (high-priority buffer) 1: STB The bit will be reset to the hardware reset value if (TTEN=1 and TTTBM=1).
14	LOM	RW	0	Listen-Only mode 0: Disable 1: Enable
13	STBY	RW	0	Transceiver standby mode This register bit is connected to the output signal CAN_STBYn which can be used to control a standby mode of a transceiver. 0: Disable 1: Enable
12	TPE	RW	0	Transmit primary enable 0: No transmission for the PTB

Bit	Field	Access	Initial	Description
				1: Transmission enable for the message in the high-priority PTB
11	TPA	RW	0	Transmit primary abort 0: No abort 1: Aborts a transmission from PTB which has been requested by TPE=1 but not started yet. (The data bytes of the message remains in the PTB.)
10	TSONE	RW	0	Transmit secondary one frame 0: No transmission for the STB 1: Transmission enable of one in the STB. TSONE stays set until the message has been transmitted successfully or it is aborted using TSA. In FIFO mode this is the oldest message and in priority mode this is the one with the highest priority. TSONE in priority mode is difficult to handle, because it is not always clear which message will be transmitted if new messages are written to the STB meanwhile.
9	TSALL	RW	0	Transmit secondary all frames 0: No transmission for the STB 1: Transmission enable of all messages in the STB. TSALL stays set until all messages have been transmitted successfully or they are aborted using TSA.
8	TSA	RW	0	Transmit secondary abort 0: No abort 1: Aborts a transmission from STB which has been requested but not started yet. For a TSONE transmission, only one frame is aborted while for a TSALL transmission, all frames are aborted. One or all message slots will be released which updates TSSTAT. All aborted messages are lost because they are not accessible any more. If in priority mode, a TSONE transmission is aborted, then it is not clear which frame will be aborted if new frames are written to the STB meanwhile.
7	RESET	RW	1	Reset request bit 0: No reset request to CAN bus controller 1: Issue reset request to CAN bus controller The some register (e.g. node configuration) can only be modified if RESET=1. Bit RESET forces several components to a reset state.
6	LBME	RW	0	External loop back mode 0: Disable 1: Enable
5	LBMI	RW	0	Internal loop back mode 0: Disable 1: Enable
4	TPSS	RW	0	Transmission primary single shot mode for PTB 0: Disable 1: Enable
3	TSSS	RW	0	Transmission secondary single shot mode for STB 0: Disable 1: Enable
2	RACTIVE	R	0	Reception active status bit 0: No receive activity 1: The controller is currently receiving a frame
1	TACTIVE	R	0	Transmission active status bit 0: No transmit activity 1: The controller is currently transmitting a frame
0	BUSOFF	RW	0	Bus off status bit 0: The controller status is “Bus On” 1: The controller status is “Bus Off”

16.11.10 CAN n Interrupt Configuration register (CANn_INTCFG) (n=0,1)

Address Offset: 0xA4

To reset an interrupt flag, the host controller needs to write a 1 to the flag. Writing a 0 has no effect.

If a new interrupt event occurs while the write access is active then this event will set the flag and override the reset. This ensures that no interrupt event is lost.

Interrupt flags will only be set if the associated interrupt enable bit is set.

Bit	Field	Access	Initial	Description
31:28	AFWL	RW	1	Receive buffer almost full warning limit AFWL is compared to the number of filled RB slots and triggers RAFIF if equal. 0: Almost Full Warning Limit = 1 1: Almost Full Warning Limit = 1 2: Almost Full Warning Limit = 2 3: Almost Full Warning Limit = 3 4: Almost Full Warning Limit = 4 5: Almost Full Warning Limit = 5 6: Almost Full Warning Limit = 6 7: Almost Full Warning Limit = 7 8: Almost Full Warning Limit = 8 9: Almost Full Warning Limit = 9 10: Almost Full Warning Limit = 10 11: Almost Full Warning Limit = 11 12: Almost Full Warning Limit = 12 13: Almost Full Warning Limit = 13 14: Almost Full Warning Limit = 14 15: Almost Full Warning Limit = 15. This is a valid value, but note that RFIF also exists
27:24	EWL	RW	0xB	Programmable error warning limit = (EWL+1)*8 0: Error Warning Limit = 8 1: Error Warning Limit = 16 2: Error Warning Limit = 24 3: Error Warning Limit = 32 4: Error Warning Limit = 40 5: Error Warning Limit = 48 6: Error Warning Limit = 56 7: Error Warning Limit = 64 8: Error Warning Limit = 72 9: Error Warning Limit = 80 10: Error Warning Limit = 88 11: Error Warning Limit = 96 12: Error Warning Limit = 104 13: Error Warning Limit = 112 14: Error Warning Limit = 120 15: Error Warning Limit = 128
23	EWARN	R	0	Error warning limit reached 0: The values in both counters are less than EWL 1: One of the error counters RECENT or TECENT is equal to or greater than EWL
22	EPASS	R	0	Error passive mode active 0: Not active (node is error active) 1: Active (node is error passive)
21	EPIE	RW	0	Error passive interrupt enable bit 0: Disable 1: Enable
20	EPIF	RW	0	Error passive interrupt flag 0: No flag 1: The error status changes from error active to error passive or vice versa and if EPIE=1
19	ALIE	RW	0	Arbitration lost interrupt enable bit 0: Disable 1: Enable

Bit	Field	Access	Initial	Description
18	ALIF	RW	0	Arbitration lost interrupt flag 0: No arbitration lost 1: Arbitration lost
17	BEIE	RW	0	Bus error interrupt enable bit 0: Disable Bus Error interrupt 1: Enable Bus Error interrupt
16	BEIF	RW	0	Bus error interrupt flag 0: No bus error 1: An error occurred on the bus
15	RIF	RW	0	Receive interrupt flag 0: No frame has been received 1: Data or a remote frame has been received and is available in the receive buffer
14	ROIF	RW	0	RB overflow interrupt flag 0: No RB overwritten 1: At least one received frame has been overwritten in the RB. In case of an overrun both ROIF and RFIF will be set.
13	RFIF	RW	0	RB full interrupt flag 0: The RB FIFO is not full 1: All RBs are full.
12	RAFIF	RW	0	RB almost full interrupt flag 0: The number of filled RB slots < AFWL 1: The number of filled RB slots ≧ AFWL
11	TPIF	RW	0	Transmission primary interrupt flag 0: No transmission of the PTB has been completed 1: The requested transmission of the PTB has been successfully completed In TTCAN mode, TPIF will never be set. Then only TSIF is valid.
10	TSIF	RW	0	Transmission secondary interrupt flag 0: No transmission of the STB has been completed successfully 1: The requested transmission of the STB has been successfully completed In TTCAN mode, TSIF will signal all successful transmissions, regardless of storage location of the frame.
9	EIF	RW	0	Error interrupt flag 0: There has been no change 1: The EWL has been crossed in either direction, or the BUSOFF bit has been changed in either direction
8	AIF	RW	0	Abort interrupt flag 0: No abort has been executed 1: After setting TPA or TSA the appropriated frame(s) have been aborted. (The AIF does not have an associated enable register)
7	RIE	RW	1	Receive interrupt enable bit 0: Disable 1: Enable
6	ROIE	RW	1	RB overflow interrupt enable bit 0: Disable 1: Enable
5	RFIE	RW	1	RB full Interrupt enable bit 0: Disable 1: Enable
4	RAFIE	RW	1	RB almost full interrupt enable bit 0: Disable 1: Enable
3	TPIE	RW	1	Transmission primary interrupt enable bit 0: Disable 1: Enable
2	TSIE	RW	1	Transmission secondary interrupt enable bit 0: Disable 1: Enable
1	EIE	RW	1	Error interrupt enable bit

Bit	Field	Access	Initial	Description
				0: Disable 1: Enable
0	TSFF	RW	0	Transmit secondary buffer slot full flag 0: When TTEN = 1 and TTTBM = 1, the buffer slot selected by TBPTR is empty. Otherwise, the STB is not filled with the maximal number of messages 1: When TTEN = 1 and TTTBM = 1, the buffer slot selected by TBPTR is filled. Otherwise, the STB is filled with the maximal number of messages The update of TSFF after setting TSNEXT=1 takes a one additional cycle. Therefore reading TSFF must not be done immediately after setting TSNEXT=1.

16.11.11 CAN n Bit Timing Configuration register (CANn_SBITCFG) (n=0,1)

Address Offset: 0xA8

The register can only be written if bit RESET in register CTRL is set.

Bit	Field	Access	Initial	Description
31:24	S_PRESC	RW	1	Prescaler setting The prescaler divides the system clock to synthesize the time quanta T_Q. $n_{prescaler} = \text{S\_PRESC} + 1$ and $T_Q = \frac{n_{prescaler}}{F_{CLOCK}}$
23	–	–	0	Reserved
22:16	S_SJW	RW	0x2	Synchronization jump width setting $t_{SJW} = (\text{S\_SJW} + 1) \times T_Q$
15	–	–	0	Reserved
14:8	S_SEG2	RW	0x2	Bit timing of segment 2 $t_{seg\_2} = (\text{S\_SEG2} + 1) \times T_Q$
7:0	S_SEG1	RW	0x3	Bit timing of segment 1 $t_{seg\_1} = (\text{S\_SEG1} + 2) \times T_Q$

16.11.12 CAN n Error Status register (CANn_ERRSTA) (n=0,1)

Address Offset: 0xB0

Bit	Field	Access	Initial	Description
31:24	TECNT	R	0	Transmit error count (number of errors during transmission) TECNT is incremented and decremented as defined in the CAN specification.
23:16	RECNT	R	0	Receive error count (number of errors during reception) RECNT is incremented and decremented as defined in the CAN specification.
15:8	–	–	0	Reserved
7:5	KOER	R	0	Kind of error (Error code) 0: No Error 1: Bit Error 2: Form Error 3: Stuff Error 4: Acknowledgement Error 5: CRC Error 6: Other Error (dominant bits after own error flag, received active Error Flag too long, dominant bit during Passive-Error-Flag after ACK error) 7: Not Used
4:0	ALC	R	0	Arbitration lost capture (bit position in the frame where the arbitration has been lost)

Bit	Field	Access	Initial	Description
				0: Arbitration lost bit = ID10/ID28 (2.0 Standard/Extended) 1: Arbitration lost bit = ID9/ID27 (2.0 Standard/Extended) 2: Arbitration lost bit = ID8/ID26 (2.0 Standard/Extended) 3: Arbitration lost bit = ID7/ID25 (2.0 Standard/Extended) 4: Arbitration lost bit = ID6/ID24 (2.0 Standard/Extended) 5: Arbitration lost bit = ID5/ID23 (2.0 Standard/Extended) 6: Arbitration lost bit = ID4/ID22 (2.0 Standard/Extended) 7: Arbitration lost bit = ID3/ID21 (2.0 Standard/Extended) 8: Arbitration lost bit = ID2/ID20 (2.0 Standard/Extended) 9: Arbitration lost bit = ID1/ID19 (2.0 Standard/Extended) 10: Arbitration lost bit = ID0/ID18 (2.0 Standard/Extended) 11: Arbitration lost bit = RTR/SRR (2.0 Standard/Extended) 12: Arbitration lost bit = IDE (2.0 Extended) 13: Arbitration lost bit = ID17 (2.0 Extended) 14: Arbitration lost bit = ID16 (2.0 Extended) 15: Arbitration lost bit = ID15 (2.0 Extended) 16: Arbitration lost bit = ID14 (2.0 Extended) 17: Arbitration lost bit = ID13 (2.0 Extended) 18: Arbitration lost bit = ID12 (2.0 Extended) 19: Arbitration lost bit = ID11 (2.0 Extended) 20: Arbitration lost bit = ID10 (2.0 Extended) 21: Arbitration lost bit = ID9 (2.0 Extended) 22: Arbitration lost bit = ID8 (2.0 Extended) 23: Arbitration lost bit = ID7 (2.0 Extended) 24: Arbitration lost bit = ID6 (2.0 Extended) 25: Arbitration lost bit = ID5 (2.0 Extended) 26: Arbitration lost bit = ID4 (2.0 Extended) 27: Arbitration lost bit = ID3 (2.0 Extended) 28: Arbitration lost bit = ID2 (2.0 Extended) 29: Arbitration lost bit = ID1 (2.0 Extended) 30: Arbitration lost bit = ID0 (2.0 Extended) 31: Arbitration lost bit = RTR (2.0 Extended)

16.11.13 CAN n Acceptance Filter Control register (CANn_ACFCTRL) (n=0,1)

Address Offset: 0xB4

The register can only be written if bit RESET in register SYSCTRL is set.

Each acceptance filter (AMASK / ACODE) can be individually enabled or disabled.

Only filter number 0 is enabled by default after hardware reset.

To accept all messages one filter x has to be enabled by setting AE_x=1, AMASK_x=0xff and ACODE_x=0x00. This is the default configuration after hardware reset for filter x=0 while all other filters are disabled.

Bit	Field	Access	Initial	Description
31	AE_15	RW	0	Acceptance filter 15 enable bit 0: Disable 1: Enable
30	AE_14	RW	0	Acceptance filter 14 enable bit 0: Disable 1: Enable
29	AE_13	RW	0	Acceptance filter 13 enable bit 0: Disable 1: Enable
28	AE_12	RW	0	Acceptance filter 12 enable bit 0: Disable 1: Enable
27	AE_11	RW	0	Acceptance filter 11 enable bit 0: Disable 1: Enable
26	AE_10	RW	0	Acceptance filter 10 enable bit

Bit	Field	Access	Initial	Description
				0: Disable 1: Enable
25	AE_9	RW	0	Acceptance filter 9 enable bit 0: Disable 1: Enable
24	AE_8	RW	0	Acceptance filter 8 enable bit 0: Disable 1: Enable
23	AE_7	RW	0	Acceptance filter 7 enable bit 0: Disable 1: Enable
22	AE_6	RW	0	Acceptance filter 6 enable bit 0: Disable 1: Enable
21	AE_5	RW	0	Acceptance filter 5 enable bit 0: Disable 1: Enable
20	AE_4	RW	0	Acceptance filter 4 enable bit 0: Disable 1: Enable
19	AE_3	RW	0	Acceptance filter 3 enable bit 0: Disable 1: Enable
18	AE_2	RW	0	Acceptance filter 2 enable bit 0: Disable 1: Enable
17	AE_1	RW	0	Acceptance filter 1 enable bit 0: Disable 1: Enable
16	AE_0	RW	1	Acceptance filter 0 enable bit 0: Disable 1: Enable
15:6	–	–	0	Reserved
5	SELMASK	RW	0	Select acceptance code or mask 0: Registers ACFID point to acceptance code 1: Registers ACFID point to acceptance mask
4	–	–	0	Reserved
3:0	ACFADR	RW	0	Acceptance filter address ACFADR points to a specific acceptance filter. The selected filter is accessible using the registers ACFID. 0: Acceptance filter 0 1: Acceptance filter 1 2: Acceptance filter 2 3: Acceptance filter 3 4: Acceptance filter 4 5: Acceptance filter 5 6: Acceptance filter 6 7: Acceptance filter 7 8: Acceptance filter 8 9: Acceptance filter 9 10: Acceptance filter 10 11: Acceptance filter 11 12: Acceptance filter 12 13: Acceptance filter 13 14: Acceptance filter 14 15: Acceptance filter 15

16.11.14 CAN n Acceptance Filter Identifier register (CANn_ACFID) (n=0,1)

Address Offset: 0xB8

The register can only be written if bit RESET=1.

Only filter 0 is affected by the power-on reset. All other filters stay uninitialized.

- **If SELMASK=0, ACFID is Acceptance code buffer:**

Bit	Field	Access	Initial	Description
31:29	–	–	0	Reserved
28:0	ACFBUF	RW	0	Acceptance code buffer for Identifier 0: ACC bit value to compare with ID bit of the received message 1: ACC bit value to compare with ID bit of the received message ACODE_x[10:0] will be used for standard frames. ACODE_x[28:0] will be used for extended frames.

- **If SELMASK=1, ACFID is Acceptance mask buffer:**

Bit	Field	Access	Initial	Description
31	–	–	0	Reserved
30	AIDEE	RW	0	Acceptance mask IDE bit check enable 0: Acceptance filter accepts both standard or extended frames 1: Acceptance filter accepts either standard or extended as defined by AIDE
29	AIDE	RW	0	Acceptance mask IDE bit value 0: Acceptance filter accepts only standard frames 1: Acceptance filter accepts only extended frames
28:0	ACFBUF	RW	0x1FFFFFFF	Acceptance mask buffer for Identifier (AMASK_x) 0: Acceptance check for these bits of receive identifier enable 1: Acceptance check for these bits of receive identifier disabled AMASK_x[10:0] will be used for standard frames. AMASK_x[28:0] will be used for extended frames. Disabled bits result in accepting the message. Therefore the default configuration after reset for filter 0 accepts all messages.

16.11.15 CAN n TTCAN Control register (CANn_TTCTRL) (n=0,1)

Address Offset: 0xBC

Bit	Field	Access	Initial	Description
31	WTIE	RW	1	TTCAN watch trigger interrupt enable bit 0: Disable 1: Enable
30	WTIF	RW	0	TTCAN watch trigger interrupt flag 0: The cycle count did not reach the limit defined by TT_WTRIG 1: The cycle count reaches the limited defined by TT_WTRIG and WTIE = 1
29	TEIF	RW	0	TTCAN trigger error interrupt flag The conditions when TEIF will be set, are defined in chapter TTCAN Trigger Types. There is no bit to enable or disable the handling of TEIF. 0: No error 1: Trigger error occurred
28	TTIE	RW	1	TTCAN time trigger interrupt enable bit 0: Disable 1: Enable. TTIF will be set if the cycle time is equal to the trigger time TT_TRIG.
27	TTIF	RW	0	TTCAN time trigger interrupt flag 0: The cycle time isn't equal to the trigger time TT_TRIG 1: The cycle time is equal to the trigger time TT_TRIG and TTIE = 1

Bit	Field	Access	Initial	Description
				Writing an one to TTIF resets it. Writing a zero has no impact. TTIF will be set only once. If TT_TRIG gets not updated, then TTIF will be not set again in the next basic cycle.
26:25	T_PRESC	RW	0	TTCAN timer prescaler The TTCAN time base is a CAN bit time defined by S_PRES, S_SEG1 and S_SEG2. With T_PRESC an additional prescaling factor of 1, 2, 4 or 8 is defined. 0: 1 1: 2 2: 4 3: 8
24	TTEN	RW	0	Time trigger enable bit 0: Disable 1: Enable TTCAN, timer is running
23	TBE	RW	0	Set TB slot to empty TBE is automatically reset to 0 as soon as the slot is marked as empty and TSFF=0. If a transmission from this slot is active, then TBE stays set as long as either the transmission completes or after a transmission error or arbitration loss the transmission is not active any more. If both TBF and TBE are set, then TBE wins. 0: No action 1: Slot selected by TBPTR shall be marked as Empty
22	TBF	RW	0	Set TB slot to filled TBF is automatically reset to 0 as soon as the slot is marked as filled and TSFF=1. If both TBF and TBE are set, then TBE wins. 0: No action 1: Slot selected by TBPTR shall be marked as Filled
21:19	–	–	0	Reserved
18:16	TBPTR	RW	0	Pointer to a TB frame slot The frame slot pointed to by TBPTR is readable / writable using the TBUF registers. Write access is only possible if TSFF=0. Setting TBF to 1 marks the selected slot as filled and setting TBE to 1 marks the selected slot as empty. If TBPTR is too big and points to a slot that is not available, then TBF and TBE are reset automatically and no action takes place. 0: Pointer to the PTB 1: Pointer to the STB1 2: Pointer to the STB2 3: Pointer to the STB3 4: Pointer to the STB4 5: Pointer to the STB5 6: Pointer to the STB6 7: Pointer to the STB7 Other: Reserved
15:0	–	–	0	Reserved

16.11.16 CAN n TTCAN Reference Message ID register (CANn_REFID) (n=0,1)

Address Offset: 0xC0

CAN bus controller recognizes the reference message only by ID. The payload is not tested.

Bit	Field	Access	Initial	Description
31	REF_IDE	RW	0	Reference message IDE bit 0: Standard frames 1: Extended frames
30:29	–	–	0	Reserved
28:0	REF_ID	RW	0	Reference message IDentifier 0: REF_ID(10:0) is valid (standard ID) 1: REF_ID(28:0) is valid (extended ID)

Bit	Field	Access	Initial	Description
				REF_ID is used in TTCAN mode to detect a reference message. REF_ID(2:0) is not tested and therefore the appropriate register bits are forced to 0. These bits are used for up to 8 potential time masters.

16.11.17 CAN n TTCAN Trigger Configuration register (CANn_TRIGCFG) (n=0,1)

Address Offset: 0xC4

Bit	Field	Access	Initial	Description
31:16	TT_TRIG	RW	0	Trigger time TT_TRIG[15:0] defines the cycle time for a trigger. For a transmission trigger the earliest point of transmission of the SOF of the appropriate frame will be TT_TRIG+1.
15:12	TEW	RW	0	Transmit enable window For a single shot transmit trigger there is a time of up to 16 ticks of the cycle time where the frame is allowed to start. TWE+1 defines the number of ticks. TEW=0 is a valid setting and shortens the transmit enable window to 1 tick. 0: 1 tick 1: 2 ticks 2: 3 ticks 3: 4 ticks 4: 5 tick 5: 6 ticks 6: 7 ticks 7: 8 ticks 8: 9 tick 9: 10 ticks 10: 11 ticks 11: 12 ticks 12: 13 tick 13: 14 ticks 14: 15 ticks 15: 16 ticks
11	–	–	0	Reserved
10:8	TTYPE	RW	0	Trigger type 0: Immediate Trigger for immediate transmission 1: Time Trigger for receive triggers 2: Single Shot Transmit Trigger for exclusive time windows 3: Transmit Start Trigger for merged arbitrating time windows 4: Transmit Stop Trigger for merged arbitrating time windows Other: No action
7:3	–	–	0	Reserved
2:0	TTPTR	RW	0	Transmit trigger TB slot pointer If TTPTR points to a slot that is not available, then TEIF is set and no new trigger can be activated after a write access to TT_TRIG[15:0]. If TTPTR points to an empty slot, then TEIF will be set at the moment, when the trigger time is reached. 0: Pointer to the PTB 1: Pointer to the STB1 2: Pointer to the STB2 3: Pointer to the STB3 4: Pointer to the STB4 5: Pointer to the STB5 6: Pointer to the STB6 7: Pointer to the STB7 Other: Reserved

16.11.18 CAN n TTCAN Watch Trigger Time register (CANn_TRIGWTR) (n=0,1)

Address Offset: 0xC8

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	TT_WTRIG	RW	0xFFFF	Watch trigger time TT_WTRIG[15:0] defines the cycle time for a watch trigger. The initial watch trigger is the maximum cycle time 0xFFFF.

16.11.19 CAN n Pin Control register (CANn_PINCTRL) (n=0,1)

Address offset: 0xD4

Bit	Field	Access	Initial	Description
31:3	–	–	0	Reserved
2	STBYEN	RW	0	CAN_STBYn transceiver standby control pin enable bit 0: Disable. 1: Enable.
1	RXEN	RW	0	CAN_RXn input pin enable bit 0: Disable. 1: Enable..
0	TXEN	RW	0	CANn_TXn output pin enable bit 0: Disable. 1: Enable.

17 CYCLIC REDUNDANCY CHECK (CRC)

17.1 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.

17.2 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

17.3 CRC REGISTERS

Base Address: 0x4002 8000

There are two access types: RW: Read or write, R: Read Only.

Offset	Register	Description
0x0000	CRC_CTRL	CRC Control register
0x0004	CRC_DATA	CRC Data register

17.3.1 CRC Control register (CRC_CTRL)

Address offset: 0x00

Bit	Field	Access	Initial	Description
31:5	–	–	0	Reserved
4	BUSY	R	0	CRC calculation busy flag 0: CRC calculation idle/finished 1: CRC calculation is in process
3	–	–	0	Reserved
2	RESET	RW	0	CRC Reset bit 0: No effect 1: Reset CRC circuit. (Reset the initial seed value and BUSY bit to 0). Clear this bit when the reset operation had finished by HW.
1:0	CRC	RW	0	CRC Polynomial 0: CRC-16-CCITT 1: CRC-16 2: CRC-32 Other: Reserved

17.3.2 CRC Data register (CRC_DATA)

Address offset: 0x04

✸ ***Note: Support 8-bit (Byte) Write ONLY!***

Bit	Field	Access	Initial	Description
31:0	DATA	RW	0	CRC Data to be input or read. Write: Support BYTE WRITE ONLY! Input 8-bit data to the CRC calcu-lator and start to calculation process. Read: Output the previous CRC calculation result depends on the CRC Polynomial.

18 SDIO

18.1 OVERVIEW

SDIO is a host controller used to access the Secure Digital (SD) card that conforms to the standard specifications, which include the SDIO specification and SD memory card physical layer specification. The SD host controller provides the programmable I/O and DMA for data transfers. DMA supports the data transfers between the memory and the SD card through the AHB Master interface without being interrupted by CPU. DMA also supports the single-block transfer, multi-block transfer, and infinite transfer. The system address register points to the address of the descriptor table and sequentially accesses the data until the end of a transfer.

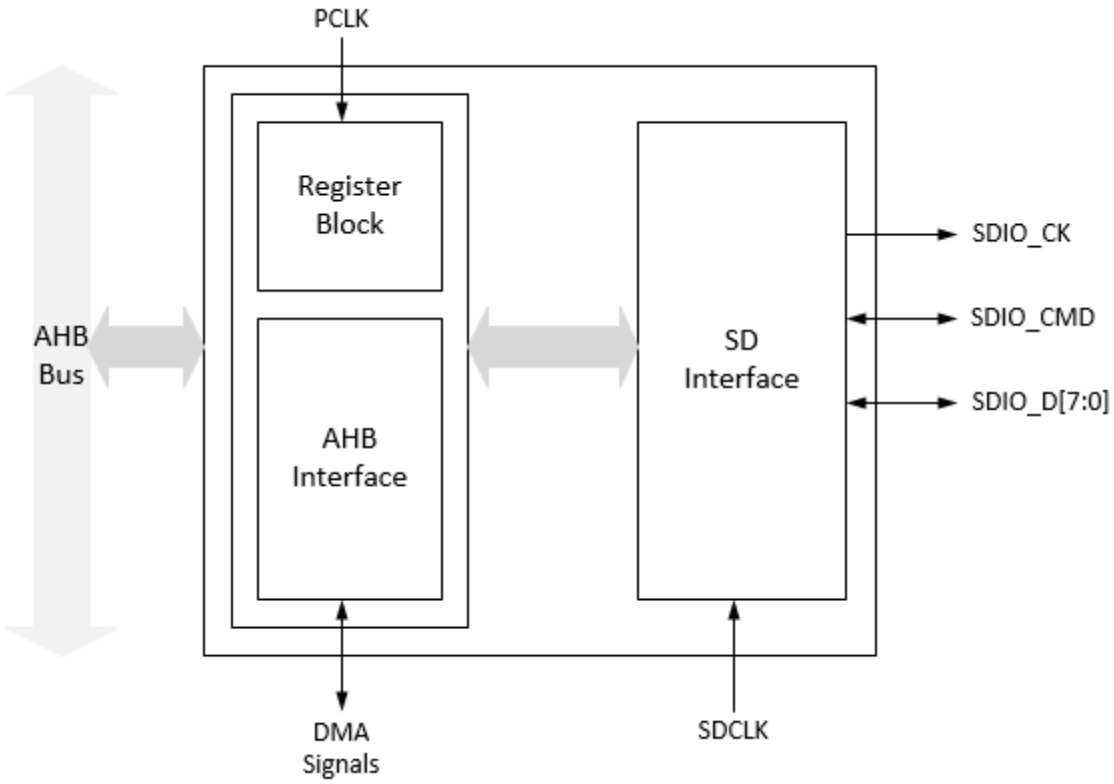
18.2 FEATURES

- Compliant with SD host controller standard specification, version 2.0
- Supports both DMA and non-DMA data transfers
- Compliant with SD physical layer specification, version 2.0
- Supports configurable SD bus modes: 4-bit mode and 8-bit mode
- Compliant with SDIO card specification, version 2.0
- Compliant with MMC card specification, version 4.2 mandatory part
- Supports configurable 1K SRAM for data FIFO
- Supports configurable 1-bit/4-bit SD card bus and 1-bit/4-bit/8-bit eMMC card bus
- Built-in generation and check for 7-bit and 16-bit CRC data
- Supports Read Wait mechanism for SDIO function

18.3 Pin DESCRIPTION

Pin Name	Type	Description	GPIO Configuration
SDIO_CMD	O	SDIO serial command.	Depends on AFIO register
	I	SDIO serial response.	Depends on AFIO register
SDIO_CK	O	SDIO serial clock.	Depends on AFIO register
SDIO_D0~7	I/O	SDIO data.	Depends on AFIO register

18.4 Block Diagram



18.5 Clock Usage

The clock frequency HCLK must be larger than or equal to 1/4 SDIO_CK (divided by SDCLK) to avoid CMD FIFO underrun or RSP FIFO overrun.
When operating in 8-bit mode, the clock frequency HCLK must be larger than or equal to 1/2 SDIO_CK (divided by SDCLK).

Bit Mode		
1-bit	**4-bit**	**8-bit**
HCLK ≥ 1/4 *SDIO_CK	HCLK ≥ 1/4 *SDIO_CK	HCLK ≥ 1/2 *SDIO_CK

18.6 SDIO REGISTERS

Base Address: 0x4005 2000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	SDIO_SDMAADD	SDIO SDMA System Address register
0x0004	SDIO_BLKCTRL	SDIO Block Size register
0x0008	SDIO_ARG1	SDIO Argument 1 register
0x000C	SDIO_TRANSCTRL	SDIO Transfer Control register
0x000E	SDIO_CMDISU	SDIO Command Issuing register
0x0010	SDIO_RESP0	SDIO Response register 0
0x0014	SDIO_RESP1	SDIO Response register 1
0x0018	SDIO_RESP2	SDIO Response register 2
0x001C	SDIO_RESP3	SDIO Response register 3
0x0020	SDIO_BUFDP	SDIO Buffer Data Port register
0x0024	SDIO_PRSSTA	SDIO Present State register
0x0028	SDIO_HSTCTRL	SDIO Host Control register
0x002C	SDIO_CLKCTRL	SDIO Clock Control register
0x002E	SDIO_DATTOCTRL	SDIO Data Timeout Control register
0x002F	SDIO_SWRST	SDIO Software Reset register
0x0030	SDIO_RIS	SDIO Interrupt Status register
0x0034	SDIO_ISTEN	SDIO Interrupt Status Enable register
0x0038	SDIO_ISGEN	SDIO Interrupt Signal Enable register
0x003C	SDIO_ATCMDERR	SDIO Auto CMD12 Error Status register
0x0040 – 0x004C	–	Reserved
0x0050	SDIO_FORCEEVT	SDIO Force Event register
0x0054	SDIO_ADMAERRSTA	SDIO ADMA Error Status register
0x0058	SDIO_ADMAADD	SDIO ADMA System Address register
0x005C – 0x0124	–	Reserved
0x0128	SDIO_DMAHSKEN	SDIO DMA Handshake Enable register

18.6.1 SDIO SDMA System Address register (SDIO_SDMAADD)

Address Offset: 0x00

This register is used to set the system memory address of an SDMA transfer.

The host driver sets the register before issuing a command to start the SDMA transfer. After the SDMA buffer boundary reaches and stops an SDMA transfer, the next system address of the next contiguous data address can be read from this register. The SDMA transfer waits at every boundary specified by BUFBOUND bits in the SDIO_BLKCTRL register. When the host driver sets the next system address to this register, the host controller will restart the SDMA transfer.

Bit	Field	Access	Initial	Description
31:0	ADDR	RW	0	SDMA system address register

18.6.2 SDIO Block Control register (SDIO_BLKCTRL)

Address Offset: 0x04

This register is used to configure the number of bytes in a data block. The SDMA system address register is updated at each system memory boundary during the SDMA transfer. When the SDMA transfer reaches each boundary, the host controller will trigger the DMA interrupt to request the host driver to update the SDMA system address.

Bit	Field	Access	Initial	Description
31:16	BLKCNT	RW	0	Block count of the current transfer The block count register is valid when the BLKCNTEN bit is set to '1'. This register is used only for the multi-block transfers. The host controller will decrease the counting number during the data transfer and stop counting when it counts down to zero.
15	–	–	0	Reserved
14:12	BUFBOUND	RW	0	Buffer boundary of the SDMA host 0: Buffer Boundary = 4KB 1: Buffer Boundary = 8KB 2: Buffer Boundary = 16KB 3: Buffer Boundary = 32KB 4: Buffer Boundary = 64KB 5: Buffer Boundary = 128KB 6: Buffer Boundary = 256KB 7: Buffer Boundary = 512KB
11:0	SIZE	RW	0	The block size of data transfer The block size of data transfers for CMD17, CMD18, CMD24, CMD25, and CMD53, and can be set with values ranging from 1 up to the maximum buffer size. In the memory, this register should be set up to 512 bytes. It can be accessed only when no transaction is executed (i.e., After a transaction has been stopped). The read operations during the transfers may return an invalid value and the write operations should be ignored.

18.6.3 SDIO Argument 1 register (SDIO_ARG1)

Address Offset: 0x08

Bit	Field	Access	Initial	Description
31:0	CMDARG	RW	0	Command argument as bit[39:8] of command field

18.6.4 SDIO Transfer Control register (SDIO_TRANSCTRL)

Address Offset: 0x0C

Bit	Field	Access	Initial	Description
15:6	–	–	0	Reserved
5	MULBLKEN	RW	0	Single/Multi-block selection 0: Single block 1: Multiple blocks
4	TRANSDIR	RW	0	Data transfer direction selection 0: Write from the host to card 1: Read from the card to host
3:2	ATCMDEN	RW	0	Auto command enable There are two methods to stop the read and write operations of multiple blocks: (1) Auto CMD12 Enable When this field is set to 1, the Host Controller will automatically issue CMD12 when the last block transfer is completed. The Auto CMD12 error is indicated to the Auto CMD Error Status register. This bit should not be set by the Host Driver if CMD12 is not required by the command. User cannot issue ACMD before receiving transfer complete interrupt of Auto CMD12.

Bit	Field	Access	Initial	Description
				(Note: The Host Controller does not check command index.) 0: Disable auto command 1: Enable auto CMD12 Other: Reserved
1	BLKCNTEN	RW	0	Block count enable This bit is only valid for a multi-block transfer. The multi-block transfer will be an infinite transfer. 0: Disable 1: Enable
0	DMAEN	RW	0	DMA enable The DMA mode can be selected by DMATYPE bits in the host controller register. When a data transfer command is issued, the DMA transfer will begin. 0: Disable 1: Enable

18.6.5 SDIO Command Issuing register (SDIO_CMDISU)

Address Offset: 0x0E

The host driver should set this register before issuing the data transfer command or resume command. The host driver should check the **Command Inhibit (CMD)** and **Command Inhibit (DAT)** bits in the present state register to determine whether the SD bus is free to transfer.

Bit	Field	Access	Initial	Description
31:14	–	–	0	Reserved
13:8	CMDIDX	RW	0	Command index as bit[45:40] of command field
7:6	CMDTYPE	RW	0	Command type There are two types of special commands: Normal and Abort. If Abort command is issued when executing a read transfer, the Host Controller should stop reading to the buffer. If this command is issued when executing a write transfer, the Host Controller should stop driving the ***DAT*** line. After issuing the Abort command, the Host Driver should issue a software reset. 0: Other command 1: Reserved for SDIO suspend command 2: Reserved for SDIO resume command 3: CMD12,CMD52 for writing I/O Abort in CCCR
5	DATPRES	RW	0	Data present select This bit is 0 under the following conditions: (1) Commands only using the ***CMD*** line (ex. CMD52) (2) Commands with no data transfer but using the busy signal on ***DAT[0]*** line (R1b or R5b ex. CMD38) (3) Resume command 0: No data present 1: Data is present and data transfer is enabled
4	CMDINDCHKEN	RW	0	Command Index check enable 0: Disable 1: Enable. The host controller will check the index field response to determine if the values are CMDIDX. If they are not the same, CMDINDEIF will be triggered.
3	CMDCRCEN	RW	0	Command CRC check enable 0: Disable 1: Enable. If an error is detected, CMDCRCEIF will be triggered.
2	–	–	0	Reserved
1:0	RSPTYPE	RW	0	Response type selection 0: No response 1: Response length 136 2: Response length 48 3: Response length 48 with busy check after response

18.6.6 SDIO Response register 0~3 (SDIO_RESP0~3)

Address Offset: 0x10, 0x14, 0x18, 0x1C

The Host Controller stores the Auto CMD12 response in the upper word of the Response register to avoid the Auto CMD12 response, which tends to be overwritten by the other command.

Bit	Field	Access	Initial	Description
31:0	RSP	R	0	Command response as bit[31:0] of response field

18.6.7 SDIO Buffer Data Port register (SDIO_BUFDP)

Address Offset: 0x20

This register uses the 32-bit Data Port register to access the internal buffer (Always access through the offset address 0x20 even when the transfer size is byte or half-word.)

Bit	Field	Access	Initial	Description
31:0	DATA	RW	0	Buffer data port

18.6.8 SDIO Present State register (SDIO_PRSSTA)

Address Offset: 0x24

Bit	Field	Access	Initial	Description
31:25	–	–	0	Reserved
24	CMDLV	R	1	Command line signal level
23:20	DATLV	R	0xF	DATA[3:0] line signal level
19:12	–	–	0x2	Reserved
11	BUFREN	R	0	Buffer read enable status This status bit is used for the non-DMA read transfer. This bit indicates that there are valid data exist in the RX buffer and is ready for read. This bit can be changed from 1 to 0 to indicate that the RX buffer data have been read from the buffer. This bit can be changed from 0 to 1 when all block data are ready in the buffer for read or when the Rx buffer is full and generates a BUFRIF interrupt in the normal interrupt status register. Note: The buffer read enable is set when the amount of data that equals the FIFO depth is ready for read when the data FIFO is implemented by the register (Please check the register offset 0x178 for the hardware attribute). When the data FIFO is implemented by SRAM, this bit will be set when at least one block of data is ready for read. 0: Disable 1: Enable
10	BUFWEN	R	0	Buffer write enable status This bit is used for the non-DMA write transfer. This bit indicates that the TX buffer is ready to receive data. This bit changes from 1 to 0 when all block data are written to the TX buffer or when the TX buffer is full. This bit changes from 0 to 1 when top of the block data can be written to the TX buffer. Note: The buffer write enable is set when the amount of space that equals the FIFO depth is available for write if the data FIFO is implemented by the register (Please check the register offset 0x178 for the hardware attribute). When the data FIFO is implemented by SRAM, this bit will be set when at least one block of space is available for written. Note: This bit is still work when stop at block gap is set. But user should not base on this bit to write data to TX buffer when stop at block gap is set. 0: Disable 1: Enable. Data can be written to the TX buffer.
9	RACT	R	0	Read transfer active

Bit	Field	Access	Initial	Description
				This status bit is used to detect the completion of a read transfer. This bit is set to 1 under the following conditions: (1) After the end bit of a read command (2) When CONREQ in the block gap control register is set to restart a transfer. This bit is set to 0 under the following conditions: (1) When all data blocks specified by the block length are transferred to the system. (2) When BLKGAPREQ in the block gap control register is set to 1 and the host controller has transferred all the valid data blocks to the system. The TRANSCPLIF interrupt is generated when this bit changes from 1 to 0. 0: Inactive/idle 1: Active/busy
8	WACT	R	0	Write transfer active This status bit indicates that a write transfer is active or not. This bit is set to 1 under the following conditions: (1) After the end bit of a write command (2) When CONREQ in the block gap control register is set to restart a transfer. This bit is set to 0 under the following conditions: (1) After getting the CRC status of the last data block specified by the transfer count. (2) After getting the CRC status of any block where data transmission is stopped by BLKGAPREQ. A BLKGAPIF interrupt will be generated when BLKGAPREQ is set to 1 and this bit changes to 0. This bit is useful in the command with a busy data line. 0: Inactive/idle 1: Active/busy
7:3	–	–	0	Reserved
2	DATACT	R	0	Data line active This bit is used to determine whether the Data line is in use or not. In a read transfer, this status bit is used to check whether a read transfer is executing on the bus. Changing this bit from 1 to 0 will generate a BLKGAPIF interrupt when BLKGAPREQ is set to 1. This bit is set to 1 under the following conditions: (1) After the end bit of a read command (2) When CONREQ in the block gap control register is set to restart a transfer. This bit is set to 0 under the following conditions: (1) When the end bit of the last data block is sent from the SD bus to the host controller. (2) When BLKGAPREQ is set to 1 and a read transfer is stopped at the block gap. In a write transfer, this status bit is used to check whether a write transfer is executing on the bus. Changing this bit from 1 to 0 will generate a TRANSCPLIF interrupt in the normal interrupt status register. This bit is set to 1 under the following conditions: (1) After the end bit of a read command (2) When CONREQ in the block gap control register is set to restart a transfer. This bit is set to 0 under the following conditions: (1) When the card release the busy signal of the last data block. (2) When BLKGAPREQ is set to 1 and the card releases the write busy at the block gap. In the command with busy data line, this bit indicates whether a command with busy is executing on the bus. This bit will be set after the end bit of the command with busy and will be cleared when busy is de-asserted or busy is not detected after the end of a response. 0: Inactive/idle 1: Active/busy
1	CMDINBDAT	R	0	Command Inhibit (DAT) This bit will be generated if DATACT bit or RACT bit is set. 0: Issue new commands with the data line

Bit	Field	Access	Initial	Description
				1: Cannot issue a new command with the data line
0	CMDINBCMD	R	0	Command Inhibit (CMD) This bit is cleared when the command response is received. Even if CMDINBDAT bit is set, the commands using the command line can be issued if this bit is 0. CMD0, CMD12, CMD13, and CMD52 can be issued when the data lines are in use during a data transfer. The CMDCPLIF interrupt in the normal interrupt register is issued when this bit is changed from 1 to 0. However, if the present command suffer some errors (CRC error, command index error, and so on), this bit will be remained 1. Note: This bit will not be set by Auto CMD12. 0: Issue a new command with the command line 1: Cannot issue command

18.6.9 SDIO Host Control register (SDIO_HSTCTRL)

Address Offset: 0x28

Bit	Field	Access	Initial	Description
31:20	–	–	0	Reserved
19	INTBLKGAP	RW	0	Interrupt at block gap This bit is only useful in the 4-bit mode of the SDIO card. This bit should be set to 0 if the SDIO card cannot issue an interrupt during the block gap. (RWAITCTRL should be enabled) 0: Disable check the interrupt at the block gap 1: Enable checking the interrupt at the block gap for the multiple block transfers.
18	RWAITCTRL	RW	0	Read wait control When the SDIO card supports the read wait function, the host controller can use the read wait protocol to control the SDIO bus. When the card does not support the read wait function, the host controller will stop SDIO_CK to hold the read transfer. 0: Disable the read wait function 1: Enable the read wait function
17	CONREQ	RW	0	Continue request Used to restart a transaction, which can be stopped by using BLKGAPREQ. To restart a transaction, this bit should be set to 1 and BLKGAPREQ should be set to 0. The host controller will automatically clear this bit under the following conditions: (1) In a read transfer, DATACT bit will be changed from 0 to 1 to start a read transfer. (2) In a write transfer, WACT bit will be changed from 0 to 1 to start a write transfer. 0: No effect 1: Restart
16	BLKGAPREQ	RW	0	Stop at block gap request Used to stop executing the read or write transfer at the next block gap. This bit is useful for the non-DMA and SDMA, but is not useful for ADMA. The host driver should keep the setting of this bit to 1 until the TRANSCPLIF interrupt is set to 1. 0: Transfer. The host controller will not write data to SDIO_BUFDP register. 1: Stop. The host controller will stop at the block gap by using RWAITCTRL or stop SDIO_CK in a read transaction.
15:9	–	–	0	Reserved
8	SDBUSPW	RW	0	SD bus power 0: Power off 1: Power on
7:6	–	–	0	Reserved
5	EXDATWID	RW	0	Extended data transfer width 0: Bus width is selected by the DATWID bit 1: 8-bit bus width
4:3	DMATYPE	RW	0	DMA type select 0: SDMA

Bit	Field	Access	Initial	Description
				2: 32-bit address ADMA2 Other: Reserved
2	HSPEN	RW	0	High speed enable 0: SDIO_CK up to 25MHz 1: SDIO_CK up to 50MHz
1	DATWID	RW	0	Data transfer width 0: 1-bit mode 1: 4-bit mode
0	–	–	0	Reserved

18.6.10 SDIO Clock Control register (SDIO_CLKCTRL)

Address Offset: 0x2C

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:8	LCLKSET	RW	0	SD clock frequency value[7:0] for the 10-bit divided clock mode The bit is used to select the frequency of the SDIO_CK pin. The base clock is specified by SDCLK. 0x0: SDIO_CK= SDCLK/1 0x1: SDIO_CK= SDCLK/2 0x2: SDIO_CK= SDCLK/4 0x3: SDIO_CK= SDCLK/6 …
7:6	UCLKSET	RW	0	SD clock frequency value[9:8] for the 10-bit divided clock mode
5	CLKGENSEL	R	0	Clock generator select (fixed to 0) 0: 10-bit divided clock mode Other: Reserved
4:3	–	–	0	Reserved
2	SDCLKEN	RW	0	SD clock enable 0: Disable providing SDIO_CK 1: Enable providing SDIO_CK
1	INTCLKSTB	R	0	Internal clock stable 0: The internal clock is not ready 1: The internal clock is ready
0	INTCLKEN	RW	0	Internal clock enable 0: Disable 1: Enable

18.6.11 SDIO Data Timeout Control register (SDIO_DATTOCTRL)

Address Offset: 0x2E

Bit	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3:0	TOCNT	RW	0xE	Data timeout counter value 0: Timeout counter value = 2^{13} x HCLK 1: Timeout counter value = 2^{14} x HCLK 2: Timeout counter value = 2^{15} x HCLK 3: Timeout counter value = 2^{16} x HCLK 4: Timeout counter value = 2^{17} x HCLK 5: Timeout counter value = 2^{18} x HCLK 6: Timeout counter value = 2^{19} x HCLK 7: Timeout counter value = 2^{20} x HCLK 8: Timeout counter value = 2^{21} x HCLK 9: Timeout counter value = 2^{22} x HCLK 10: Timeout counter value = 2^{23} x HCLK 11: Timeout counter value = 2^{24} x HCLK 12: Timeout counter value = 2^{25} x HCLK 13: Timeout counter value = 2^{26} x HCLK 14: Timeout counter value = 2^{27} x HCLK 15: Reserved

18.6.12 SDIO Software Reset register (SDIO_SWRST)

Address Offset: 0x2F

Bit	Field	Access	Initial	Description
7:3	–	–	0	Reserved
2	SWRSTDAT	RW	0	Software reset for data line 0: No effect 1: Reset SDIO_BUFDP register, BUFREN/BUFWEN/RACT/WACT/DATACT/ CMDINBDAT bits in SDIO_PRSSTA register, CONREQ/BLKGAPREQ bits in SDIO_HSTCTRL register, and BUFRIF/BUFWIF/DMAIF/BLKGAPIF/TRANSCPLIF bits in SDIO_RIS register
1	SWRSTCMD	RW	0	Software reset for command line 0: No effect 1: Reset CMDINBCMD bit in SDIO_PRSSTA register and CMDCPLIF bit in SDIO_RIS register
0	SWRSTALL	RW	0	Software reset 0: No effect 1: Reset all registers except for the card detection

18.6.13 SDIO Interrupt Status register (SDIO_RIS)

Address Offset: 0x30

Bit	Field	Access	Initial	Description
31:26	–	–	0	Reserved
25	ADMAEIF	RW	0	ADMA error interrupt flag 0: No error 1: ADMA error
24	ATCMDEIF	RW	0	Auto CMD12 error interrupt flag 0: No error 1: Auto CMD12 error
23	–	–	0	Reserved
22	DATEBEIF	RW1C	0	Data end bit error interrupt flag 0: No error 1: Data end bit error (This status is set when the host controller detects 0 at the end bit of the read data, which uses the data line or at the end bit of the write CRC status)
21	DATCRCEIF	RW1C	0	Data CRC error interrupt flag 0: No error 1: Data CRC error (the host controller detects the CRC error during the read transfer or detects the write CRC status that is not of the value of 010b)
20	DATTOEIF	RW1C	0	Data timeout error interrupt flag 0: No error 1: Timeout (Wait to read data timeout/Wait write CRC status timeout/Busy timeout after write CRC status/Busy timeout for the R1b and R5b types)
19	CMDINDEIF	RW1C	0	Command index error interrupt flag 0: No error 1: Command index error (The command index in the command response is different from the command)
18	CMDEBEIF	RW1C	0	Command end bit error interrupt flag 0: No error 1: Command end bit error (The end bit of the command response is 0)
17	CMDCRCEIF	RW1C	0	Command CRC error interrupt flag 0: No error 1: Command CRC error
16	CMDTOEIF	RW1C	0	Command timeout error interrupt flag 0: No error 1: Response timeout error (No response is sent from the card within 64 SD card clocks)
15	ERRIF	RC	0	Error Interrupt

Bit	Field	Access	Initial	Description
				0: No error 1: Error
14:9	–	–	0	Reserved
8	CARDIF	RC	0	Card interrupt flag In the 4-bit mode, the card interrupt is sampled during the interrupt cycle. When this bit is set to 1, the interrupt will need to be handled by the Host Driver. The host driver needs to set CARDIE bit to 0 and clear the CARDIF status to avoid driving the interrupt signal to the host system again. If the Host Driver has handled the card interrupt completely, the CARDIE bit should be set to 1 to re-start sampling. Note: When issuing abort command with card interrupt is enabled, there might be some dummy card interrupts. Users are recommended enabling card interrupt only when card interrupt function is needed. 0: No card interrupt 1: Generate card interrupt
7:6	–	–	0	Reserved
5	BUFRIF	RW1C	0	Buffer read ready interrupt flag 0: Not ready to read buffer 1: Ready to read buffer
4	BUFWIF	RW1C	0	Buffer write ready interrupt flag 0: Not ready to write buffer 1: Ready to write buffer
3	DMAIF	RW1C	0	DMA interrupt flag This status bit will be set if the host controller detects that the SDMA buffer boundary is reached during a transfer, except in the last block is completed. In case of ADMA, the int field will be set in the descriptor line and the host controller will generate the interrupt once the descriptor line is done. 0: No DMA interrupt 1: DMA interrupt is generated
2	BLKGAPIF	RW1C	0	Block gap event interrupt flag By setting BLKGAPREQ to 1, this bit will be set when a read or write transaction is stopped at the block gap. (1) Read transaction This bit will be set when DATACT bit changes from 1 to 0. The read wait should be supported for this function in the SDIO card. (2) Write transaction This bit will be set when WACT bit changes from 1 to 0. 0: No block gap event 1: Transaction stopped at block gap (not completed)
1	TRANSCPLIF	RW1C	0	Transfer complete interrupt flag A data transfer will be completed or a data transfer will be stopped at the block gap during a transmission to generate a transfer complete interrupt. (1) Read transaction This bit will be set when RACT bit changes from 1 to 0. (2) Write transaction This bit will be set when DATACT bit changes from 1 to 0. (3) Command with busy This bit will be set when the busy data line is released. 0: Not complete 1: Command execution is completed
0	CMDCPLIF	RW1C	0	Command complete interrupt flag This bit will be set when the command response is received and the CRC check is okay. Auto CMD12 and Auto CMD23 will not generate the command complete interrupt. The following table shows that the Command Timeout Error bit has higher priority than the Command Complete bit. When both bits are set to 1, it indicates that the response has not been correctly received. 0: No command complete 1: Command complete

Table. Relationship between Command complete and Command Timeout Error

Command Complete	Command Timeout Error	Meaning of status

0	0	Interrupt by another factor
Don't care	1	Response not received within 64 SDIO_CK cycles
1	0	Response received

18.6.14 SDIO Interrupt Status Enable register (SDIO_ISTEN)

Address Offset: 0x34

Bit	Field	Access	Initial	Description
31:26	–	–	0	Reserved
25	ADMAEIE	RW	0	ADMA error interrupt enable 0: Disable 1: Enable
24	ATCMDEIE	RW	0	Auto CMD12 error interrupt enable 0: Disable 1: Enable
23	–	–	0	Reserved
22	DATEBEIE	RW	0	Data end bit error interrupt enable 0: Disable 1: Enable
21	DATCRCEIE	RW	0	Data CRC error interrupt enable 0: Disable 1: Enable
20	DATTOEIE	RW	0	Data timeout error interrupt enable 0: Disable 1: Enable
19	CMDINDEIE	RW	0	Command index error interrupt enable 0: Disable 1: Enable
18	CMDEBEIE	RW	0	Command end bit error interrupt enable 0: Disable 1: Enable
17	CMDCRCEIE	RW	0	Command CRC error interrupt enable 0: Disable 1: Enable
16	CMDTOEIE	RW	0	Command timeout error interrupt enable 0: Disable 1: Enable
15:9	–	–	0	Reserved
8	CARDIE	RW	0	Card interrupt enable If this bit is set to 1, CARDIF will be served by the Host Driver. The Host Driver should set this bit to 0 before serving CARDIF and should reset this bit after CARDIF completes requesting card. The detection of CARDIF will be stopped when this bit is set to 0 and will be restarted when this bit is set to 1. 0: Disable 1: Enable
7:6	–	–	0	Reserved
5	BUFRIE	RW	0	Buffer read ready interrupt enable 0: Disable 1: Enable
4	BUFWIE	RW	0	Buffer write ready interrupt enable 0: Disable 1: Enable
3	DMAIE	RW	0	DMA interrupt enable 0: Disable 1: Enable
2	BLKGAPIE	RW	0	Block gap event interrupt enable 0: Disable 1: Enable
1	TRANSCPLIE	RW	0	Transfer complete interrupt enable

Bit	Field	Access	Initial	Description
				0: Disable 1: Enable
0	CMDCPLIE	RW	0	Command complete interrupt enable 0: Disable 1: Enable

18.6.15 SDIO Normal Interrupt Signal Enable register (SDIO_ ISGEN)

Address Offset: 0x38

Bit	Field	Access	Initial	Description
31:26	–	–	0	Reserved
25	ADMAEISE	RW	0	ADMA error interrupt signal enable 0: Disable 1: Enable
24	ATCMDEISE	RW	0	Auto CMD12 error interrupt signal enable 0: Disable 1: Enable
23	–	–	0	Reserved
22	DATEBEISE	RW	0	Data end bit error interrupt signal enable 0: Disable 1: Enable
21	DATCRCEISE	RW	0	Data CRC error interrupt signal enable 0: Disable Data CRC error interrupt signal 1: Enable
20	DATTOEISE	RW	0	Data timeout error interrupt signal enable 0: Disable 1: Enable
19	CMDINDEISE	RW	0	Command index error interrupt signal enable 0: Disable 1: Enable
18	CMDEBEISE	RW	0	Command end bit error interrupt signal enable 0: Disable 1: Enable
17	CMDCRCEISE	RW	0	Command CRC error interrupt signal enable 0: Disable 1: Enable
16	CMDTOEISE	RW	0	Command timeout error interrupt signal enable 0: Disable 1: Enable
15:9	–	–	0	Reserved
8	CARDISE	RW	0	Card interrupt signal enable 0: Disable 1: Enable
7:6	–	–	0	Reserved
5	BUFRISE	RW	0	Buffer read ready interrupt signal enable 0: Disable 1: Enable
4	BUFWISE	RW	0	Buffer write ready interrupt signal enable 0: Disable 1: Enable
3	DMAISE	RW	0	DMA interrupt signal enable 0: Disable 1: Enable
2	BLKGAPISE	RW	0	Block gap event interrupt signal enable 0: Disable 1: Enable
1	TRANSCPLISE	RW	0	Transfer complete interrupt signal enable 0: Disable 1: Enable
0	CMDCPLISE	RW	0	Command complete interrupt signal enable 0: Disable 1: Enable

18.6.16 SDIO Auto CMD12 Error Status register (SDIO_ATCMDERR)

Address Offset: 0x3C

When the auto_cmd12_en register is set to 1 and the Auto CMD12 error status register is set, the host driver will check this register to identify what kind of error happens during executing AUTO CMD12.

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7	AUTCMDEXCT	RC	0	Command not executed by Auto CMD12 error This bit indicates that the command followed Auto CMD12 is not executed due to an Auto CMD12 error (Bit 1 to bit 4) in this register. The bit will be set to 0 when Auto CMD Error is generated by Auto CMD23. 0: No error 1: Not issued
6:5	–	–	0	Reserved
4	ATCMDINDE	RC	0	Auto CMD index error 0: No error 1: Auto command index error generated
3	ATCMDEBE	RC	0	Auto CMD end bit error 0: No error 1: End bit error generated
2	ATCMDCRCE	RC	0	Auto CMD CRC error 0: No error 1: CRC error generated
1	ATCMDTOE	RC	0	Auto CMD time out error 0: No error 1: Time out
0	ATCMDEXCT	RC	0	Auto CMD12 not executed If the memory multiple block data transfer is not started due to the command error, this bit will not be set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means that the Host Controller cannot issue Auto CMD12 to stop the memory multiple block data transfer due to some errors. If this bit is set to 1, other error status bits D0[4:1] are meaningless. This bit is set to 0 when Auto CMD Error is generated by Auto CMD23. 0: Auto CMD12 executed 1: Not executed

Table. Relationship between CRC Error and Timeout Error for Auto CMD

Auto CMD CRC Error	Auto CMD Timeout Error	Error Type
0	0	No Error
0	1	Response Timeout Error
1	0	Response CRC Error
1	1	CMD line conflict

18.6.17 SDIO Force Event register (SDIO_FORCEEVT)

Address Offset: 0x50

The Force Event register is not a physical register. It is an address to which the Auto CMD error status register can be written. The force event register is only for debugging.

Bit	Field	Access	Initial	Description
31:29	–	–	0	Reserved
28	FAHBRSPE	W	0	Force event for the AHB response error 0: No affected

Bit	Field	Access	Initial	Description
				1: Response error is set
27:26	–	–	0	Reserved
25	FADMAE	W	0	Force event for ADMA error 0: No affected 1: ADMA error status is set
24	FATCMDE	W	0	Force event for Auto CMD error 0: No affected 1: Auto CMD error status is set
23	–	–	0	Reserved
22	FDATEBE	W	0	Force event for Data end bit error 0: No affected 1: Data end bit error status is set
21	FDATCRCE	W	0	Force event for Data CRC error 0: No affected 1: Data CRC error status is set
20	FDATTOE	W	0	Force event for Data timeout error 0: No affected 1: Data timeout error status is set
19	FCMDINDE	W	0	Force event for Command index error 0: No affected 1: Command index error status is set
18	FCMDEBE	W	0	Force event for Command end bit error 0: No affected 1: Command end bit error status is set
17	FCMDCRCE	W	0	Force event for Command CRC error 0: No affected 1: Command CRC error status is set
16	FCMDTOE	W	0	Force event for Command timeout error 0: No affected 1: Command timeout error status is set
15:8	–	–	0	Reserved
7	FAUTCMDEXCT	W	0	Force event for Command not executed by Auto CMD12 error 0: No affected 1: Command not executed by Auto CMD12 error status is set
6:5	–	–	0	Reserved
4	FATCMDINDE	W	0	Force event for Auto CMD index error 0: No affected 1: Auto CMD index error status is set
3	FATCMDEBE	W	0	Force event for Auto CMD end bit error 0: No affected 1: Auto CMD end bit error status is set
2	FATCMDCRCE	W	0	Force event for Auto CMD CRC error 0: No affected 1: Auto CMD CRC error status is set
1	FATCMDTOE	W	0	Force event for Auto CMD time out error 0: No affected 1: Auto CMD time out error status is set
0	FATCMDEXCT	W	0	Force event for Auto CMD12 not executed 0: No affected 1: Auto CMD12 not executed status is set

18.6.18 SDIO ADMA Error Status register (SDIO_ADMAERRSTA)

Address Offset: 0x54

When the ADMA error interrupt is occurred, ADMMERR bits holds the ADMA state and the ADMA system address register holds the address around the error descriptor. When an error occurs, the host driver should watch the ADMA state to identify the error descriptor address.

Bit	Field	Access	Initial	Description
31:3	–	–	0	Reserved
2	ADMALENERR	R	0	ADMA length mismatch error

Bit	Field	Access	Initial	Description
				(1) When Block Count Enable is set, the total data length specified by the Descriptor table will be different from the one specified by the Block Count and Block Length. (2) The total data length cannot be divided by the block length. 0: No error 1: ADMA length mismatch error
1:0	ADMMERR	R	0	ADMA error state 0: ST_STOP (Stop DMA) 1: ST_FDS (Fetch descriptor) 2: Not used 3: ST_TFR (Transfer data)

18.6.19 SDIO ADMA System Address register (SDIO_ ADMAADD)

Address Offset: 0x58

Bit	Field	Access	Initial	Description
31:0	ADMALADD	RW	0	Lower 32-bit ADMA system address The 32-bit address descriptor only uses the lower 32-bit of the ADMA system address register. ADMA ignores the lower 2-bit of this register and assumes it to be 00b.

18.6.20 SDIO DMA Handshake Enable register (SDIO_DMAHSKEN)

Address Offset: 0x128

Bit	Field	Access	Initial	Description
31:1	–	–	0	Reserved
0	DMAHSKEN	RW	0	DMA handshake enable 0: Disable 1: Enable. BUFRIF and BUFWIF bits will not be set and become ineffective. Data will be transferred by the external DMA and the flow control will be controlled by the DMA handshake protocol.

19 LCD Module (LCM)

19.1 OVERVIEW

LCD module is a peripheral device which interfaces to LCD.
The LCM controlling function is compatible with 8080 interfaces to communicate with the display panels and supports the 8-bit, 9=bit, 16-bit, and 18-bit types of the LCD display panel module.

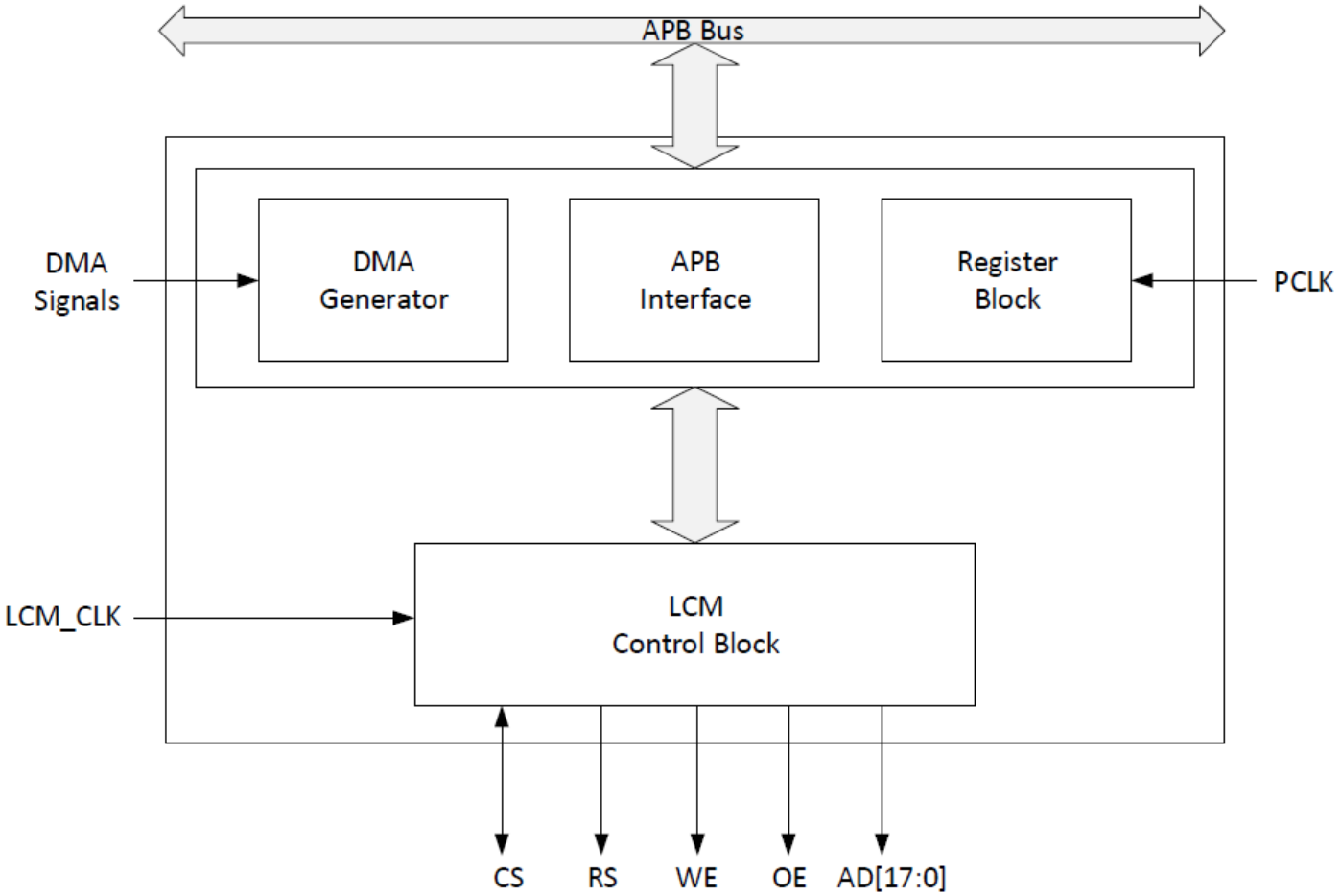
19.2 FEATURES

- Support 8-bit, 9-bit, 16-bit and 18-bit data bus panels
- Support 8080 modes
- Support chip select pin
- LCM write data FIFO function

19.3 PIN DESCRIPTION

Pin Name	Type	Description	GPIO Configuration
AD[17:0]	I/O	Data of LCM	Depends on AFIO register
CS	O	Chip select	Depends on AFIO register
A0 (RS)	O	The signal indicates that the data on data bus is command or data. High = Data , Low = Command	Depends on AFIO register
WE	O	The signal identifies a write operation when it is active low.	Depends on AFIO register
OE	O	The signal identifies a read operation when it is active low.	Depends on AFIO register

19.4 BLOCK DIAGRAM

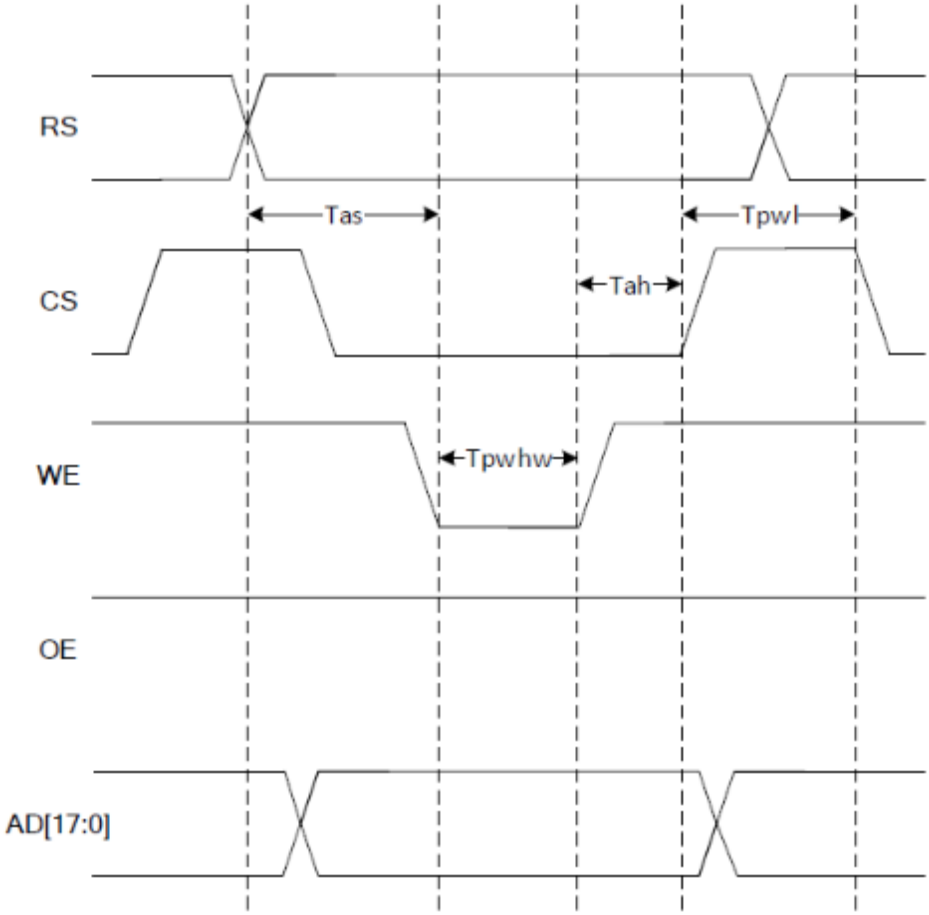


19.5 INTERFACE DESCRIPTION

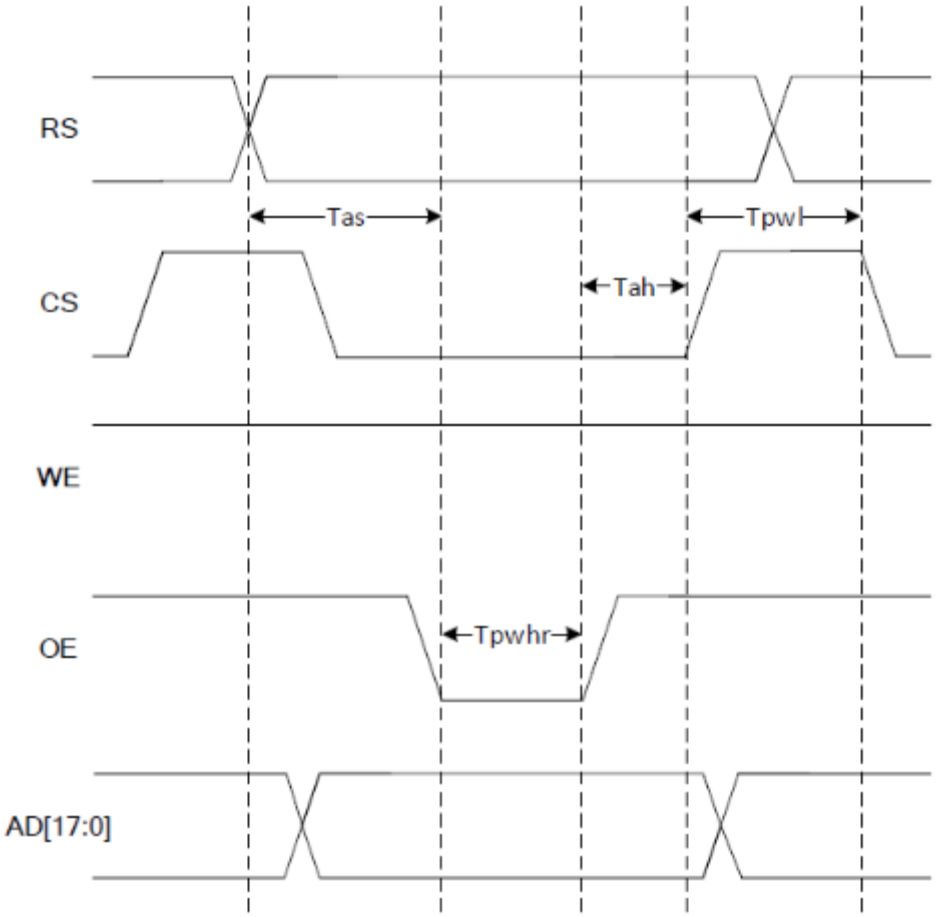
The LCM controlling function provides the 8080 interfaces to communicate with the display panels, such as the 8-bit, 16-bit, and 18-bit mono-STN, CSTN, and TFT display panels.
This controller provides the flexible programming timing registers to fit different panels.

19.5.1 8080 Mode

- Write operation in 8080 mode:



- Read operation in 8080 mode:



19.6 LCM CONTROL MODE

19.6.1 CPU mode

The CPU mode is to move data/command to/from LCM through register programming.
In the CPU mode, the register setting sequence of the read/write data/command is shown below.

■ Read operation

Step 1 : If RDYF = 1, go to step 2.
Step 2 : Write address to the LCM_CMD register.
Step 3 : If RDYF = 1, go to step 4.
Step 4 : Write DATCMDSEL = 1.
Step 5 : If RDYF = 1, go to step 6.
Step 6 : Read data in the LCM_CMD register, users can obtain data from the LCM_CMD register.

■ Write operation

Step 1 : If RDYF = 1, go to step 2.
Step 2 : Write address to the LCM_CMD register.
Step 3 : If RDYF = 1, go to step 4.
Step 4 : Write data to the LCM_DAT register.

19.6.2 DMA Mode

The DMA mode is to have the internal FIFO and DMA hand-shake signals, it can perform data/command write function to LCM through external DMA engine.
In the DMA mode, the read data from LCM will be inhibited
In the DMA mode, it allows to transfer data/command to LCM with several data format.

- Write DAMFORMAT enable in LCM_MODE register.
- Write FIFOCLR = 1 in LCM_DMAFIFOCTRL register, wait FIFOCLR becomes 0.
- Write WATERMARK & REQSRC enable in LCM_DMACFG register.
- Write DMAEN enable in LCM_ENABLE register.

19.7 LCM REGISTERS

Base Address: 0x4002 6000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000 – 0x01FC	-	Reserved
0x0200	LCM_TMCTRL	LCM Timing Control register
0x0204	LCM_RDY	LCM Ready register
0x0208	LCM_RS	LCM Data Command Read register
0x020C	LCM_DAT	LCM Date Window register
0x0210	LCM_CMD	LCM Command register
0x0214	LCM_MODE	LCM Mode register
0x0218 – 0x0224	-	Reserved
0x0228	LCM_ENABLE	LCM Enable register
0x022C – 0x023C	-	Reserved
0x0240	LCM_DMADAT	LCM DMA Date register
0x0244	LCM_DMAFIFOCTRL	LCM DMA FIFO Control register
0x0248	-	Reserved
0x024C	LCM_DMACFG	LCM DMA Configuration register
0x0250	LCM_DMAIS	LCM DMA Interrupt Status register
0x0254	LCM_DMAIE	LCM DMA Interrupt Enable register
0x0258 – 0x0260	-	Reserved
0x0264	LCM_DMACMD	LCM DMA Command register

19.7.1 LCM Timing Control register (LCM_TMCTRL)

Address offset: 0x200

Bit	Field	Access	Initial	Description
31:20	–	–	0	Reserved
19:16	TPWHR	RW	0	Tpwh width for read cycle = (TPWHR+1) * LCMCLK clock cycle The programmable cycles is 1 up to 16 cycles
15:12	TAS	RW	0	Tas width = (TAS+1) * LCMCLK clock cycle The programmable cycles is 1 up to 16 cycles
11:8	TAH	RW	0	Tah width = (TAH+1) * LCMCLK clock cycle The programmable cycles is 1 up to 16 cycles
7:4	TPWL	RW	0	Tpwl width = (TPWL+1) * LCMCLK clock cycle The programmable cycles is 1 up to 16 cycles
3:0	TPWHW	RW	0	Tpwh width for write cycle = (TPWHW+1) * LCMCLK clock cycle The programmable cycles is 1 up to 16 cycles

19.7.2 LCM Ready register (LCM_RDY)

Address offset: 0x204

Bit	Field	Access	Initial	Description
31:1	–	–	0	Reserved
0	RDYF	R	0	Ready flag for access 0: Not ready for access 1: Ready for access

19.7.3 LCM Data Command Read register (LCM_RS)

Address offset: 0x208

Bit	Field	Access	Initial	Description
31:1	–	–	0	Reserved
0	DATCMDSEL	W	0	Data/Command select 0: LCM_CMD is used to read/write command. 1: LCM_CMD is used to read data.

19.7.4 LCM Data register (LCM_DAT)

Address offset: 0x20C

Bit	Field	Access	Initial	Description
31:0	DATA	RW	0	LCM data

19.7.5 LCM Command register (LCM_CMD)

Address offset: 0x210

Bit	Field	Access	Initial	Description
31:0	CMD	RW	0	LCM command or data(read)

19.7.6 LCM Mode register (LCM_MODE)

Address offset: 0x214

Bit	Field	Access	Initial	Description
31:24	DMAFORMAT	RW	0	DMA data format enable bit 0: Disable 0x30: Enable
23:18	–	–	0	Reserved
17	CSPOL	RW	0	Chip select pin polarity select bit 0: CS active low 1: CS active high
16:8	–	–	0	Reserved
7:6	TRANSMODE	RW	0	16 bits mode control bits 0: One transfer/pixel (65536colors) 1: Two transfers/pixel (262144colors) First transfer = 16bits (MSB), other transfers = 2bits (LSB) 2: Two transfers/pixel (262144colors) First transfer = 2bits (MSB), other transfers = 16bits (LSB) Other: Reserved
5:4	BUSWIDTH	RW	0	Bus width select bits 0: 8 bits 1: 9 bits 2: 16 bits 3: 18 bits
3:2	PANEL	RW	0	Panel interface select bits 1: 16 bits 2: 18 bits Other: Reserved
1:0	–	–	0	Reserved

19.7.7 LCM Enable register (LCM_ENABLE)

Address offset: 0x228

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1	DMAEN	RW	0	LCM DMA operation mode select bit 0: CPU mode. The data read/write is through CPU programming. 1: DMA mode. The data write is through DMA.
0	LCMEN	RW	0	LCM enable 0: Disable 1: Enable

19.7.8 LCM DMA Data register (LCM_DMADAT)

Address offset: 0x240

Bit	Field	Access	Initial	Description
31:0	DATA	W	0	Write data to DMA

19.7.9 LCM DMA FIFO Control register (LCM_DMAFIFOCTRL)

Address offset: 0x244

Bit	Field	Access	Initial	Description
31:2	–	–	0x3010000	Reserved
1	FIFOCLR	RW	0	Clear LCM FIFO - Write 0: Don't care 1: Enable FIFO clear status - Read 0: FIFO clear is done 1: In clear process
0	–	–	0	Reserved

19.7.10 LCM DMA Configuration register (LCM_DMACFG)

Address offset: 0x24C

Bit	Field	Access	Initial	Description
31:20	–	–	0	Reserved
19:16	WATERMARK	RW	0	FIFO water mark setting for DMA enable bit 0: Disable 4: Enable Other: Reserved
15:3	–	–	0	Reserved
2:0	REQSRC	RW	0	DMA Request source enable bit 0: Disable 4: Enable Other: Reserved

19.7.11 LCM DMA Interrupt Status register (LCM_DMAIS)

Address offset: 0x250

Bit	Field	Access	Initial	Description
31:9	–	–	0	Reserved
8	DMAERRIF	RW1C	0	Interrupt status when DMA error When DMA request was assert, and the DMA response grant signal, but No read transaction at data register. To write 1 for clear status. 0: No interrupt 1: DMA error
7	CLRIF	RW1C	0	Interrupt status when clear procedure is complete. To write 1 for clear status. 0: No interrupt 1: Clear procedure is complete
6	UDRUNIF	RW1C	0	Interrupt status when FIFO underrun was found. To write 1 for clear status. 0: No interrupt 1: TX underrun was found
5	FIFOFIF	RW1C	0	Interrupt status when the data count of FIFO is full. To write 1 for clear status. 0: No interrupt 1: The data count of FIFO is full
4	–	–	0	Reserved
3	FIFOHFIF	RW1C	0	Interrupt status when the data count of FIFO over the half of FIFO. To write 1 for clear status. 0: No interrupt 1: The data count of FIFO over the half of FIFO
2:1	–	–	0x3	Reserved
0	FIFOEIF	RW1C	1	Interrupt status when FIFO is empty. To write 1 for clear status. 0: No interrupt 1: FIFO is empty

19.7.12 LCM DMA Interrupt Enable register (LCM_DMAIE)

Address offset: 0x254

Bit	Field	Access	Initial	Description
31:9	–	–	0	Reserved
8	DMAERRIE	RW	0	Interrupt enable for DMAERRIF 0: Disable 1: Enable
7	CLRIE	RW	0	Interrupt enable for CLRIF 0: Disable 1: Enable
6	UDRUNIE	RW	0	Interrupt enable for UDRUNIF 0: Disable 1: Enable
5	FIFOFIE	RW	0	Interrupt enable for FIFOFIF 0: Disable 1: Enable
4	–	–	0	Reserved
3	FIFOHFIE	RW	0	Interrupt enable for FIFOHFIF 0: Disable 1: Enable
2:1	–	–	0	Reserved
0	FIFOEIE	RW	0	Interrupt enable for FIFOEIF 0: Disable 1: Enable

19.7.13 LCM DMA Command register (LCM_DMACMD)

Address offset: 0x264

Bit	Field	Access	Initial	Description
31:0	CMD	W	0	Write command to DMA

20 ETHERNET (ETHMAC)

20.1 OVERVIEW

ETHMAC provides an efficient interface between the device and a network. It includes a DMA engine, on-chip memories (TXFIFO and RXFIFO), MAC, and MII/RMII/GMII/RGMII interfaces.

ETHMAC is an Ethernet controller that provides the AHB master capability and is fully compliant with the IEEE 802.3 specification for the Ethernet. ETHMAC supports MII/RMII interfaces. The ETHMAC Ethernet controller with the DMA function handles all data transfers between the system memory and the on-chip memories. With the DMA engine, this controller reduces the CPU loading, maximizes the performance, and minimizes the FIFO size. The MII/RMII interfaces support two specific data rates, 10 Mbps and 100 Mbps.

In order to reduce the processing load of the host CPU, ETHMAC implements TCP, UDP, IPv4, and IPv6 checksum generations and validations, and supports VLAN tagging.
ETHMAC provides a Wake-on-LAN function. It supports three wake-up events: Link status change, magic packet, and wake-up frame. This function allows a system containing ETHMAC to be woken up from the remote side.

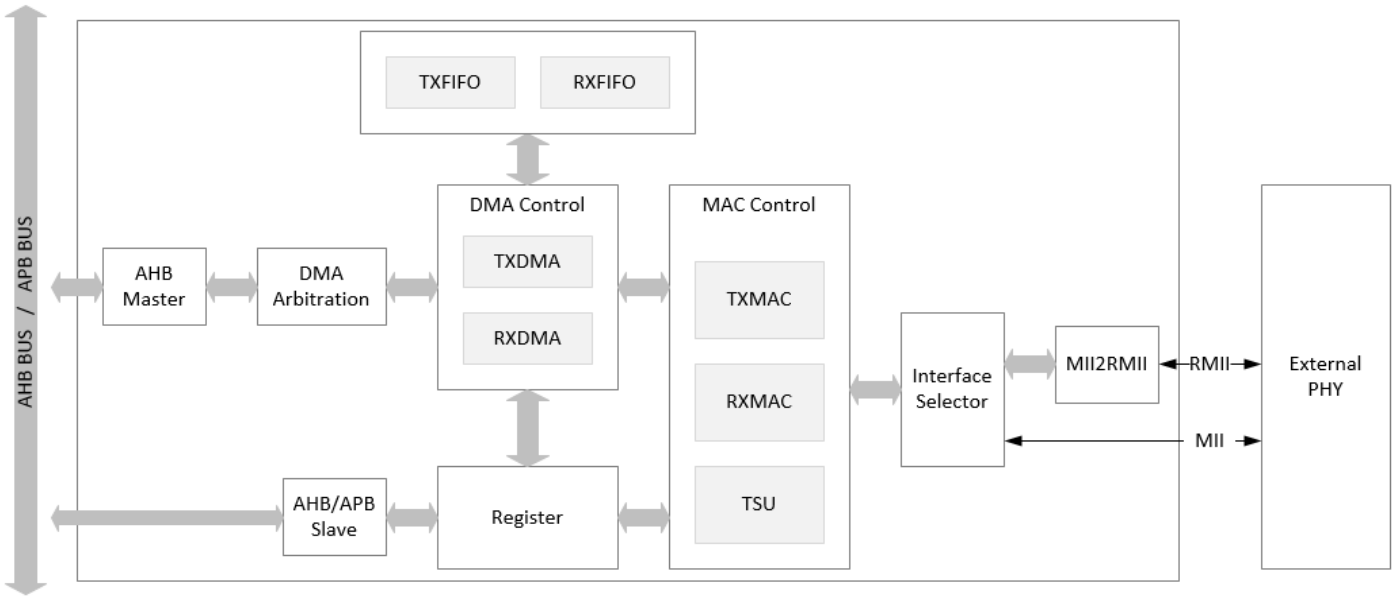
20.2 FEATURES

- DMA engine for transmitting and receiving packets
- Supports TCP, UDP, IPv4, and IPv6 checksum offloads
- Supports IEEE 802.1Q VLAN tag insertion and removal
- Supports high-priority queues for QoS and CoS applications
- Supports Wake-on-LAN function and three wake-up events: Link status change, magic packet, and wake-up frame
- Independent TX/RX FIFOs
- Supports half and full duplex operations
- Supports flow control for full duplex and back pressure for half duplex
- Supports MII/RMII interfaces
- Supports jumbo packets (9K bytes)
- Supports IEEE 1588v1 and 1588v2 frame recognition and timestamp generation
- Supports IEEE 802.3az LPI assertion and detection

20.3 PIN DESCRIPTION

Pin Name	Type	Description	GPIO Configuration
ETH_MDIO	I/O	MII or RMII Data input/output of PHY management	Depends on AFIO register
ETH_MDC	O	MII or RMII Clock of PHY management	Depends on AFIO register
ETH_MII_COL	I	MII Collision detect	Depends on AFIO register
ETH_MII_CRS	I	MII Carrier sense	Depends on AFIO register
ETH_TXER	O	MII Transmit error	Depends on AFIO register
ETH_MII_TX_EN	O	MII Transmit enable	Depends on AFIO register
ETH_MII_TX_CLK	I	MII Transmit clock	Depends on AFIO register
ETH_MII_TXD[3:0]	O	MII Transmit data	Depends on AFIO register
ETH_MII_RX_DV	I	MII Receive data valid	Depends on AFIO register
ETH_MII_RX_ER	I	MII Receive error	Depends on AFIO register
ETH_MII_RX_CLK	I	MII Receive clock	Depends on AFIO register
ETH_MII_RXD[3:0]	I	MII Receive data	Depends on AFIO register
ETH_RMII_CRS_DV	I	RMII Carrier sense/data valid	Depends on AFIO register
ETH_RMII_REF_CLK	I/O	RMII 50MHz REF_CLK	Depends on AFIO register
ETH_RMII_TX_EN	O	RMII Transmit enable	Depends on AFIO register
ETH_RMII_TXD[1:0]	O	RMII Transmit data	Depends on AFIO register
ETH_RMII_RX_ER	I	RMII Receive error	Depends on AFIO register
ETH_RMII_RXD[1:0]	I	RMII Receive data	Depends on AFIO register
ETH_PHY_LINKSTS	I	PHY links status	Depends on AFIO register
ETH_PHY_PDN	O	PHY power-down control	Depends on AFIO register

20.4 Block Diagram



20.4.1 DMA Arbitrator

The block acts as the bridge between two sets of control signals from TXDMA and RXDMA, and the control signal to AHB Master. It acts as an arbiter to decide if TXDMA or RXDMA is entitled to use AHB Master.

20.4.2 TXDMA

The block performs four main functions:
1. Read the transmit descriptor.
2. Move the transmit packet data from the transmit buffer to TXFIFO.
3. Control the read/write actions of TXFIFO.
4. Do the TCP/UDP/IP checksum calculation.

When TXDMA transmits packet to Ethernet, it will fetch the descriptor information and transmit the buffer base address and size. Then, TXDMA moves the transmit packet data from the corresponding transmit buffer to TXFIFO and requests TXMAC to read the transmit packet data with the help of TXDMA and sends it to the network. When the packet has been transmitted, TXMAC sends the transmit status to TXDMA and writes the transmit status back to the transmit descriptor.

20.4.3 RXDMA

RXDMA performs three main functions:
1. Read the receive descriptor and write the receive status to the receive descriptor
2. Move the receive packet data from RXFIFO to the receive buffer
3. Control the read/write actions of RXFIFO

When a packet is incoming, it will save the received packet into the RXFIFO. Then RXDMA moves the received packet from RXFIFO to the receiving buffer and writes the receive status to the receive descriptor.

20.4.4 TXMAC

TXMAC transmits packets from TXFIFO to Ethernet with CRC, preamble, jam generator, and transmit state machine included.

When TXMAC transmits a packet, it detects the Ethernet status and halts the transmission until Ethernet is free. Then, TXMAC adds preamble and CRC to this packet and sends the packet to Ethernet. If TXMAC detects the collision during a transmission, it sends the jam to Ethernet and determines whether the collision is excessive. If not, it waits for backing to off-time and transmits the packet again. TXMAC also performs the VLAN tag insertion.

Besides, if TXMAC can transmit no packets, then it will generate LPI (Low Power Idle) pattern when SEND_TX_LPI function is turn on. For IEEE 1588 PTP application, TXMAC can recognize the PTP frame to record precise time stamp.

20.4.5 RXMAC

RXMAC receives packets from Ethernet to RXFIFO, with address recognition circuit, CRC check circuit, and receive state machine included.

When a packet is incoming, RXDMA will pass the received packet data to RXFIFO from RXMAC. RXDMA will save the received packet in RXFIFO if both the CRC result and packet address are correct; otherwise, the packet will be discarded. RXMAC also performs the TCP/UDP/IP checksum verification and VLAN tag removal.

Besides, RXMAC can recognize LPI pattern from PHY RX channel. For IEEE 1588 PTP application, RXMAC can recognize the PTP frame to record precise timestamp for users.

20.4.6 TSU

TSU (Time Stamp Unit) control time stamp timer that users can program and adjust. Besides, it records the time stamp timer value when TX PTP frame or RX PTP frame is detected.

20.4.7 Interface Selector

The block acts as selector that switches the Ethernet interface by user programming. When users select to RMII, it will switch the data path to MII2RMII block. Otherwise, MII interface pass through to IP boundary.

20.4.8 MI2RMI

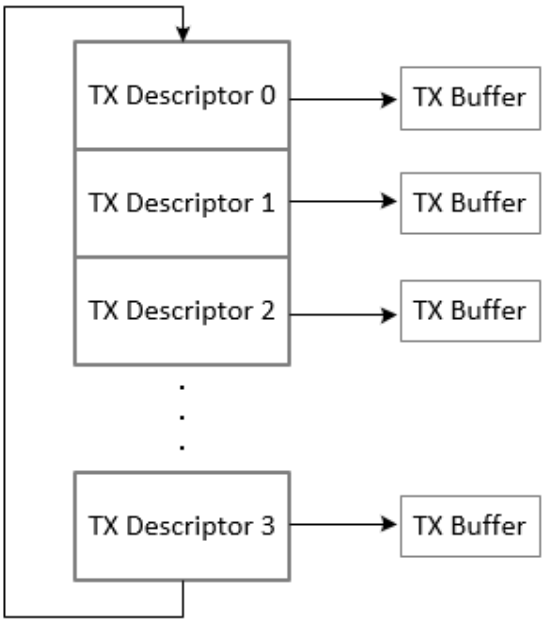
The block is a wrapper that translates the TXMAC/RXMAC MII signals to the RMII signals.

20.5 Function Description

20.5.1 Transmit Descriptors and Data Buffers

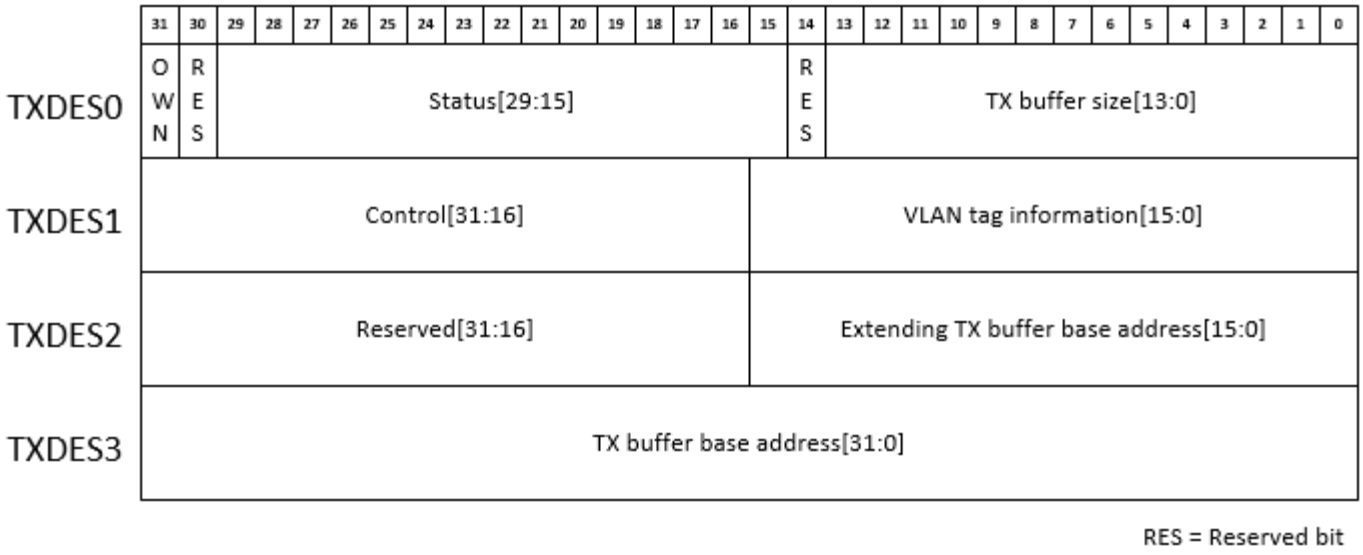
ETHMAC uses a descriptor ring to manage the transmit buffers. The transmit descriptors and data buffers are all located in the system memory. ETHMAC moves the transmit packet data from the transmit buffers in the system memory to TX FIFO inside ETHMAC and then transmits the packet to Ethernet. The transmit descriptors that reside in the system memory act as pointers to the transmit buffers.

Each transmit descriptor contains a transmit buffer (TX Buffer). A transmit buffer consists of either an entire frame or part of a frame, but not multiple frames. The transmit descriptor contains the transmit buffer status and the transmit buffer that can only contain the transmit data. ETHMAC supports two descriptor rings for transmission. These descriptor rings are the normal-priority transmit ring and high-priority transmit ring. The normal-priority transmit ring is for the normal packet transmission; the high-priority transmit ring is for the high priority packet transmission. Higher priority packets can be put into the high-priority transmit ring for quicker transmission.



The transmit descriptor structure is as follows.
Notes:

- The transmit descriptor ring must have at least two or more descriptors.
- The start address of each transmit descriptor must be 16-byte aligned.
- The maximum transmit packet size including CRC is 9216 bytes (9220 bytes if the VLAN tag is inserted).
- ETHMAC supports the IPV4/6 checksum offload. Software must be certain that the transmit packet is an IPV4/6 packet when software requests ETHMAC to do checksum offload. Besides, users must be make sure that the IPv4/6 packet is smaller than TX FIFO size when enabling the checksum offload.
- LLC packet is IEEE 802.3/802.2/SNAP format packet.
- ETHMAC does not support the following two packets for the checksum offload:
  - IEEE 802.3 with IEEE 802.2 packet
  - IEEE 802.3 with 802.1Q and 802.2 packets.
- When preparing a transmit packet which contains more than one transmit descriptor, the first transmit descriptor must be the last set descriptor of the transmit packet.
- When setting the transmit descriptor, TXDES0 must be set last. Thus, the setting procedure should be as follows:
  - Set TXDES3
  - Set TXDES2
  - Set TXDES1
  - Set TXDES0



RES = Reserved bit

TXDES0 contains the status bits and TX buffer size and descriptor ownership information.

Bit	Field	Description
31	OWN	TXDMA Ownership bit When set, it indicates that the descriptor is owned by ETHMAC. When reset, it indicates that the software owns the descriptor. ETHMAC clears this bit when it completes the frame transmission.
30	–	Reserved
29	FTS	First Transmit Segment descriptor When set, it indicates that this is the first descriptor of a TX packet.
28	LTS	Last Transmit Segment descriptor When set, it indicates that this is the last descriptor of a TX packet.
27	BUS_ERR	BUS response Error When set, it indicates that the system bus respond ERROR for TX descriptor and buffer reading. It is valid only when FTS = 1.
26:20	–	Reserved
19	CRC_ERR	CRC Error When CRC_ERR = 1 and DISCARD_CRCERR of ETHMAC_MACCTRL(Offset: 0x50) = 1, TXDMA will discard the transmit packet and will not send it to Ethernet.
18:16	–	Reserved
15	EDOTR	End Descriptor of Transmit Ring When set, it indicates that the descriptor is the last descriptor of the transmit ring.
14	–	Reserved
13:0	TXBUF_SIZE	Transmit buffer size in byte The transmit buffer size cannot be zero.

TXDES1 contains the control bits and VLAN tag information.

Bit	Field	Description
31	TXIC	Transmit Interrupt on Completion When set, ETHMAC will assert the transmit interrupt after the present frame has been transmitted. It is valid only when FTS = 1, TXINT_THR and TXINT_CYC of ETHMAC_TXITC(Offset: 0x30) = 0.
30	TX2FIC	Transmit to FIFO Interrupt on Completion When set, ETHMAC will assert the transmit interrupt after the present frame has been moved into TX FIFO. It is valid only when FTS = 1.
29:23	–	Reserved
22	LLC_PKT	The LLC packet is IEEE 802.3/802.2/SNAP format packet. When set, ETHMAC will treat the packet as the IEEE 802.3/802.2/SNAP LLC packet.

Bit	Field	Description
		It is valid only when FTS = 1.
21:20	PKT_TYPE	Packet Type When set, ETHMAC will treat the packet as IPv4 or IPv6 or other packet type. 0x0: IPv4 packet type 0x1: IPv6 packet type 0x2: Other packet type 0x3: Reserved It is valid only when FTS = 1.
19	IPCS_EN	IP Checksum offload enable When set, ETHMAC will offload the IP checksum. It is valid only when FTS = 1.
18	UDPCS_EN	UDP Checksum offload enable When set, ETHMAC will offload the UDP checksum. It is valid only when FTS = 1.
17	TCPCS_EN	TCP Checksum offload enable. When set, ETHMAC will offload the TCP checksum. It is valid only when FTS = 1.
16	INS_VLAN	Insert VLAN tag When set, 0x8100 (IEEE 802.1Q VLAN Tag Type) will be inserted after the source address, and two bytes of VLAN_TAGC will be inserted after the IEEE 802.1Q VLAN tag type. When cleared, the packet content will not be changed when transmitting to network. It is valid only when FTS = 1.
15:0	VLAN_TAGC	VLAN tag control information The 2-byte VLAN tag control information contains information from the upper layer of user priority, canonical format indicator, and VLAN ID. Please refer to IEEE 802.1Q for more VLAN tag information. Bit[15:13]: User priority Bit[12]: CFI (Canonical Format Indicator) Bit[11:0]: VID (VLAN Identifier) It is valid only when FTS = 1.

TXDES2 contains the extending transmit buffer base address if the system address is larger than 32 bits. Besides, if the system address is 32 bits, then user can ignore TXDES2 and ETHMAC will not read the field.

Bit	Field	Description
31:16	–	Reserved
15:0	EXT_TXBUF_BADR	Extending Transmit Buffer Base Address When the system address is larger than 4G Bytes mapping, then the field needs to specify the extending address field. For example, when the system address is 8G Bytes address mapping, then the address bit[32] is set to bit[0] of the field. And other case, 16G Bytes address => address bit [33:32] is set to bit[1:0] of the field. 32G Bytes address => address bit [34:32] is set to bit[2:0] of the field. 64G Bytes address => address bit [35:32] is set to bit[3:0] of the field. ... 128T Bytes address => address bit [46:32] is set to bit[14:0] of the field. 256T Bytes address => address bit [47:32] is set to bit[15:0] of the field.

TXDES3 contains the transmit buffer base address.

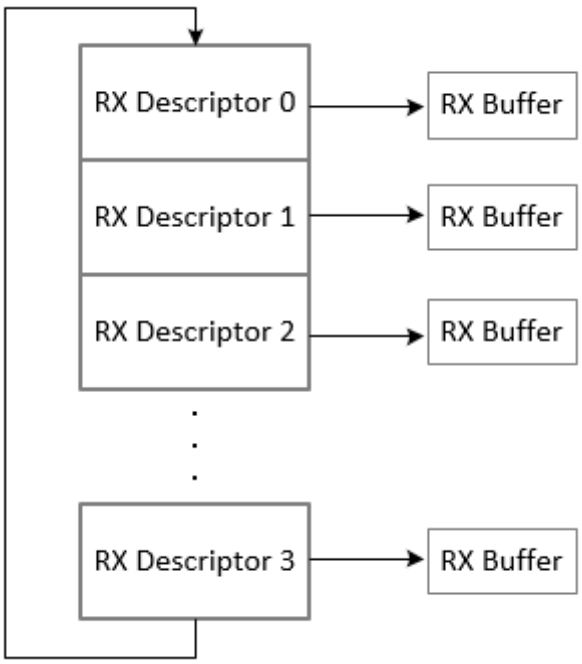
Bit	Field	Description
31:0	TXBUF_BADR	Transmit Buffer Base Address

20.5.2 Receiver Descriptors and Data Buffers

ETHMAC uses a descriptor ring to manage the receive buffers. The receive descriptors and data buffers are all located in the system memory. ETHMAC first stores the packet received from the network in the RX FIFO and then moves the received packet data to the receive buffers in system memory. The receive descriptors that reside in the system memory act as pointers to the receive buffers.

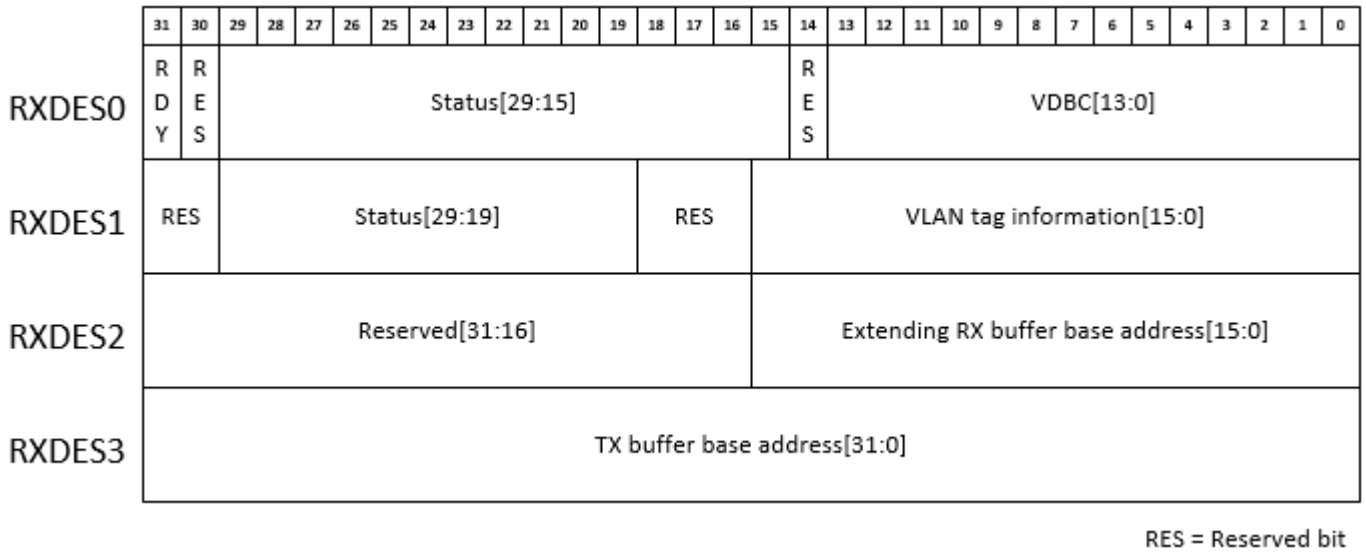
There is a descriptor ring for reception. The base address of the receive ring is in the ETHMAC_RXR_BADR (offset: 0x24) and ETHMAC_EXT_RXR_BADR (offset: 0x154).
Each receive descriptor contains a receive buffer. A receive buffer consists of either an entire frame or part of a frame, but not the multiple frames. The receive descriptor contains the receive buffer status and the receive buffer only contains the receive packet data.

The receive descriptor structure is as follows:



Notes:

- The start address of each receive descriptor must be 16-byte aligned.
- The maximum receive packet size is 9216 bytes (9220 bytes for packets with the VLAN tag).
- ETHMAC supports the IPV4/6 checksum offload. If the incoming packet is not an IPV4/6 packet, ETHMAC will not perform the checksum verification. The IPCS_FAIL, UDPCS_FAIL, and TCPCS_FAIL fields are always zeros.
- LLC packet is IEEE 802.3/802.2/SNAP format packet.
- ETHMAC does not support the following two packets for the checksum offload. They are IEEE 802.3 with IEEE 802.2 packet and IEEE 802.3 with 802.1Q and 802.2 packets.



RXDES0 contains the receive frame status, VDBC and descriptor ownership information.

Bit	Field	Description
31	RDY	RX packet ready When clear, it indicates that the descriptor is owned by ETHMAC. When set, it indicates that the descriptor is owned by software. ETHMAC sets this bit when it completes the frame reception or when the receive buffer of the receive descriptor is full.
30	–	Reserved
29	FRS	First Receive Segment descriptor When set, it indicates that this is the first descriptor of a received packet. Note: Bit[25:0] are valid only when FRS = 1.
28	LRS	Last Receive Segment descriptor When set, it indicates that this is the last descriptor of a received packet.
27	BUS_ERR	BUS response Error When set, it indicates that the system bus responds ERROR for RX descriptor reading and RX buffer writing. It is valid only when FTS = 1.
26	–	Reserved
25	PAUSE_FRAME	Pause Frame When set, it indicates that the receive packet is a pause frame.
24	PAUSE_OPCODE	Pause frame OP code When set, it indicates that there is pause frame OP code in the receive packet.
23	FIFO_FULL	FIFO Full When set, it indicates that RX FIFO is full when the packet is received.
22	RX_ODD_NB	Receive Odd Nibbles When set, it indicates that a packet is received with odd nibbles.
21	RUNT	Runt packet When set, it indicates that the received packet length is less than 64 bytes.
20	FTL	Frame Too Long When set, it indicates that the received packet length exceeds the long frame length. If JUMBO_LF = 0, the long frame length will be 1518 (1522 for the receive packet with the VLAN tag) bytes. If JUMBO_LF = 1, the long frame length will be 9216 (9220 for the receive packet with the VLAN tag) bytes.
19	CRC_ERR	CRC Error When set, it indicates that the CRC error occurs on the received packet.
18	RX_ERR	Receive Error When set, it indicates that receive error happens when receiving a packet.
17	BROADCAST	Broadcast Frame When set, it indicates that the received packet is a broadcast frame.
16	MULTICAST	Multicast Frame When set, it indicates that the received packet is a multicast frame.

Bit	Field	Description
15	EDORR	End Descriptor of Receive Ring When set, it indicates that the descriptor is the last descriptor of the receive ring.
14	–	Reserved
13:0	VDBC	Valid Data Byte Count The field indicates the valid data in the receive buffer. The unit is 1 byte.

RXDES1 contains the status bits and VLAN tag information.

Bit	Field	Description
31:30	–	Reserved
29:28	PTP_RX_TIMESTAMP	PTP frame Timestamp type 0x0: It is non-PTP frame. 0x1: It is PTP frame without timestamp. 0x2: It is PTP event frame and timestamp value at register offset 0x118 and 0x11C. 0x3: It is PTP peer frame and timestamp value at register offset 0x128 and 0x12C. The field is valid only when enabling the PTP function.
27	IPCS_FAIL	IP Checksum Failure When set, ETHMAC detects IP checksum failure. The field is valid only when the FRS = 1.
26	UDPCS_FAIL	UDP Checksum Failure The field is valid only when the FRS = 1.
25	TCPCS_FAIL	TCP Checksum Failure The field is valid only when the FRS = 1.
24	VLAN_AVA	VLAN Tag Available The field is valid only when the FRS = 1.
23	MF	IPv4 More Fragment (MF) 0x0: Last Fragment 0x1: More Fragments
22	LLC_PKT	The LLC packet is IEEE 802.3/802.2/SNAP format packet. When set, it indicates that the receive packet is the IEEE 802.3/802.2/SNAP LLC packet. The field is valid only when the FRS = 1.
21:20	PROTL_TYPE	Protocol Type These two bits indicate which protocol in the receive packet. 0x0: Not IP protocol 0x1: IP protocol 0x2: TCP/IP protocol 0x3: UDP/IP protocol The field is valid only when the FRS = 1.
19	IP6_TYPE	IP Version 6 Protocol Type When set, the IP Type is version 6; others are version 4. It is valid only when FTS = 1 and PROTL_TYPE is not 0x0.
18:16	–	Reserved
15:0	VLAN_TAGC	VLAN Tag Control information If the receive packet contains the VLAN tag, ETHMAC will extract four bytes from the receive packet. The 4-bytedata contains 0x8100 and 2-byte VLAN tag control information. ETHMAC will move the 2-byte VLAN tag control information to this field. The 2-byte VLAN tag control information contains information, from the upper layer of user priority, canonical format indicator, and VLAN ID. Please refer to IEEE 802.1Q for more VLAN tag information. Bit[15:13]: User priority Bit[12]: CFI (Canonical Format Indicator) Bit[11:0]: VID (VLAN Identifier) The field is valid only when the FRS = 1.

RXDES2 contains the extending transmit buffer base address if the system address is larger than 32 bits.
Besides, if the system address is 32 bits, then user can ignore RXDES2 and ETHMAC will not read the field.

Bit	Field	Description
31:16	–	Reserved
15:0	EXT_RXBUF_BADR	Extending Receive Buffer Base Address When the system address is larger than 4G Bytes mapping, then the field needs to specify

Bit	Field	Description
		the extending address field. For example, when the system address is 8G Bytes address mapping, then the address bit[32] is set to bit[0] of the field. And other case, 16G Bytes address => address bit [33:32] is set to bit[1:0] of the field. 32G Bytes address => address bit [34:32] is set to bit[2:0] of the field. 64G Bytes address => address bit [35:32] is set to bit[3:0] of the field. ... 128T Bytes address => address bit [46:32] is set to bit[14:0] of the field. 256T Bytes address => address bit [47:32] is set to bit[15:0] of the field.

RXDES3 contains the receive buffer base address.

Bit	Field	Description
31:0	RXBUF_BADR	Receive Buffer Base Address Receive buffer base address must be at least 8-byte alignment.

20.5.3 Transmitting Packets

When software wants to transmit a packet to Ethernet, it moves the packet data into the transmit buffer first. Then, software writes the packet length and position into the transmit descriptor and triggers ETHMAC to send the packet. After the entire packet has been moved into TX FIFO, ETHMAC begins to transmit it to Ethernet.
When the packet has been transmitted, ETHMAC asserts interrupt to notify software that the packet has been transmitted successfully. Higher priority packets can be put into the high priority descriptor for quicker transmission.

20.5.4 Receiving Packets

When there is an incoming packet, ETHMAC first saves the received packet in RX FIFO if the address check result is correct. After the incoming packet is successfully saved in RX FIFO, ETHMAC initiates Direct Memory Access (DMA) function to move the received packet data from RX FIFO to the system memory.
Then, ETHMAC asserts interrupt to notify software that the packet has been received successfully.

20.5.5 Ethernet Address Filtering

ETHMAC can be set to recognize any one of the Ethernet receive address groups described in the following table.

- RX_BROADPKT_EN: bit[11] of ETHMAC_MACCTRL (Offset: 0x50)
- RX_MULTIPKT_EN: bit[10] of ETHMAC_MACCTRL (Offset: 0x50)
- RX_HT_EN: bit[9] of ETHMAC_MACCTRL (Offset: 0x50)
- RX_ALLADR: bit[8] of MAC ETHMAC_MACCTRL (offset: 0x50)

DA = Destination address of the received frame.

RX_ALLADR	RX_MULTIPKT_EN	RX_BROADPKT_EN	RX_HT_EN	Received Frame Rule
0	0	0	0	- DA exactly matches ETHMAC_MAC_MADR (Offset: 0x8).
0	0	0	1	- DA exactly matches ETHMAC_MAC_MADR (Offset: 0x8). - DA is a multicast address. - DA passes the address filtering of the multicast address hash table.
0	0	1	0	- DA exactly matches ETHMAC_MAC_MADR (Offset: 0x8). - DA is a broadcast address.
0	0	1	1	- DA exactly matches ETHMAC_MAC_MADR (Offset: 0x8). - DA is a multicast address. - DA passes the address filtering of the multicast address hash table. - DA is a broadcast address
0	1	X	X	- DA exactly matches ETHMAC_MAC_MADR (Offset: 0x8). - DA is a multicast address.
1	X	X	X	- Reception of all frames on the network regardless of DA

20.5.6 DMA Arbitration Scheme

The DMA arbitration scheme is decided by RX_THR_EN **(**bit[6] of ETHMAC_DBLAC, offset: 0x3C).
When RX_THR_EN = 0, the DMA arbitration scheme does a fair arbitration between TXDMA and RXDMA. If both TXDMA and RXDMA request the DMA channel at the same time, the one last using the DMA channel has lower priority to get the DMA channel.

When RX_THR_EN is set, if the used space in RX FIFO is larger than or equal to RXFIFO_HTHR (bit[5:3] of ETHMAC_DBLAC, offset: 0x3C), RXDMA has higher priority over TXDMA by using the DMA channel.
RXDMA keeps the higher priority until the used space in RX FIFO is less than or equal to RXFIFO_LTHR (bit[2:0] of ETHMAC_DBLAC, offset: 0x3C). Then, TXDMA gets higher priority over RXDMA. So software must set RXFIFO_HTHR to be larger than RXFIFO_LTHR to keep ETHMAC working correctly.

20.5.7 Wake-On-LAN

ETHMAC supports the Wake-On-LAN function. The Wake-On-LAN function supports three wake-up events:
Link status change, magic packet, and wake-up frame.

20.5.7.1 Link Status Change

Link status change refers to the event where the link state to Ethernet changes. PHY offers a ETH_PHY_LINKSTS signal. If the link state to Ethernet changes, the state of ETH_PHY_LINKSTS signal will also change.

If ETHMAC enables the link status change mode, the link status change will be treated as a wake-up event.

20.5.7.2 Magic Packet

A magic packet contains a specific sequence consisting of 16 duplications of network the adaptor node address without breaks. The specific sequence must be preceded by 6 bytes of 0xFF. The format of a magic packet goes as follows:

DA + SA + … + 6 * (0xFFh) + …+ 16 * (network adaptor's node address) + …

DA = Destination address
SA = Source address

If ETHMAC enables the magic packet mode, a magic packet will be treated as a wake-up event.

20.5.7.3 Wake-up Frame

The purpose of the wake-up frame is to wake up a system when another machine on the network needs to communicate with the system. It does not require the application running on the remote machine to send a special wake-up frame pattern.
Instead, when ETHMAC is in the wake-up frame mode, it tries to identify certain interesting frames that are sent by the existing network protocols. Some examples are NETBIOS name lookups and ARP requests.

Before putting ETHMAC into the wake-up frame mode, the system should pass to the driver a list of wake-up frames that will wake up the system and the corresponding byte masks that correspond to bits will set to 1.
Each byte mask defines which bytes of the incoming frames should be compared with the corresponding wake-up frame in order to determine whether or not to accept the incoming frame as a wake-up event.

Besides, the packet length of wake-up frame must be at least 128 bytes.

Table. Wake-up Frame Format Example

Wake-up Frame Format								
Byte content	0x00	0x25	0x67	0x90	0x44	0xA3	0x6C	……

Byte mask	0	0	0	1	1	0	1	……

There are two ways to identify if a received packet is a wake-up frame. Signature matching is used in ETHMAC.

- Exact matching: ETHMAC needs many registers to store the entire byte content and byte mask for each wake-up frame. When a packet arrives from the network, ETHMAC checks those bytes of the incoming frame that correspond to bits that set to 1 in the byte mask for each wake-up frame. If the check result is ok for any wake-up frame and if the incoming frame passes the standard CRC check, ETHMAC would treat it as a wake-up event.

- Signature matching: ETHMAC needs a CRC generation circuit and registers to save the entire byte mask and 4-byte CRC register for each wake-up frame. The driver calculates a CRC value based on those bytes of any wakeup frame correspond to bits that set to 1 in the byte mask. Besides, the start offset of byte mask field is Destination Address (DA) field. The driver stores the resulting value and corresponding byte mask into ETHMAC. When the wake-up frame mode is enabled as a frame arriving from the network, each CRC generator calculates a CRC value based on those bytes of the incoming frame which correspond to bits that set to 1 in the CRC generator byte mask. If the calculated value matches the stored value for any wake-up frame and the incoming frame passes the standard CRC check, ETHMAC would treat it as a wake-up event.

20.5.8 Flow Control

ETHMAC implements the flow control function. It supports IEEE802.3x flow control for the full-duplex mode and backpressure for the half-duplex mode.

The IEEE802.3x flow control is used in the full-duplex mode. When A and B are reciprocally transmitting and receiving packets in the full-duplex mode, if RX FIFO in B is nearly full, B sends a pause frame to A in order to avoid loss of the received packet. Then A is inhibited from transmitting packets for a specified period of time. B consumes the received data during the specified period of time. A continues to send packets to B after the pause time has lapsed. Here is a brief account of the features of the flow control in the full-duplex mode:

- Software configures the pause time of the pause frame.
- ETHMAC sends the pause frame according to the low/high threshold of RX FIFO.
- Software sends the pause frame by writing the register.

The back pressure mode is used in the half-duplex operation. When A is transmitting and receiving packets in the half-duplex mode, if RX FIFO in A is nearly full, A will send a jam pattern to generate collisions to avoid incoming packets from being saved into RX FIFO. A consumes the received data as soon as possible during the period of time. A does not send a jam pattern to receive packets again when RX FIFO is not nearly full. Here is a brief account of the features of the back pressure mode:

- Software configures the length of the jam.
- ETHMAC sends jam according to the low/high threshold of RX FIFO.

20.5.9 Supported Ethernet Frame Type of Checksum Offload

Two Ethernet frame formats are supported in order to correctly identify the IP and TCP/UDP headers. ETHMAC supports Ethernet Type II format and IEEE 802.3/802.2/SNAP format as shown below.
VLAN tagging is also supported on top of these two frame formats.

- Ethernet Type II format

Destination Address	Source Address	Type	Data	CRC
6-byte	6-byte	2-byte	n-byte	4-byte

- IEEE 802.3/802.2/SNAP format

Destination Address	Source Address	Length	DSAP	SSAP	Control	OUI	Type	Data	CRC
6-byte	6-byte	2-byte	1-byte	1-byte	1-byte	3-byte	2-byte	n-byte	4-byte
SNAP : DSAP=0xAA, SSAP=0xAA, Control=0x03, OUI=0x000000.									

20.5.10 IEEE 1588 PTP Frame Reorganization

If users configure the PTP function, then ETHMAC supports IEEE 1588 PTP frame recognition.
The IEEE 1588 versions 1 and 2 specify different PTP frame types on Ethernet. ETHMAC can recognize the following PTP frames:

For IEEE 1588 version 1:
- PTP over UDP over IPv4 with multicast address

For IEEE 1588 version 2:
- PTP over UDP over IPv4 with multicast address
- PTP over UDP over IPv4 with unicast address
- PTP over UDP over IPv6 with multicast address
- PTP over Ethernet with multicast address

For the detailed format information, please refer to the IEEE 1588-2002 and IEEE 1588-2008 specifications.
Meanwhile, the PTP clock cycle value must specify to the PTP Period Increment 0/1 register (offset 0x13C, 0x140) and PTP Period Offset Increment register (offset 0x144).

Example 1, the frequency of PTPClk is 125 MHz, period is 8 ns. Then,
- PTP Period Increment 0 register is 0x8,
- PTP Period Increment 1 register is 0x0,
- PTP Period Offset Increment register is 0x0.

Example 2, the frequency of PTPClk is 160 MHz, period is 6.25 ns. Then,
- PTP Period Increment 0 register is 0x6,
- PTP Period Increment 1 register is 0xEE6B280, due to 0.25ns = 250000000nns.
- PTP Period Offset Increment register is 0x0.

Example 3, the frequency of PTPClk is 133 MHz, period is 7.5187…ns. Due to the period is indivisibility, so the clock has 133 cycles at every 1us. So the timer with a 133MHz clock source is constructed by incrementing by 7ns for 132 cycles and then incrementing by 76 (132 x 7 + 76 = 1000). It means that the period is 7 ns and increment is 76 ns at every 133th clock cycle.
- PTP Period Increment 0 register is 0x7,
- PTP Period Increment 1 register is 0x0
- PTP Period Offset Increment register is 0x0084004C.

20.5.11 Software Reset

The software reset function is at bit[31] of ETHMAC_MACCTRL (offset 0x50). The register value can be software reset to zero except for the following registers:

- Keep the register value:
  - Normal Priority Transmit Poll Demand register (Offset 0x18)
  - Receive Poll Demand register (Offset 0x1C)
  - High Priority Transmit Poll Demand register (Offset 0x28)
  - DMA/FIFO State register (Offset 0x40)
  - LOOP_EN bit of MAC Control register (Bit[21], Offset 0x50)
  - MAC_SPEED bits of MAC Control register (Bit[25:24], Offset 0x50)
  - Wake-On-LAN Status register (Offset 0x74)
  - Normal Priority Transmit Ring Pointer register (Offset 0x90)
  - High Priority Transmit Ring Pointer register (Offset 0x94)
  - Receive Ring Pointer register (Offset 0x98)
  - RGMII In-Band Status register (Offset 0xD4)
  - GMAC Interface Selection register (Offset 0xE8)
  - SW Reset Cycle Count register (Offset 0xEC)
  - PTP Timer register (Offset 0x130, 0x134, and 0x138)

- Reset to default value:
  - Normal Priority Transmit Ring Base Address register (Offset 0x20)
  - Receive Ring Base Address register (Offset 0x24)
  - High Priority Transmit Ring Base Address register (Offset 0x2C)
  - DMA Burst Length and Arbitration Control register (Offset 0x3C)
  - Transmit Priority Arbitration and FIFO Control register (Offset 0x48)
  - Receive Buffer Size register (Offset 0x4C)
  - PHY Control register (Offset 0x60)
  - Flow Control register (Offset 0x68)
  - Back Pressure register (Offset 0x6C)
  - EEE Control register (Offset 0xF0)

20.6 ETHERNET REGISTERS

Base Address: 0x4005 5000

There are seven access types: RW: Read or write, R: Read Only, W: Write only, RWS: Read and Write one to set and automatically clear by hardware, RW1C: Read and Set by Hardware and write 1 to clear, RW0C: Read and Set by Hardware and write 0 to clear, PW: Write by protected sequence.

Offset	Register	Description
0x0000	ETHMAC_IS	Interrupt Status register
0x0004	ETHMAC_IE	Interrupt Enable register
0x0008	ETHMAC_MAC_MADR	MAC Most Significant Address register
0x000C	ETHMAC_MAC_LADR	MAC Least Significant Address register
0x0010	ETHMAC_MAHT0	Multicast Address Hash Table 0 register
0x0014	ETHMAC_MAHT1	Multicast Address Hash Table 1 register
0x0018	ETHMAC_NPTXPD	Normal Priority Transmit Poll Demand register
0x001C	ETHMAC_RXPD	Receive Poll Demand register
0x0020	ETHMAC_NPTXR_BADR	Normal Priority Transmit Ring Base Address register
0x0024	ETHMAC_RXR_BADR	Receive Ring Base Address register
0x0028	ETHMAC_HPTXPD	High Priority Transmit Poll Demand register
0x002C	ETHMAC_HPTXR_BADR	High Priority Transmit Ring Base Address register
0x0030	ETHMAC_TXITC	TX Interrupt Timer Control register
0x0034	ETHMAC_RXITC	RX Interrupt Timer Control register
0x0038	ETHMAC_APTC	Automatic Polling Timer Control register
0x003C	ETHMAC_DBLAC	DMA Burst Length and Arbitration Control register
0x0040	ETHMAC_DMAFIFOS	DMA/FIFO State register
0x0044	–	Reserved
0x0048	ETHMAC_TPAFC	Transmit Priority Arbitration and FIFO Control register
0x004C	ETHMAC_RBSZ	Receive Buffer Size register
0x0050	ETHMAC_MACCR	MAC Control register
0x0054	ETHMAC_MACSR	MAC Status register
0x0058	ETHMAC_TM	PHY Control register
0x005C	–	Reserved
0x0060	ETHMAC_PHYCTRL	PHY Control register
0x0064	ETHMAC_PHYDATA	PHY Data register
0x0068	ETHMAC_FCTRL	Flow Control register
0x006C	ETHMAC_BP	Back Pressure register
0x0070	ETHMAC_WOLCTRL	Wake-On-LAN Control register
0x0074	ETHMAC_WOLST	Wake-On-LAN Status register
0x0078	ETHMAC_WFCRC	Wake-up Frame CRC register
0x007C	–	Reserved
0x0080	ETHMAC_WFBM1	Wake-up Frame Byte Mask 1st Double-word register
0x0084	ETHMAC_WFBM2	Wake-up Frame Byte Mask 2nd Double-word register
0x0088	ETHMAC_WFBM3	Wake-up Frame Byte Mask 3rd Double-word register
0x008C	ETHMAC_WFBM4	Wake-up Frame Byte Mask 4th Double-word register
0x0090	ETHMAC_NPTXR_PTR	Normal Priority Transmit Ring Pointer register
0x0094	ETHMAC_HPTXR_PTR	High Priority Transmit Ring Pointer register
0x0098	ETHMAC_RXR_PTR	Receive Ring Pointer register
0x009C	–	Reserved
0x00A0	ETHMAC_TX_CNT1	TPKT_CNT Counter register
0x00A4	ETHMAC_TX_CNT2	TXMCOL_CNT and TXSCOL_CNT Counter register
0x00A8	ETHMAC_TX_CNT3	TXMCOL_CNT and TXSCOL_CNT Counter register
0x00AC	ETHMAC_TX_CNT4	TXLCOL_CNT and TXUNDERUN_CNT Counter register
0x00B0	ETHMAC_RX_CNT1	RPKT_CNT Counter register
0x00B4	ETHMAC_RX_CNT2	BROPKT_CNT Counter register
0x00B8	ETHMAC_RX_CNT3	MULPKT_CNT Counter register
0x00BC	ETHMAC_RX_CNT4	RPF_CNT and AEP_CNT Counter register
0x00C0	ETHMAC_RX_CNT5	RUNT_CNT Counter register
0x00C4	ETHMAC_RX_CNT6	CRCER_CNT and FTL_CNT Counter register
0x00C8	ETHMAC_RX_CNT7	RCOL_CNT and RLOST_CNT Counter register
0x00CC	ETHMAC_BIST	BIST Pattern Control register

0x00D0	ETHMAC_BMRCTRL	Broadcast and Multicast Receiving Control register
0x00D4	ETHMAC_RGMII_IBS	RGMII In-band Status register
0x00D8	ETHMAC_RX_CNT8	RCEE_CNT Counter register
0x00DC	ETHMAC_RX_CNT9	RPKTB_CNT Counter register
0x00E0	ETHMAC_ERCTRL	Error Response Control register
0x00E4	–	Reserved
0x00E8	ETHMAC_GIS	GMAC Interface Selection register
0x00EC	ETHMAC_SWRCC	SW Reset Cycle Count register
0x00F0	ETHMAC_EEECTRL	EEE Control register
0x00F4 – 0x00FC	–	Reserved
0x0100	ETHMAC_PTP_RXUIPDA	PTP RX Unicast IP Destination Address register
0x0104	ETHMAC_PTP_TXUIPDA	PTP TX Unicast IP Destination Address register
0x0108 – 0x010C	–	Reserved
0x0110	ETHMAC_PTP_TX	PTP TX Event Frame register
0x0114	ETHMAC_PTP_TX2	PTP TX Event Frame register
0x0118	ETHMAC_PTP_RX	PTP RX Event Frame register
0x011C	ETHMAC_PTP_RX2	PTP RX Event Frame register
0x0120	ETHMAC_PTP_TXP	PTP TX Peer Frame register
0x0124	ETHMAC_PTP_TXP2	PTP TX Peer Frame register
0x0128	ETHMAC_PTP_RXP	PTP RX Peer Frame register
0x012C	ETHMAC_PTP_RXP2	PTP RX Peer Frame register
0x0130	ETHMAC_PTP_TMR	PTP Timer register
0x0134	ETHMAC_PTP_TMR2	PTP Timer register
0x0138	ETHMAC_PTP_TMR3	PTP Timer register
0x013C	ETHMAC_PTP_PER0	PTP Period Increment 0 register
0x0140	ETHMAC_PTP_PER1	PTP Period Increment 1 register
0x0144	ETHMAC_PTP_OFF	PTP Period Offset Increment register
0x0148	ETHMAC_PTP_ADJ	PTP Timer Adjustment register

20.6.1 Ethernet Interrupt Status register (ETHMAC_IS)

Address Offset: 0x00

Bit	Field	Access	Initial	Description
31:25	–	–	0	Reserved
24	PDELAY_RESP_OUT	RW1C	0	PTP pdelay_resp frame transmitted status 0: No effect 1: PTP pdelay_resp frame transmitted
23	PDELAY_RESP_IN	RW1C	0	PTP pdelay_resp frame received status 0: No effect 1: PTP pdelay_resp frame received
22	PDELAY_REQ_OUT	RW1C	0	PTP pdelay_req frame transmitted status 0: No effect 1: PTP pdelay_req frame transmitted
21	PDELAY_REQ_IN	RW1C	0	PTP pdelay_req frame received status 0: No effect 1: PTP pdelay_req frame received
20	DELAY_REQ_OUT	RW1C	0	PTP delay_req frame transmitted status 0: No effect 1: PTP delay_req frame transmitted
19	DELAY_REQ_IN	RW1C	0	PTP delay_req frame received status 0: No effect 1: PTP delay_req frame received
18	SYNC_OUT	RW1C	0	PTP sync frame transmitted status 0: No effect 1: PTP sync frame transmitted
17	SYNC_IN	RW1C	0	PTP sync frame received status 0: No effect 1: PTP sync frame received
16	TSU_SEC_INC	RW1C	0	PTP TSU second register status 0: No effect 1: PTP TSU second register indication

Bit	Field	Access	Initial	Description
15:13	–	–	0	Reserved
12	RX_LPI_IN	RW1C	0	RX LPI mode enter status 0: No effect 1: Enter RX LPI mode
11	RX_LPI_EXIT	RW1C	0	RX LPI mode exit status 0: No effect 1: Exit RX LPI mode
10	HPTXBUF_UNAVA	RW1C	0	High-priority transmit buffer unavailable status 0: No effect 1: High-priority transmit buffer is unavailable
9	PHYSTS_CHG	RW1C	0	PHY link status 0: No effect 1: PHY link status changing
8	BUS_ERR	RW1C	0	Bus error status 0: No effect 1: Master receives the ERROR response
7	TPKT_LOST	RW1C	0	Packets transmitted lost status 0: No effect 1: Packets transmitted to Ethernet are lost due to late collision, excessive collision or under-run
6	NPTXBUF_UNAVA	RW1C	0	Normal priority transmit buffer unavailable status 0: No effect 1: Unavailable
5	TPKT2F	RW1C	0	TXDMA moved data to FIFO status 0: No effect 1: TXDMA has moved data into the TX FIFO
4	TPKT2E	RW1C	0	Packets transmitted to Ethernet status 0: No effect 1: Packets transmitted to Ethernet successfully
3	RPKT_LOST	RW1C	0	Received packet lost status 0: No effect 1: Received packet is lost due to RX FIFO full
2	RXBUF_UNAVA	RW1C	0	Receiving buffer unavailable status 0: No effect 1: Unavailable
1	RPKT2F	RW1C	0	Packets received to FIFO status 0: No effect 1: Packets received to RX FIFO successfully
0	RPKT2B	RW1C	0	RXDMA received packets to buffer status 0: No effect 1: RXDMA has received packets to the RX buffer successfully

20.6.2 Ethernet Interrupt Enable register (ETHMAC_IE)

Address Offset: 0x04

Bit	Field	Access	Initial	Description
31:25	–	–	0	Reserved
24	PDELAY_RESP_OUT_EN	RW	0	PTP pdelay_resp frame transmitted interrupt enable bit 0: Disable 1: Enable
23	PDELAY_RESP_IN_EN	RW	0	PTP pdelay_resp frame received interrupt enable bit 0: Disable 1: Enable
22	PDELAY_REQ_OUT_EN	RW	0	PTP pdelay_req frame transmitted interrupt enable bit 0: Disable 1: Enable
21	PDELAY_REQ_IN_EN	RW	0	PTP pdelay_req frame received interrupt enable bit 0: Disable 1: Enable
20	DELAY_REQ_OUT_EN	RW	0	PTP delay_req frame transmitted interrupt enable bit 0: Disable 1: Enable

Bit	Field	Access	Initial	Description
19	DELAY_REQ_IN_EN	RW	0	PTP delay_req frame received interrupt enable bit 0: Disable 1: Enable
18	SYNC_OUT_EN	RW	0	PTP sync frame transmitted interrupt enable bit 0: Disable 1: Enable
17	SYNC_IN_EN	RW	0	PTP sync frame received interrupt enable bit 0: Disable 1: Enable
16	TSU_SEC_INC_EN	RW	0	PTP TSU second register interrupt enable bit 0: Disable 1: Enable
15:13	–	–	0	Reserved
12	RX_LPI_IN_EN	RW	0	RX LPI mode enter interrupt enable bit 0: Disable 1: Enable
11	RX_LPI_EXIT_EN	RW	0	RX LPI mode exit interrupt enable bit 0: Disable 1: Enable
10	HPTXBUF_UNAVA_EN	RW	0	High-priority transmit buffer unavailable interrupt enable bit 0: Disable 1: Enable
9	PHYSTS_CHG_EN	RW	0	PHY link status changing interrupt enable bit 0: Disable 1: Enable
8	BUS_ERR_EN	RW	0	Bus error interrupt enable bit 0: Disable 1: Enable
7	TPKT_LOST_EN	RW	0	Packets transmitted lost interrupt enable bit 0: Disable 1: Enable
6	NPTXBUF_UNAVA_EN	RW	0	Normal priority transmit buffer unavailable interrupt enable bit 0: Disable 1: Enable
5	TPKT2F_EN	RW	0	TXDMA moved data to FIFO interrupt enable bit 0: Disable 1: Enable
4	TPKT2E_EN	RW	0	Packets transmitted to Ethernet interrupt enable bit 0: Disable 1: Enable
3	RPKT_LOST_EN	RW	0	Received packet lost interrupt enable bit 0: Disable 1: Enable
2	RXBUF_UNAVA_EN	RW	0	Receiving buffer unavailable interrupt enable bit 0: Disable 1: Enable
1	RPKT2F_EN	RW	0	Packets received to FIFO interrupt enable bit 0: Disable 1: Enable
0	RPKT2B_EN	RW	0	RXDMA received packets to buffer interrupt enable bit 0: Disable 1: Enable

20.6.3 Ethernet MAC Most Significant Address register (ETHMAC_MAC_MADR)

Address Offset: 0x08

Bit	Field	Access	Initial	Description
31:0	MAC_MADR	RW	0	The most significant two bytes of the MAC address.

20.6.4 Ethernet MAC Least Significant Address register (ETHMAC_MAC_LADR)

Address Offset: 0x0C

Bit	Field	Access	Initial	Description
31:0	MAC_MADR	RW	0	The least significant two bytes of the MAC address.

20.6.5 Ethernet MAC Multicast Address Hash Table 0 register (ETHMAC_MAHT0)

Address Offset: 0x10

Bit	Field	Access	Initial	Description
31:0	MAHT	RW	0	Multicast address hash table bytes 3 ~ 0 (Hash table 31:0)

20.6.6 Ethernet MAC Multicast Address Hash Table 0 register (ETHMAC_MAHT1)

Address Offset: 0x14

Bit	Field	Access	Initial	Description
31:0	MAHT	RW	0	Multicast address hash table bytes 3 ~ 0 (Hash table 63:32).

20.6.7 Ethernet MAC Normal Priority Transmit Poll Demand register (ETHMAC_NPTXPD)

Address Offset: 0x18

Bit	Field	Access	Initial	Description
31:0	NPTXPD	W	0	When writing a value to this register, ETHMAC reads the normal priority transmit descriptor and checks the TXDMA OWN bit. If TXDMA OWN bit = 1, it will move the transmit buffer data to TX FIFO.

20.6.8 Ethernet MAC Receive Poll Demand register (ETHMAC_RXPD)

Address Offset: 0x1C

Bit	Field	Access	Initial	Description
31:0	RXPD	W	0	When writing a value to this register, ETHMAC reads the receive descriptor and checks the RXDMA RDY bit. If RXDMA RDY bit = 1, it will move the receive packet data from the RX FIFO to the receiving buffer in the system memory.

20.6.9 Ethernet MAC Normal Priority Transmit Ring Base Address register (ETHMAC_NPTXR_BADR)

Address Offset: 0x20

Bit	Field	Access	Initial	Description
31:0	NPTXR_BADR	RW	0	Base address of the normal priority transmit ring. The base address must be 16-byte aligned. ETHMAC treats the base address bits 3 ~ 0 as 0 when reading descriptors if bits 3 ~ 0 are not zero.

20.6.10 Ethernet MAC Receive Ring Base Address register (ETHMAC_RXR_BADR)

Address Offset: 0x24

Bit	Field	Access	Initial	Description
31:0	RXR_BADR	RW	0	Base address of the receive ring. The base address must be 16-byte aligned. ETHMAC treats the base address bits 3 ~ 0 as 0 when reading descriptors if bits 3 ~ 0 are not zero.

20.6.11 Ethernet MAC High Priority Transmit Poll Demand register (ETHMAC_HPTXPD)

Address Offset: 0x28

Bit	Field	Access	Initial	Description
31:0	HPTXPD	W	0	When writing a value to the register, ETHMAC reads the high-priority transmit descriptor process and checks the TXDMA OWN bit. If TXDMA OWN bit = 1, it will move the transmit buffer data into TX FIFO.

20.6.12 Ethernet MAC High Priority Transmit Ring Base Address register (ETHMAC_HPTXR_BADR)

Address Offset: 0x2C

Bit	Field	Access	Initial	Description
31:0	HPTXR_BADR	RW	0	Base address of the high-priority transmit ring. The base address must be 16-byte aligned. ETHMAC treats the base address bits 3 ~ 0 as 0 when reading descriptors if bits 3~ 0 are not zero.

20.6.13 Ethernet MAC TX Interrupt Timer Control register (ETHMAC_TXITC)

Address Offset: 0x30

Bit	Field	Access	Initial	Description
31:17	–	–	0	Reserved
16	TXINT_TIME_SEL	RW	0	The period of TX cycle time 0: 100 Mbps mode → 5.12 us/10 Mbps mode → 51.2 us 1: 100 Mbps mode → 81.92 us/10 Mbps mode → 819.2 us
15:8	TXINT_CYC	RW	0	The maximum wait time to issue the transmit interrupt after a packet has been transmitted by ETHMAC. The time unit is 1 TX cycle time. The function will be disabled if TXINT_CYC = 0. When TXINT_THR = 0 and TXINT_CYC = 0, ETHMAC will issue a transmit interrupt or will not depend on TXIC in TXDES1.
7	–	–	0	Reserved
6:4	TXINT_THR	RW	0	The maximum number of transmit interrupts that are pending before an interrupt is generated. When TXINT_THR is not equal to 0, ETHMAC will issue a transmit interrupt if the transmit packet number transmitted by ETHMAC reaches TXINT_THR. When TXINT_THR = 0 and TXINT_CYC = 0, ETHMAC will issue a transmit interrupt or will not depend on TXIC in TXDES1.
3:2	–	–	0	Reserved
1:0	TXINT_THR_UNIT	RW	0	This field defines the unit of TXINT_THR. 0: 1 packets 1: 4 packets 2: 16 packets 3: 64 packets

The TX Interrupt Timer Control register allows the software driver to reduce the number of transmit interrupts (TPKT2E, bit 4 of ETHMAC_IS) by setting the register. This lowers the CPU utilization for handling a large number of interrupts.

The register defines the threshold values for the transmit packet number, and one associated timer. The threshold value defines the maximum number of the transmit interrupts that can be pending before an interrupt is generated.

The timer defines the maximum wait time to issue the transmit interrupt after a packet has been transmitted by ETHMAC. The threshold value and timer combination allow batching of several packets into a single interrupt with a limit for how long it will be pending. The combination prevents throughput from being impeded in the heavy traffic and the time limit prevents resources from being held for too long in the low traffic .

The mitigation mechanism is similar for both the receive and transmit interrupts. There is a “transmitted packet counter” in ETHMAC to count the packets transmitted by ETHMAC. When the counter reaches TXINT_THR, and TXINT_THR is not equal to 0, ETHMAC will issue the transmit interrupt. And the counter will be cleared when the transmit interrupt is issued.

Below lists the condition for ETHMAC to issue a transmit interrupt.

TXINT_THR	TXINT_CYC	Action
Zero	Zero	● The transmit interrupt will be issued after a packet is transmitted and TXIC of the packet is set. ● Clear the transmitted packet counter.
Zero	Non-Zero	● The transmit interrupt will be issued after a packet is transmitted and timer reaches the value of TXINT_CYC. ● Clear the transmitted packet counter.
Non-Zero	Zero	● The transmit interrupt will be issued if TPKT_CNT = TXINT_THR. ● Clear the transmitted packet counter.
Non-Zero	Non-Zero	● The transmit interrupt will be issued if the following condition holds: - TPKT_CNT = TXINT_THR. - TPKT_CNT = 1 and timer reaches the value of TXINT_CYC. ● Clear the transmitted packet counter.

20.6.14 Ethernet RX Interrupt Timer Control register (ETHMAC_RXITC)

Address Offset:0x34

Bit	Field	Access	Initial	Description
31:28	–	–	0	Reserved
27:20	RXINT_RST	RW	0	The refresh time. The reset counter will be cleared once one packet is received by ETHMAC. The time unit is 1 RX cycle time. When RXINT_CYC = 0, the function will be disabled. If RXINT_THR = 0 and RXINT_CYC = 0, a receive interrupt will be issued when a packet is received by ETHMAC.
19:17	–	–	0	Reserved
16	RXINT_TIME_SEL	RW	0	The RX cycle time 0: 100 Mbps mode → 5.12 us/10 Mbps mode → 51.2 us 1: 100 Mbps mode → 81.92 us/10 Mbps mode → 819.2 us
15:8	RXINT_CYC	RW	0	The maximum wait time to issue a receive interrupt after a packet has been received by ETHMAC. The time unit is 1 RX cycle time. When RXINT_CYC = 0, the function is disabled. If RXINT_THR = 0 and RXINT_CYC = 0, a receive interrupt will be issued when a packet is received by ETHMAC.
7	–	–	0	Reserved
6:4	RXINT_THR	RW	0	The maximum number of receive interrupts that are pending before an interrupt is generated. When RXINT_THR is not equal to 0, ETHMAC will issue a receive interrupt if the receive packet number received by ETHMAC reaches RXINT_THR. If RXINT_THR = 0 and RXINT_CYC = 0, a receive interrupt will be issued when ETHMAC finishes receiving a receive packet.
3:2	–	–	0	Reserved
1:0	RXINT_THR_UNIT	RW	0	This field defines the unit of RXINT_THR. 0: 1 packets 1: 4 packets 2: 16 packets 3: 64 packets

The RX Interrupt Timer Control register allows the software driver to reduce the number of the receive interrupts (RPKT2B , bit 0 of ETHMAC_IS) by setting the register. This lowers the CPU utilization for handling a large number of interrupts.

The register define threshold values for the receive packet number, and one associated timer. The threshold value defines the maximum number of receive interrupts that can be pending before an interrupt is generated. The timer defines the maximum wait time to issue the receive interrupt after a packet has been received by ETHMAC. The threshold value and timer combination allow batching of several packets into a single interrupt with a limit for how long it will be pending. The combination prevents throughput from being impeded in the heavy traffic and the time limit prevents resources from being held for too long in the low traffic .

There is a “Received packet counter” in ETHMAC to count the packets received by ETHMAC. When the counter reaches RXINT_THR, and RXINT_THR is not equal to 0, ETHMAC will issue a receive interrupt. And the counter will be cleared when the receive interrupt is issued.

Below lists the condition for ETHMAC to issue a receive interrupt.

RXINT_THR	RXINT_CYC	Action
Zero	Zero	● The receive interrupt will be issued after a packet is received by ETHMAC. ● Clear the received packet counter.
Zero	Non-Zero	● The receive interrupt will be issued after a packet is received by ETHMAC and timer reaches the value of RXINT_CYC. ● Clear the received packet counter.
Non-Zero	Zero	● The receive interrupt will be issued if the received packet counter = RXINT_THR. ● Clear the received packet counter.
Non-Zero	Non-Zero	● The receive interrupt will be issued if the following conditions: - The received packet counter = RXINT_THR. - The received packet counter = 1 and timer reaches the value of RXINT_CYC. ● Clear the received packet counter.

20.6.15 Ethernet Automatic Polling Timer Control register (ETHMAC_APTC)

Address Offset:0x38

This register allows ETHMAC to automatically poll the descriptors. This lowers the CPU utilization. If the transmit automatic poll function is enabled. ETHMAC will automatically poll the transmit descriptor each time the transmit automatic poll timer expires. If the function is disabled, software will write the Transmit Poll Demand register to trigger ETHMAC for reading the transmit descriptors after software has prepared the transmit packets in the transmit buffers.

If the receive automatic poll function is enabled, ETHMAC will automatically poll the receive descriptor each time the receive automatic poll timer expires. If the function is disabled, software will write the Receive Poll Demand register to trigger ETHMAC for reading the receive descriptors after software has released the receive descriptors to ETHMAC.

Bit	Field	Access	Initial	Description
31:13	–	–	0	Reserved
12	TXPOLL_TIME_SEL	RW	0	The period of TX poll time 0: 100 Mbps mode → 5.12us/10 Mbps mode → 51.2us 1: 100 Mbps mode → 81.92us/10 Mbps mode → 819.2us
11:8	TXPOLL_CNT	RW	0	This field defines the period of transmit automatic poll time. The unit is 1 TX poll time. When TXPOLL_CNT is not equal to 0, ETHMAC will automatically poll the transmit descriptor. If TXPOLL_CNT = 0, ETHMAC will not automatically poll the transmit descriptor
7:5	–	–	0	Reserved
4	RXPOLL_TIME_SEL	RW	0	The period of RX poll time 0: 100 Mbps mode → 5.12 us/10 Mbps mode → 51.2 us 1: 100 Mbps mode → 81.92 us/10 Mbps mode → 819.2 us
3:0	RXPOLL_CNT	RW	0	This field defines the period of receive automatic poll time. The unit is 1 RX poll time. When RXPOLL_CNT is not equal to 0, ETHMAC will automatically poll the receive descriptor. If RXPOLL_CNT = 0, ETHMAC will not automatically poll the receive descriptor.

20.6.16 Ethernet DMA Burst Length and Arbitration Control register (ETHMAC_DBLAC)

Address Offset:0x3C

Bit	Field	Access	Initial	Description
31:24	–	–	0	Reserved
23	IFG_INC	RW	0	The increase or decrease of IFG (Inter-Frame Gap) in Ethernet 0: Decrease 1: Increase
22:20	IFG_CNT	RW	0	The increased or decreased number of IFG (Inter-Frame Gap) count in Ethernet. The unit is 1 transmit clock in Ethernet (40 ns in 100Mbps mode, and 400 ns in 10Mbps mode)
19:12	–	–	0	Reserved
11:10	TXBST_SIZE	RW	0x3	TXDMA maximum burst size per TXDMA burst 0: 64 Bytes 1: 128 Bytes 2: 256 Bytes 3: 512 Bytes
9:8	RXBST_SIZE	RW	0x3	RXDMA maximum burst size per RXDMA burst 0: 64 Bytes 1: 128 Bytes 2: 256 Bytes 3: 512 Bytes
7	–	–	0	Reserved
6	RX_THR_EN	RW	0	RX FIFO threshold arbitration enable bit 0: Disable 1: Enable
5:3	RXFIFO_HTHR	RW	0	RX FIFO high threshold value for arbitration. When the used space in RX FIFO is larger than or equal to the RX FIFO high threshold value, RXDMA will be higher priority over TXDMA by using the DMA channel. RXDMA keeps the higher priority until the used space in RX FIFO is less than or equal to the RX FIFO low threshold value. Then, TXDMA gets higher priority over RXDMA. Consequently, software must set RXFIFO_HTHR to be larger than RXFIFO_LTHR to keep ETHMAC work correctly 0: 0/8 space of RX FIFO 1: 1/8 space of RX FIFO 2: 2/8 space of RX FIFO 3: 3/8 space of RX FIFO 4: 4/8 space of RX FIFO 5: 5/8 space of RX FIFO 6: 6/8 space of RX FIFO 7: 7/8 space of RX FIFO
2:0	RXFIFO_LTHR	RW	0	RX FIFO low threshold value for arbitration. When the used space in RX FIFO is less than or equal to the RX FIFO low threshold value, TXDMA will have higher priority over RXDMA by using the DMA channel 0: 0/8 space of RX FIFO 1: 1/8 space of RX FIFO 2: 2/8 space of RX FIFO 3: 3/8 space of RX FIFO 4: 4/8 space of RX FIFO 5: 5/8 space of RX FIFO 6: 6/8 space of RX FIFO 7: 7/8 space of RX FIFO

20.6.17 Ethernet DMA FIFO Status register (ETHMAC_DMAFIFOS)

Address Offset:0x40

Bit	Field	Access	Initial	Description
31	TXD_REQ	R	0	TXDMA request status 0: No TX Request 1: TX Request
30	RXD_REQ	R	0	RXDMA request status 0: No RX Request 1: RX Request
29	DARB_TXGNT	R	0	TXDMA grant status
28	DARB_RXGNT	R	0	RXDMA grant status
27	TXFIFO_EMPTY	R	0	TX FIFO empty status 0: TX FIFO is not empty 1: TX FIFO is empty
26	RXFIFO_EMPTY	R	0	RX FIFO empty status 0: RX FIFO is not empty 1: RX FIFO is empty
25:22	–	–	0	Reserved
21:18	TXDMA3_SM	R	0	TXDMA 3 state machine Don't care.
17:16	TXDMA2_SM	R	0	TXDMA 2 state machine Don't care.
15:12	TXDMA1_SM	R	0	TXDMA 1 state machine Don't care.
11	–	–	0	Reserved
10:8	RXDMA3_SM	R	0	RXDMA 3 state machine Don't care.
7:4	RXDMA2_SM	R	0	RXDMA 2 state machine Don't care.
3:0	RXDMA1_SM	R	0	RXDMA 1 state machine. Don't care.

20.6.18 Ethernet Transmit Priority Arbitration and FIFO Control register (ETHMAC_TPAFC)

Address Offset:0x48

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:4	HPKT_THR	RW	0xF	High Priority Transmit Packet Threshold The hardware behavior is the same whether writing 0 or 1 to this field
3:0	NPKT_THR	RW	0x1	Normal Priority Transmit Packet Threshold The hardware behavior is the same whether writing 0 or 1 to this field

20.6.19 Ethernet Receive Buffer Size register (ETHMAC_RBSZ)

Address Offset:0x4C

Bit	Field	Access	Initial	Description
31:0	RBSR	RW	0x640	Receive Buffer size

20.6.20 Ethernet MAC Control register (ETHMAC_MACCR)

Address Offset:0x50

Bit	Field	Access	Initial	Description
31	SW_RST	RW	0	Software reset 0: None

Bit	Field	Access	Initial	Description
				1: Enables (at last 560 system bus clocks), HW clears when SW reset done
30:27	–	–	0	Reserved
26	FULLDUP	RW	0	Full duplex mode selection bit 0: half-duplex mode 1: full-duplex mode
25:24	MAC_SPEED	RW	0	MAC operation mode 0: 10 Mbps 1: 100 Mbps Other: Reserved
23	–	–	0	Reserved
22	HPTXR_EN	RW	0	High priority transmit ring enable bit 0: Disable. Software will not use the high-priority transmit ring 1: Enable. Software will use the high-priority transmit ring
21	LOOP_EN	RW	0	Internal loopback enable bit. This field cannot be software reset. 0: Disable 1: Enable
20	PTP_EN	RW	0	IEEE 1588 PTP TX/RX frame detection enable bit 0: Disable 1: Enable
19	–	–	0	Reserved
18	REMOVE_VLAN	RW	0	Remove the VLAN tag from the packets received with the VLAN tag
17	CRC_APD	RW	0	Append CRC to transmitted packets
16	DISCARD_CRCERR	RW	0	Discard the CRC error packet if there is CRC error status in the transmit packet
15	–	–	0	Reserved
14	ENRX_IN_HALFTX	RW	0	Enable bit for packet reception when transmitting packets in the half duplex mode 0: Disable 1: Enable
13	JUMBO_LF	RW	0	Jumbo Long Frame status 0: Packets with length of more than 1518 (1522 for packets with VLAN tag) are treated as long frames 1: Packets with length of more than 9216 (9220 for packets with VLAN tag) are treated as long frames
12	RX_RUNT	RW	0	Receive the incoming packet even if its length is less than 64 bytes. The incoming packet length must be longer than or equal to 10 bytes
11	RX_BROADPKT_EN	RW	0	Enable bit for receiving broadcast packets 0: Disable 1: Enable
10	RX_MULTIPKT_EN	RW	0	Enable bit for receiving all multicast packets 0: Disable 1: Enable
9	RX_HT_EN	RW	0	Enable bit for storing incoming packet if the packet passes hash table address filtering and is a multicast packet 0: Disable 1: Enable
8	RX_ALLADR	RW	0	Destination address of the incoming packet checking status 0: Not checked 1: Checked
7	DIS_IPV6_PKT_REC	RW	0	IPV6 packet reorganization at RXMAC disable bit 0: Enable 1: Disable
6:4	–	–	0	Reserved
3	RXMAC_EN	RW	0	RXMAC enable bit 0: Disable 1: Enable
2	TXMAC_EN	RW	0	TXMAC enable bit 0: Disable 1: Enable
1	RXDMA_EN	RW	0	Receive DMA channel enable bit 0: Disable 1: Enable

Bit	Field	Access	Initial	Description
0	TXDMA_EN	RW	0	Transmit DMA channel enable bit 0: Disable 1: Enable

20.6.21 Ethernet MAC Status register (ETHMAC_MACST)

Address Offset:0x54

Bit	Field	Access	Initial	Description
31:13	–	–	0	Reserved
12	AFIFO_UNDERUN	RW1C	0	TX Asynchronous FIFO underrun status When the system frequency does not meet the clock limitation, then TX Asynchronous FIFO will cause the underrun condition. 0: No effect 1: TX Asynchronous FIFO is underrun
11	COL_EXCEED	RW1C	0	Collision exceeded status 0: No effect 1: Collision exceeded amount exceeds 16
10	LATE_COL	RW1C	0	Late collision status 0: No effect 1: Transmitter detects late collision
9	XPKT_LOST	RW1C	0	Packets lost status 0: No effect 1: Packets transmitted to Ethernet are lost due to late collision or excessive collision
8	XPKT_OK	RW1C	0	Packets transmitted status 0: No effect 1: Packets transmitted to Ethernet successfully
7	AROUND	RW1C	0	Runt packet received status 0: No effect 1: Receive a runt packet
6	FTL	RW1C	0	Too long frame detected status 0: No effect 1: Receive a frame that is too long
5	CRC_ERR	RW1C	0	CRC error status 0: No effect 1: CRC error. The CRC check result of the incoming packet is invalid.
4	RPKT_LOST	RW1C	0	Received packets are lost due to RX FIFO full 0: No effect 1: Lost
3	RPKT_SAVE	RW1C	0	Packets received to RX FIFO successfully 0: No effect 1: Success
2	COL	RW1C	0	Incoming packet dropped due to collision 0: No effect 1: Dropped
1	BROADCAST	RW1C	0	Incoming packet for broadcast address 0: No effect 1: Incoming
0	MULTICAST	RW1C	0	Incoming packet for multicast address status 0: No effect 1: Incoming

20.6.22 Ethernet MAC Test Mode register (ETHMAC_TM)

Address Offset:0x58

Bit	Field	Access	Initial	Description
31:21	–	–	0	Reserved
20	PTIMER_TEST	RW	0	Automatic polling timer test mode 0: Disable Automatic polling timer test mode 1: TX and RX POLL TIME unit (offset 0x38) will increase the step.

Bit	Field	Access	Initial	Description
				100Mbps=1.3107ms/10Mbps=13.107ms
19	ITIMER_TEST	RW	0	Interrupt timer test mode 0: Disable Interrupt timer test mode 1: TX and RX INT TIME unit (offset 0x30 and 0x34) will increase the step. 100Mbps=1.3107ms/10Mbps=13.107ms
18:16	–	–	0	Reserved
15	TEST_COL	RW	0	When set to 1, the collision upper limit can be specified by TEST_EXSTHR. Besides, the back-off value can be specified by TEST_BKOFF. 0: Disable Transmit collision test mode 1: Enable Transmit collision test mode
14:5	TEST_BKOFF	RW	0	Back-off value in the transmission collision test mode. When TEST_COL is set to 1, then the value must not set to 0.
4:0	TEST_EXSTHR	RW	0	Retry upper limit in the transmit collision test mode. When TEST_COL is set to 1, then the value must not set to 0.

20.6.23 Ethernet MAC PHY Control Mode register (ETHMAC_PHYCTRL)

Address Offset:0x60

Bit	Field	Access	Initial	Description
31:28	–	–	0	Reserved
27	PHYWR	RW	0	Setting this bit to 1 initializes a write sequence to PHY. 0: Automatically cleared after the write operation is finished 1: Initializes a write sequence to PHY
26	PHYRD	RW	0	Setting this bit to 1 initializes a read sequence to PHY. 0: Automatically cleared after the read operation is finished 1: Initializes a read sequence to PHY
25:21	REG_DEV_AD	RW	0	Register address for 802.3 Clause 22/Device address for 802.3 Clause 45
20:16	PHY_PRT_AD	RW	0	PHY address for 802.3 Clause 22/Port address for 802.3 Clause 45
15:14	OP	RW	0	Operation code
13:12	ST	RW	0	Start of frame
11:8	–	–	0	Reserved
7:0	MDC_CYCTHR	RW	0x34	MDC cycle threshold. The MDC period = MDC_CYCTHR x system clock period (HCLK)

20.6.24 Ethernet MAC PHY Data register (ETHMAC_PHYDATA)

Address Offset:0x64

Bit	Field	Access	Initial	Description
31:16	RDATA	R	0	Read data from PHY
15:0	WDATA	RW	0	Write data to PHY

20.6.25 Ethernet MAC Flow Control register (ETHMAC_FCTRL)

Address Offset:0x68

Bit	Field	Access	Initial	Description
31:16	PAUSE_TIME	RW	0	Pause time in the pause frame. The unit is 1 slot time.
15:9	FC_HIGH_FC_LOW	RW	0x02	RX FIFO free space high threshold/RX FIFO free space low threshold.
8	FC_HTHR_SEL	RW	0	RX FIFO free space high threshold select 0: the RX FIFO free space low threshold is selected 1: RX FIFO free space high threshold is selected
7:5	–	–	0	Reserved
4	RX_PAUSE	RW1C	0	Receive pause frame 0: No effect 1: Clear RX_PAUSE
3	TX_PAUSED	R	0	Packet transmission paused due to the receive pause frame.
2	FCTHR_EN	RW	0	Flow control threshold mode enable bit 0: Disable

Bit	Field	Access	Initial	Description
				1: Enable
1	TX_PAUSE	RW	0	Transmit pause frame 0: Auto-cleared after the pause frame has been transmitted 1: Send the pause frames
0	FC_EN	RW	0	Flow control mode enable bit 0: Disable 1: Enable

20.6.26 Ethernet MAC Back Pressure register (ETHMAC_BP)

Address Offset:0x6C

Bit	Field	Access	Initial	Description
31:15	–	–	0	Reserved
14:8	BK_LOW	RW	0x2	RX FIFO free space low threshold. MAC generates a jam pattern if RX FIFO free space is less than the low threshold when packets are incoming. (The unit is 256 bytes, and the default value is 2)
7:4	BKJAM_LEN	RW	0	Back pressure jam length 0: 4 bytes 1: 8 bytes 2: 16 bytes 3: 32 bytes 4: 64 bytes 5: 128 bytes 6: 256 bytes 7: 512 bytes 8: 1024 bytes 9: 1518 bytes 10: 2048 bytes Other: Reserved
3:2	–	–	0	Reserved
1	BKADR_MODE	RW	0	Back pressure address mode 0: Generate jam pattern when any packet is incoming. 1: Generate jam pattern when packet address matches.
0	BK_EN	RW	0	Back pressure mode enable bit 0: Disable 1: Enable

20.6.27 Ethernet MAC Wake-On-LAN Control register (ETHMAC_WOLCTRL)

Address Offset:0x70

Bit	Field	Access	Initial	Description
31:26	–	–	0	Reserved
25:24	WOL_TYPE	RW	0	WOL output signal type 0: Active high 1: Active low 2: Positive pulse 3: Negative pulse
23:19	–	–	0	Reserved
18	SW_PDNPHY	RW	0	Software power down PHY. It will only inform PHY into power down and will not affect ETHMAC. 0: No effect 1: Power down PHY
17:16	WAKEUP_SEL	RW	0	Wake-up frame selection bits 0: Wake-up frame 1 1: Wake-up frame 2 2: Wake-up frame 3 3: Wake-up frame 4
15:7	–	–	0	Reserved
6	WAKEUP4_EN	RW	0	Wake-up frame 4 event enable bit 0: Disable

Bit	Field	Access	Initial	Description
				1: Enable
5	WAKEUP3_EN	RW	0	Wake-up frame 3 event enable bit 0: Disable 1: Enable
4	WAKEUP2_EN	RW	0	Wake-up frame 2 event enable bit 0: Disable 1: Enable
3	WAKEUP1_EN	RW	0	Wake-up frame 1 event enable bit 0: Disable 1: Enable
2	MAGICPKT_EN	RW	0	Magic packet event enable bit 0: Disable 1: Enable
1	LINKCHG1_EN	RW	0	Link change to 1 event enable bit 0: Disable 1: Enable
0	LINKCHG0_EN	RW	0	Link change to 0 event enable bit 0: Disable 1: Enable

20.6.28 Ethernet MAC Wake-On-LAN Status register (ETHMAC_WOLST)

Address Offset:0x74

Bit	Field	Access	Initial	Description
31:7	–	–	0	Reserved
6	WAKEUP4_STS	RW1C	0	Wake-up frame 4 event status 0: No effect 1: Wake-up frame 4 event happens
5	WAKEUP3_STS	RW1C	0	Wake-up frame 3 event status 0: No effect 1: Wake-up frame 3 event happens
4	WAKEUP2_STS	RW1C	0	Wake-up frame 2 event status 0: No effect 1: Wake-up frame 2 event happens
3	WAKEUP1_STS	RW1C	0	Wake-up frame 1 event status 0: No effect 1: Wake-up frame 1 event happens
2	MAGICPKT_STS	RW1C	0	Magic packet event status 0: No effect 1: Magic packet event happens
1	LINKCHG1_STS	RW1C	0	Link change to 1 event status 0: No effect 1: Link change to 1 event happens
0	LINKCHG0_STS	RW1C	0	Link change to 0 event status 0: No effect 1: Link change to 0 event happens

20.6.29 Ethernet MAC Wake-up Frame CRC register (ETHMAC_WFCRC)

Address Offset:0x78

Bit	Field	Access	Initial	Description
31:0	WFCRC	RW	0	Wake-up Frame CRC value

20.6.30 Ethernet MAC Wake-up Frame Byte Mask 1st Double-word register (ETHMAC_WFBM1)

Address Offset:0x80

Bit	Field	Access	Initial	Description
31:0	WFBM	RW	0	Wake-up Frame Byte Mask 1st Double-word value

20.6.31 Ethernet MAC Wake-up Frame Byte Mask 2nd Double-word register (ETHMAC_WFBM2)

Address Offset:0x84

Bit	Field	Access	Initial	Description
31:0	WFBM	RW	0	Wake-up Frame Byte Mask 2nd Double-word value

20.6.32 Ethernet MAC Wake-up Frame Byte Mask 3rd Double-word register (ETHMAC_WFBM3)

Address Offset:0x88

Bit	Field	Access	Initial	Description
31:0	WFBM	RW	0	Wake-up Frame Byte Mask 3rd Double-word value

20.6.33 Ethernet MAC Wake-up Frame Byte Mask 4th Double-word register (ETHMAC_WFBM4)

Address Offset:0x8C

Bit	Field	Access	Initial	Description
31:0	WFBM	RW	0	Wake-up Frame Byte Mask 4th Double-word value

20.6.34 Ethernet MAC Normal Priority Transmit Ring Pointer register (ETHMAC_NPTXR_PTR)

Address Offset:0x90

Bit	Field	Access	Initial	Description
31:0	PTR	R	0	Normal Priority Transmit Ring Pointer value When all descriptors of NPTXR are invalid (All TXDMA OWN bits are zeros), then NPTXR_PTR can decrease one TX descriptor size.

20.6.35 Ethernet MAC High Priority Transmit Ring Pointer register (ETHMAC_HPTXR_PTR)

Address Offset:0x94

Bit	Field	Access	Initial	Description
31:0	PTR	R	0	High Priority Transmit Ring Pointer value When all descriptors of HPTXR are invalid (All TXDMA OWN bits are zeros), then HPTXR_PTR can decrease one TX descriptor size.

20.6.36 Ethernet MAC Receiver Ring Pointer register (ETHMAC_RXR_PTR)

Address Offset:0x98

Bit	Field	Access	Initial	Description
31:0	PTR	R	0	Receive Ring Pointer value When all descriptors of RXR are invalid (All RXPKT RDY bit are ones), then RXR_PTR can decrease one RX descriptor size.

20.6.37 Ethernet MAC Transmitted Packet Counter 1 register (ETHMAC_TX_CNT1)

Address Offset:0xA0

Bit	Field	Access	Initial	Description
31:0	CNT	R	0	Counter for counting packets transmitted successfully

20.6.38 Ethernet MAC Transmitted Packet Counter 2 register (ETHMAC_TX_CNT2)

Address Offset:0xA4

Bit	Field	Access	Initial	Description
31:16	TXMCOL_CNT	R	0	Counter for counting packets transmitted OK with 2 ~ 15 collisions
15:0	TXSCOL_CNT	R	0	Counter for counting packets transmitted OK with single collision

20.6.39 Ethernet MAC Transmitted Packet Counter 3 register (ETHMAC_TX_CNT3)

Address Offset:0xA8

Bit	Field	Access	Initial	Description
31:16	TXECOL_CNT	R	0	Counter for counting packets failed in transmission (Due to the late collision or collision count ≧16 or transmit underrun)
15:0	TXFAIL_CNT	R	0	Counter for counting packets failed in transmission (Due to the collision count ≧16)

20.6.40 Ethernet MAC Transmitted Packet Counter 4 register (ETHMAC_TX_CNT4)

Address Offset:0xAC

Bit	Field	Access	Initial	Description
31:16	TXUNDERUN_CNT	R	0	Counter for counting packets failed in transmission (Due to transmit under-run)
15:0	TXLCOL_CNT	R	0	Counter for counting the packets failed in transmission (Due to late collision)

20.6.41 Ethernet MAC Received Packet Counter 1 register (ETHMAC_RX_CNT1)

Address Offset:0xB0

Bit	Field	Access	Initial	Description
31:0	CNT	R	0	Counter for counting the packets received successfully

20.6.42 Ethernet MAC Received Packet Counter 2 register (ETHMAC_RX_CNT2)

Address Offset:0xB4

Bit	Field	Access	Initial	Description
31:0	BROPKT_CNT	R	0	Counter for counting the received broadcast packets

20.6.43 Ethernet MAC Received Packet Counter 3 register (ETHMAC_RX_CNT3)

Address Offset:0xB8

Bit	Field	Access	Initial	Description
31:0	MULPKT_CNT	R	0	Counter for counting the received multicast packets

20.6.44 Ethernet MAC Received Packet Counter 4 register (ETHMAC_RX_CNT4)

Address Offset:0xBC

Bit	Field	Access	Initial	Description
31:16	RPF_CNT	R	0	Receive pause frame counter
15:0	AEP_CNT	R	0	Counter for counting the packets with alignment error. The counter is to count packets with CRC error and no octet-boundary discarded by ETHMAC.

20.6.45 Ethernet MAC Received Packet Counter 5 register (ETHMAC_RX_CNT5)

Address Offset:0xC0

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	RUNT_CNT	R	0	Counter for counting the received runt packets

20.6.46 Ethernet MAC Received Packet Counter 6 register (ETHMAC_RX_CNT6)

Address Offset:0xC4

Bit	Field	Access	Initial	Description
31:16	CRCER_CNT	R	0	Received CRC error packet counter The counter counts the number of the following received packets: 1. The octet-boundary frames discarded due to the CRC error. 2. The frames truncated due to RX FIFO full and it leads to CRC error.
15:0	FTL_CNT	R	0	Counter for counting received FTL packets

20.6.47 Ethernet MAC Received Packet Counter 7 register (ETHMAC_RX_CNT7)

Address Offset:0xC8

Bit	Field	Access	Initial	Description
31:16	RLOST_CNT	R	0	Counter for counting the loss of the received packets. (Due to RX FIFO full)
15:0	RCOL_CNT	R	0	Receive collision counter

20.6.48 Ethernet MAC BIST Pattern Control register (ETHMAC_BIST)

Address Offset:0xCC

The BIST function can send one specific broadcast pattern from TX and compare the pattern from RX without DMA operation. When the BIST function is enabled, ETHMAC will force to the full-duplex mode.

The BIST function running step:
- Step 1. Users must set to the correct MAC_SPEED (bit [25:24], offset 0x50) to synchronize PHY and disable TX/RX DMA_EN and MAC_EN (bit [3:0], offset 0x50).
- Step 2. Users set the BIST pattern control register for the test purpose.

Bit	Field	Access	Initial	Description
31:10	–	–	0	Reserved
9	RX_CRC_FAIL	R	0	RX CRC compared status 0: None 1: Fail
8	RX_COMPARE	RW	0	BIST pattern comparison from RX side 0: Users must wait the bit to clear to 0, and then users can update the bit again. 1: TX and RX sides must have the loopback path either the internal loopback or external loopback. The internal loopback is LOOP_EN (bit [21], offset 0x50); and the external loopback is PHY loopback mode.
7:1	–	–	0	Reserved
0	TX_SEND	RW	0	BIST pattern sending enable 0: Users must wait the bit to clear to 0, and then users can update the bit again. 1: Sending the BIST pattern

20.6.49 Ethernet MAC Broadcast and Multicast Receiving Control register (ETHMAC_BMRCTRL)

Address Offset:0xD0

The register is used to control the broadcast and the multicast packet receiving function.

Bit	Field	Access	Initial	Description
31:25	–	–	0	Reserved
24	TIME_STEP	RW	0	Timer step setting 0: 10M:419.43ms/100M:41.94ms 1: 10M:409.6us/100M:40.96us
23:16	TIME_THR_NUM	RW	0	Timer threshold number If PKT_THR_VAL is not zero, then ETHMAC will monitor the increasing number per TIME_THR_NUM; if the increasing number is larger than PKT_THR_VAL, then ETHMAC automatically disables the broadcast and multicast packet receiving. The unit is the time step and depends on the speed mode.
15:5	–	–	0	Reserved
4:0	PKT_THR_VAL	RW	0	Multicast and broadcast packets threshold value When the value is zero, then it will not affect the broadcast and multicast packet receiving function. The unit is 256 packets.

20.6.50 Ethernet RGMII In-band Status register (ETHMAC_ RGMII_IBS)

Address Offset:0xD4

Bit	Field	Access	Initial	Description
31:4	–	–	0	Reserved
3	DUPLEX_MODE	R	0	PHY duplex status 0: Half-duplex mode 1: Full duplex mode

Bit	Field	Access	Initial	Description
2:1	RXC_CLK	R	0	PHY clock speed 0: 2.5MHz 1: 25MHz Other: Reserved
0	LINK_STS	R	0	Indicates link status When link status is down, PHY speed and duplex are defined by the PHY's internal setting. 0: Down 1: Up

20.6.51 Ethernet MAC Received Packet Counter 8 register (ETHMAC_RX_CNT8)

Address Offset:0xD8

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	RCEE_CNT	R	0	Receive carrier extension error counter. It is valid half duplex.

20.6.52 Ethernet MAC Received Packet Counter 9 register (ETHMAC_RX_CNT9)

Address Offset:0xDC

Bit	Field	Access	Initial	Description
31:0	RPKTB_CNT	R	0	Counter for RXDMA has received packets into RX BUF successfully.

20.6.53 Ethernet MAC Error Response Control register (ETHMAC_ERCTRL)

Address Offset:0xE0

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1	RONLY_ERR_EN	RW	0	Read-only Error enable bit 0: Disable 1: Enable. When users write the read-only register, then ETHMAC will respond ERROR.
0	DEC_ERR_EN	RW	0	Decode Error enable bit When user write does not exist in the register, then ETHMAC will respond ERROR if the bit set to 1. When user read does not exist in the register, then ETHMAC responds OKAY no matter the value is 0 or 1. 0: Disable 1: Enable

20.6.54 Ethernet MAC Interface Selection register (ETHMAC_GIS)

Address Offset:0xE8

Bit	Field	Access	Initial	Description
31:2	–	–	0	Reserved
1:0	IF	RW	1	MAC interface selection 0: MII Interface 1: RMII Interface Other: Reserved

20.6.55 Ethernet MAC SW Reset Cycle Count register (ETHMAC_SWRCC)

Address Offset:0xEC

Bit	Field	Access	Initial	Description
31:16	–	–	0	Reserved
15:0	SW_CYC	RW	0xF0	SW reset cycle count.(The value must be non-zero.) Soft Reset Cycle counter at 10Mbps The Cycle Counter must be meet the Ethernet 10M mode (2.5MHz) 10M period is 400ns , syn. circuit need to 2 cycles So Reset cycle must large than 400ns x 2 / system period

20.6.56 Ethernet MAC EEE Control register (ETHMAC_EEECTRL)

Address Offset:0xF0

Bit	Field	Access	Initial	Description
31:20	TX_ENTER_CNT	RW	0xFFF	TX EEE enter counter. The time unit is 40 ns at 100M.
19:17	–	–	0	Reserved
16	SEND_TX_LPI	RW	0	Send TX LPI pattern 0: ETHMAC do not issue TX LPI pattern 1: When TX FIFO is empty, it will wait TX_ENTER_CNT period and then issue TX LPI pattern. When TX packet is ready, it will stop TX LPI pattern and wait TX_WKP_CNT period, and then issue TX packet.
15:0	TX_WKP_CNT	RW	0xFFFF	TX EEE wake up counter. The time unit is 40 ns at 100M

20.6.57 Ethernet MAC PTP RX Unicast IP Destination Address register (ETHMAC_PTP_RXUIPDA)

Address Offset:0x100

Bit	Field	Access	Initial	Description
31:0	RDA	RW	0	Unicast IP destination address used for detection of PTP frames on the receive path.

20.6.58 Ethernet MAC PTP TX Unicast IP Destination Address register (ETHMAC_PTP_TXUIPDA)

Address Offset:0x104

Bit	Field	Access	Initial	Description
31:0	TDA	RW	0	Unicast IP destination address used for detection of PTP frames on the transmit path.

20.6.59 Ethernet MAC PTP TX Event Frame register (ETHMAC_PTP_TX)

Address Offset:0x110

ETHMAC detects the transmitted PTP frame and records the timestamp at the PTP TX Event Frame register. The following PTP frames will be detected:

- Sync Frame
- Delay_Req Frame

Bit	Field	Access	Initial	Description
31:0	PSEC	R	0	PTP TX frame timestamp for seconds

20.6.60 Ethernet MAC PTP TX Event Frame 2 register (ETHMAC_PTP_TX2)

Address Offset:0x114

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:0	PNSEC	R	0	PTP TX frame timestamp for nanoseconds

20.6.61 Ethernet PTP RX Event Frame register (ETHMAC_PTP_RX)

Address Offset:0x118

ETHMAC detects the received PTP frame and records the timestamp at the PTP RX Event Frame register.

The following PTP frames will be detected:
- Sync Frame
- Delay_Req Frame

Bit	Field	Access	Initial	Description
31:0	PSEC	R	0	PTP RX frame timestamp for seconds.

20.6.62 Ethernet PTP RX Event Frame register 2 (ETHMAC_PTP_RX2)

Address Offset:0x11C

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:0	PNSEC	R	0	PTP RX frame timestamp for nanoseconds.

20.6.63 Ethernet MAC PTP TX Peer Frame register (ETHMAC_PTP_TXP)

Address Offset:0x120

ETHMAC detects the transmitted PTP peer frame and record the timestamp at the PTP TX Peer Frame register.

The following PTP peer frames will be detected:
- Pdelay_Req Frame
- Pdelay_Resp Frame

Bit	Field	Access	Initial	Description
31:0	PSEC	R	0	PTP TX peer frame timestamp for seconds.

20.6.64 Ethernet MAC PTP TX Peer Frame 2 register (ETHMAC_PTP_TXP2)

Address Offset:0x124

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:0	PNSEC	R	0	PTP TX peer frame timestamp for nanoseconds.

20.6.65 Ethernet MAC PTP RX Peer Frame register (ETHMAC_PTP_RXP)

Address Offset:0x128

ETHMAC detects the received PTP Peer frame and records the timestamp at the PTP RX Peer Frame register.

The following PTP peer frames will be detected:
- Pdelay_Req Frame
- Pdelay_Resp Frame

Bit	Field	Access	Initial	Description
31:0	PSEC	R	0	PTP RX peer frame timestamp for seconds.

20.6.66 Ethernet MAC PTP RX Peer Frame 2 register (ETHMAC_PTP_RXP2)

Address Offset:0x12C

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:0	PNSEC	R	0	PTP RX peer frame timestamp for nanoseconds.

20.6.67 Ethernet MAC PTP Timer register (ETHMAC_PTP_TMR)

Address Offset:0x130

Bit	Field	Access	Initial	Description
31:0	NNSEC	RW	0	Nano-Nanosecond timer. Unit is 2-9 ns. When counting to 999999999 and then it returns to 0.

20.6.68 Ethernet MAC PTP Timer 2 register (ETHMAC_PTP_TMR2)

Address Offset:0x134

Bit	Field	Access	Initial	Description
31:30	–	–	0	Reserved
29:0	NSEC	RW	0	Nanosecond timer. When counting to 999999999 and then it returns to 0.

20.6.69 Ethernet MAC PTP Timer 3 register (ETHMAC_PTP_TMR3)

Address Offset:0x138

Note: When updating 0x130, 0x134, and 0x138, the register 0x138 must be last one

Bit	Field	Access	Initial	Description
31:0	SEC	RW	0	Second timer. Once the register is written, then update PTP_TMR to the PTP timer.

20.6.70 Ethernet MAC PTP Period Increment 0 register (ETHMAC_PTP_PER0)

Address Offset:0x13C

Bit	Field	Access	Initial	Description
31:8	–	–	0	Reserved
7:0	NS_PER	RW	0x06	PTP Clock Period. The unit is nanosecond. (Max. is 255 ns.)

20.6.71 Ethernet MAC PTP Period Increment 1 register (ETHMAC_PTP_PER1)

Address Offset:0x140

Bit	Field	Access	Initial	Description
31:0	NNS_PER	RW	0x00	PTP Clock Period. The unit is 10-9 ns.(Max. is 0.999999999 ns.)

20.6.72 Ethernet MAC PTP Period Offset Increment register (ETHMAC_PTP_OFF)

Address Offset:0x144

Bit	Field	Access	Initial	Description
31:16	INC_CYC	RW	0x95	Cycle number. Set to 0 then disable the function.
15:0	NS_OFF	RW	0x6A	Offset. The unit is nanosecond.

20.6.73 Ethernet MAC Timer Adjustment register (ETHMAC_PTP_ADJ)

Address Offset:0x148

Bit	Field	Access	Initial	Description
31	INC	RW	0	Subtract or Add 0: Subtract 1: Add
30	–	–	0	Reserved
29:0	ADJ_VAL	RW	0	30-bit adjust value(1 unit/ns) 0: Disable

21 FLASH

21.1 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 **0**.
- 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.

21.2 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 and information block.

User ROM	504K Bytes divided into 1008 pages of 512 Bytes
Boot ROM	8K Bytes divided into 16 pages of 512 Bytes
Information Block	2K Bytes divided into 4 pages of 512 Bytes

Flash access times can be configured through a register in the Flash memory controller.

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.

21.3 FEATURES

- Read interface (32-bit)
- Flash Program / Erase operation
- Read Protect for Code Security (CS)
- Boot Byte for remapping selection
- Flash Access Frequency Modify
- Checksum

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 protected against by RDP bit control.

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.

The Flash memory can be programmed and erased using ICP and ISP.

21.4 ORGANIZATION

Memory	Block	Name	Physical Read Address	FMC Program/Erase Address	Size (Byte)
Main Memory (MM)	Boot ROM	Page 0	0x00200000 ~ 0x002001FF	0x00000000 ~ 0x000001FF	512
		Page 1	0x00200200 ~ 0x002003FF	0x00000200 ~ 0x000003FF	512
		. . .	. . .	. . .	. . .
		Page 15	0x00201E00 ~ 0x00201FFF	0x00001E00 ~ 0x00001FFF	512
	User ROM	Page 16	0x00202000 ~ 0x002021FF	0x00002000 ~ 0x000021FF	512
		Page 17	0x00202200 ~ 0x002023FF	0x00002200 ~ 0x000023FF	512
		. . .	. . .	. . .	. . .
		Page 1022	0x0027FC00 ~ 0x0027FDFF	0x0007FC00 ~ 0x0007FDFF	512
		Page 1023	0x0027FE00 ~ 0x0027FFFF	0x0007FE00 ~ 0x0007FFFF	512
Specific Memory (SM)	Information Block	Page 0	0x00400000 ~ 0x004001FF	SM: 0x00000000 ~ 0x000001FF	512
		Page 1	0x00400200 ~ 0x004003FF	SM: 0x00000200 ~ 0x000003FF	512
		Page 2	0x00400400 ~ 0x004005FF	SM: 0x00000400 ~ 0x000005FF	512
		Page 3	0x00400600 ~ 0x004007FF	SM: 0x00000600 ~ 0x000007FF	512

✸ ***Note:***
1. ***Specific Memory page 0~1 reserved. Do not modify to avoid mistakes.***
2. ***The shadow memory range is 0x00000000 to 0x001FFFFF. Shadow memory content is controlled by the REMAP bit.***

21.4.1 Memory Map of FMC and Flash

The first 2MB space (0x000000 ~ 0x1FFFFF) is the shadow memory for remapping function. REMAP 0/1 could be decided either statically by boot flag, which can be modified via message for boot flag write, or dynamically via Bit 24 and Bit 7 in SCU_PWRMODE register.

The next space (0x2000000 ~ 0x3FFFFF) is mapped to MM of Flash whose actual size is 512KB, followed by 2KB of SM (0x400000 ~ 0x4007FF). Starting from 0x600000, BUF of 512B (0x600000 ~0x6001FF) is available, as its name implies, for data exchange with Flash. FLASH (0x601000 ~ 0x6013FF) is a set of flash memory controller registers for communicating with Flash via message handshaking mechanism.

Start Address	End Address	Size	Target
0x00000000	0x001FFFFF	2MB	Shadow Memory
0x00200000	0x003FFFFF	2MB	Main Memory (MM)
0x00400000	0x004007FF	2KB	Specific Memory (SM)
0x00600000	0x006001FF	512B	Data Buffer (BUF)
0x00601000	0x006013FF	1KB	FMC register (FLASH)

21.4.2 SM Layout

The section illustrates what SM 4 pages are allocated for, as follows.

Byte	511	~	40	39	38	37	36	35	34	33	32	31	~	0
Page 0	Reserved. Do not modify to avoid mistakes.													
Page 1	Reserved. Do not modify to avoid mistakes.													
Page 2	Reserved			mBoot Byte	Boot Byte	mBoot Byte	Boot Byte	mRDP	RDP	mRDP	RDP	Reserved		
Page 3	Customer use													

The Pages 2 (not including reserved area) stores data in one byte error correction coding (ECC), which encodes each byte of data as a 32-bit word, where mData is ones' complement of Data, e.g., Data = 0x38 gives mData = 0xC7. By inverting Bytes 1 and 3 (mData) and majority rule, one byte data (Data) could be corrected if three of 4 bytes match, i.e., at least 3 bytes are the same for Data to be deemed valid.

32-bit			
Byte 3	Byte 2	Byte 1	Byte 0
mData	Data	mData	Data
mBoot Byte	Boot Byte	mBoot Byte	Boot Byte
mRDP	RDP	mRDP	RDP
-	-	-	-

Note that Page 2 must be modified via Messages for MM read protect and boot flag write. RDP and Boot Byte are encoded into one byte ECC automatically via Messages while customer use on Page 2 is stored directly without involvement of one byte ECC.

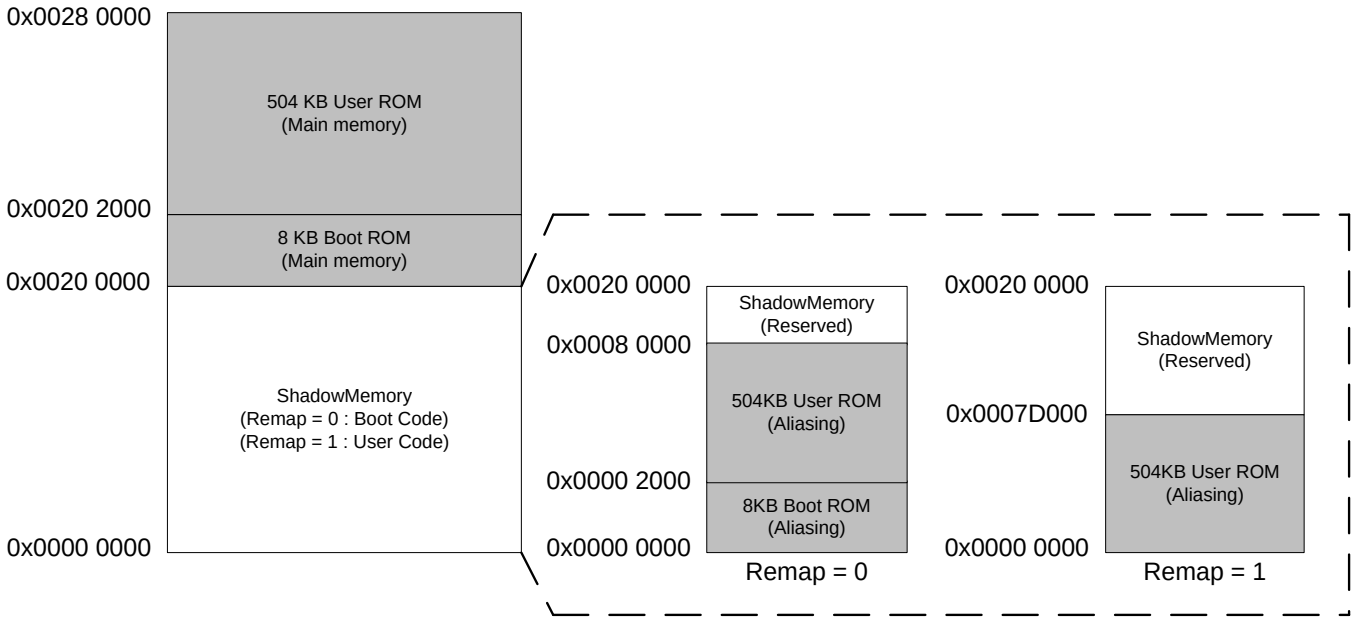
✸ ***Note: Specific Memory page 0~1 reserved. Do not modify to avoid mistakes.***

21.4.3 SHADOW MEMORY

The default value of REMAP bit is latched from Boot Byte.

- When Boot Byte = 0x00, REMAP bit = 0 after reset on.
- When Boot Byte = 0x5A, REMAP bit = 1 after reset on

The shadow memory for remapping function as below.



- The 0x00200000 ~0x0027FFFF is the physical read address of ROM.
- The Shadow memory content is controlled by the REMAP bit.
- The Shadow memory contains Boot ROM & User ROM When REMAP as 0. The Shadow memory only contains User ROM When REMAP as 1,.
- The Shadow memory provides the memory data that the PC reads from 0 after the CPU restarts.
- The shadow memory range can only be read, but cannot be written or erased.

21.5 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.

Data can be read directly from the physical read address, or from the shadow memory range. The function call must determine whether the address has shifted to avoid calling the wrong address.

21.6 PROGRAM/ERASE

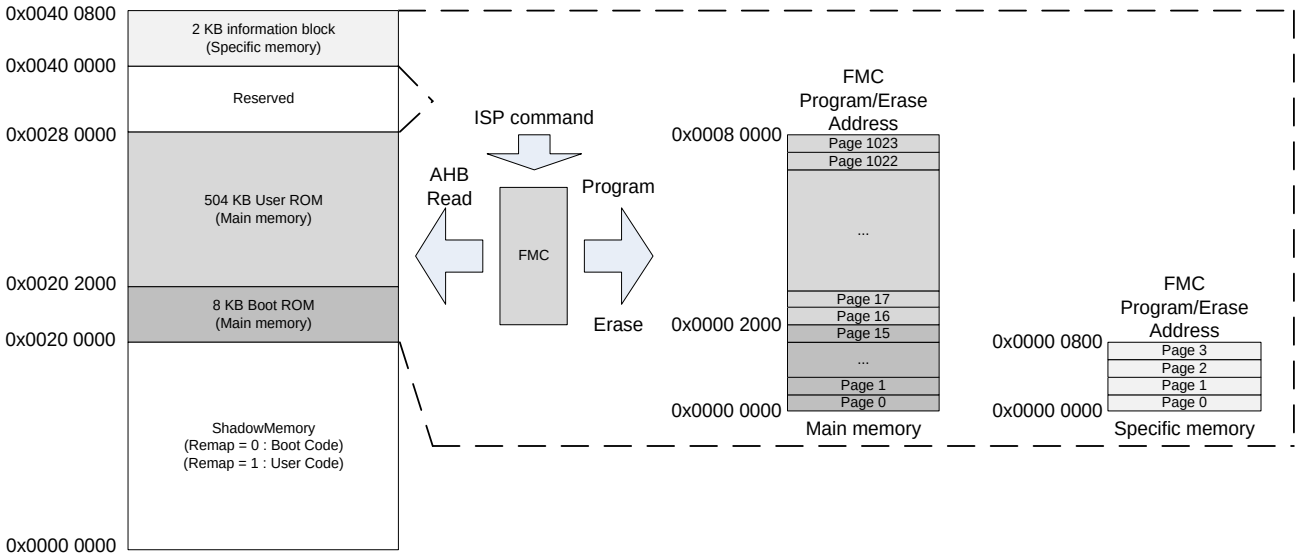
The Flash memory erase operation can be performed at page level or on the whole Flash area (mass erase).

To ensure that there is no over-programming, the Flash programming and erase controller blocks are clocked by a fixed clock.

Both program and erase operations are at the FMC program/erase address, because the FMC operation ignores the AHB read address. The FMC program/erase addresses are not affected by the REMAP bit. The address cannot exceed 0x001FFFFF when ISP is executed, otherwise an error will occur.

21.6.1 Physical Read Address (REAMP don't care)

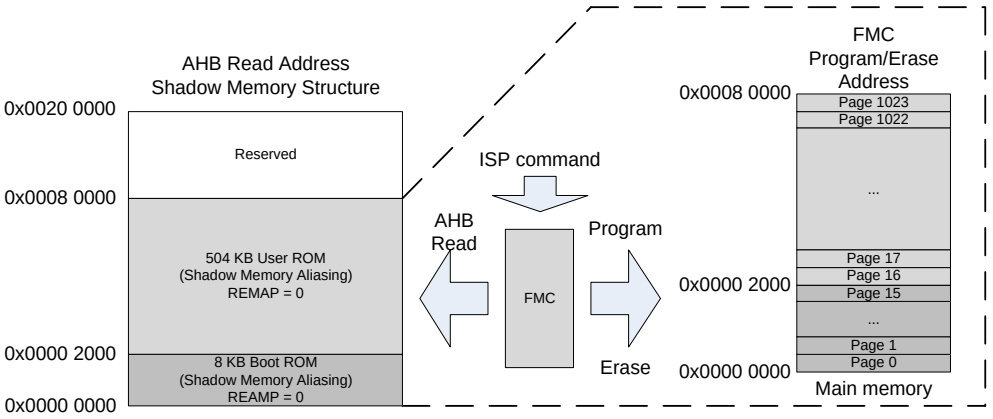
- Main Memory Physical Read Address = FMC Program/Erase Address + 0x00200000
- Specific Memory Physical Read Address = FMC Program/Erase Address + 0x00400000



Memory	Block	Name	Physical Read Address	FMC Program/Erase Address	Size (Byte)
Main Memory (MM)	Boot ROM	Page 0	0x00200000 ~ 0x002001FF	0x00000000 ~ 0x000001FF	512
		Page 1	0x00200200 ~ 0x002003FF	0x00000200 ~ 0x000003FF	512
		. . .	. . .	. . .	. . .
		Page 15	0x00201E00 ~ 0x00201FFF	0x00001E00 ~ 0x00001FFF	512
	User ROM	Page 16	0x00202000 ~ 0x002021FF	0x00002000 ~ 0x000021FF	512
		Page 17	0x00202200 ~ 0x002023FF	0x00002200 ~ 0x000023FF	512
		. . .	. . .	. . .	. . .
		Page 1022	0x0027FC00 ~ 0x0027FDFF	0x0007FC00 ~ 0x0007FDFF	512
		Page 1023	0x0027FE00 ~ 0x0027FFFF	0x0007FE00 ~ 0x0007FFFF	512
Specific Memory (SM)	Information Block	Page 0	0x00400000 ~ 0x004001FF	SM: 0x00000000 ~ 0x000001FF	512
		Page 1	0x00400200 ~ 0x004003FF	SM: 0x00000200 ~ 0x000003FF	512
		Page 2	0x00400400 ~ 0x004005FF	SM: 0x00000400 ~ 0x000005FF	512
		Page 3	0x00400600 ~ 0x004007FF	SM: 0x00000600 ~ 0x000007FF	512

21.6.2 When REMAP = 0, Shadow Memory read structure

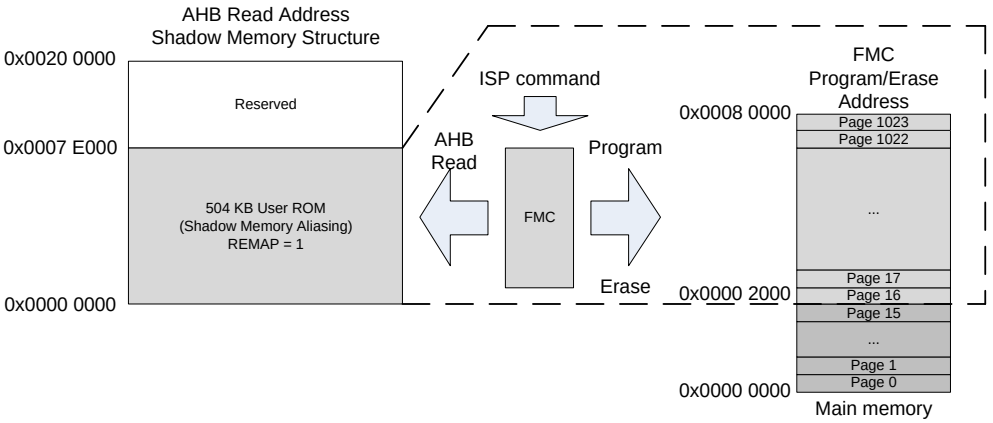
- Shadow Memory Read Address = FMC Program/Erase Address



Memory	Block	Name	Shadow Memory Read Address	FMC Program/Erase Address	Size (Byte)
Main Memory (MM)	Boot ROM	Page 0	0x00000000 ~ 0x000001FF	0x00000000 ~ 0x000001FF	512
		Page 1	0x00000200 ~ 0x000003FF	0x00000200 ~ 0x000003FF	512
		. . .	. . .	. . .	. . .
		Page 15	0x00001E00 ~ 0x00001FFF	0x00001E00 ~ 0x00001FFF	512
	User ROM	Page 16	0x00002000 ~ 0x000021FF	0x00002000 ~ 0x000021FF	512
		Page 17	0x00002200 ~ 0x000023FF	0x00002200 ~ 0x000023FF	512
		. . .	. . .	. . .	. . .
		Page 1022	0x0007FC00 ~ 0x0007FDFF	0x0007FC00 ~ 0x0007FDFF	512
		Page 1023	0x0007FE00 ~ 0x0007FFFF	0x0007FE00 ~ 0x0007FFFF	512

21.6.3 When REMAP = 1, Shadow Memory read structure

- Shadow Memory Read Address = FMC Program/Erase Address – 0x2000



Memory	Block	Name	Shadow Memory Read Address	FMC Program/Erase Address	Size (Byte)
Main Memory (MM)	User ROM	Page 16	0x0000**0**000 ~ 0x0000**0**1FF	0x00002000 ~ 0x000021FF	512
		Page 17	0x0000**0**200 ~ 0x0000**0**3FF	0x00002200 ~ 0x000023FF	512
		. . .	. . .	. . .	. . .
		Page 1022	0x0007**D**C00 ~ 0x0007**D**DFF	0x0007FC00 ~ 0x0007FDFF	512
		Page 1023	0x0007**D**E00 ~ 0x0007**D**FFF	0x0007FE00 ~ 0x0007FFFF	512

✷ ***Note: When REMAP is 1, user needs to pay attention to the function address in the USER ROM area. It is recommended to plan the function address and call behavior to conform to the shadow memory area read operation.***

21.7 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.

21.8 FLASH MEMORY CONTROLLER (FMC)

The FMC handles the program and erase operations of the Flash memory. The FMC executes ISP operations through message registers.

21.8.1 ISP Flow

1. Check PROC_BUSY bit
2. Set IE for wakeup/interrupt
3. Program MSG0 ~ MSG2 registers
4. Program data buffer
5. Set PROC_START bit
6. Wait flash
   - Auto hold = 0: WFI command for CPU sleep
   - Auto hold = 0: execute in SRAM
   - Auto hold = 1: CPU hold-up
7. Wait interrupt/Check status (DONE/FAIL)
8. Check status code in MSG3 register

21.8.2 Message Description

This section describes the messages available in FMC. Each message represents an operation, which tells FMC what to do and returns a status code, as shown in the table below to indicate if the operation succeeded or failed due to some error.

Message Status Code

Code	Description
0x0000	Pass
0x0001	Invalid page address
0x0002	Invalid memory type
0x0003	Attempted bulk/sector/subsector erase on SM
0x0004	Erase failed
0x0005	Program failed
0x0006	Read protect change failed
0x0008	Arbitrary write failed
0x0009	Invalid magic number
0x000A	Write boot flag write failed
0x000B	Write customer use data write failed
0x000C	Invalid offset
0x000D	Invalid length
0xFFFF	Failure

21.8.3 Message Function

Function	MSG0					MSG1	MSG2	MSG3		BUF 512 Byte
	11	10	9	8	[7:0]			[31:16]	[15:0]	
MM/SM erase	0x0: Page 0x1: Subsector 0x2: Sector 0x3: Bulk all		0x0: MM 0x1: SM		**0x1**	Page address MM: 0~1023 SM: Only 3	-	-	Status code.	-
MM/SM page write	-		0x0: MM 0x1: SM		**0x2**	Page address MM: 0~1023 SM: Only 3	-	-	Status code.	Program Data
MM/SM arbitrary write	-		0x0: MM 0x1: SM		**0x3**	Page address	[17:9]: Offset [8:0]: Length	-	Status code.	Program Data
MM Read protect	-		-		**0x4**	0x5A: CS1 0xA5: CS3 Others : CS0	-	-	Status code.	-
MM/SM Protect status	-		-		**0x6**	-	-	[31:24]: MM RDP status. 0x5A: CS1. 0xA5: CS3 Others: CS0.	Status code.	-
boot flag read/ write	-		-	0:Read. 1: Write.	**0x7**	[7:0]: Boot flag 0x5A: User code Others: Boot code.	-	[23:16]: Boot flag status. 0x5A : User code Others : Boot code.	Status code.	-
Flash clock frequency read/write	-		-	0:Read. 1: Write.	**0x9**	[31:0]: Flash FREQ	-	-	Status code.	Byte 0~3 Flash FREQ
MM Checksum calculate	-		-	-	**0xA**	[31:0]: Start offset	[31:0]: Length	[31:16]: 16 bit Checksum	Status code.	-

21.8.3.1 Message for MM/SM erase

The Flash memory can be erased page by page/ subsector (8 pages) / sector(512 pages) / Bulk (Mass Erase).

Pages of the Flash memory can be erased using the Erase feature of the FMC. To erase pages, the procedure below should be followed:

1. Set the MSG0 register to select memory type / erase range / erase operation.
2. Program the MSG1 register to select page address to erase
3. Set the PROC_START bit.
4. Wait for the DONE_STATUS bit in INT_STATUS register and check message status code in MSG3 register.
5. Read the erased page and verify.

✷ ***Note:***
1. When the Flash memory read protection is changed from CS1 to CS0, a Page Erase of the USER ROM is performed (except BOOT ROM).
2. When the Flash memory read protection is changed from CS3 to CS0, a Mass Erase of the main Flash memory is performed.

The start address of Page Erase / Subsector Erase / Sector Erase is fixed.

1. Page Erase (1 page): 0x0000_0000 / 0x0000_0200 / 0x0000_0400…
2. Subsector (8 pages): 0x0000_0000 / 0x0000_1000 / 0x0000_2000…
3. Sector (512 pages): 0x0000_0000 / 0x0004_0000
4. Bulk (all pages): 0x0000_0000

Message Register	Bit	Description
MSG0	31:12	Reserved
	11:10	Erase range 0x0: Page 0x1: Subsector (MM only) 0x2: Sector (MM only) 0x3: Bulk all (MM only)
	9:8	Memory type. 0x0: Main memory (MM) 0x1: Specific memory (SM)
	7:0	Operation. Erase = 0x1
MSG1	31:12	Reserved
	11:0	Page address. Each page of MM and SM has a unique page number and is numbered in order starting from 0. Flash has 2 sectors of 512 pages each in MM, and 4 pages in SM. The pages of MM are numbered from 0 to 1023, and the pages of SM are numbered from 0 to 3.
MSG2	31:0	Reserved
MSG3	31:16	Reserved
	15:0	Status code. The code indicates if the operation succeeded or failed due to error.

✷ ***Note:***
1. Subsector, sector and bulk erase are only supported for main memory.
2. The specific memory page 0~2 is reserved. Do not erase to avoid mistakes.

21.8.3.2 Message for MM/SM page program

In this mode the CPU programs the main Flash memory by performing standard page write operations.

The main Flash memory programming sequence in standard mode is as follows:

1. Set the MSG0 register to select memory type & program operation.
2. Program the MSG1 register to select page address to program
3. Set the PROC_START bit.
4. Wait for the DONE_STATUS bit in INT_STATUS register and check message status code in MSG3 register.

Message Register	Bit	Description
MSG0	31:10	Reserved
	9:8	Memory type. 0x0: Main memory (MM) 0x1: Specific memory (SM)
	7:0	Operation. Program = 0x2 This operation programs data from BUF into a given page.
MSG1	31:12	Reserved
	11:0	Page address. Each page of MM and SM has a unique page number and is numbered in order starting from 0. Flash has 2 sectors of 512 pages each in MM, and 4 pages in SM. The pages of MM are numbered from 0 to 1023, and the pages of SM are numbered from 0 to 3.
MSG2	31:0	Reserved
MSG3	31:16	Reserved
	15:0	Status code. The code indicates if the operation succeeded or failed due to error.

BUF	Address	Description
Data buffer	0x00600000 ~ 0x006001FF	Program data 512 bytes

✸ ***Note:***

1. ***The erase operations should be done before the program operation.***
2. ***The specific memory page 0~2 is reserved. Do not program to avoid mistakes.***

21.8.3.3 Message for MM/SM arbitrary write

In this mode the CPU programs the main Flash memory by performing standard byte write operations.

The main Flash memory programming sequence in standard mode is as follows:

1. Set the MSG0 register to select memory type & program operation.
2. Program the MSG1 register to select page address to program.
3. Program the MSG2 register to select offset & length.
4. Set the PROC_START bit.
5. Wait for the DONE_STATUS bit in INT_STATUS register and check message status code in MSG3 register.

Message Register	Bit	Description
MSG0	31:10	Reserved
	9:8	Memory type. 0x0: Main memory (MM) 0x1: Specific memory (SM)
	7:0	Operation. Arbitrary write = 0x3 This operation writes data in a length of bytes (1~512) beginning at 0 in BUF into the space starting at an offset (0~511) in a page. For example, length = 17 (MSG2[8:0] = 16) and offset = 43 (MSG2[17:9] = 43) mean that 17 bytes beginning at 0 in BUF will be written into the space starting at the 44th byte (Offset 43) in a page without change of the rest of the page. If the space at an offset in page couldn't accommodate data in a length, the exceeded data will be truncated unconditionally. That is, length + offset <= 512.
MSG1	31:12	Reserved
	11:0	Page address. Each page of MM and SM has a unique page number and is numbered in order starting from 0. Flash has 2 sectors of 512 pages each in MM, and 4 pages in SM. The pages of MM are numbered from 0 to 1023, and the pages of SM are numbered from 0 to 3.
MSG2	31:0	Reserved
	17:9	Offset in a page. 0: Starting at 0 in a page 1: Starting at 1 in a page … 511: Starting at 511 in a page
	8:0	Length of bytes in BUF (minus 1). 0: 1 byte 1: 2 bytes 2: 3 bytes … 511: 512 bytes
MSG3	31:16	Reserved
	15:0	Status code. The code indicates if the operation succeeded or failed due to error.

BUF	Address	Description
Data buffer	0x00600000 ~ 0x006001FF	Program data 1~512 bytes

✸ ***Note:***

1. ***The Arbitrary write operation runs without necessity of erase operations in advance.***
2. ***The specific memory page 0~2 is reserved. Do not operate to avoid mistakes.***

21.8.3.4 Message for MM read protect

The read protection is activated by setting the RDP byte.

To set read protection sequence is as follows:

1. Set the MSG0 register to select read protect operation.
2. Program the MSG1 register to select read protection level.
3. Set the PROC_START bit.
4. Wait for the DONE_STATUS bit in INT_STATUS register and check message status code in MSG3 register.
5. MM read protection (RDP/ mRDP) will show in information block page 2. mRDP is RDP inverting byte.

Message Register	Bit	Description
MSG0	31:8	Reserved
	7:0	Operation. Read protect = 0x4 This operation takes effect only on MM.
MSG1	31:8	Reserved
	7:0	Page address. Read protection (RDP). 0x5A: CS1 Prevents SRAM from reading and writing when ICE mode is engaged. Prevents all MM pages from reading when ICE mode is engaged. Allows SM pages to read. ● If current RDP is CS3, it is forbidden to change into CS1. ● It is permitted to change into CS0 or CS2. 0xA5: CS3 Prevents SRAM from reading and writing when ICE mode is engaged. Prevents all MM pages from reading when ICE mode is engaged. Allows SM pages to read. ● It is forbidden to change into CS1. ● It is permitted to change into CS0. Others: CS0 Allows SRAM to read and write when ICE mode is engaged. Allows MM pages to read when ICE mode is engaged. Allows SM pages to read. ● If current RDP is CS1, erase all MM pages except for boot code. ● If current RDP is CS2, erase all MM pages. ● It is permitted to change into CS1 or CS2.
MSG2	31:0	Reserved
MSG3	31:16	Reserved
	15:0	Status code. The code indicates if the operation succeeded or failed due to error.

21.8.3.5 Message for MM/SM protect status

Read protection status is checking by protect status mode.

To check read protection sequence is as follows:

1. Set the MSG0 register to select protect status operation.
2. Set the PROC_START bit.
3. Wait for the DONE_STATUS bit in INT_STATUS register and check message status code in MSG3 register.
4. MM read protection status shows in MSG3[31:24]

Message Register	Bit	Description
MSG0	31:8	Reserved
	7:0	Operation. Protect status = 0x6 This operation reads all protection status including MM read protection.
MSG1	31:0	Reserved
MSG2	31:0	Reserved
MSG3	31:24	Show MM read protection status. 0x5A: Indicates RDP is CS1. 0xA5: Indicates RDP is CS3. Others: Indicates RDP is CS0.
	23:16	Reserved
	15:0	Status code. The code indicates if the operation succeeded or failed due to error.

21.8.3.6 Message for boot flag read/ write

In this mode the boot byte modify by performing boot flag operations.

Boot flag read / write sequence is as follows:

1. Set the MSG0 register to select access type & boot flag operation.
2. Program the MSG1 register to select boot flag.
3. Set the PROC_START bit.
4. Wait for the DONE_STATUS bit in INT_STATUS register and check message status code in MSG3 register.
5. Boot flag can be read in MSG3[23:16] in read operation.
6. Boot flag (Boot / mBoot Byte) will show in information block page 2. mBoot is Boot inverting byte.

Message Register	Bit	Description
MSG0	31:9	Reserved
	8	Access type. 0: Read. 1: Write.
	7:0	Operation. Read/Write boot flag = 0x7 This operation reads or writes boot flag. Read boot flag (MSG0[8] = 0x0): Boot flag will be read out and put into MSG3[23:16]. Write boot flag (MSG0[8] = 0x1): Boot flag (MSG1[7:0]) will be written.
MSG1	31:8	Reserved
	7:0	Boot flag to write. 0x5A: Boot from User ROM. Others: Boot from Boot ROM.
MSG2	31:0	Reserved
MSG3	31:24	Reserved
	23:16	Boot flag read out. 0x5A: Boot from User ROM. Others: Boot from Boot ROM.
	15:0	Status code. The code indicates if the operation succeeded or failed due to error.

21.8.3.7 Message for Flash clock frequency (FREQ) read/write

In this mode the Flash clock is modified by performing clock frequency operations.

The Flash clock frequency read/ write sequence is as follows:

1. Set the MSG0 register to select access type and clock frequency read/write operation.
2. Program the MSG1 register to write Flash FREQ.
3. Set the PROC_START bit.
4. Wait for the DONE_STATUS bit in INT_STATUS register and check message status code in MSG3 register.
5. Flash FREQ can be read in BUF byte0~byte3.

Message Register	Bit	Description
MSG0	31:9	Reserved
	8	Access type. 0: Read. 1: Write.
	7:0	Operation. Read/Write Flash FREQ = 0x9 This operation reads or writes Flash FREQ in MHz. Read Flash FREQ (MSG0[8] = 0x0): Flash clock frequency in MHz will be read out and put into Bytes 0~3 in BUF. Write Flash FREQ (MSG0[8] = 0x1): Flash clock frequency in MHz (MSG1[31:0]) will be written and applied into Flash.
MSG1	31:0	Flash FREQ to write (Unit MHz).
MSG2	31:0	Reserved
MSG3	31:16	Reserved
	15:0	Status code. The code indicates if the operation succeeded or failed due to error.

BUF	Address	Description
Data buffer	0x00600000 ~ 0x00600003	Flash FREQ after read out. (Unit MHz).

✷ ***Note:***

1. The system efficiency is highest when the Flash clock frequency is equal to the current HCLK frequency.

2. If HCLK exceeds the Flash clock frequency, the system may have a hard fault.

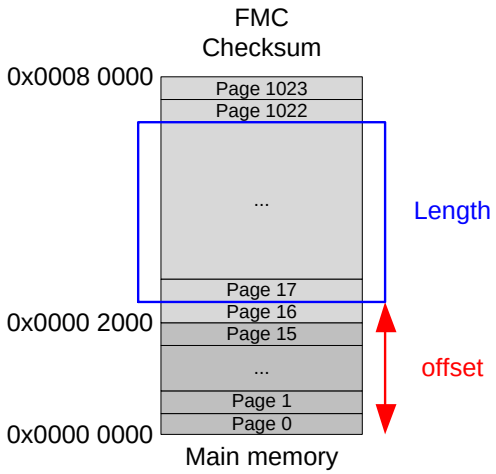
3. It is recommended to increase the Flash clock frequency before modifying HCLK to ensure sufficient access time.

21.8.3.8 Message for MM checksum calculate

HW checksum is the checksum of User/Boot ROM. If the read protection is enabled, the users can still readout the HW checksum through Writer or ISP AP.

The HW checksum sequence is as follows:

1. Set the MSG0 register to select checksum operation.
2. Program the MSG1 register to set start offset.
3. Program the MSG2 register to set length.
4. Set the PROC_START bit.
5. Wait for the DONE_STATUS bit in INT_STATUS register and check message status code in MSG3 register.
6. The 16-bit checksum shows in MSG3[31:16].



Message Register	Bit	Description
MSG0	31:8	Reserved
	7:0	Operation. Calculate checksum = 0xA This operation calculates the checksum of MM data in a length of bytes (MSG2[31:0]) starting at an offset (MSG1[31:0]) and puts the checksum into MSG3[31:16]. If start offset + length > MM size, the exceeded part will be skipped unconditionally.
MSG1	31:0	Start offset in MM (0x00000000 ~ 0x0007FE00).
MSG2	31:0	Length of bytes (0x200 ~ 0x80000).
MSG3	31:16	16-bit Checksum
	15:0	Status code. The code indicates if the operation succeeded or failed due to error.

✷ ***Note: The length must be greater than or equal to 0x200 and will be converted, if it is not a multiple of 4, into a multiple of 4, which is greater than or equal to the length.***

21.8.4 Auto-hold function

This section depicts how to communicate with Flash via messages, such as erase, program, etc. There are interrupt and polling modes, which can be used to request a message to ask Flash to do some operations, such as erase, program, arbitrary write, checksum calculate, etc.

Both interrupt and polling modes can go along with auto-hold feature, which is enabled by setting AUTO_HOLD[0] (0x601058) to 1. With auto-hold enabled (AUTO_HOLD[0] = 1), interrupt and polling modes can execute in MM or SRAM. Once PROC_START[0] is written by host CPU, it will be held immediately and other masters, e.g., DMA, trying to access Flash memory region, will also be held until the message processing is done. If auto-hold is disabled (AUTO_HOLD[0] = 0), polling mode must not execute in MM because access path to MM is switched to Flash during the message processing, leading to unknown data accessed by host CPU, which might cause Hard Fault exception. With auto-hold disabled, interrupt mode can execute in MM, but host CPU must go to sleep as soon as PROC_START[0] is written. Besides, all interrupts, except for Flash interrupt, must be disabled during the message processing. Otherwise, host CPU will be woken up by other interrupts and might access unknown data. The following table delineates the access behavior during message processing with auto-hold enabled and disabled.

Access Behavior during Message Processing

PROC_START[0] written	AUTO_HOLD[0] = 0	AUTO_HOLD[0] = 1
CPU running in MM	CPU must go to idle (WFI); otherwise, unpredictable	CPU held
CPU running in SRAM	OK	CPU held
CPU accessing MM or SM	Unpredictable	CPU held
CPU accessing 0x600000~0x6013FF	OK	CPU held
CPU accessing SRAM	OK	CPU held
Other masters accessing MM or SM	Unpredictable	Other master held
Other master accessing 0x600000~0x6013FF	OK	Other master held
Other master accessing SRAM	OK	OK
Other interrupt happening and running in SRAM	OK	CPU held
Other interrupt happening and running in MM	Unpredictable	CPU held

Note that Flash can only handle one message at a time. Once a message is requested to handle, message registers (MSG0~2) cannot be changed, and setting PROC_START[0] takes no effect until the message is done.

✸ ***Note:***

1. ***If the CPU runs in Flash, it is recommended to set AUTO_HOLD[0] to avoid errors.***
2. ***If the CPU runs in SRAM and other master don't read flash, it is recommended to clear AUTO_HOLD[0]. This allows continued operation during the ISP, increasing system efficiency.***
3. ***AUTO_HOLD[0] can only be cleared when flash is not read by any master.***

21.8.5 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.

The read protection is activated by setting the RDP bytes. The RDP bytes must be modified via MM read protect message function. The read protection has three level: CS0, CS1 and CS3.

- 0x5A: CS1
  Prevents SRAM from reading and writing when ICE mode is engaged.
  Prevents all MM pages from reading when ICE mode is engaged.
  Allows SM pages to read.
  - If current RDP is CS3, it is forbidden to change into CS1.
  - It is permitted to change into CS0 or CS3.
- 0xA5: CS3
  Prevents SRAM from reading and writing when ICE mode is engaged.
  Prevents all MM pages from reading when ICE mode is engaged.
  Allows SM pages to read.
  - It is forbidden to change into CS1.
  - It is permitted to change into CS0.
- Others: CS0
  Allows SRAM to read and write when ICE mode is engaged.
  Allows MM pages to read when ICE mode is engaged.
  Allows SM pages to read.
  - If current RDP is CS1, erase all MM pages except for boot code.
  - If current RDP is CS3, erase all MM pages.
  - It is permitted to change into CS1 or CS3.

After power on, the RDP status as below

- The default value of MM RDP is latched from RDP Byte in SM page 2.

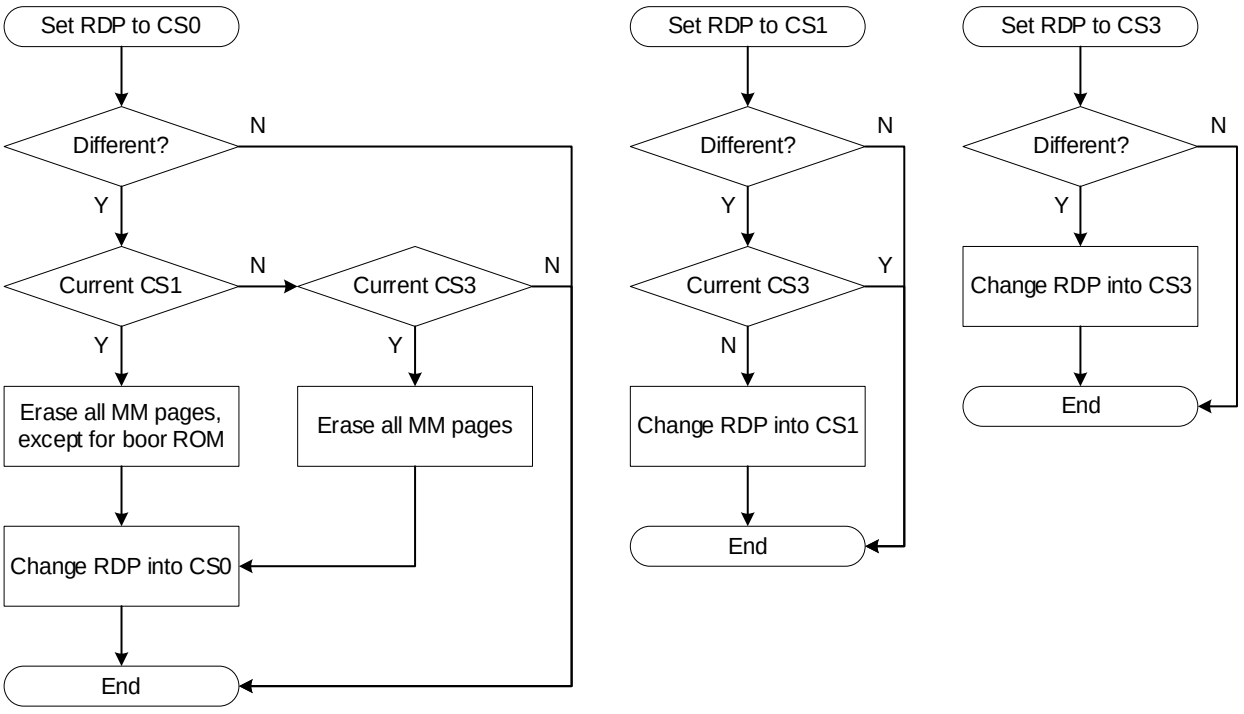
> ✸ ***Note: Any Code Security change becomes effective only after the MCU has been Reboot.***

Block	Security	SWD/Writer			Normal mode		
	Level	CS0	CS1	CS3	CS0	CS1	CS3
	RDP	RDP=Other	RDP=0x5A	RDP=0xA5	RDP=Other	RDP=0x5A	RDP=0xA5
	Status	Emulating is OK	Cannot emulate	Cannot emulate	Free run		
MM (Boot ROM & User ROM)	Read	O	X	X	O	O	O
	Program	O	O	O	O	O	O
	Erase	O	O	O	O	O	O
SM ROM	Read	O	O	O	O	O	O
	Program	O	O	O	O	O	O
	Erase	O	O	O	O	O	O
SRAM	Read	O	X	X	O	O	O
	Write	O	X	X	O	O	O
Note			Ref. Note Table	Ref. Note Table		Ref. Note Table	Ref. Note Table

Note Table

RDP	Description
CS1	1. RDP can be upgraded. CS1 → CS3 OK 2. RDP can be downgraded. CS1 → CS0 (Erase all MM pages except for boot ROM)
CS3	1. RDP cannot be downgraded to CS1. Cannot CS3 → CS1 2. RDP can be downgraded to CS0. CS3 → CS0 (Erase all MM pages)

✸ ***Note: User may try to change read security level from CS3 to CS0, or from CS1 to CS0. HW shall:***
1. ***Mass erase the User ROM first.***
2. ***Mass erase the boot ROM when original RDP is CS3.***
3. ***Update security level.***



21.9 FLASH REGISTERS

Base Address: 0x0060 1000

There are seven access types: RW: Read or write, R: Read Only, RW1C: Read and Set by Hardware and write 1 to clear.

Offset	Register	Description
0x0000	FLASH_MSG0	FLASH Message 0 register
0x0004	FLASH_MSG1	FLASH Message 1 register
0x0008	FLASH_MSG2	FLASH Message 2 register
0x000C	FLASH_MSG3	FLASH Message 3 register
0x0010 – 0x003C	-	Reserved
0x0040	FLASH_PROC_START	FLASH Process Start Control register
0x0044	-	Reserved
0x0048	FLASH_PROC_BUSY	FLASH Process Busy Status register
0x004C	FLASH_IE	FLASH Interrupt Enable register
0x0050	FLASH_IST	FLASH Interrupt Status register
0x0054	-	Reserved
0x0058	FLASH_AUTO_HOLD	FLASH Auto-hold Control register

21.9.1 FLASH Message 0 register (FLASH_MSG0)

Address offset: 0x00

MSG0~2 register are locked and read only when PROC_BUSY is 1. Please refer to FLASH MEMORY CONTROL section for detailed description.

Bit	Field	Access	Initial	Description
31:12	-	-	0	Reserved
11:10	RANGE	RW	0	Memory range used ONLY for Erase/Program operation 0: Page (MM and SM) 1: Subsector (MM only) 2: Sector (MM only) 3: Bulk all (MM only)
9:8	TYPE	RW	0	Memory type 0: Main memory/Read boot flag/Read FLASH ROM frequency 1: Supervisory memory/Write boot flag/Write FLASH ROM frequency Other: Reserved
7:0	OPERATION	RW	0	FMC operation 1: Erase operation 2: Program operation 3: Arbitrary write operation 4: Read protect operation 6: Read protect status operation 7: Read/Write boot flag operation 9: Read/Write FLASH ROM frequency operation 10: Calculate MM checksum operation Other: Reserved

21.9.2 FLASH Message 1 register (FLASH_MSG1)

Address offset: 0x04

MSG0~2 register are locked and read only when PROC_BUSY is 1. Please refer to FLASH MEMORY CONTROL section for detailed description.

Bit	Field	Access	Initial	Description
31:0	MSG1	RW	0	FMC Message 1 Page address[11:0] ONLY for Erase/Program/Arbitrary write operation.

21.9.3 FLASH Message 2 register (FLASH_MSG2)

Address offset: 0x08

MSG0~2 register are locked and read only when PROC_BUSY is 1. Please refer to FLASH MEMORY CONTROL section for detailed description.

Bit	Field	Access	Initial	Description
31:0	MSG2	RW	0	FMC Message 2

21.9.4 FLASH Message 3 register (FLASH_MSG3)

Address offset: 0x0C

Please refer to FLASH MEMORY CONTROL section for detailed description.

Bit	Field	Access	Initial	Description
31:16	STATUS_CODE2	RW	0	MM read protection status or boot flag or MM checksum.
15:0	STATUS_CODE	RW	0	Status code 0: Pass 1: Invalid page address 2: Invalid memory type 3: Erase operation 4: Erase failed 5: Program failed 6: Read protect change failed 8: Arbitrary write failed 9: Invalid magic number 10: Write boot flag failed 11: Write customer-used data failed 12: Invalid offset 13: Invalid length 65535: Failure Other: Reserved

21.9.5 FLASH Process Start Control register (FLASH_PROC_START)

Address offset: 0x40

Writing to this bit takes no effect when PROC_BUSY is 1.

Bit	Field	Access	Initial	Description
31:1	-	-	0	Reserved
0	PROC_START	RW	0	Process start control bit 0: Idle or finish process 1: Start process

21.9.6 FLASH Process Busy Status register (FLASH_PROC_BUSY)

Address offset: 0x48

Bit	Field	Access	Initial	Description
31:1	-	-	0	Reserved
0	PROC_BUSY	R	0	Process busy status 0: Process is idle 1: Process is busy

21.9.7 FLASH Interrupt Enable register (FLASH_IE)

Address offset: 0x4C

Bit	Field	Access	Initial	Description
31:1	-	-	0	Reserved
0	IE	RW	0	Interrupt enable bit 0: Disable 1: Enable

21.9.8 FLASH Interrupt status register (FLASH_IST)

Address offset: 0x50

Bit	Field	Access	Initial	Description
31:2	-	-	0	Reserved
1	FAIL	RW1C	0	Process fail status 0: No effect 1: Process is done but failed
0	DONE	RW1C	0	Process done status 0: No effect 1: Process is done

21.9.9 FLASH Auto-hold Control register (FLASH_AUTO_HOLD)

Address offset: 0x58

Bit	Field	Access	Initial	Description
31:1	-	-	0	Reserved
0	AUTO_HOLD	RW	1	AHB interface auto-hold enable bit 0: Disable 1: Enable → The auto-hold function is enabled and PROC_START[0] written, host CPU will be held immediately and other masters (e.g., DMA) will also be held when accessing flash memory region (0x600000 ~ 0x6013FF).

22 SERIAL-WIRE DEBUG (SWD)

22.1 OVERVIEW

SWD functions are integrated into the ARM Cortex-M4. The ARM Cortex-M4 is configured to support up to four breakpoints and two watch points.

22.2 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.

22.3 PIN DESCRIPTION

Pin Name	Type	Description	GPIO Configuration
SWCLK	I	Serial Wire Clock pin in SWD mode.	Depends on AFIO register
SWDIO	I/O	Serial Wire Data Input/Output pin in SWD mode.	Depends on AFIO register

22.4 DEBUG NOTE

22.4.1 LIMITATIONS

Debug mode changes the way in which reduced power modes work internal to the ARM Cortex-M4 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.

22.4.2 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.***

22.4.3 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.

23 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)

SN-LINK-V3 IDE Tools

SONiX 32-bit series Embedded ICE Emulator Feature:

- Target's Operating Voltage: 2.4V~3.6V.
- Up to 4 hardware break points.
- System clock rate up to 192MHz.
- Oscillator supports IHRC, ILRC, EHS/ELS X'tal.

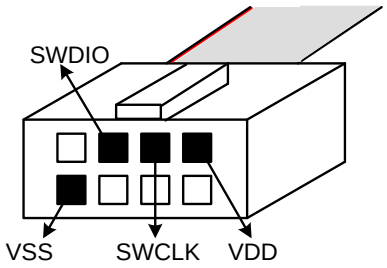
SONiX 32-bit series Embedded ICE Emulator Limitation:

- SWCLK and SWDIO pins are shared with GPIO pins. In embedded ICE mode, the shared GPIO function can't work.

23.1 SN-LINK-V3

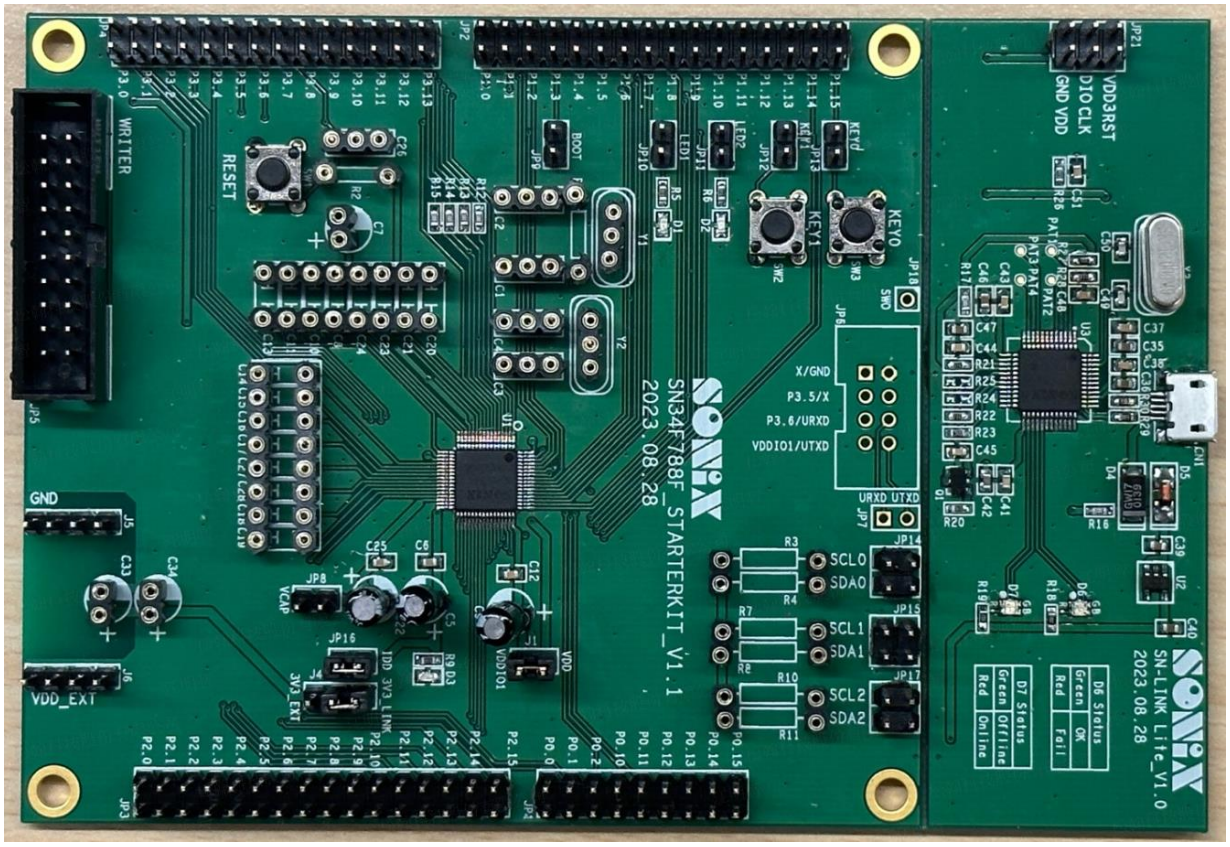
SN-LINK-V3 is a high speed emulator for SONiX 32-bit MCU. It debugs and programs based on SWD protocol. In addition to debugger functions, the SN-LINK-V3 also may be used as a programmer to load firmware from PC to MCU for engineering production, even mass production.

SN-LINK-V3 communicates with SONiX 32-bit MCU through SWD interface. The pin definition of the Modular cable is as following:



23.2 SN34F780 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.



- U1 : SN34F788FG real chip
- JP5 : WRITER connector
- U2 : 3.3V LDO (3V3_LINK) from Micro USB 5V.
- J4 : VDD power connector: Choose the source of VDD(VDD_EXT or 3.3V LDO on board).
- J6 : VDD_EXT is 3V3_EXT.
- JP16 : For measuring current. Default short.
- J1 : VDDIO1 power connector. Default short VDD.
- RESET button : External reset trigger source.
- JP6 : SN-LINK connector when SN-LINK Lite is not connected
- JP9 : Short to force MCU stay in Boot loader.
- Y1: External high-speed X'tal for SN34F788FG
- Y2: External low-speed 32.768KHz X'tal for SN34F788FG
- Y3: External high-speed X'tal for SN-LINK Lite

24 ELECTRICAL CHARACTERISTIC

24.1 ABSOLUTE MAXIMUM RATING

Supply voltage (Vdd) ..... - 0.3V ~ 3.6V
Input in voltage (Vin) ..... Vss – 0.2V ~ Vdd + 0.2V
Operating ambient temperature (Topr) ..... - 40°C ~ + 125°C
Storage ambient temperature (Tstor) ..... - 40°C ~ + 125°C

24.2 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		2.4	3.3	3.6	V
VDD rise rate	V_{POR}	VDD rise rate to ensure internal power-on reset		0.05	-	-	V/ms
VDDIO1	V_{DDIO1}	I/O driver power for P0.12~P0.15, P3.0~P3.6, P1.6~P1.11.		1.8		Vdd	V
VCAP1 capacitor	C_{LOAD}	Core power regulator output external capacitor		1.1	-	-	uF
Power Consumption							
Supply Current	Idd1	Normal mode	System clock = 12MHz [1][2][3]	-	2.1	-	mA
			System clock = 24MHz [1][3][4]		7.5		mA
			System clock = 48MHz [1][3][4]		10		mA
			System clock = 72MHz [1][3][4]	-	13	-	mA
			System clock = 96MHz [1][3][4]	-	15	-	mA
			System clock = 120MHz [1][3][4]	-	17	-	mA
			System clock = 144MHz [1][3][4]	-	20	-	mA
			System clock = 168MHz [1][3][4]	-	22.5	-	mA
			System clock = 192MHz [1][3][4]	-	25.5	-	mA
	Idd2	Sleep Mode	System clock = 12MHz [1][2][3][5]	-	1.1	-	mA
	Idd3	Deep sleep Mode	Vdd=3.3V [1][3][5]	-	300	-	uA
	Idd4	Deep power-down Mode	Vdd=3.3V [1][3][5]	-	10	-	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 = 3.3V		30	40	50	KΩ
I/O port pull-down resistor	R_{PD}	Vin = 3.3V		30	40	50	KΩ
I/O High-level output source current	I_{OH}	Standard port and RESET pins	Level 0	-	17	-	mA
			Level 1	-	19	-	mA

		V_{OP} = Vdd – 0.4V;	Level 2	-	21	-	mA
			Level 3	-	23	-	mA
I/O Low-level output sink current	I_{OL1}	Standard port and RESET pins V_{OP} = Vss + 0.4V	Level 0	-	17	-	mA
			Level 1	-	19	-	mA
			Level 2	-	21	-	mA
			Level 3	-	23	-	mA
P0.12~P0.15, P3.0~P3.6 and P1.6~P1.11 Pins [8]							
High-level input voltage	V_{IH1}			0.7 V_{DDIO1}	-	V_{DDIO1}	V
Low-level input voltage	V_{IL1}			Vss	-	0.3 V_{DDIO1}	V
Input voltage	V_{i1}			0	-	V_{DDIO1}	V
Output voltage	V_{o1}			0	-	V_{DDIO1}	V
I/O port pull-up resistor	R_{PU1}	Vin = VSS, V_{DDIO1} = 3.3V		30	40	50	KΩ
I/O port pull-down resistor	R_{PD1}	Vin = 3.3V		30	40	50	KΩ
I/O High-level output source current	I_{OH1}	Standard port and RESET pins V_{OP} = V_{DDIO1} – 0.4V;	Level 0	-	17	-	mA
			Level 1	-	19	-	mA
			Level 2	-	21	-	mA
			Level 3	-	23	-	mA
I/O Low-level output sink current	I_{OL1}	Standard port and RESET pins V_{OP} = Vss + 0.4V	Level 0	-	17	-	mA
			Level 1	-	19	-	mA
			Level 2	-	21	-	mA
			Level 3	-	23	-	mA
ADC							
ADC Operating Voltage	V_{ADC}			2.4	-	3.6	V
Internal Reference 2.5V/2V/1.5V Operating Voltage	V_{REF}	V_{REF} = VDDA		3		3.6	V
Internal Reference Capacitor	C_{REF}	Internal reference output external capacitor		1.0			uF
External reference voltage	V_{EREF}	VDDA=3.6V		2.4	3.3	VDDA	V
Internal VDD Reference Voltage	V_{IREF1}	VDDA=2.4V~3.6V			VDDA		V
Internal 2.5V Reference Voltage	V_{IREF2}	*T=25*℃, VDDA=3.3V		2.48	2.5	2.52	V
Internal 2V Reference Voltage	V_{IREF3}	*T=25*℃, VDDA=3.3V		1.98	2	2.02	V
Internal 1.5V Reference Voltage	V_{IREF4}	*T=25*℃, VDDA=3.3V		1.48	1.5	1.52	V
AVREFH pin input voltage	V_{REFH}	VDDA=3.6V		2.4	3.3	VDDA	V
ADC current consumption	I_{ADC}	VDDA=3.3V		-	450	-	uA
Internal Reference 2.5V/2V/1.5V current consumption	I_{REF}	VDDA=3.3V		-	310	-	uA
Resolution	Nr	No missing code.		-	11	12	bit
AIN0 ~ AIN15 input voltage	V_{AIN}			0	-	V_{REFH}	V
Integral Nonlinearity	INL [*]	Vdd=3.3V		-1	-	+1	LSB
Differential Nonlinearity	DNL [*]	Vdd=3.3V		-1	-	+1	LSB
ADC Clock Frequency	F_{ADCLK}	VDDA=3.3V. (F_{ADCLK} = MCLK)		1M	-	16M	Hz
ADC Offset Voltage	V_{OFFSET} [*]	Non-trimmed		-11	0	+11	mV
		Trimmed		-1	0	+1	mV
ADC enable time	T_{ADCEN}	Ready to start convert after set ADCEN = “1” [6]		3	-	-	Conv.
Internal reference enable time	T_{ADCEN}	Ready after PD_LDO = “0”		-		0.5	ms
ADC Conversion Cycle Time	F_{ADCYL}	VDDA=2.4V~3.6V		14	16	-	1/F_{ADCLK}
FLASH							
Supply Voltage	Vdd1			0.99	1.1	1.21	V
Endurance time	T_{EN}	Erase + Program		10K	100K	-	Cycle
Page Erase current	I_{PER}			-	6	-	mA
Program current	I_{PG}			-	6	-	mA
Page erase time	T_{PE}	1-Page (512 bytes) [7]		-	10.6	-	ms
Subsector erase time	T_{SSE}	8-Page (4096 bytes) [7]		-	10.6	-	ms
Sector erase time	T_{SE}	512-Page (256KB) [7]		-	10.6	-	ms

Mass erase time	T_{MER}	Main Memory all erase (1024-Page, 512KB) [7]		-	10.6	-	ms
Page Programming time	T_{PG}	1-Page (512 bytes) [7]		-	3.8	-	ms
MISC							
Low Voltage Detector	LVD	Interrupt/Reset	Level 0	2.30	2.40	2.50	V
			Level 1	2.50	2.60	2.70	V
			Level 2	2.70	2.80	2.90	V
			Level 3	2.90	3.00	3.10	V
IHRC Freq.	F_{IHRC}	*T=25*℃, Vdd=2.4V~ 3.6V		11.76	12	12.24	MHz

*** These parameters are for design reference, not tested.**

[1] I_{DD} measurements were performed with all pins configured as GPIO input and pull-up resistors enabled, code while(1); executed and VDD=3.3V
[2] IHRC and ILRC are enabled, external X'tal is 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] At least 3 idle conversion cycles are required for the ADC engine warm-up.
[7] Flash clock frequency must be set equal to AHB clock.
[8] When the package does not have a VDDIO1 pin, the voltage source of the P0.12~P0.15, P3.0~P3.6 and P1.6~P1.11 pins is VDD.

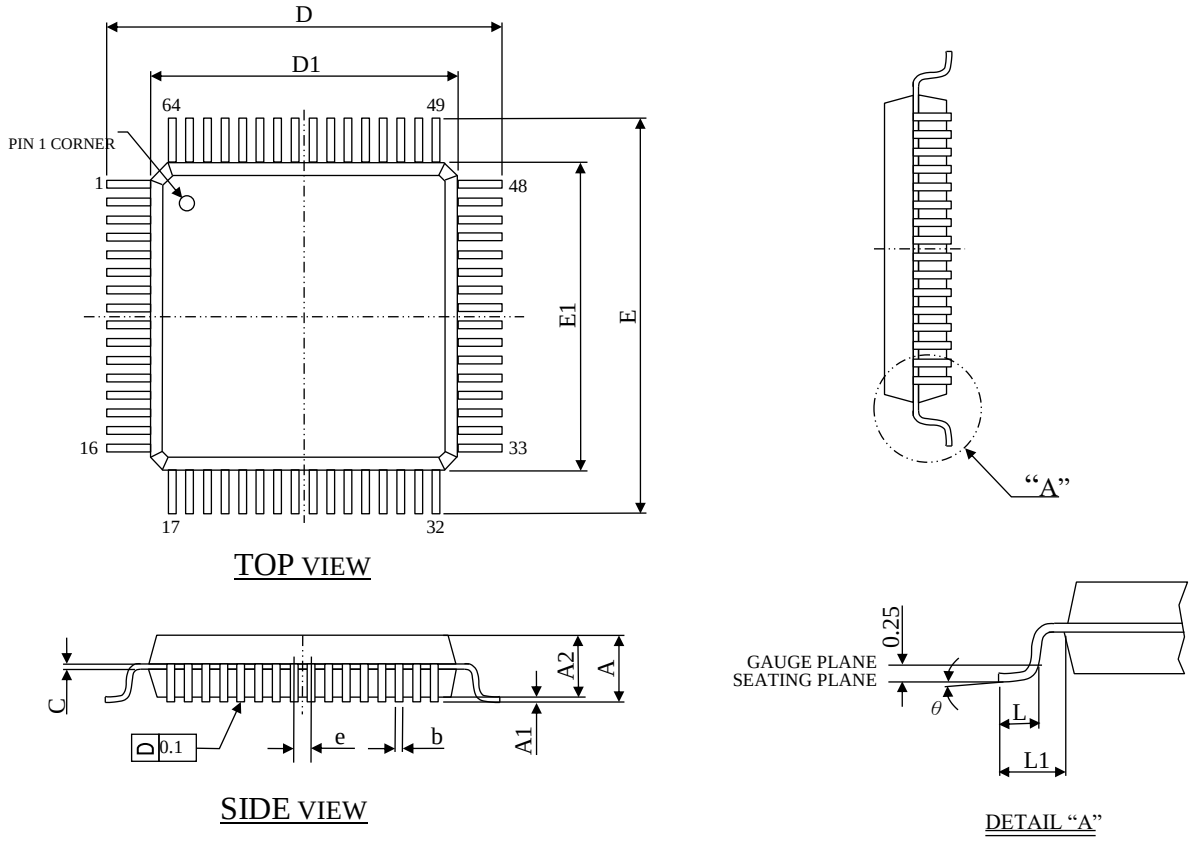
25 FLASH ROM PROGRAMMING PIN

Programming Information of SN34F780 Series							
Chip Name		SN34F788F		SN34F787F		SN34F785J	
Writer Connector JP5							
Number	Name	Number	Pin	Number	Pin	Number	Pin
1	VDD	13 19 32 48 64	VDDA VDD VDD VDDIO1 VDD	9 24 36 48	VDDA VDD VDDIO1 VDD	1 5 17	VDD VDDA VDD
2	GND	12 18 31	VSSA VSS VSS	8 23	VSSA VSS	16	VSS
3	CLK	55	P1.0	39	P1.0	26	P1.0
4	CE						
5	PGM	49	P3.6	37	P3.6	24	P3.6
6	OE	46	P3.5	34	P3.5	23	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	56	P1.1	40	P1.1	27	P1.1

✷ ***Note: The VCAP1 pin needs to connect external capacitor 1.1uF.***

26 PACKAGE INFORMATION

26.1 LQFP 64 PIN

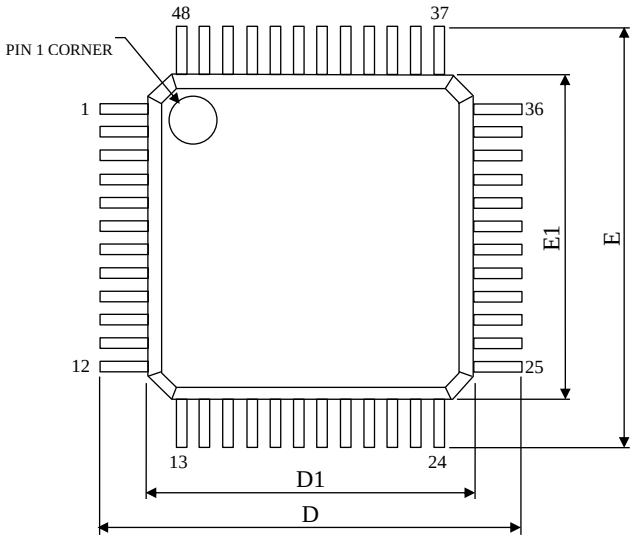


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.60	--	--	0.063
A1	0.05	--	0.25	0.002	--	0.01
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.13	0.19	0.25	0.005	0.007	0.010
c	0.09	--	0.20	0.004	--	0.008
D	9.00 BSC			0.354 BSC		
D1	7.00 BSC			0.276 BSC		
e	0.40 BSC			0.016 BSC		
E	9.00 BSC			0.354 BSC		
E1	7.00 BSC			0.276 BSC		
L	0.4	0.60	0.8	0.016	0.024	0.032
L1	1.00 REF			0.039 REF		
θ	0˚	3.5˚	7˚	0˚	3.5˚	7˚

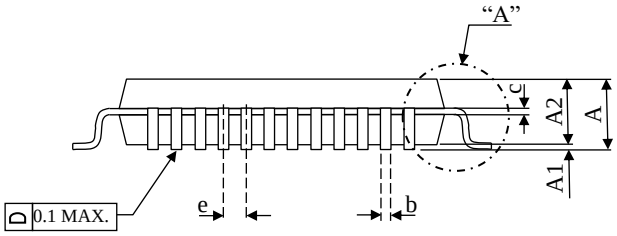
Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

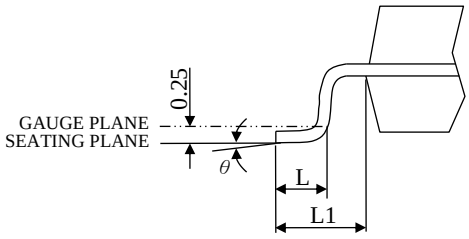
26.2 LQFP 48 PIN



TOP VIEW



SIDE VIEW



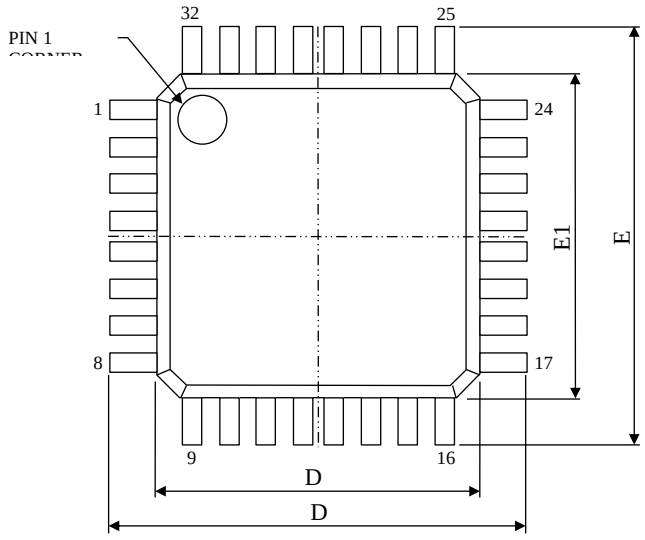
DETAIL "A"

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°

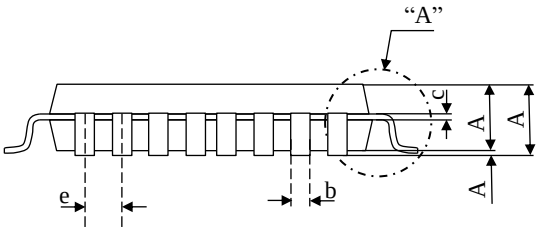
Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

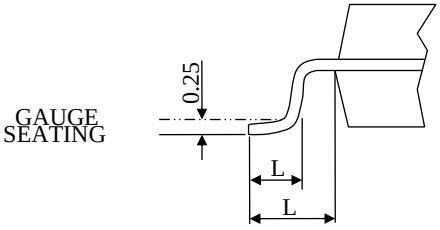
26.3 LQFP 32 PIN



TOP VIEW



SIDE VIEW



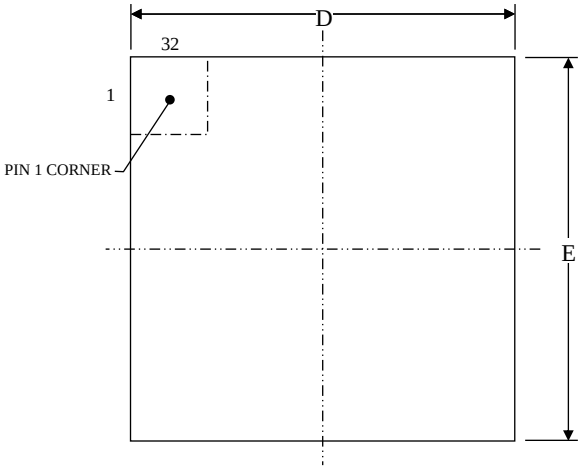
DETAIL

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		

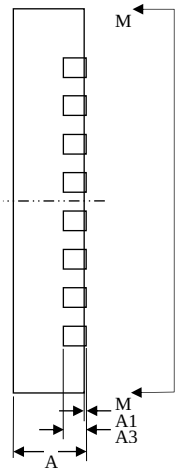
Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

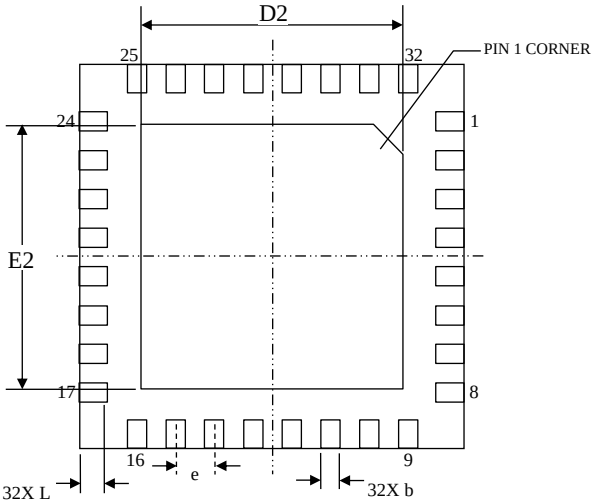
26.4 QFN 32 PIN 4x4



TOP VIEW



SIDE VIEW



BOTTOM VIEW
VIEW M－M

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	2.90	0.080	0.096	0.114
E2	2.00	2.45	2.90	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)

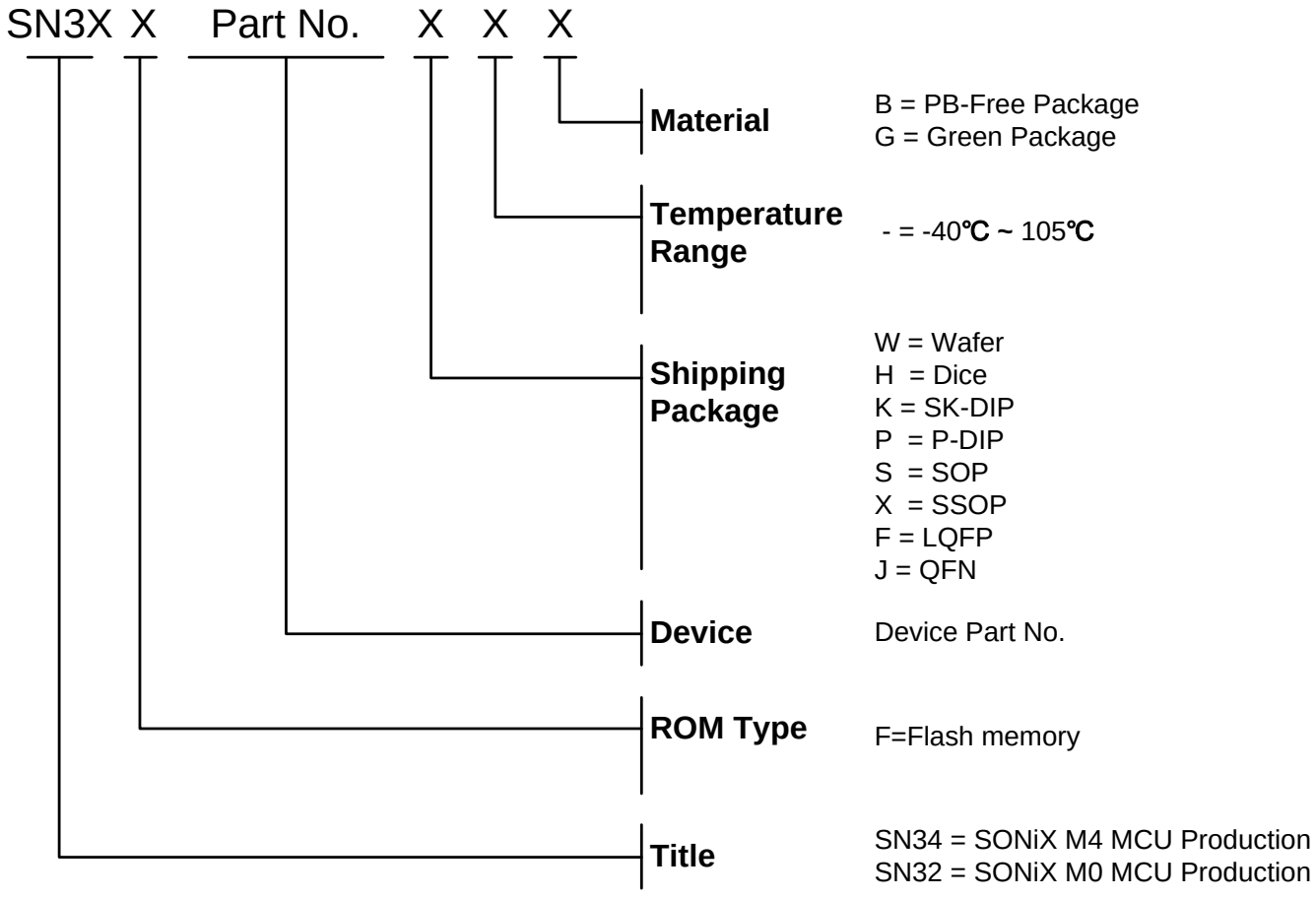
27 MARKING DEFINITION

27.1 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.

27.2 MARKING INDETIFICATION SYSTEM

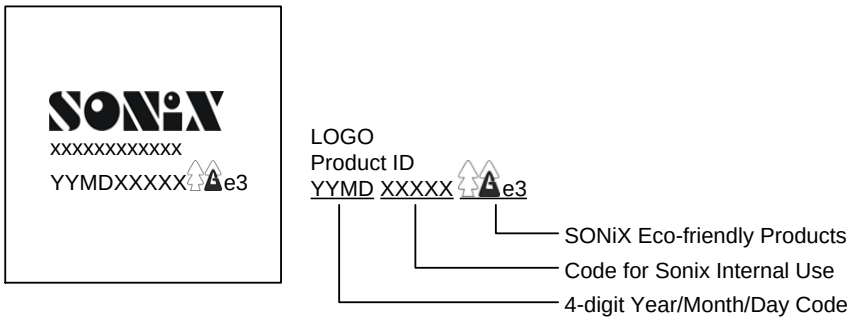


27.3 MARKING EXAMPLE

Name	ROM Type	Device	Package	Temperature	Material
SN34F788FG	Flash memory	788	LQFP	-40℃~125℃	Green Package
SN34F787FG	Flash memory	788	LQFP	-40℃~125℃	Green Package
SN34F785FG	Flash memory	788	LQFP	-40℃~125℃	Green Package
SN34F785JG	Flash memory	788	QFN	-40℃~125℃	Green Package

27.4 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