PAN271x series

User Manual

V1.1 Mar. 2026

Panchip Microelectronics Co., Ltd.

2.4GHz High-speed SoC Transceiver

General Description

PAN271x integrates 2.4GHz wireless SoC transceiver. The wireless transceiver circuit works in the 2.400-2.483GHz universal ISM frequency band. There is a 16KB OTP internal program memory and a built-in 3KB SRAM memory. In addition, PAN271x is equipped with a wealth of peripherals, including up to 19 GPIOs, 6-channel PWM, 1 25-bit timer, 1 32-bit SLPTMR, 1 I2C, 2 UARTs, 1 SPI, 1 ADC, WDT, etc.

Key Features

- **MCU**
  - 32-bit MCU core running up to 48 MHz
- **Memory**
  - Up to 16KB OTP supporting deep sleep mode
  - Up to 3KB SRAM
  - 2K bits EEPROM (PAN2713 only)
- **Low Power**
  - Active mode RX (whole chip): 10mA
  - Active mode TX at 0dBm (whole chip): 13mA
  - Standby mode0 (external interrupts): 0.82uA
  - Standby mode1 (external interrupts, SLPTMR, 3KB SRAM retention): 1.12uA
  - Deep sleep mode1 (external interrupts, SLPTMR, chip retention): 1.68uA
- **Clock**
  - 32M RC / 32.768kHz RC
  - 32M XTAL / 32.768kHz XTAL
  - DPLL(Two channels: 32M/48M)
- **RF**
  - Mode
    - 2.4G private protocol: 2Mbps / 1Mbps / 500kbps /250kbps / 125kbps, supporting hardware ACK
  - Output power: -55 to 11dBm
  - Receiver
    - -94dBm @ 1Mbps
    - -91dBm @ 2Mbps
    - -99dBm @ 250kbps
    - -98dBm @ 500kbps
    - -100dBm @ 125kbps
  - RSSI
    - Resolution: 0.25dB
    - Accuracy: ±2dB
    - Range: -90 to -15dBm
  - Single antenna supported
  - Safety regulations: ETSI / FCC
- **Peripheral**
  - Up to 19 GPIOs
  - 6-channel PWM
  - 1 25-bit timer
  - 1 32-bit SLPTMR
  - 1 I2C / 1 SPI
  - 2 UARTs
  - Up to 21-channel ADC (18 ext, bandgap, VDD/2, temp)
  - 1 WDT / 1 Keyscan
  - IO / BOD / POR / System reset
  - 1 Clock measurement
- **Power Management**
  - Integrated voltage regulator
  - Support USB 5V supply
  - Operating voltage (VBAT): 1.8 to 3.8V
  - Operating voltage (VUSB): 4.5 to 5.5V
- **Package:**
  - SSOP24 / SOP16 / SOP14 / MSOP10
- **Operating Condition**
  - Operating temperature: -40 to 85°C
  - Storage temperature: -60 to 150°C
  - ESD
    - HBM: ±2KV
    - MM: ±200V
    - CDM: ±2000V
    - Latch-up: ±100mA

Typical Applications

- Wireless remote control
- Smart home & security
- Wireless mouse & keyboard
- Wireless game controller
- Toys and wireless audio
- Active tag

Content

List of Tables

List of Figures

1 Block Diagram

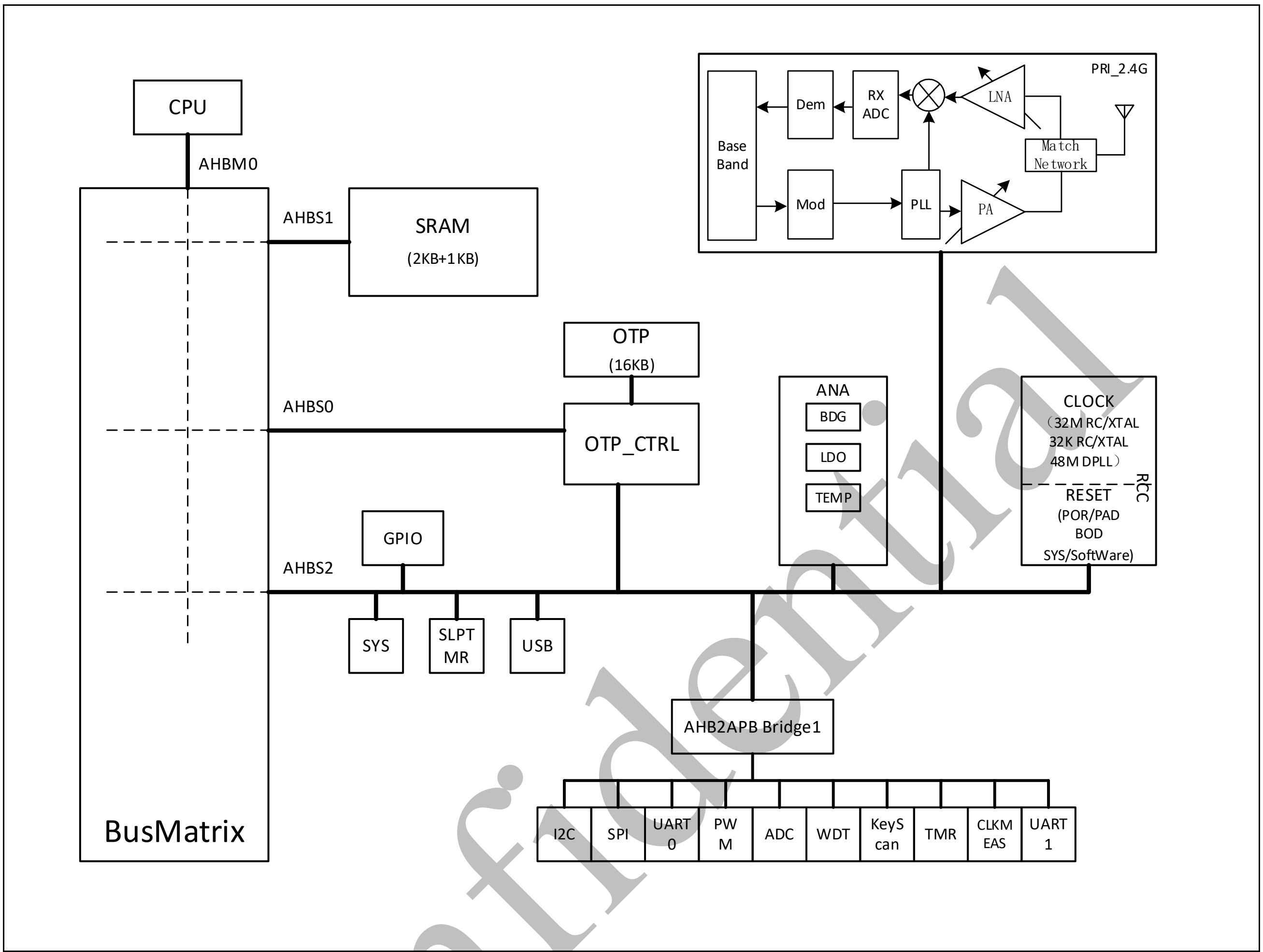


Figure 1-1 Block Diagram

2 Pin Information

2.1 SSOP24 Package

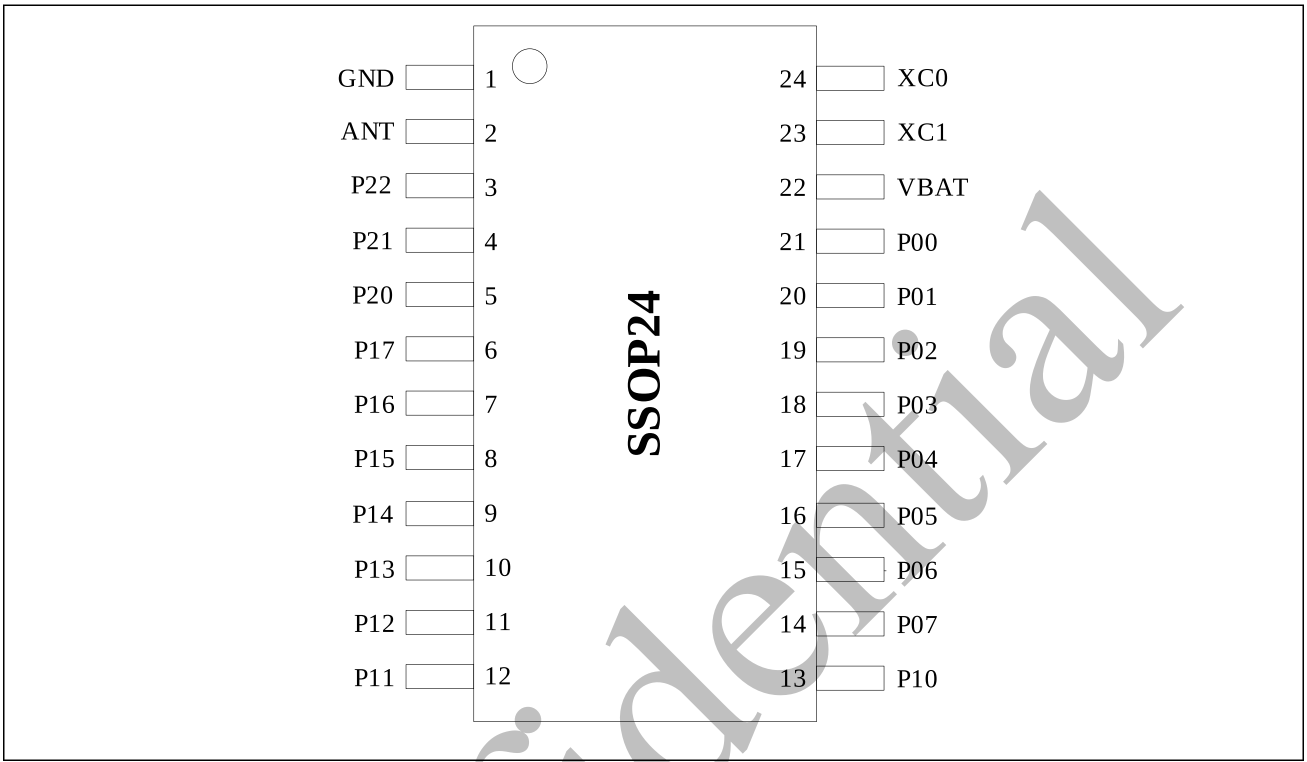


Figure 2-1 SSOP24 Diagram

Table 2-1 Pin Descriptions for SSOP24

Pin No. SSOP24	Pin Name	Pin Type	Description
1	GND	P	Ground (VSS)
2	ANT	AI/AO	RF antenna , an external antenna is required for use.
3(1)	P22	I/O	General-purpose digital input and output (The pull-up voltage must not exceed VBAT.)
	VPP	AI	OTP program VPP (6.25V~6.75V). Support input and open-drain output only. When used as an output pin, the IO port must be connected with a pull-up or pull-down resistor.
	KS_I0	I	Channel 0 keyscan input
	UART1_RX	I	UART1 RX
	EXT_STADC	I	ADC External pin trigger
	EXT_MEAS_CLK	I	External measurement clock
	TM0_EXT	I	Timer0 external input
	UART0_RX	I	UART0 RX

4	P21	I/O	General-purpose digital input and output
	ADC_CH21	AI	Channel 21 ADC input
	KS_O0	O	Channel 0 keyscan output
	UART1_RX	I	UART1 RX
	I2C0_SDA	I/O	I2C0 SDA
	XTL1	AI	External 32.768kHz clock source input
5	P20	I/O	General-purpose digital input and output
	ADC_CH20	AI	Channel 20 ADC input
	KS_I2	I	Channel 2 keyscan input
	UART1_TX	O	UART1 TX
	EXT_MEAS_CLK	I	External measurement clock
	XTL2	AO	External 32.768kHz clock source output
6	P17	I/O	General-purpose digital input and output
	ADC_CH17	AI	Channel 17 ADC input
	UART0_RX	I	UART0 RX
	PWM_CH1	O	Channel 1 PWM output
	KS_O4	O	Channel 4 keyscan output
7	P16	I/O	General-purpose digital input and output
	ADC_CH16	AI	Channel 16 ADC input
	UART0_TX	O	UART0 TX
	KS_I1	I	Channel 1 keyscan input
	PWM_CH0	O	Channel 0 PWM output
	I2C0_SCL	I/O	I2C0 SCL
8	P15	I/O	General-purpose digital input and output
	ADC_CH15	AI	Channel 15 ADC input
	SPI0_CS	I/O	SPI0 CS
	PWM_CH5	O	Channel 5 PWM output
	TM0_EXT	I	Timer0 external input
	UART0_RX	I	UART0 RX
	KS_O5	O	Channel 5 keyscan output
9	P14	I/O	General-purpose digital input and output
	ADC_CH14	AI	Channel 14 ADC input
	KS_O4	O	Channel 4 keyscan output

	I2C0_SCL	I/O	I2C0 SCL
	PWM_CH2	O	Channel 2 PWM output
	SPI0_CLK	I/O	SPI0 clock
	UART1_RX	I	UART1 RX
10	P13	I/O	General-purpose digital input and output
	ADC_CH13	AI	Channel 13 ADC input
	KS_O3	O	Channel 3 keyscan output
	I2C0_SDA	I/O	I2C0 SDA
	PWM_CH3	O	Channel 3 PWM output
	UART1_TX	O	UART1 TX
	SPI0_CS	I/O	SPI0 CS
11	P12	I/O	General-purpose digital input and output
	ADC_CH12	AI	Channel 12 ADC input
	UART0_RX	I	UART0 RX
	PWM_CH4	O	Channel 4 PWM output
	KS_O2	O	Channel 2 keyscan output
	SPI0_MISO	I/O	SPI0 MISO
12	P11	I/O	General-purpose digital input and output
	ADC_CH11	AI	Channel 11 ADC input
	KS_O1	O	Channel 1 keyscan output
	SPI0_MOSI	I/O	SPI0 MOSI
	EXT_MEAS_CLK	I	External measurement clock
	KS_I0	I	Channel 0 keyscan input
	UART0_TX	O	UART0 TX
13	P10	I/O	General-purpose digital input and output
	ADC_CH10	AI	Channel 10 ADC input
	KS_O0	O	Channel 0 keyscan output
	I2C0_SDA	I/O	I2C0 SDA
	SPI0_MISO	I/O	SPI0 MISO
	PWM_CH5	O	Channel 5 PWM output
14	P07	I/O	General-purpose digital input and output
	ADC_CH7	AI	Channel 7 ADC input
	KS_I5	I	Channel 5 keyscan input

	I2C0_SCL	I/O	I2C0 SCL
	SPI0_MOSI	I/O	SPI0 MOSI
	PWM_CH0	O	Channel 0 PWM output
15	P06	I/O	General-purpose digital input and output
	ADC_CH6	AI	Channel 6 ADC input
	KS_I4	I	Channel 4 keyscan input
	UART0_RX	I	UART0 RX
	SPI0_MISO	I/O	SPI0 MISO
	PWM_CH5	O	Channel 5 PWM output
16	P05	I/O	General-purpose digital input and output
	ADC_CH5	AI	Channel 5 ADC input
	KS_I3	I	Channel 3 keyscan input
	UART0_TX	O	UART0 TX
	SPI0_MISO	I/O	SPI0 MISO
	PWM_CH4	O	Channel 4 PWM output
17	P04	I/O	General-purpose digital input and output
	ADC_CH4	AI	Channel 4 ADC input
	KS_I2	I	Channel 2 keyscan input
	SPI0_CLK	I/O	SPI0 clock
	PWM_CH3	O	Channel 3 PWM output
	KS_O3	O	Channel 3 keyscan output
	UART1_RX	I	UART1 RX
18	P03	I/O	General-purpose digital input and output
	ADC_CH3	AI	Channel 3 ADC input
	KS_I1	I	Channel 1 keyscan input
	PWM_CH2	O	Channel 2 PWM output
	SPI0_CS	I/O	SPI0 CS
	UART1_TX	O	UART1 TX
	PIN RESET	I	Reset pin
19	P02	I/O	General-purpose digital input and output
	ADC_CH2	AI	Channel 2 ADC input
	KS_O1	O	Channel 1 keyscan output
	PWM_CH0	O	Channel 0 PWM output

	SPI0_MOSI	I/O	SPI0 MOSI
20[1]	P01	I/O	General-purpose digital input and output
	ADC_CH1	AI	Channel 1 ADC input
	SWD_DAT	I/O	SWD data input and output
	UART0_TX	O	UART0 TX
	I2C0_SDA	I/O	I2C0 SDA
	SPI0_CS	I/O	SPI0 CS
21[1]	P00	I/O	General-purpose digital input and output
	ADC_CH0	AI	Channel 0 ADC input
	SWD_CLK	I	SWD clock input
	UART0_RX	I	UART0 RX
	I2C0_SCL	I/O	I2C0 SCL
	SPI0_CLK	I/O	SPI0 clock
22	VBAT	P	Power input (VDD)
23	XC1	AO	External 32MHz clock source output
24	XC0	AI	External 32MHz clock source input

Note[1]: Pins for programming.

2.2 SOP16 / SOP14 Package

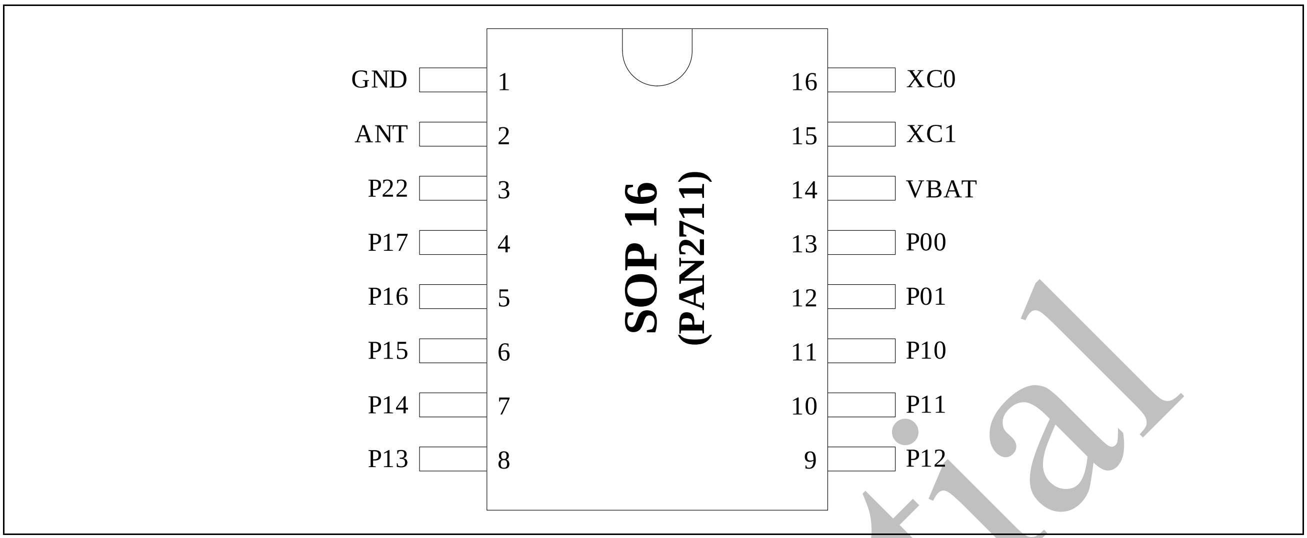


Figure 2-2 SOP16 Diagram (PAN2711P3DA)

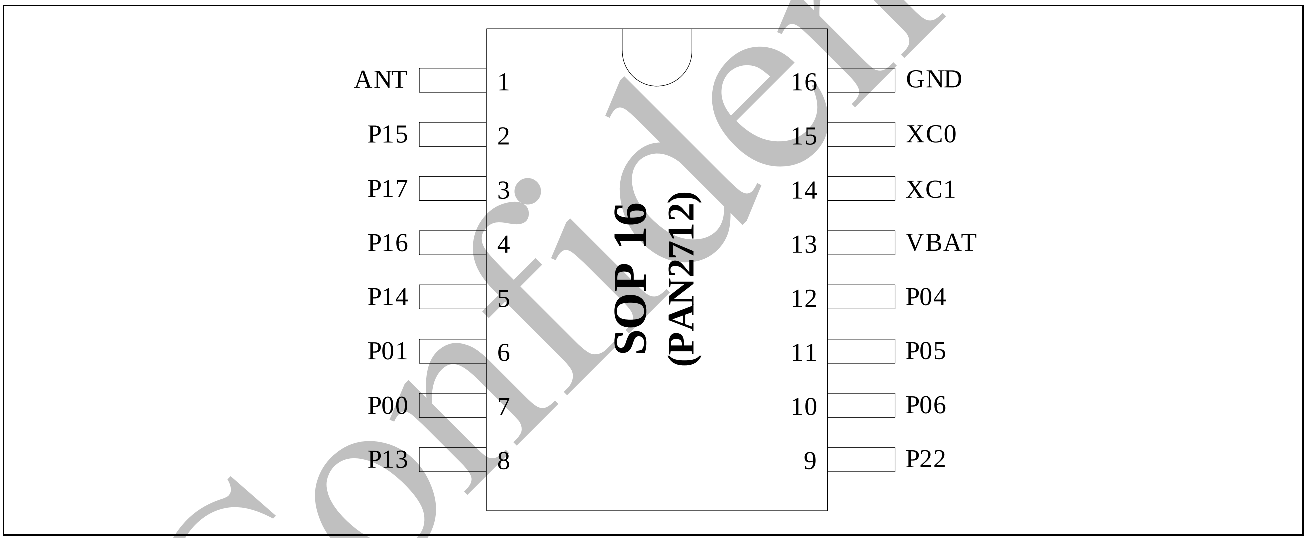


Figure 2-3 SOP16 Diagram (PAN2712P3DA)

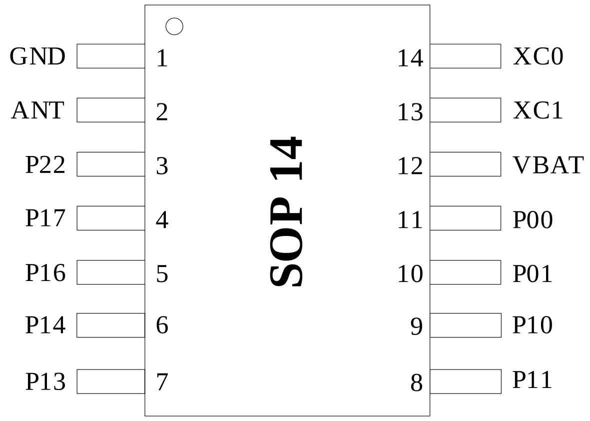


Figure 2-4 SOP14 Diagram

Table 2-2 Pin Descriptions for SOP16 and SOP14

Pin No.			Pin Name	Pin Type	Description
SOP16 (PAN2711)	**SOP16 (PAN2712)**	**SOP14**			
1	16	1	GND	P	Ground (VSS)
2	1	2	ANT	AI/AO	RF antenna , an external antenna is required for use.
3(1)	9(1)	3(1)	P22	I/O	General-purpose digital input and output (The pull-up voltage must not exceed VBAT.)
			VPP	AI	OTP program VPP (6.25V~6.75V). Support input and open-drain output only. When used as an output pin, the IO port must be connected with a pull-up or pull-down resistor.
			KS_I0	I	Channel 0 keyscan input
			UART1_RX	I	UART1 RX
			EXT_STADC	I	ADC External pin trigger
			EXT_MEAS_CLK	I	External measurement clock
			TM0_EXT	I	Timer0 external input
			UART0_RX	I	UART0 RX
4	3	4	P17	I/O	General-purpose digital input and output
			ADC_CH17	AI	Channel 17 ADC input
			UART0_RX	I	UART0 RX
			PWM_CH1	O	Channel 1 PWM output
			KS_O4	O	Channel 4 keyscan output
5	4	5	P16	I/O	General-purpose digital input and output
			ADC_CH16	AI	Channel 16 ADC input

			UART0_TX	O	UART0 TX
			KS_I1	I	Channel 1 keyscan input
			PWM_CH0	O	Channel 0 PWM output
			I2C0_SCL	I/O	I2C0 SCL
6	2	-	P15	I/O	General-purpose digital input and output
			ADC_CH15	AI	Channel 15 ADC input
			SPI0_CS	I/O	SPI0 CS
			PWM_CH5	O	Channel 5 PWM output
			TM0_EXT	I	Timer0 external input
			UART0_RX	I	UART0 RX
			KS_O5	O	Channel 5 keyscan output
7	5	6	P14	I/O	General-purpose digital input and output
			ADC_CH14	AI	Channel 14 ADC input
			KS_O4	O	Channel 4 keyscan output
			I2C0_SCL	I/O	I2C0 SCL
			PWM_CH2	O	Channel 2 PWM output
			SPI0_CLK	I/O	SPI0 clock
			UART1_RX	I	UART1 RX
8	8	7	P13	I/O	General-purpose digital input and output
			ADC_CH13	AI	Channel 13 ADC input
			KS_O3	O	Channel 3 keyscan output
			I2C0_SDA	I/O	I2C0 SDA
			PWM_CH3	O	Channel 3 PWM output
			UART1_TX	O	UART1 TX
			SPI0_CS	I/O	SPI0 CS
9	-	-	P12	I/O	General-purpose digital input and output
			ADC_CH12	AI	Channel 12 ADC input
			UART0_RX	I	UART0 RX
			PWM_CH4	O	Channel 4 PWM output
			KS_O2	O	Channel 2 keyscan output
			SPI0_MISO	I/O	SPI0 MISO
10	-	8	P11	I/O	General-purpose digital input and output
			ADC_CH11	AI	Channel 11 ADC input

			KS_O1	O	Channel 1 keyscan output
			SPI0_MOSI	I/O	SPI0 MOSI
			EXT_MEAS_CLK	I	External measurement clock
			KS_I0	I	Channel 0 keyscan input
			UART0_TX	O	UART0 TX
11	-	9	P10	I/O	General-purpose digital input and output
			ADC_CH10	AI	Channel 10 ADC input
			KS_O0	O	Channel 0 keyscan output
			I2C0_SDA	I/O	I2C0 SDA
			SPI0_MISO	I/O	SPI0 MISO
			PWM_CH5	O	Channel 5 PWM output
12(1)	6(1)	10(1)	P01	I/O	General-purpose digital input and output
			ADC_CH1	AI	Channel 1 ADC input
			SWD_DAT	I/O	SWD data input and output
			UART0_TX	O	UART0 TX
			I2C0_SDA	I/O	I2C0 SDA
			SPI0_CS	I/O	SPI0 CS
13(1)	7(1)	11(1)	P00	I/O	General-purpose digital input and output
			ADC_CH0	AI	Channel 0 ADC input
			SWD_CLK	I	SWD clock input
			UART0_RX	I	UART0 RX
			I2C0_SCL	I/O	I2C0 SCL
			SPI0_CLK	I/O	SPI0 clock
14	13	12	VBAT	P	Power input (VDD)
15	14	13	XC1	AO	External 32MHz clock source output
16	15	14	XC0	AI	External 32MHz clock source input
-	10	-	P06	I/O	General-purpose digital input and output
			ADC_CH6	AI	Channel 6 ADC input
			UART0_RX	I	UART0 RX
			SPI0_MISO	I/O	SPI0 MISO
			PWM_CH5	O	Channel 5 PWM output
			KS_I4	I	Channel 4 keyscan input
-	11	-	P05	I/O	General-purpose digital input and output

			ADC_CH5	AI	Channel 5 ADC input
			KS_I3	I	Channel 3 keyscan input
			UART0_TX	O	UART0 TX
			SPI0_MISO	I/O	SPI0 MISO
			PWM_CH4	O	Channel 4 PWM output
-	12	-	P04	I/O	General-purpose digital input and output
			ADC_CH4	AI	Channel 4 ADC input
			KS_I2	I	Channel 2 keyscan input
			SPI0_CLK	I/O	SPI0 clock
			PWM_CH3	O	Channel 3 PWM output
			KS_O3	O	Channel 3 keyscan output
			UART1_RX	I	UART1 RX

Note(1): Pins for programming.

2.3 MSOP10 Package

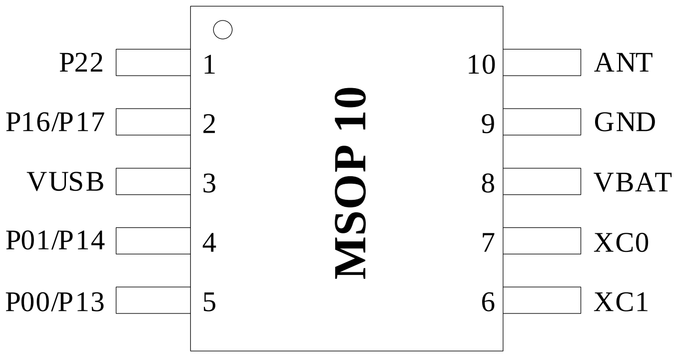


Figure 2-5 MSOP10 Diagram

Table 2-3 MSOP10 Pin Descriptions

Pin No. MSOP10	Pin Name	Pin Type	Description
1(1)	P22	I/O	General-purpose digital input and output (The pull-up voltage must not exceed VBAT.)
	VPP	AI	OTP program VPP (6.25V~6.75V). Support input and open-drain output only. When used as an output pin, the IO port must be connected with a pull-up or pull-down resistor.
	KS_I0	I	Channel 0 keyscan input
	UART1_RX	I	UART1 RX
	EXT_STADC	I	ADC External pin trigger
	EXT_MEAS_CLK	I	External measurement clock
	TM0_EXT	I	Timer0 external input
	UART0_RX	I	UART0 RX
2(2)	**P16**	**I/O**	**General-purpose digital input and output**
	ADC_CH16	AI	Channel 16 ADC input
	UART0_TX	O	UART0 TX
	KS_I1	I	Channel 1 keyscan input
	PWM_CH0	O	Channel 0 PWM output
	I2C0_SCL	I/O	I2C0 SCL
	P17	**I/O**	**General-purpose digital input and output**
	ADC_CH17	AI	Channel 17 ADC input
	UART0_RX	I	UART0 RX
	PWM_CH1	O	Channel 1 PWM output

	KS_O4	O	Channel 4 keyscan output
3	VUSB	P	USB 5V input
4(2)	**P01(1)**	**I/O**	**General-purpose digital input and output**
	ADC_CH1	AI	Channel 1 ADC input
	SWD_DAT	I/O	SWD data input and output
	UART0_TX	O	UART0 TX
	I2C0_SDA	I/O	I2C0 SDA
	SPI0_CS	I/O	SPI0 CS
	P14	**I/O**	**General-purpose digital input and output**
	ADC_CH14	AI	Channel 14 ADC input
	KS_O4	O	Channel 4 keyscan output
	I2C0_SCL	I/O	I2C0 SCL
	PWM_CH2	O	Channel 2 PWM output
	SPI0_CLK	I/O	SPI0 clock
	UART1_RX	I	UART1 RX
	USB_DP	AI/AO	USB dp
5(2)	**P00(1)**	**I/O**	**General-purpose digital input and output**
	ADC_CH0	AI	Channel 0 ADC input
	SWD_CLK	I	SWD clock input
	UART0_RX	I	UART0 RX
	I2C0_SCL	I/O	I2C0 SCL
	SPI0_CLK	I/O	SPI0 clock
	P13	**I/O**	**General-purpose digital input and output**
	ADC_CH13	AI	Channel 13 ADC input
	KS_O3	O	Channel 3 keyscan output
	I2C0_SDA	I/O	I2C0 SDA
	PWM_CH3	O	Channel 3 PWM output
	UART1_TX	O	UART1 TX
	SPI0_CS	I/O	SPI0 CS
	USB_DM	AI/AO	USB dm
6	XC1	AO	External 32MHz clock source output
7	XC0	AI	External 32MHz clock source input
8	VBAT	P	Power input (VDD)

9	GND	P	Ground (VSS)
10	ANT	AI/AO	RF antenna , an external antenna is required for use.

Note(1): Pins for programming.

Note(2): This pin has two sets of pads. You can choose any set of pads, but the other set must be in analog state.

2.4 Internal Connection

Table 2-4 Internal Connection for PAN2713M5BA

Pin Status	EEPROM	RF
I S	PAD_SDA	P10
I S	PAD_SCL	P07

3 Function Description

3.1 System Manager

3.1.1 Overview

System management includes the following sections:

- System Memory Map and description
- System Timer (SysTick)
- Nested Vectored Interrupt Controller (NVIC)
- System Control registers (SCB)

3.1.2 Memory Organization

3.1.2.1 Overview

The PAN271x series provides 4G-byte addressing space. The addressing space assigned to each on-chip controllers is shown in the Table 3-1. The detailed register definition, addressing space, and programming details will be described in the following sections for each on-chip peripheral. The PAN271x series only supports little-endian data format.

3.1.2.2 System Memory Map

The memory locations assigned to each on-chip controllers are shown in the Table 3-1.

Address	Peripheral	BUS	
0xE0000000-0xE00FFFFF	M0 (M0 core registers)	-	
0x50020000-0x5003FFFF	MAC	AHB	
0x400A0000-0x400AFFFF	USB	-	
0x40080000-0x4008FFFF	Chip_Security	AHB	
0x40070000-0x4007FFFF	ANACTL		
0x40050000-0x4005FFFF	OTP_CTRL		
0x40040000-0x4004FFFF	RCC		
0x40030000-0x4003FFFF	SYSTEM		
0x40020000-0x4002FFFF	GPIO		
0x4000A000-0x4000AFFF	UART1	AHB	APB1
0x40009000-0x40009FFF	KeyScan		
0x40008000-0x40008FFF	TIMER		
0x40007000-0x40007FFF	CLKMEAS		
0x40006000-0x40006FFF	WDT		
0x40005000-0x40005FFF	ADC		
0x40004000-0x40004FFF	PWM		
0x40003000-0x40003FFF	UART0		
0x40001000-0x40001FFF	SPI		

0x40000000-0x40000FFF	I2C		
0x20000000-0x2000FFFF	SRAM (2+1KB)	-	-
0x00000000-0x007FFFFF	OTP (16KB)	-	-

Table 3-1 Address Space Assignments for On-Chip Modules

3.1.3 System Register Map

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

Register	Offset	R/W	Description	Reset Value
SYS Base Address: SYS_BA = 0x4003_0000				
SYS_P0_MFP	SYS_BA+0x00	R/W	P0 Multiple Function and Input Type Control Register	0x0000_0003
SYS_P1_MFP	SYS_BA+0x04	R/W	P1 Multiple Function and Input Type Control Register	0x0000_0000
SYS_P2_MFP	SYS_BA+0x08	R/W	P2 Multiple Function and Input Type Control Register	0x0000_0000
SYS_REGCTRL	SYS_BA+0x0C	R/W	Register Write-Protection Control Register	0x0000_0000
SYS_CTRL	SYS_BA+0x10	R/W	System Control Register	0x0000_0004
ANA_RFLDO	SYS_BA+0x14	R/W	ADCLDO control	0x0000_2080
ANA_DFT	SYS_BA+0x18	R/W	Analog DFT control	0x0000_0000

3.1.4 System Register Description

3.1.4.1 SYS_P0_MFP

Register	Offset	R/W	Description	Reset Value
SYS_P0_MFP	SYS_BA+0x00	R/W	P0 Multiple Function and Input Type Control Register	0x0000_0003

Bits	Description	
[31:24]	Reserved	Reserved.
[23:16]	EXT[7:0]	P0[7:0] Alternate Function Selection Extension The pin function of P0 depends on EXT, MFP and ALT.
[15:8]	ALT[7:0]	P0[7:0] Alternate Function Select Bit The pin function of P0 depends on EXT, MFP and ALT.
[7:2]	Reserved	Reserved.
[1]	MFP [1]	P0.1 Alternate Function Select Bit Bits EXT[1] (SYS_P0_MFP[17]), ALT[1] (SYS_P0_MFP[9]), and MFP[1] (SYS_P0_MFP[1]) determine the P0.1 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = swd_dat function is selected. (0, 1, 0) = uart0_tx function is selected. (0, 1, 1) = i2c0_sda function is selected. (1, 0, 0) = spi0_cs function is selected. (1, 1, 1) = Reserved.
[0]	MFP [0]	P0.0 Alternate Function Select Bit Bits EXT[0] (SYS_P0_MFP[16]), ALT[0] (SYS_P0_MFP[8]), and MFP[0] (SYS_P0_MFP[0]) determine the P0.0 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = swd_clk function is selected. (0, 1, 0) = uart0_rx function is selected. (0, 1, 1) = i2c0_scl function is selected. (1, 0, 0) = spi0_clk function is selected. (1, 1, 1) = Reserved.

3.1.4.2 SYS_P1_MFP

Register	Offset	R/W	Description	Reset Value
SYS_P1_MFP	SYS_BA+0x04	R/W	P1 Multiple Function and Input Type Control Register	0x0000_0000

Bits	Description	
[31:24]	Reserved	Reserved.
[23:16]	EXT[7:0]	P1[7:0] Alternate Function Selection Extension The pin function of P1 depends on EXT, MFP and ALT.
[15:8]	ALT[7:0]	P1[7:0] Alternate Function Select Bit The pin function of P1 depends on EXT, MFP and ALT.
[7]	MFP [7]	P1.7 Alternate Function Select Bit Bits EXT[7] (SYS_P1_MFP[23]), ALT[7] (SYS_P1_MFP[15]), and MFP[7] (SYS_P1_MFP[7]) determine the P1.7 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = uart0_rx function is selected. (0, 1, 1) = pwm_ch1 function is selected. (1, 0, 0) = ks_o4 function is selected.
[6]	MFP [6]	P1.6 Alternate Function Select Bit Bits EXT[6] (SYS_P1_MFP[22]), ALT[6] (SYS_P1_MFP[14]), and MFP[6] (SYS_P1_MFP[6]) determine the P1.6 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = uart0_tx function is selected. (0, 1, 0) = ks_i1 function is selected. (0, 1, 1) = pwm_ch0 function is selected. (1, 1, 1) = i2c0_scl function is selected.
[5]	MFP [5]	P1.5 Alternate Function Select Bit Bits EXT[5] (SYS_P1_MFP[21]), ALT[5] (SYS_P1_MFP[13]), and MFP[5] (SYS_P1_MFP[5]) determine the P1.5 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = spi0_cs function is selected. (0, 1, 0) = pwm_ch5 function is selected. (0, 1, 1) = tm0_ext function is selected. (1, 0, 0) = uart0_rx function is selected. (1, 1, 1) = ks_o5 function is selected.
[4]	MFP [4]	P1.4 Alternate Function Select Bit Bits EXT[4] (SYS_P1_MFP[20]), ALT[4] (SYS_P1_MFP[12]), and MFP[4] (SYS_P1_MFP[4]) determine the P1.4 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = ks_o4 function is selected. (0, 1, 0) = i2c0_scl function is selected. (0, 1, 1) = pwm_ch2 function is selected. (1, 1, 0) = spi0_clk function is selected. (1, 1, 1) = uart1_rx function is selected.

[3]	MFP [3]	P1.3 Alternate Function Select Bit Bits EXT[3] (SYS_P1_MFP[19]), ALT[3] (SYS_P1_MFP[11]), and MFP[3] (SYS_P1_MFP[3]) determine the P1.3 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = ks_o3 function is selected. (0, 1, 0) = i2c0_sda function is selected. (0, 1, 1) = pwm_ch3 function is selected. (1, 1, 0) = uart1_tx function is selected. (1, 1, 1) = spi0_cs function is selected.
[2]	MFP [2]	P1.2 Alternate Function Select Bit Bits EXT[2] (SYS_P1_MFP[18]), ALT[5] (SYS_P1_MFP[10]), and MFP[2] (SYS_P1_MFP[2]) determine the P1.2 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = uart0_rx function is selected. (0, 1, 0) = pwm_ch4 function is selected. (0, 1, 1) = Reserved. (1, 0, 0) = ks_o2 function is selected. (1, 1, 1) = spi0_miso function is selected.
[1]	MFP [1]	P1.1 Alternate Function Select Bit Bits EXT[1] (SYS_P1_MFP[17]), ALT[1] (SYS_P1_MFP[9]), and MFP[1] (SYS_P1_MFP[1]) determine the P1.1 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = ks_o1 function is selected. (0, 1, 0) = spi0_mosi function is selected. (0, 1, 1) = ext_meas_clk function is selected. (1, 0, 1) = ks_i0 function is selected. (1, 1, 1) = uart0_tx function is selected.
[0]	MFP [0]	P1.0 Alternate Function Select Bit Bits EXT[0] (SYS_P1_MFP[16]), ALT[0] (SYS_P1_MFP[8]), and MFP[0] (SYS_P1_MFP[0]) determine the P1.0 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = ks_o0 function is selected. (0, 1, 0) = i2c0_sda function is selected. (0, 1, 1) = spi0_miso function is selected. (1, 0, 0) = pwm_ch5 function is selected.

3.1.4.3 SYS_P2_MFP

Register	Offset	R/W	Description	Reset Value
SYS_P2_MFP	SYS_BA+0x08	R/W	P2 Multiple Function and Input Type Control Register	0x0000_0000

Bits	Description	
[31:24]	Reserved	Reserved.
[23:16]	EXT[7:0]	P2[7:0] Alternate Function Selection Extension The pin function of P2 depends on EXT, MFP and ALT.
[15:8]	ALT[7:0]	P2[7:0] Alternate Function Select Bit The pin function of P2 depends on EXT, MFP and ALT.
[7:3]	MFP [7:3]	Reserved.
[2]	MFP [2]	P2.2 Alternate Function Select Bit Bits EXT[2] (SYS_P2_MFP[18]), ALT[5] (SYS_P2_MFP[10]), and MFP[2] (SYS_P2_MFP[2]) determine the P2.2 function. (0, 0, 0) = GPIO function is selected. (0, 0, 1) = ks_i0 function is selected. (0, 1, 0) = uart1_rx function is selected. (0, 1, 1) = ext_stadc function is selected. (1, 0, 0) = ext_meas_clk function is selected. (1, 0, 1) = tm0_ext function is selected. (1, 1, 0) = uart0_rx function is selected.
[1:0]	Reserved	Reserved.

3.1.4.4 SYS_REGCTRL

Register	Offset	R/W	Description	Reset Value
SYS_REGCTRL	SYS_BA+0x0C	R/W	Register Write-Protection Control Register	0x0000_0000

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:8]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[7:0]</td><td>REGCTRL</td><td>Register Write-protection Code (Write Only)
Some registers have write-protection function. Writing these registers have to disable the protected function by writing the sequence value 0x59, 0x16, 0x88 to this field. After this sequence is completed, the REGLCTL bit will be set to 1 and write-protection registers can be normal write.
Register Write-protection Disable Index (Read Only)
0 : Write-protection Enabled for writing protected registers. Any write to the protect-ed register is ignored.
1 : Write-protection Disabled for writing protected registers.
Protected registers are listed below:
<table>
<tr><td>Register</td><td>Address</td><td>Note</td></tr>
<tr><td>WDT_CTL</td><td>WDT_BA+0x00</td><td></td></tr>
<tr><td>WDT_ALTCTL</td><td>WDT_BA+0x04</td><td></td></tr>
<tr><td>IPRST0</td><td>RCC_BA+0x04</td><td></td></tr>
<tr><td>EFUSE_CTL</td><td>EF_BA+0x00</td><td></td></tr>
<tr><td>EFUSE_TRG</td><td>EF_BA+0x1C</td><td></td></tr>
<tr><td>EFUSE_FLASH_PERMISSION</td><td>EF_BA+0x78</td><td></td></tr>
</table>
Note: The bits which are write-protected will be noted as” (Write Protect)” beside the description.</td></tr>
</table>

3.1.4.5 SYS_CTRL

Register	Offset	R/W	Description	Reset Value
SYS_CTRL	SYS_BA+0x10	R/W	System Control Register	0x0000_0004

Bits	Description	
[31:27]	Reserved	Reserved.
[26]	TMR_CAPT_SEL	The source of timer capture select 0: mac_acc_match 1: usb_sof Note: When tm0_ext is selected, this bit is nouse
[25]	CLK_OUT_SEL	The function selected of apb1_clk 0: apb1_clk is selected 1: clk_32k is selected
[24]	MDM_DEBUG_SEL	mdm_dbueg sel control 0: mdm_debug are selected as other func 1: mdm_debug are selected adc iq data
[23:19]	Reserved	Reserved.
[18]	USB_PU2	USB PAD pullup2 enable, 150K
[17]	USB_PU	USB PAD pullup enable1, 1.5K
[16]	USB_EN	USB PAD enable control
[15:0]	VALID_REMOVAL_CYCLES	For usb insert detection usage When USB_PU2 is set to 1, after the USB device is removed, both USB_DP and USB_DM are at high level. The removal of the USB device can be detected by DIP=DIM=1. This register is used to configure the debounce time for removal detection, in 48MHz cycles.

3.1.4.6 ANA_RFLDO

Register	Offset	R/W	Description	Reset Value
ANA_RFLDO	SYS_BA+0x14	R/W	ADCLDO control	0x0000_2080

Bits	Description	
[31:14]	Reserved	Reserved.
[13:10]	LDO_ADC_TRIM	The output voltage trim of LDO ADC 4‘b0000 0.986V 4’b1000 1.2V 4‘b1111 1.38V
[9:8]	ADCLDO_LOAD_SEL	ADCLDO load selection 00: High resistance 01: 2K 11: 1K
[7:4]	LDO_RFFE_TRIM	The output voltage trim of LDO RFFE 4‘b0000: 0.986V 4’b1000: 1.2V 4‘b1111: 1.38V
[3:1]	Reserved	Reserved.
[0]	EN_LDO_ADC	The enable control RXADC LDO 1: Enable 0: Disable

3.1.4.7 ANA_DFT

Register	Offset	R/W	Description	Reset Value
ANA_DFT	SYS_BA+0x18	R/W	Analog DFT control	0x0000_0000

Bits	Description	
[31:15]	Reserved	Reserved.
[14]	EN_DFT_V	Enable DFT voltage output mode
[13:11]	Reserved	Reserved.
[10:7]	DFT_V_SEL	Registers for selecting proper output voltage
[6:1]	Reserved	Reserved.
[0]	DFT_V_BYPASS	Register for DFT voltage output mode buffer bypass function

3.1.5 System Timer (SysTick)

The PAN271x series MCU includes an integrated system timer, SysTick, which provides a simple, 24-bit clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. The counter can be used as a Real Time Operating System (RTOS) tick timer or as a simple counter.

When system timer is enabled, it will count down from the value in the SysTick Current Value Register (SYST_VAL) to zero, and reload (wrap) to the value in the SysTick Reload Value Register (SYST_LOAD) on the next clock edge, and then decrement on subsequent clocks. When the counter transitions to zero, the COUNTFLAG status bit is set. The COUNTFLAG bit clears on reads.

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

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

3.1.5.1 System Timer Control Register Map

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

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

3.1.5.2 System Timer Control Register Description

SysTick Control and Status Register (SYST_CTRL)

Register	Offset	R/W	Description	Reset Value
SYST_CTRL	SCS_BA+0x10	R/W	SysTick Control and Status Register	0x0000_0000

Bits	Description	
[31:17]	Reserved	Reserved.
[16]	COUNTFLAG	System Tick Counter Flag Returns 1 if timer counted to 0 since last time this register was read. COUNTFLAG is set by a count transition from 1 to 0. COUNTFLAG is cleared on read or by a write to the Current Value register.
[15:3]	Reserved	Reserved.
[2]	CLKSRC	System Tick Clock Source Select Bit 0 : Clock source is optional, refer to STCLKSEL. 1 : Core clock used for SysTick timer.
[1]	TICKINT	System Tick Interrupt Enable Bit 0 : Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to 0 has occurred. 1 : Counting down to 0 will cause the SysTick exception to be pended. Clearing the SysTick Current Value register by a register write in software will not cause SysTick to be pended.
[0]	ENABLE	System Tick Counter Enable Bit 0 : Counter Disabled. 1 : Counter Enabled and will operate in a multi-shot manner.

SysTick Reload Value Register (SYST_LOAD)

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

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

SysTick Current Value Register (SYST_VAL)

Register	Offset	R/W	Description	Reset Value
SYST_VAL	SCS_BA+0x18	R/W	SysTick Current Value Register	0x00XX_XXXX

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

3.1.6 Nested Vectored Interrupt Controller (NVIC)

3.1.6.1 Overview

The PAN271x series MCU provides an interrupt controller as an integral part of the exception mode, named as “Nested Vectored Interrupt Controller (NVIC)”, which is closely coupled to the processor core and provides following features.

3.1.6.2 Features

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

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

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

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

3.1.6.3 Exception Model and System Interrupt Map

The following table lists the exception model supported by NuMicro® Mini58 series. Software can set four levels of priority on some of these exceptions as well as on all interrupts. The highest user-configurable priority is denoted as 0 and the lowest priority is denoted as 3. The default priority of all the user-configurable interrupts is 0. Note that the priority 0 is treated as the fourth priority on the system, after three system exceptions "Reset", "NMI" and "Hard Fault".

Table 3-2 Exception Model

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

Table 3-3 System Interrupt Map Vector Table

Exception Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Source Module	Interrupt Description
1 ~ 15	31	SLPTMR1	SLPTMR1	SLPTMR1 interrupt
16	30	LP	PMU	LP interrupt
17	29	SLPTMR	SLPTMR	SLPTMR interrupt
18	28	BOD	BOD	BOD interrupt
19	27	Reserved	-	-
20	26	Reserved	-	-
21	25	Reserved	-	-
22	24	Reserved	-	-
23	23	Reserved	-	-
24	22	USB	USB	USB interrupt
25	21	Reserved	-	-
26	20	Reserved	-	-
27	19	Reserved	-	-
28	18	Reserved	-	-
29	17	GPIO_P2	GPIO_P2	GPIO_P2 interrupt
30	16	GPIO_P1	GPIO_P1	GPIO_P1 interrupt
31	15	GPIO_P0	GPIO_P0	GPIO_P0 interrupt
32	14	Reserved	-	-
33	13	Reserved	-	-

34	12	Reserved	-	-
35	11	MAC	MAC	MAC interrupt
36	10	UART1	UART1	UART1 interrupt
37	9	KeyScan	KeyScan	KeyScan interrupt
38	8	TIMER	TIMER	Timer interrupt
39	7	CLKMEAS	CLKMEAS	CLKMEAS interrupt
40	6	WDT	WDT	WDT interrupt
41	5	ADC	ADC	ADC interrupt
42	4	PWM	PWM	PWM interrupt
43	3	UART0	UART0	UART0 interrupt
44	2	Reserved	-	-
45	1	SPI	SPI	SPI interrupt
46	0	I2C	I2C	I2C interrupt

3.1.6.4 Vector Table

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

Table 3-4 Vector Table Format

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

3.1.6.5 Operation Description

NVIC interrupts can be enabled and disabled by writing to their corresponding Interrupt Set-Enable or Interrupt Clear-Enable register bit-field. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current enabled state of the corresponding interrupts. When an interrupt is disabled, interrupt assertion will cause the interrupt to become Pending; however, the interrupt will not be activated. If an interrupt is Active when it is disabled, it remains in its Active state until cleared by reset or an exception return. Clearing the enable bit prevents new activations of the associated interrupt.

NVIC interrupts can be pended/un-pended using a complementary pair of registers to those used to enable/disable the interrupts, named the Set-Pending Register and Clear-Pending Register respectively. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current pended state of the corresponding interrupts. The Clear-Pending Register has no effect on the execution status of an Active interrupt.

NVIC interrupts are prioritized by updating an 8-bit field within a 32-bit register (each register supporting four interrupts).

The general registers associated with the NVIC are all accessible from a block of memory in the System Control Space and will be described in next section.

3.1.6.6 NVIC Control Register Map

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

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

3.1.6.7 NVIC Control Register Description

IRQ0 ~ IRQ31 Set-Enable Control Register (NVIC_ISER)

Register	Offset	R/W	Description	Reset Value
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0 ~ IRQ31 Set-Enable Control Register	0x0000_0000

Bits	Description	
[31:0]	SETENA	Interrupt Enable Bits Enable one or more interrupts. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Write Operation: 0 : No effect. 1 : Write 1 to enable associated interrupt. Read Operation: 0 : Associated interrupt status Disabled. 1 : Associated interrupt status Enabled. **Note:** Read value indicates the current enable status.

IRQ0 ~ IRQ31 Clear-Enable Control Register (NVIC_ICER)

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

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

IRQ0 ~ IRQ31 Set-Pending Control Register (NVIC_ISPR)

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0 ~ IRQ31 Set-Pending Control Register	0x0000_0000

Bits	Description	
[31:0]	SETPEND	Set Interrupt Pending Bits Write Operation: 0 : No effect. 1 : Write 1 to set pending state. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Read Operation:

		0 : Associated interrupt in not in pending status. 1 : Associated interrupt is in pending status. **Note:** Read value indicates the current pending status.

IRQ0 ~ IRQ31 Clear-Pending Control Register (NVIC_ICPR)

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

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

IRQ0 ~ IRQ3 Interrupt Priority Register (NVIC_IPR0)

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

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

IRQ4 ~ IRQ7 Interrupt Priority Register (NVIC_IPR1)

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

Bits	Description	
[31:30]	PRI_7	Priority of IRQ7 0 denotes the highest priority and 3 denotes the lowest priority.

[29:24]	Reserved	Reserved.
[23:22]	PRI_6	Priority of IRQ6 0 denotes the highest priority and 3 denotes the lowest priority.
[21:16]	Reserved	Reserved.
[15:14]	PRI_5	Priority of IRQ5 0 denotes the highest priority and 3 denotes the lowest priority.
[13:8]	Reserved	Reserved.
[7:6]	PRI_4	Priority of IRQ4 0 denotes the highest priority and 3 denotes the lowest priority.
[5:0]	Reserved	Reserved.

IRQ8 ~ IRQ11 Interrupt Priority Register (NVIC_IPR2)

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

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

IRQ12 ~ IRQ15 Interrupt Priority Register (NVIC_IPR3)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12 ~ IRQ15 Interrupt Priority Control Register	0x0000_0000

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

		0 denotes the highest priority and 3 denotes the lowest priority.
[5:0]	Reserved	Reserved.

IRQ16 ~ IRQ19 Interrupt Priority Register (NVIC_IPR4)

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

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

IRQ20 ~ IRQ23 Interrupt Priority Register (NVIC_IPR5)

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

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

IRQ24 ~ IRQ27 Interrupt Priority Register (NVIC_IPR6)

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

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

IRQ28 ~ IRQ31 Interrupt Priority Register (NVIC_IPR7)

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

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

3.1.6.8 System Control Block Registers (SCB)

The PAN271x series MCU status and operating mode control are managed System Control Block Registers. Including CPUID, PAN271x series MCU interrupt priority and PAN271x series MCU power management can be controlled through these system control registers.

3.1.6.9 System Control Block Register Map

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

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

3.1.6.10 System Control Block Register Description

CPUID Base Register (SCS_CPUID)

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

Bits	Description	
[31:24]	IMPLEMENTER	Implementer Code Implementer code assigned by M6_core.
[23:20]	Reserved	Reserved.
[19:16]	PART	Architecture of the Processor Read as 0xC for M6_core parts.
[15:4]	PARTNO	Part Number of the Processor Read as 0xC20.
[3:0]	REVISION	Revision Number Read as 0x0.

Interrupt Control State Register (SCS_ICSR)

Register	Offset	R/W	Description	Reset Value
SCS_ICSR	SCS_BA+0xD04	R/W	Interrupt Control State Register	0x0000_0000

Bits	Description	
[31]	NMIPENDSET	NMI Set-pending Bit Write Operation: 0 : No effect. 1 : Changes NMI exception state to pending. Read Operation: 0 : NMI exception is not pending. 1 : NMI exception is pending. **Note**: Because NMI is the highest-priority exception, normally the processor enters the NMI exception handler as soon as it detects a write of 1 to this bit. Entering the handler then clears this bit to 0. This means a read of this bit by the NMI exception handler returns 1 only if the NMI signal is reasserted while the processor is executing that handler.
[30:29]	Reserved	Reserved.
[28]	PENDSVSET	PendSV Set-pending Bit Write Operation: 0 : No effect. 1 : Changes PendSV exception state to pending. Read Operation: 0 : PendSV exception is not pending. 1 : PendSV exception is pending. **Note**: Writing 1 to this bit is the only way to set the PendSV exception state to pending.
[27]	PENDSVCLR	PendSV Clear-pending Bit Write Operation: 0 : No effect. 1 : Removes the pending state from the PendSV exception. **Note**: This bit is write-only. To clear the PENDSV bit, you must “write 0 to PENDSVSET and write 1 to PENDSVCLR” at the same time.
[26]	PENDSTSET	SysTick Exception Set-pending Bit Write Operation: 0 : No effect. 1 : Changes SysTick exception state to pending. Read Operation: 0 : SysTick exception is not pending. 1 : SysTick exception is pending.
[25]	PENDSTCLR	SysTick Exception Clear-pending Bit Write Operation: 0 : No effect. 1 : Removes the pending state from the SysTick exception. **Note**: This bit is write-only. When you want to clear PENDST bit, you must “write 0 to PENDSTSET and write 1 to PENDSTCLR” at the same time.
[24]	Reserved	Reserved.
[23]	ISRPREEMPT	Interrupt Preemption Bit If set, a pending exception will be serviced on exit from the debug halt state.

		Note: This bit is read-only.
[22]	ISRPENDING	Interrupt Pending Flag, Excluding NMI and Faults 0 : Interrupt not pending. 1 : Interrupt pending. **Note**: This bit is read-only.
[21]	Reserved	Reserved.
[20:12]	VECTPENDING	Exception Number of the Highest Priority Pending Enabled Exception 0 : No pending exceptions. Non-zero : Exception number of the highest priority pending enabled exception. **Note**: These bits are read-only.
[11:9]	Reserved	Reserved.
[8:0]	VECTACTIVE	Contains the Active Exception Number 0 : Thread mode. Non-zero : Exception number of the currently active exception. **Note**: These bits are read-only.

Application Interrupt and Reset Control Register (SCS_AIRCR)

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

Bits	Description	
[31:16]	VECTORKEY	Register Access Key Write Operation: When writing to this register, the VECTORKEY field need to be set to 0x05FA, otherwise the write operation would be ignored. The VECTORKEY filed is used to prevent accidental write to this register from resetting the system or clearing of the exception status. Read Operation: Read as 0xFA05.
[15:3]	Reserved	Reserved.
[2]	SYSRESETREQ	System Reset Request Writing this bit 1 will cause a reset signal to be asserted to the chip to indicate a reset is requested. The bit is a write only bit and self-clears as part of the reset sequence.
[1]	VECTCLRACTIVE	Exception Active Status Clear Bit Reserved for debug use. When writing to the register, user must write 0 to this bit, otherwise behavior is unpredictable.
[0]	Reserved	Reserved.

System Control Register (SCS_SCR)

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

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

System Handler Priority Register 2 (SCS_SHPR2)

Register	Offset	R/W	Description	Reset Value
SCS_SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000

Bits	Description	
[31:30]	PRSH_11	Priority Of System Handler 11 – SVCall 0 denotes the highest priority and 3 denotes the lowest priority.
[29:0]	Reserved	Reserved.

System Handler Priority Register 3 (SCS_SHPR3)

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

Bits	Description	
[31:30]	PRSH_15	Priority of System Handler 15 – SysTick 0 denotes the highest priority and 3 denotes the lowest priority.
[29:24]	Reserved	Reserved.
[23:22]	PRSH_14	Priority of System Handler 14 – PendSV 0 denotes the highest priority and 3 denotes the lowest priority.
[21:0]	Reserved	Reserved.

3.2 Power Management Unit (PMU)

3.2.1 Overview

The PAN271x series' low power design contains the following function:

- Multiple low-power modes
- Low power timer
- Analog power control
- Analog clock reset control

3.2.2 Power Supply

Table 3-5 Power Supply Illustration

Name	Direction	Voltage(V)	Note
VBAT	I	1.7 to 3.6	System 3V power supply input
DVDD_PAD	O	1.2 (typ)	HLDO output, connect with a capacitor
VBUS	I	5 (typ)	USB 5V input
VBUS_3V	O	3.3V	USB 5V converted to 3.3V output, connected to VBAT
P22 (OTP_VPP)	I	6.25 to 6.75	OTP programming voltage, can be multiplexed as GPIO

3.2.3 Functional Description

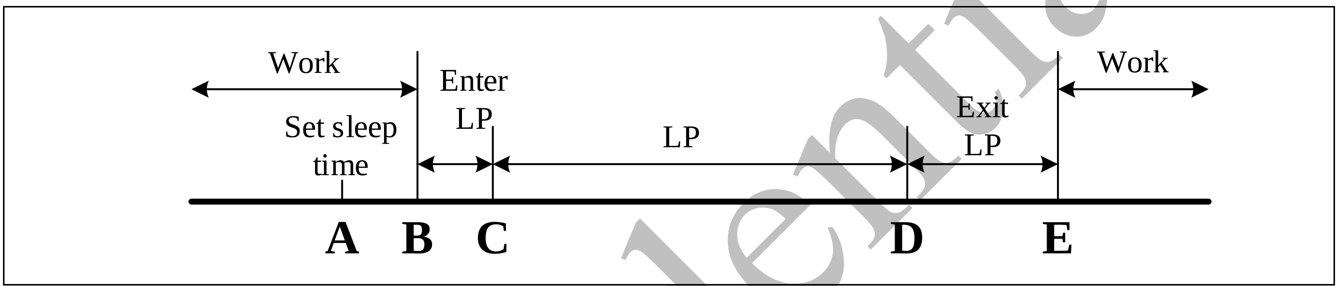
3.2.3.1 LPCTRL

The system has four low-power modes: STANDBY-M0, STANDBY-M1, DEEPSLEEP, and SLEEP. The different modes are described in the table below:

Mode	Entrance	Wake up	Clock	Power
STANDBY_M0	SLEEP_MODE = 3 SW0 = 0 SW1 = 0 LPLDOL_EN = 0 WFI	All GPIO BOD, LVR (optional, and the slow clock need to be on) PIN RESET	All off	LPLDOH: PMU, GPIO, WDT, BOD, LVR
STANDBY_M1	SLEEP_MODE = 2 SW0 = 0 SW1 = 1/0 LPLDOL_EN = 0/1(Based on SW1) WFI	All GPIO (Edge deblurring) SLPTMR WDT BOD, LVR (optional) PIN RESET	Slow clock (the others off)	LPLDOH: PMU, GPIO, WDT, BOD, LVR LPLDOL/LPLDOH: PHY REG, MAC REG, SRAM, RCC REG, SYSTEM REG

DEEPSLEEP	SLEEP_MODE = 1 SW0 = 1 SW1 = 1/0 LPLDOL_EN = 0/1(Based on SW1) WFI	All GPIO SLPTMR WDT BOD, LVR (optional) PIN RESET	Slow clock (the others off)	LPLDOH: PMU, GPIO, WDT, BOD, LVR LPLDOL/LPLDOH: ALL OTHER DIG MODULES
SLEEP	SLEEP_MODE = 0 SW0 = 1 SW1 = 1 LPLDOL_EN = 0/1 WFI	All peripheral interrupts BOD, LVR (optional) PIN RESET	Slow clock CPU_CLK off RCH/XTH/DPLL enabled based on software configuration	HP_LDO: ALL DIG MODULES

The following diagram illustrates the timing sequence for entering and exiting low-power mode:



A. Set the sleep time or other wake-up sources.
B. Send WFI instruction, CPU stops working.
C. Hardware completes the shutdown of the clock power supply module, fully entering low-power mode.
D. The wake source takes effect, and the system begins to power on the clock.
E. System fully awakened, CPU starts working.

3.2.3.2 SLPTMR

SLPTMR is a 32-bit counter, enabled by default and continuously counting. The compare function is disabled by default and must be manually enabled. After counting to the compare value, an interrupt is generated, requiring software to reset the next compare value. There are a total of 2 comparators. Support software loading function. Clock sources include internal 32kHz RC, external 32.768kHz crystal, and ACT 32K (external 32MHz crystal with frequency division).

3.2.3.3 CPU REMAP

The 0-255 addresses of the CPU can be mapped to addresses of other memory in the system. CPU_ADDR_REMAP_EN (CPU_ADDR_REMAP_CTRL[31]) is the enable control signal for this function, while CPU_REMAP_ADDR (CPU_ADDR_REMAP_CTRL[24:0]) specifies the corresponding starting address. This feature is primarily used to map the VECTOR TABLE to RAM or other power-preserving regions where fast execution is required.

3.2.3.4 RESET

The BOD module supports selection of different voltage thresholds and can be configured for either reset or interrupt mode.

Supports disabling the PAD RESET function.

3.2.4 Register Map

Register	Offset	R/W	Description	Reset Value
ANA Base Address: PMU_BA = 0x4007_0000				
LP_CTRL0	PMU_BA+0x00	R/W	Low power control	0x00F4_243C
LP_FLAG_CTRL	PMU_BA+0x04	R/W	Low power interrupt control	0x0080_0000
SLPTMR_CTRL	PMU_BA+0x08	R/W	RCL clock hardware calibration control	0x0000_0001
SLPTMR_CURR	PMU_BA+0x0C	R	SLPTMR value	0x0000_0000
SLPTMR_CMP0	PMU_BA+0x10	R/W	Compare value of SLPTMR	0x0000_0000
SLPTMR_CMP1	PMU_BA+0x14	R/W	Compare value of SLPTMR	0x0000_0000
SLPTMR_LOAD	PMU_BA+0x18	R/W	Compare value of SLPTMR	0x0000_0000
CPU_ADDR_REMAP_CTRL	PMU_BA+0x1C	R/W	Cpu address remap control	0x0000_0000
BLD_CTRL	PMU_BA+0x20	R/W	Brown-out Detector Control Register	0x0000_00CC
SYS_CLK_CTRL	PMU_BA+0x24	R/W	System Clock Control Register	0x0000_4900
RCL_CTRL	PMU_BA+0x28	R/W	RCL Control Register	0x0108_8081
RCH_CTRL	PMU_BA+0x2C	R/W	RCH Control Register	0x0103_4021
XTL_CTRL	PMU_BA+0x30	R/W	XTL Control Register	0x0003_0020
XTH_CTRL	PMU_BA+0x34	R/W	XTH Control Register	0x0000_2A0C
DPLL_CTRL	PMU_BA+0x38	R/W	DPLL Control Register	0x0000_1480
ANA_MISC	PMU_BA+0x3C	R/W	PTAT & POLY & SW control	0xE0F7_0049
ANA_HPLDO	PMU_BA+0x40	R/W	HPLDO control	0x0000_03A5
ANA_LPLDO	PMU_BA+0x44	R/W	LPLDO control	0x00F0_001F
ANA_ANALDO	PMU_BA+0x48	R/W	ANALDO control	0x0000_0081
ANA_RESERVED	PMU_BA+0x4C	R/W	Analog reserved control	0xFFFF_0000

3.2.5 Register Description

3.2.5.1 LP_CTRL0

Register	Offset	R/W	Description	Reset Value
LP_CTRL0	PMU_BA+0x00	R/W	Low power control	0x00F4_243C

Bits	Description		
[31:28]	Reserved		Reserved.
[27]	OTP_LOAD_MASK	R/W	The control of mask OTP load information 0: Mask disable 1: Mask enable Note: Just can config at cpu debug mode. Should delete for customer.
[26:25]	HPLDO_REDUCE_TRIM	R/W	The reduce trim of HPLDO at wakeup
[24:20]	DLY3_TIME	R/W	Used for wait OTP CE (TCRU) ready. Unit 1us Suggest: 0xA
[19:16]	DLY2_TIME	R/W	Used for wait OTP power (TVDC, TVCD) ready. Unit 1us Suggest: 0x2
[15]	SW1_LP_CTRL	R/W	The Sw1 enable control at low power mode 0: Disable 1: Enable
[14]	SW0_LP_CTRL	R/W	The Sw0 enable control at low power mode 0: Disable 1: Enable
[13]	SLOW_CLK_LP_CTRL	R/W	The slow clock enable control at low power mode 0: Disable 1: Enable
[12]	HPLDO_LP_CTRL	R/W	The HPLDO LP mode enable control at low power mode 0: Disable 1: Enable
[11:10]	OTP_LP_CTRL	R/W	The OTP LP mode enable at low power mode 00: PSD,1; CE,1; REEN,1 01: PSD,1; CE,1; REEN,0 10: PSD,0; CE,0; REEN,0
[9:6]	Reserved		Reserved.
[5]	DEBUGREQ_LP_EN	R/W	The debugreq active control at low power mode 0: Debugreq can not hold chip to enter low power 1: Debugreq can hold chip to enter low power
[4]	PMU_ISOLATE_EN	R/W	The isolate enable control between digital PMU and other module at low power mode 0: Disable 1: Enable
[3]	CPU_LP_RET_EN	R/W	Cpu retention enable 0: Disable 1: Enable
[2]	LP_INT_EN	R/W	Low power interrupt enable control

[1:0]	SLEEP_MODE	R/W	Low power mode select 00: Sleep 01: Deepsleep 10: Standby_M1 11: Standby_M0 Active after WFI

3.2.5.2 LP_FLAG_CTRL

Register	Offset	R/W	Description	Reset Value
LP_FLAG_CTRL	PMU_BA+0x04	R/W	Low power interrupt control	0x0080_0000

Bits	Description		
[31:24]	Reserved		Reserved.
[23]	BODOUT	R	Brown-out Detector Output Status 1: Brown-out Detector status output is 0, the detected voltage is higher than BODVL setting. 0: Brown-out Detector status output is 1, the detected voltage is lower than BODVL setting.
[22]	BODIF	R/W	Brown-out Detector Interrupt Flag 0: Brown-out Detector does not detect any voltage draft at VDD down through or up through the voltage of BODVL setting. 1: When Brown-out Detector detects the VDD is dropped through the voltage of BODVL setting or the VDD is raised up through the voltage of BODVL setting, this bit is set to 1 and the Brown-out interrupt is requested if Brown-out interrupt is enabled.
[21]	Reserved		Reserved.
[20]	ACT32K_UP_DONE	R	Active clock update done flag 0: Slow clk is being used 1: Done, active clock is being used
[19:18]	Reserved		Reserved.
[17]	SLPTMR_FLG1	R/W	SLPTMR flag1, write 1 to clear this flag to 0
[16]	SLPTMR_FLG0	R/W	SLPTMR flag0, write 1 to clear this flag to 0
[15:3]	Reserved		Reserved
[2]	STANDBY_M0_FLG	R/W	Standby_m0 mode flag, write 1 to clear this flag to 0
[1]	STANDBY_M1_FLG	R/W	Standby_m1 mode flag, write 1 to clear this flag to 0
[0]	DEEPSLEEP_FLG	R/W	Deepsleep mode flag, write 1 to clear this flag to 0

3.2.5.3 SLPTMR_CTRL

Register	Offset	R/W	Description	Reset Value
SLPTMR_CTRL	PMU_BA+0x08	R/W	RCL clock hardware calibration control	0x0000_0001

Bits	Description		
[31:8]	Reserved		Reserved
[7]	SLPTMR_LOAD_TRG	R/W	The trigger of SLPTMR counter load This bit will be clear by hardware.
[6:5]	Reserved		Reserved
[4]	SLPTMR_CMP1_EN	R/W	SLPTMR compare1 enable control 0: Disable 1: Enable
[3]	SLPTMR_CMP0_EN	R/W	SLPTMR compare0 enable control 0: Disable 1: Enable
[2]	SLPTMR_WK_EN	R/W	SLPTMR wakeup enable control 0: Disable 1: Enable
[1]	SLPTMR_INT_EN	R/W	SLPTMR interrupt enable control
[0]	SLPTMR_EN	R/W	SLPTMR enable control 0: Disable 1: Enable

3.2.5.4 SLPTMR_CURR

Register	Offset	R/W	Description	Reset Value
SLPTMR_CURR	PMU_BA+0x0C	R	SLPTMR value	0x0000_0000

Bits	Description		
[31:0]	CURR_VAL	R	SLPTMR current value

3.2.5.5 SLPTMR_CMP0

Register	Offset	R/W	Description	Reset Value
SLPTMR_CMP0	PMU_BA+0x10	R/W	Compare value of SLPTMR	0x0000_0000

Bits	Description		
[31:0]	CMP_VAL	R/W	The compare value of SLPTMR. Units: 31.2us

3.2.5.6 SLPTMR_CMP1

Register	Offset	R/W	Description	Reset Value
SLPTMR_CMP1	PMU_BA+0x14	R/W	Compare value of SLPTMR	0x0000_0000

Bits	Description		
[31:0]	CMP_VAL	R/W	The compare value of SLPTMR. Units: 31.2us

3.2.5.7 SLPTMR_LODA

Register	Offset	R/W	Description	Reset Value
SLPTMR_LOAD	PMU_BA+0x18	R/W	Update value of SLPTMR	0x0000_0000

Bits	Description		
[31:0]	LOAD_VAL	R/W	The update value of SLPTMR. Units: 31.2us

3.2.5.8 CPU_ADDR_REMAP_CTRL

Register	Offset	R/W	Description	Reset Value
CPU_ADDR_REMAP_CTRL	PMU_BA+0x1C	R/W	Cpu address remap control	0x0000_0000

Bits	Description		
[31]	CPU_ADDR_REMAP_EN	R/W	Cpu address remap enable control 0: Disable 1: Enable, cpu address 0-255Byte will be remap to the releated region define by cpu_remap_addr.
[30:24]	Reserved		Reserved
[23:0]	CPU_REMAP_ADDR	R/W	The start address of remap to cpu address 0.(unit is 256B)

3.2.5.9 BLD_CTRL

Register	Offset	R/W	Description	Reset Value
BLD_CTRL	PMU_BA+0x20	R/W	Brown-out Detector Control Register	0x0000_00CC

Bits	Description		
[31:29]	Reserved		Reserved.
[28:23]	LVR_DB_SEL	R/W	LVR De-Bounce (glitch) time Control register(should enable high voltage sync) [0]=1: about 2^0 SLOW clock [1]=1: about 2^1 SLOW clock [2]=1: about 2^2 SLOW clock [3]=1: about 2^3 SLOW clock [4]=1: about 2^4 SLOW clock [5]=1: about 2^5 SLOW clock(default) Note: If software enables more than one bit, the bit with the smallest number will be selected and the other enabled channels will be ignored.
[22]	PMU_LVR_TEST_EN_AON	R/W	LVR analog test enable control 0: LVR test disable 1: LVR test enable
[21]	PMU_LVR_EN_AON	R/W	LVR analog enable control 0: LVR disable 1: LVR enable
[20:16]	PMU_BOD_VSEL_V2_AON	R/W	BOD high gear selection. Each bit represents one gear, and only one “1” can be configured at a time. 00001: 2.45V 00010: 2.55V 00100: 2.65V 01000: 2.75V 10000: 2.85V
[15:10]	BOD_DB_SEL	R/W	BOD De-Bounce (glitch) time Control register(should enable high voltage sync) [0]=1: about 2^0 SLOW clock [1]=1: about 2^1 SLOW clock [2]=1: about 2^2 SLOW clock [3]=1: about 2^3 SLOW clock [4]=1: about 2^4 SLOW clock [5]=1: about 2^5 SLOW clock(default) Note: If software enables more than one bit, the bit with the smallest number will be selected and the other enabled channels will be ignored.
[9]	BOD_RST_EN	R/W	Brown-out Reset Enable Bit (Write Protect) 0: Brown-out “INTERRUPT” function Enabled; when the Brown-out Detector function is enable and the detected voltage is lower than the threshold, then assert a signal to interrupt the MCU 1: Brown-out “RESET” function Enabled; when the Brown-out Detector function is enable and the detected voltage is lower than

			the threshold then assert a signal to reset the chip. Note: When the BOD_EN is enabled and the interrupt is asserted, the interrupt will be kept till the BOD_EN is set to 0. The interrupt for CPU can be blocked by disabling the NVIC in CPU for BOD interrupt or disable the interrupt source by disabling the BOD_EN and then re-enabling the BOD_EN function if the BOD function is required.
[8:6]	PMU_BOD_VREF_TRIM_AON	R/W	BOD rst vref trim, 30mV each step
[5]	Reserved		Reserved.
[4:2]	PMU_BOD_VSEL_AON	R/W	000: 1.75 001: 1.85 010: 1.95 011: 2.05 100: 2.15 101: 2.25 110: 2.35
[1]	PMU_BOD_TEST_EN_AON	R/W	BOD analog test enable control 0: BOD test disable 1: BOD test enable
[0]	PMU_BOD_EN_AON	R/W	BOD enable control 0: BOD is disable 1: BOD is enable

3.2.5.10 SYS_CLK_CTRL

Register	Offset	R/W	Description	Reset Value
SYS_CLK_CTRL	PMU_BA+0x24	R/W	System Clock Control Register	0x0000_4900

Bits	Description		
[31:16]	OTP_TR_TRIM	R/W	The value of OTP TR TRIM The default is load from OTP. The value can be overwrite by software.
[15:13]	OTP_CLK_HIGH_CYCLE	R/W	Output to otp_ctl. This register indicates the number of CPU clock cycles during which the OTP_PAD_SCLK signal remains high. 3’b000: OTP_PAD_SCLK high for 1 CPU clock cycle 3’b001: OTP_PAD_SCLK high for 2 CPU clock cycles 3’b010: OTP_PAD_SCLK high for 3 CPU clock cycles 3’b011: OTP_PAD_SCLK high for 4 CPU clock cycles .... 3’b111:OTP_PAD_SCLK high for 8 CPU clock cycles For example, when the system clock is 32 MHz and clk_high_cycle=low_cycle=3'b000, the actual frequency of the OTP memory PAD_SCLK is 16 MHz. When the system clock is 32 MHz and clk_high_cycle=low_cycle=2'b001, the actual frequency of the OTP memory PAD_SCLK is 8 MHz.
[12:10]	OTP_CLK_LOW_CYCLE	R/W	Output to otp_ctl This register indicates the number of CPU clock cycles during which the OTP_PAD_SCLK signal remains low. 3’b000: OTP_PAD_SCLK low for 1 CPU clock cycle 3’b001: OTP_PAD_SCLK low for 2 CPU clock cycles 3’b010: OTP_PAD_SCLK low for 3 CPU clock cycles 3’b011: OTP_PAD_SCLK low for 4 CPU clock cycles .... 3’b111:OTP_PAD_SCLK low for 8 CPU clock cycles For example, when the system clock is 32 MHz and clk_high_cycle=low_cycle=3'b000, he actual frequency of the OTP memory PAD_SCLK is 16 MHz. When the system clock is 32 MHz and clk_high_cycle=low_cycle=2'b001, the actual frequency of the OTP memory PAD_SCLK is 8 MHz.
[9]	OTP_SW_UP_TR_EN	R/W	OTP software update tr value enable control 0: Disable 1: Enable Should sw clear to 0 before enable.
[8]	OTP_WR_EN	R/W	OTP WEEN,REEN congtol 0: Disable 1: Enable
[7]	OTP_SW_UP_CYCLE_EN	R/W	OTP software update CYCLE value enable control 0: Disable

			1: Enable. Hardware Auto-Zero.
[6]	OTP_READ_MODE_SEL	R/W	OTP read mode select 0: READ2, in this mode the OTP peak clock frequency is 8 MHz (default) 1: READ, in this mode the OTP peak clock frequency is 16 MHz Note: config after the OTP TRIM is load with the right value
[5]	PAD_RST_MASK	R/W	PAD RESET function mask (P03) 0: Use as RESET function 1: Use as a nomal gpio
[4]	ACK32K_CLK_EN	R/W	Active 32k clock enable 0: Clock disable 1: Clock enable
[3:2]	SLOW_CLK_SEL	R/W	Slow clock source select 00: RCL 01: XTL 10: ACT_32K 11: ACT_32K
[1:0]	SYS_CLK_SEL	R/W	System Clock source select 00: RCH (default) 01: XTH 10: DPLL (RCH should not be disable) 11: DPLL (XTH should not be disable)

3.2.5.11 RCL_CTRL

Register	Offset	R/W	Description	Reset Value
RCL_CTRL	PMU_BA+0x28	R/W	RCL Control Register	0x0108_8081

Bits	Description		
[31:25]	Reserved		Reserved
[24]	CLK_RCL32K_RDY_OUT_AON	R	RCO_LOSC clock stable flag
[23:21]	Reserved		Reserved
[20:19]	CLK_RCL_TEMPTRIM_AON	R/W	The temperature tune of RCL
[18:17]	CLK_RCL_DELAY_AON	R/W	RCO_LOSC startup counter register 00: 16 01: 32 10: 64 11: 128
[16]	Reserved		Reserved
[15:8]	CLK_RCL_FREQ_FINE_AON	R/W	Fine calibration trim of RCL Default: 8'b100000000
[7:4]	CLK_RCL_FREQ_COARSE_AON	R/W	Coarse calibration trim of RCL Default: 4'b1000
[3:2]	Reserved		Reserved
[1]	CLK_RCL_TEST_EN_AON	R/W	RCO_LOSC test enable 0: Disable 1: Enable
[0]	CLK_RCL_EN_AON	R/W	RCL enable control 0: Disable 1: Enable

3.2.5.12 RCH_CTRL

Register	Offset	R/W	Description	Reset Value
RCH_CTRL	PMU_BA+0x2C	R/W	RCH Control Register	0x0103_4021

Bits	Description		
[31:25]	Reserved		Reserved
[24]	CLK_RCH32M_RDY_OUT_AON	R	RCO_HOSC clock stable flag
[23:18]	Reserved		Reserved
[17:16]	CLK_RCH_RDY_TRIM_AON	R/W	RCO_HOSC startup counter register 00: 16 cycle 01: 32 cycle 10: 64 cycle 11: 128cycle
[15:8]	CLK_RCH_FREQ_TRIM_AON	R/W	Fine: 85kHz per step. Default: 01000000
[7:6]	Reserved		Reserved
[5:4]	CLK_RCH_FREQ_COARSE_AON	R/W	
[3:2]	Reserved		Reserved
[1]	CLK_RCH_TEST_EN_AON	R/W	RCO_HOSC test enable
[0]	CLK_RCH_EN_AON	R/W	RCO_HOSC enable control 0: Disable 1: Enable

3.2.5.13 XTL_CTRL

Register	Offset	R/W	Description	Reset Value
XTL_CTRL	PMU_BA+0x30	R/W	XTL Control Register	0x0003_0020

Bits	Description		
[31:25]	Reserved		Reserved
[24]	CLK_XTL_RDY_OUT_AON	R	XO_LOSC clock stable flag
[23:18]	Reserved		Reserved
[17:16]	CLK_XTL_DELAY_AON	R/W	XO_LOSC startup counter register 00: 64 01: 128 10: 256 11: 4096
[15:7]	Reserved		Reserved
[6:4]	CLK_XTL_ICORE_AON	R/W	XO_LOSC core bias register
[3:2]	Reserved		Reserved
[1]	CLK_XTL_TEST_EN_AON	R/W	XO_LOSC test enable
[0]	CLK_XTL_EN_AON	R/W	XTL enable control 0: Disable 1: Enable

3.2.5.14 XTH_CTRL

Register	Offset	R/W	Description	Reset Value
XTH_CTRL	PMU_BA+0x34	R/W	XTH Control Register	0x0000_2A0C

Bits	Description		
[31:25]	Reserved		Reserved
[24]	FSYNXO_CLKRDY_AON	R	XO_FSYN clock stable flag
[23:21]	Reserved		Reserved
[20]	FSYNXO_AMPSEL_AON	R/W	FSYNXO Output Amplitude Control: 1: High Amplitude 750mV 0: Low Amplitude 500mV
[19]	FSYNXO_CAP2_AON	R/W	FSYNXO adds an additional on-chip capacitor of 6pF 1: Increase 0: No increase
[18:17]	FSYNXO_HYS_AON	R/W	XTH outputs buffer and inputs offset and hysteresis control. 00: No hysteresis and input offset 01: ±75mV hysteresis 10: 60mV input offset 11: ±75mV hysteresis + 60mV input offset
[16]	FSYNXO_STARTUP_COUNTER_AON	R/W	FSYNXO RDY Delay Control Bit
[15]	FSYNXO_RES_AON	R/W	FSYNXO Input/Output Feedback Resistor Control: 1: 500K 0: 250K
[14]	FSYNXO_BUFEN_AON	R/W	FSYNXO Buffer Enable: 0: Disable 1: Enable
[13:12]	FSYNXO_ICORE_AON	R/W	FSYNXO Current Range Adjustment: 00: 800uA 01: 1mA 10: 1.2mA 11: 1.4mA
[11]	FSYNXO_FB_AON	R/W	FSYNXO Open-Loop Status: 0: Open-Loop 1: Closed-Loop
[10]	FSYNXO_FAST_DLY_AON	R/W	Does the FSYNXO fast-start buffer increase the operating time? 0: No delay 1: Delay present
[9:4]	FSYNXO_CAPSEL_AON	R/W	FSYNXO Frequency Control Bit Adjustment Range: ±30 ppm, 1 ppm/step
[3]	EN_FSYNXO_DEGLITCH_AON	R/W	FSYNXO Jitter Reduction Enable 0: Disabled 1: Enabled
[2]	FSYNXO_STARTUP_FAST_AON	R/W	FSYNXO Fast Start-up Enable (Software must be disabled after start-up completes)

			0: Disabled 1: Enabled
[1]	Reserved		Reserved
[0]	FSYNXO_EN_AON	R/W	XO_FSYN Enable Control 0: Disabled 1: Enabled

3.2.5.15 DPLL_CTRL

Register	Offset	R/W	Description	Reset Value
DPLL_CTRL	PMU_BA+0x38	R/W	DPLL Control Register	0x0000_1480

Bits	Description		
[31:25]	Reserved		Reserved
[24]	CLK_DPLL_RDY_OUT_AON	R	DPLL stable flag
[23:18]	Reserved		Reserved
[17:16]	CLK_DPLL_DELAY_AON	R/W	PLL startup counter register 00: 256, 01: 512, 10: 1024, 11: 2048
[15:14]	Reserved		Reserved
[13]	CLK_DPLL_16M_MODE_AON	R/W	DPLL Switching 16M Crystal Mode 1: 16M, 0: 32M
[12:11]	CLK_DPLL_VCO_ITRIM_AON	R/W	DPLL VCO frequence trim control 00: 65M~193M 01: 99M~220M 10: 129M~246M 11: 158M~271M
[10:8]	CLK_DPLL_KVCO_CTRL_AON	R/W	DPLL KVCO control 000: 70MHz/V 111: 275MHz/V step: 30MHz/V
[7:6]	CLK_DPLL_ICP_CTRL_AON	R/W	DPLL charge pump current control 00: 0.5*I 01: I 10: 1.5*I 11: 2*I
[5:4]	CLK_DPLL_ICP_BIAS_AON	R/W	DPLL charge pump bias current control 00: 5uA 01: 7.5uA 10: 10uA 11: 12.5uA
[3]	CLK_DPLL_REFCLK_SEL_AON	R/W	PLL reference clock register 0/1: RCO/XO
[2]	CLK_DPLL_FREQ_SEL_AON	R/W	PLL frequency register 0/1: 48MHz/32MHz
[1]	CLK_DPLL_TEST_EN_AON	R/W	Register for pll test mode
[0]	CLK_DPLL_EN_AON	R/W	DPLL enable control 0: Disable 1: Enable

3.2.5.16 ANA_MISC

Register	Offset	R/W	Description	Reset Value
ANA_MISC	PMU_BA+0x3C	R/W	PTAT & POLY & SW control	0xE0F7_0049

Bits	Description		
[31]	OTP_REF_EN_AON	R/W	OTP REF Enable
[30]	OTP_CE_AON	R/W	OTP CE Enable
[29]	OTP_PSD_AON	R/W	OTP PSD Enable
[28]	TST_LDO_OUT_AON	R/W	LDO Enable 1: Enable 0: Disable
[27:24]	Reserved		Reserved.
[23]	PMU_SW1_LPLDOH_LPLDOL_EN_AON	R/W	
[22]	PMU_SW0_DVDD_LPLDOL_EN_AON	R/W	
[21]	LEVELSHIFT_EN_HPLDO_AON	R/W	The enable control of level shift between HPLDO region and analog supply voltage region 1: Enable 0: Disable
[20]	LEVELSHIFT_EN_GLOBAL_AON	R/W	The enable control of level shift between LPLDOH region and analog supply voltage region 1: Enable 0: Disable
[19:17]	PMU_IPOLY_TRIM_AON	R/W	The current tune of poly
[16]	PMU_IPOLY_EN_AON	R/W	Enable POLY 1: Enable 0: Disable
[15:7]	Reserved		Reserved.
[6:4]	PMU_PTAT_VTRIM_AON	R/W	Adjustment control signal for BG reference voltage, 680m-730mV, 8mV per step
[3]	PMU_PTAT_BUF_EN_AON	R/W	VBG BUF Enable
[2:1]	PMU_PTAT_TEMPTRIM_AON	R/W	Adjustment control signal for BG reference voltage, 680m-730mV, 8mV per step
[0]	PMU_PTAT_EN_AON	R/W	PMU-PTAT Enable 0: Disable 1: Enable

3.2.5.17 ANA_HPLDO

Register	Offset	R/W	Description	Reset Value
ANA_HPLDO	PMU_BA+0x40	R/W	HPLDO control	0x0000_03A5

Bits	Description		
[31:11]	Reserved		Reserved.
[10:8]	PMU_LDO_RDY_VREF_TRIM_AON	R/W	
[7:4]	PMU_HPLDO_VTRIM_AON	R/W	The voltage trim of HPLDO
[3]	Reserved		Reserved.
[2]	PMU_HPLDO_CAP_SEL_AON	R/W	HPLDO External Capacitor Selection for BUCK Mode and Non-BUCK Mode Switching
[1]	PMU_HPLDO_SOFTSTART_TIME_AON	R/W	HPLDO Soft Start Time Selection 0: 20µs, 1: 40µs
[0]	PMU_HPLDO_EN_AON	R/W	The enable control of HPLDO 1: Enable 0: Disable

3.2.5.18 ANA_LPLDO

Register	Offset	R/W	Description	Reset Value
ANA_LPLDO	PMU_BA+0x44	R/W	LPLDO control	0x00F0_001F

Bits	Description		
[31:24]	Reserved		Reserved.
[23:20]	PMU_LPLDOH_VSEL_AON	R/W	
[19:18]	Reserved		Reserved.
[17]	PMU_LPLDOH_MODE2_EN_AON	R/W	LPLDOH mode 2 Enable 1: Enable 0: Disable
[16:1]	Reserved		Reserved.
[0]	PMU_LPLDOL_EN_AON	R/W	The enable control of LPLDOL 1: Enable 0: Disable

3.2.5.19 ANA_ANALDO

Register	Offset	R/W	Description	Reset Value
ANA_ANALDO	PMU_BA+0x48	R/W	ANALDO control	0x0000_0081

Bits	Description		
[31:17]	Reserved		Reserved.
[16]	EN_LDO_FSYN_AON	R/W	Enable FSYN LDO 1: Enable 0: Disable
[15:8]	Reserved		Reserved.
[7:4]	LDO_ANA_TRIM_AON	R/W	the voltage trim of ANA LDO
[3:1]	Reserved		Reserved.
[0]	EN_LDO_ANA_AON	R/W	Enable ANA LDO 1: Enable 0: Disable

3.2.5.20 ANA_RESERVED

Register	Offset	R/W	Description	Reset Value
ANA_RESERVED	PMU_BA+0x4C	R/W	Analog reserved control	0xFFFF_0000

Bits	Description		
[31:21]	RESERVED1_H_AON	R/W	Reserved
[20:17]	PMU_LPLDOL_VSEL_AON	R/W	LPLDOL voltage fine-tuning: 0.4–0.8V, 10mV per step
[16]	RESERVED_H_AON	R/W	Reserved
[15:0]	RESERVED_L_AON	R/W	Reserved

3.3 Reset and Clock Controller (RCC)

3.3.1 Overview

The RCC module can implement reset function and generate clocks for the whole chip, including system clocks and all peripheral clocks.

3.3.2 Reset

3.3.2.1 System Reset Range

Reset Source	Trigger Condition	Reset Region	Flag
POR	Power on reset and power down reset (threshold voltage: 1.65V)	Whole chip	PORRF
PAD_RESET	Pull down Reset PIN	Whole chip without LEVELSHIFT_GLOBAL，HPLDO, PTAT, IPOLY disable	PINRF
LVR	Low voltage reset (threshold voltage: 1.8V)	Whole chip without LEVELSHIFT_GLOBAL，HPLDO，PTAT, IPOLY disable and itself control logic (include slow clock)	LVRRF
BOD	Brown-out Detector Reset (threshold voltage: 1.75V-2.95V)	Whole chip without LEVELSHIFT_GLOBAL，HPLDO，PTAT, IPOLY disable and itself control logic (include slow clock)	BODRF
CHIP0	Software config releated reg	whole chip	CHIP0RF
CHIP1	Software config releated reg	Whole chip without LEVELSHIFT_GLOBAL，HPLDO，PTAT, IPOLY disable	CHIP1RF
WDT	Count reachs the target time	whole chip without LEVELSHIFT_GLOBAL，HPLDO，PTAT, IPOLY disable	WDTRF
WWDT	Count reachs the target time	whole chip without LEVELSHIFT_GLOBAL，HPLDO，PTAT, IPOLY disable	WDTRF
SYS	Software config releated reg	digital region without cpu debug and PMU (except PMU regs) modules	SYSRF
CPU	Software config releated reg	Just reset cpu, icache, fmc and decryption modules (Not recommended for use)	CPURF
MODULE RESET	Software config releated reg	Reset of each function module	-

3.3.2.2 Reset Block Diagram

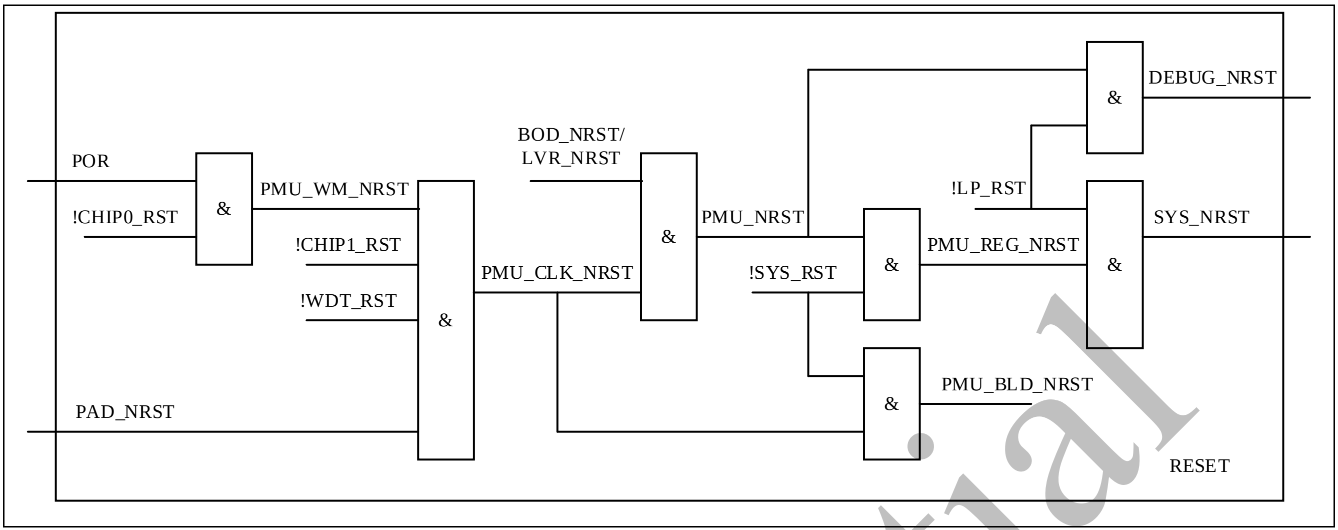


Figure 3-1 System Reset Block Diagram

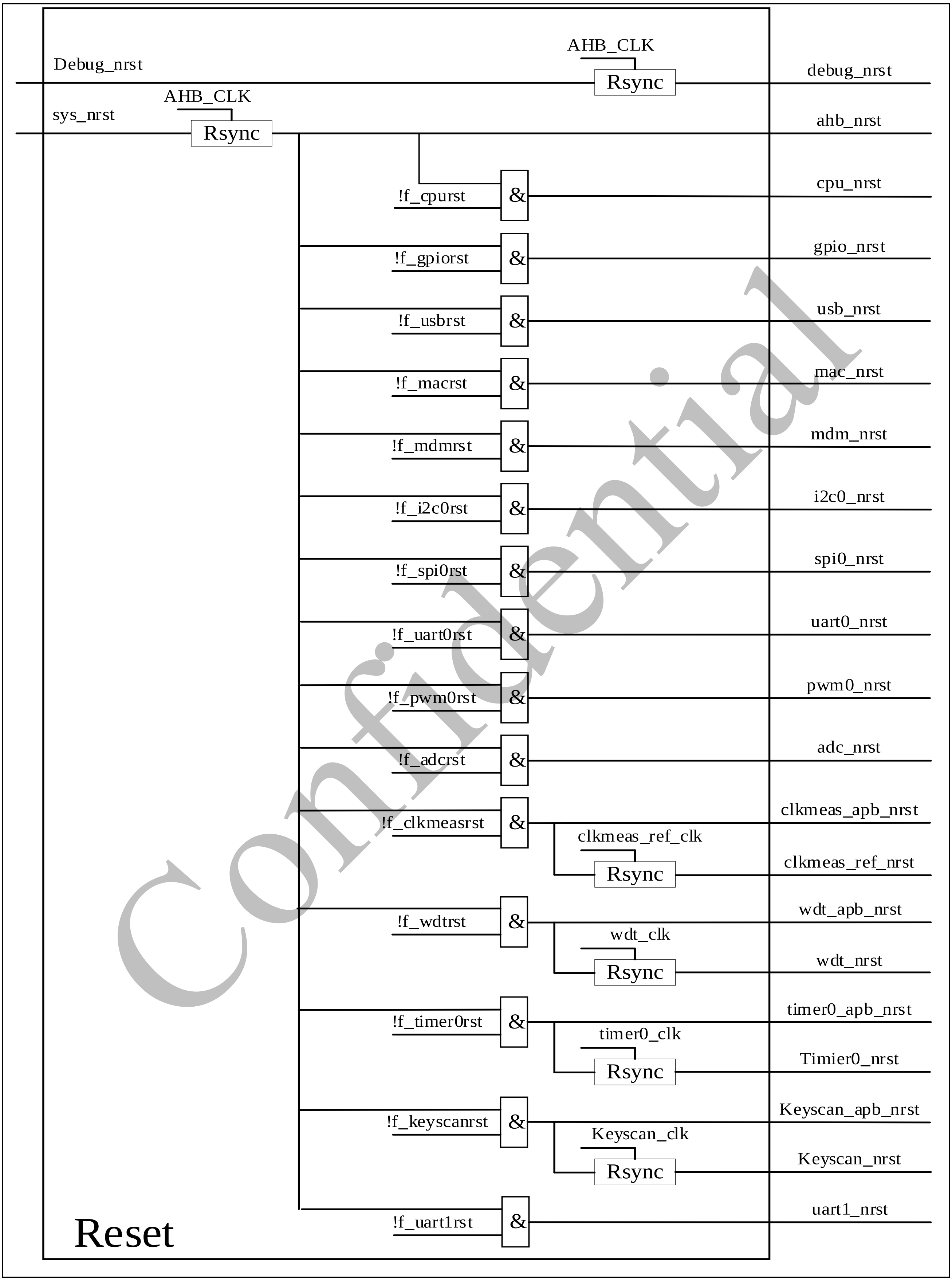


Figure 3-2 Reset Block Diagram of Each Module

3.3.3 Clock Controller

3.3.3.1 Analog Clock

The analog clocks are listed below:

Table 3-6 Analog Clock Source

Name	Source	Frequency	Function
CLK_XTH_DIG	XTH	32MHz	System clock
CLK_XTH_RF	XTH	32MHz	MAC / PHY module clock
CLK_XTL	XTL	32.768kHz	System low speed clock
CLK_RCH	RCH	32MHz	System clock / MAC / PHY module clock
CLK_RCL	RCL	32kHz	System low speed clock
CLK_DPLL	DPLL	48MHz	System clock
CLK_USB_DPLL	DPLL	48MHz	USB clock

3.3.3.2 Analog Clock Block Diagram

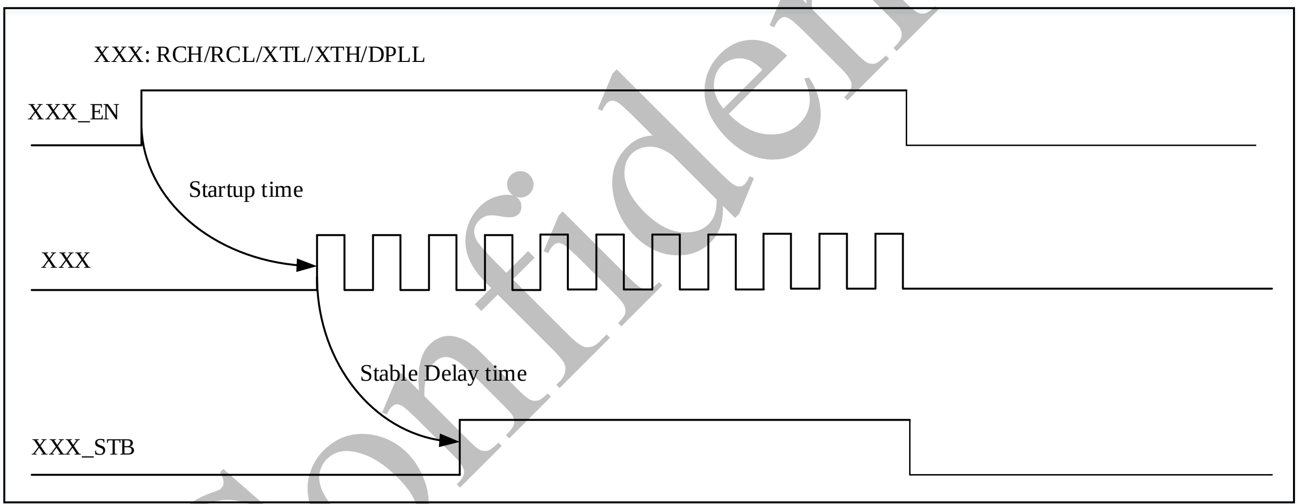


Figure 3-3 Analog Clock Block Diagram

XXX_EN represents the enable signal of the corresponding clock. If the enable signal is pulled high, the clock is output to the digital circuit after a clock startup time. The setup time is as follows: RCL 200us, XTL 50ms, RCH 2us, XTH 100us, DPLL 20us.

The STB Delay time is controlled by two bits to control the time of XXX_STB is pulled high. More information please refer to the register definition.

After each clock reaches the digital part, it is used inside the digital after passing through the XXX_STB Gate. the Gate is added on the digital side.

3.3.3.3 Digital Clock

System Clock Block Diagram

Figure 3-4 shows the system clock source.

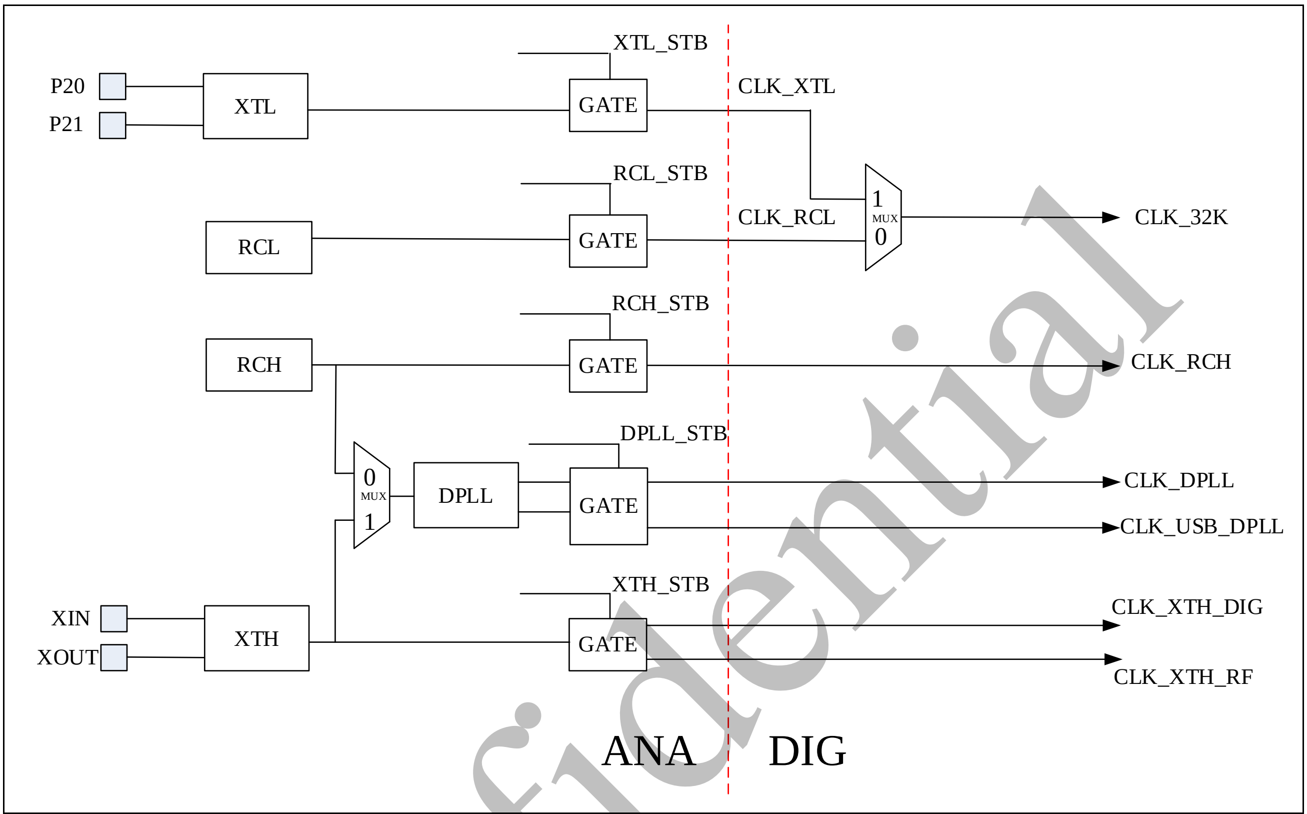


Figure 3-4 System Clock Source

Figure 3-5 shows the digital system clock.

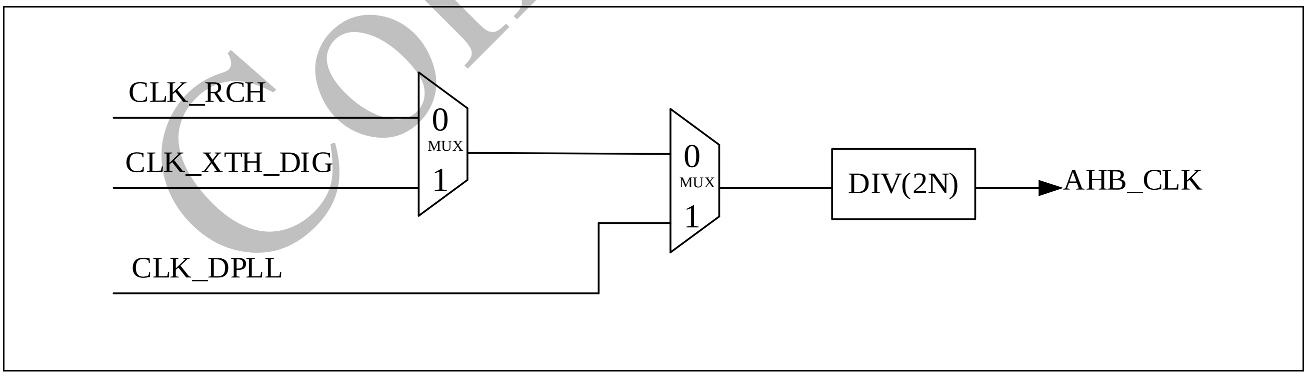


Figure 3-5 Digital System Clock

Figure 3-6 shows the system periphral clock.

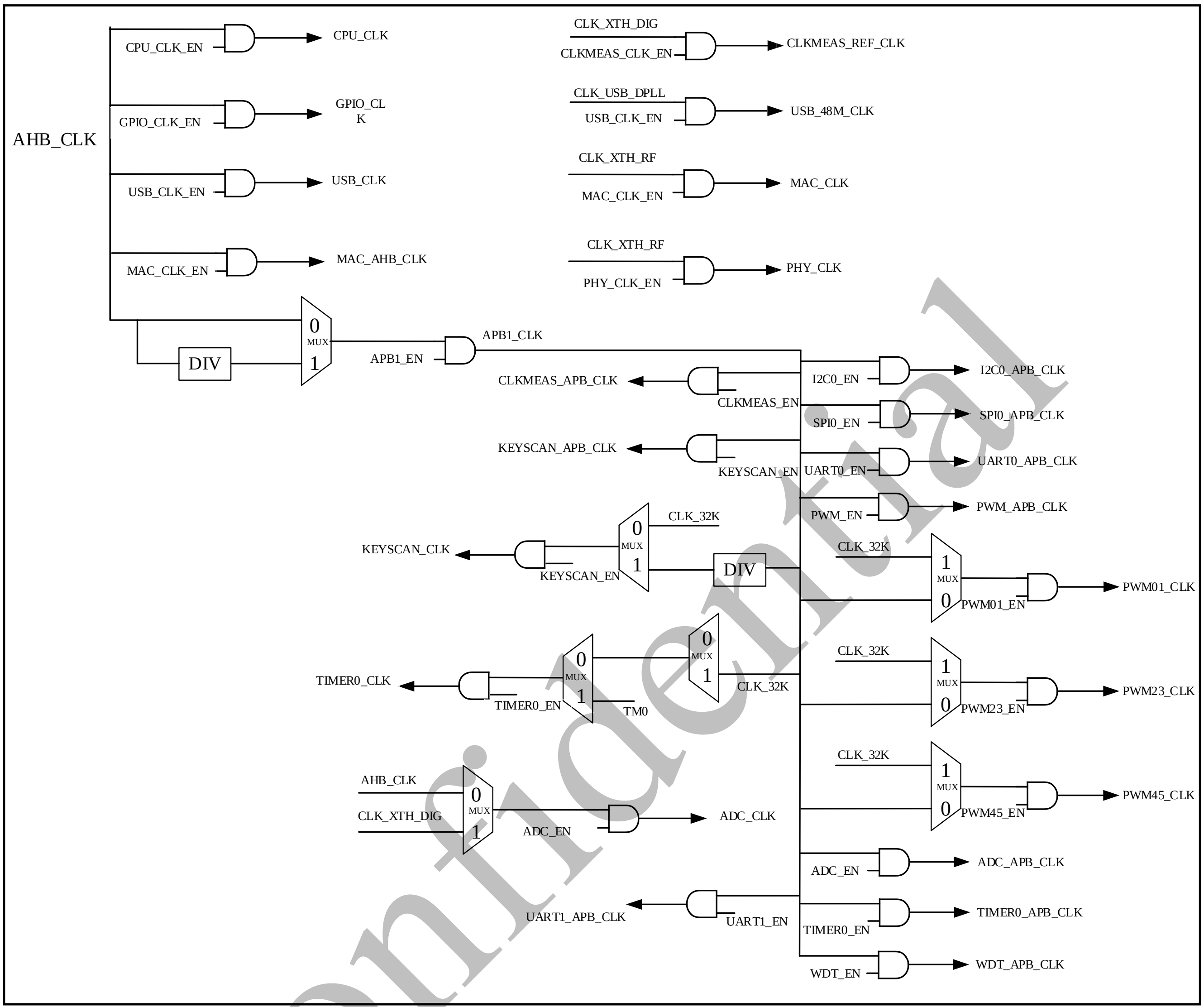


Figure 3-6 System Periphral Clock

3.3.4 RCC Register Map

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

Register	Offset	R/W	Description	Reset Value
RCC Base Address: RCC_BA = 0x4004_0000				
RSTSTS	RCC_BA+0x00	R/W	System Reset Status Register	0x0000_0040
IPRST0	RCC_BA+0x04	R/W	Peripheral Reset Control Register 0	0x0000_0000
IPRST1	RCC_BA+0x08	R/W	Peripheral Reset Control Register 1	0x0000_0000
AHB_CLK_CTRL	RCC_BA+0x0C	R/W	AHB Periphral Clock Control Register	0xF400_E000
APB1_CLK_CTRL0	RCC_BA+0x10	R/W	APB1 Periphral Clock Control Register	0x0000_0000

3.3.5 RCC Register Description

3.3.5.1 RSTSTS

Register	Offset	R/W	Description	Reset Value
RSTSTS	RCC_BA+0x00	R/W	System Reset Status Register	0x0000_0040

Bits	Description		
[31:9]	Reserved		Reserved.
[8]	CHIP1RF	R/W	CHIP1 Reset Flag The CHIP1 reset flag is set by the "Reset Signal" from the CHIP1RST (IPRST0[8]) to indicate the previous reset source. 0: CHIP1RST. 1: CHIP1RST had issued the reset signal to reset the system. Note: Software can write 1 to clear this bit to zero.
[7]	CPURF	R/W	CPU Reset Flag The CPU reset flag is set by hardware if software writes CPURST (IPRST0[1]) 1 to reset MCU and Flash Memory Controller (FMC). 0: No reset from CPU. 1: The MCU and FMC are reset by software setting CPURST to 1. Note: Software can write 1 to clear this bit to zero.
[6]	PORRF	R/W	POR Reset Flag The POR reset flag is set by the "Reset Signal" from the Power-on Reset (POR) Controller to indicate the previous reset source. 0: No reset from POR. 1: POR had issued the reset signal to reset the system. Note: Software can write 1 to clear this bit to zero.
[5]	SYSRF	R/W	System Reset Flag The system reset flag is set by the "Reset Signal" from the MCU to indicate the previous reset source. 0: No reset from MCU 1: The MCU had issued the reset signal to reset the system by writing 1 to the bit RCCRESETREQ (SCS_AIRCR[2]), Application Interrupt and Reset Control Register, address = 0xE000ED0C) in system control registers of MCU. Note: Software can write 1 to clear this bit to zero.
[4]	BODRF	R/W	BOD Reset Flag The BOD reset flag is set by the "Reset Signal" from the Brown-out Detector to indicate the previous reset source. 0: No reset from BOD. 1: The BOD had issued the reset signal to reset the system. Note: Software can write 1 to clear this bit to zero.
[3]	LVRRF	R/W	LVR Reset Flag 0: No reset from LVR. 1: The LVR had issued the reset signal to reset the system. Note: Software can write 1 to clear this bit to zero.
[2]	WDTRF	R/W	WDT Reset Flag The WDT reset flag is set by the "Reset Signal" from the Watchdog Timer or

			Window Watchdog Timer to indicate the previous reset source. 0: No reset from watchdog timer or window watchdog timer. 1: The watchdog timer or window watchdog timer had issued the reset signal to reset the system. Note: Software can write 1 to clear this bit to zero.
[1]	PINRF	R/W	NRESET Pin Reset Flag The nRESET pin reset flag is set by the “Reset Signal” from the nRESET pin Controller to indicate the previous reset source. 0: No reset from nRESET pin. 1: Pin nRESET had issued the reset signal to reset the system. Note: Software can write 1 to clear this bit to zero.
[0]	CHIP0RF	R/W	CHIP0 Reset Flag The CHIP0 reset flag is set by the “Reset Signal” from the CHIP0RST (IPRST0[0]) to indicate the previous reset source. 0: CHIP0RST. 1: CHIP0RST had issued the reset signal to reset the system. Note: Software can write 1 to clear this bit to zero.

3.3.5.2 IPRST0

Register	Offset	R/W	Description	Reset Value
IPRST0	RCC_BA+0x04	R/W	Peripheral Reset Control Register 0	0x0000_0000

Bits	Description		
[31:9]	Reserved		Reserved.
[8]	CHIP1RST	R/W	CHIP One-shot Reset (Write Protect) Setting this bit will reset the whole chip, including Processor core and all peripherals, and this bit will automatically return to 0. The CHIPRST is the same as the PAD reset, all the chip controllers is reset and the chip settings from flash are also reload. 0: Chip normal operation. 1: CHIP one-shot reset. Note: This bit is write protected. Refer to the RCC_REGLCTL register.
[7]	MDMRST	R/W	MDM Reset 0: MDM normal operation. 1: MDM reset.
[6]	Reserved		Reserved.
[5]	USBRST	R/W	USB Reset 0: USB normal operation. 1: USB reset.
[4]	Reserved		Reserved.
[3]	MACRST	R/W	MAC Reset 0: MAC normal operation. 1: MAC reset.
[2]	Reserved		Reserved.
[1]	CPURST	R/W	Processor Core One-shot Reset (Write Protect) Setting this bit will only reset the processor core and Flash Memory Controller (FMC), and this bit will automatically return to 0. 0: Processor core normal operation. 1: Processor core one-shot reset. Note: This bit is write protected. Refer to the RCC_REGLCTL register. Note: must set AHB_CLK_CTRL[20](spi_flash_clk_sel) reg to 0 before set cpurst to 1
[0]	CHIPRST	R/W	Chip One-shot Reset (Write Protect) Setting this bit will reset the whole chip, including Processor core and all peripherals, and this bit will automatically return to 0. The CHIPRST is the same as the POR reset, all the chip controllers is reset and the chip settings from flash are also reload. 0: Chip normal operation. 1: Chip one-shot reset. Note: This bit is write protected. Refer to the RCC_REGLCTL register.

3.3.5.3 IPRST1

Register	Offset	R/W	Description	Reset Value
IPRST1	RCC_BA+0x08	R/W	Peripheral Reset Control Register 1	0x0000_0000

Bits	Description		
[31:17]	Reserved		Reserved.
[16]	UART1RST	R/W	UART1 Controller Reset 0: UART1 controller normal operation. 1: UART1 controller reset.
[15]	GPIORST	R/W	GPIO Controller Reset 0: GPIO controller normal operation. 1: GPIO controller reset.
[14]	KEYSCANRST	R/W	KEYSCAN Controller Reset 0: KEYSCAN controller normal operation. 1: KEYSCAN controller reset.
[13]	CLKMEASRST	R/W	CLKMEAS Controller Reset 0: CLKMEAS controller normal operation. 1: CLKMEAS controller reset.
[12]	TMR0RST	R/W	Timer0 Controller Reset 0: Timer0 controller normal operation. 1: Timer0 controller reset.
[11]	Reserved		Reserved
[10]	WDTRST	R/W	WDT Controller Reset 0: WDT controller normal operation. 1: WDT controller reset.
[9]	ADCRST	R/W	ADC Controller Reset 0: ADC controller normal operation. 1: ADC controller reset.
[8]	PWM0RST	R/W	PWM Controller Reset 0: PWM controller normal operation. 1: PWM controller reset.
[7]	Reserved		Reserved
[6]	UART0RST	R/W	UART0 Controller Reset 0: UART0 controller normal operation. 1: UART0 controller reset.
[5:3]	Reserved		Reserved
[2]	SPI0RST	R/W	SPI0 Controller Reset 0: SPI0 controller normal operation. 1: SPI0 controller reset.
[1]	Reserved		Reserved
[0]	I2C0RST	R/W	I2C0 Controller Reset 0: I2C0 controller normal operation. 1: I2C0 controller reset.

3.3.5.4 AHB_CLK_CTRL

Register	Offset	R/W	Description	Reset Value
AHB_CLK_CTRL	RCC_BA+0x0C	R/W	AHB Periphral Clock Control Register	0xF40_E000

Bits	Description		
[31:23]	XTH_DIV	R/W	XTH divider for xtl quick setup, default value: 0x1e8
[22]	XTL_QUICK_EN	R/W	XTL quick setup enable control 0: Disable 1: Enable
[21]	Reserved		Reserved.
[20]	USB_48M_CLK_EN	R/W	USB 48MHz clock enable control 0: Disable 1: Enable
[19]	USB_AHB_CLK_EN	R/W	USB AHB clock enable control 0: Disable 1: Enable
[18]	PHY_CLK_EN	R/W	BLE PHY clock enable control 0: Disable 1: Enable
[17]	MAC_CLK_EN	R/W	MAC clock enable control 0: Disable 1: Enable
[16]	Reserved		Reserved.
[15]	APB1_CLK_EN	R/W	APB1 clock enable control 0: Disable 1: Enable
[14]	ST_CLK_EN	R/W	System Tick clock enable 0: Disable 1: Enable
[13]	GPIO_CLK_EN	R/W	GPIO clock enable control 0: Disable 1: Enable
[12]	Reserved		Reserved.
[11:8]	SLP_AHB_CLK_DIV	R/W	AHB clock divider control at sleep mode (sleep, deepsleep, standby_m1) [3:0]=0: clock does not divide [3:0]=N: clock is divided, clock = 1/(N+1), (0<N<16)
[7:4]	APB1_DIV	R/W	APB1 clock divider control [3:0]=0: clock does not divide [3:0]=N: clock is divided, clock = 1/(2*N), (0<N<16) Note: should first set these divider bits if need; then enable AHB_CLK_CTRL[3].
[3:0]	AHB_DIV	R/W	AHB clock divider control [3:0]=0: clock does not divide [3:0]=N: clock is divided, clock = 1/(N+1), (0<N<16)

3.3.5.5 APB1_CLK_CTRL0

Register	Offset	R/W	Description	Reset Value
APB1_CLK_CTRL0	RCC_BA+0x10	R/W	APB1 Periphral Clock Control Register	0x0000_0000

Bits	Description		
[31:23]	KEYSCAN_DIV	R/W	Keyscan clock source divisor [31:23] = 0: No use [31:23] = N: Clock is divided, clock = 1/(2N), (0<N<512)
[22]	ADC_CLK_SEL	R/W	Adc sample clock select 0: AHB clk 1: XTH Note: should first select the clk source if need; then enable Adccken.
[21]	KEYSCAN_CLK_SEL	R/W	keyscan clock select 0: Keyscan div 1: RCL/XTL Note: should first select the clk source if need; then enable PWM0_67cken.
[20]	PWM0_45CKSEL	R/W	PWM45 clock select 0: APB 1: RCL/XTL Note: should first select the clk source if need; then enable PWM0_45cken.
[19]	PWM0_23CKSEL	R/W	PWM23 clock select 0: APB 1: RCL/XTL Note: should first select the clk source if need; then enable PWM0_23cken.
[18]	PWM0_01CKSEL	R/W	PWM01 clock select 0: APB 1: RCL/XTL Note: should first select the clk source if need; then enable PWM0_01cken.
[17:16]	Reserved		Reserved
[15]	UART1CKEN	R/W	Uart1 clock enable 0: Clock disable 1: Clock enable
[14]	KEYSCANCKEN	RW	KeyScan clock enable 0: Clock disable 1: Clock enable
[13]	CLKMEASCKEN	RW	CLKMEAS clock enable 0: Clock disable 1: Clock enable
[12]	TMR0CKEN	R/W	Timer0 clock enable 0: Clock disable 1: Clock enable
[11]	Reserved		Reserved
[10]	WDTCKEN	R/W	Wdt clock enable 0: Clock disable 1: Clock enable
[9]	ADCCKEN	R/W	ADC clock enable

			0: Clock disable 1: Clock enable
[8]	PWM0CKEN	R/W	PWM APB clock enable 0: Clock disable 1: Clock enable
[7]	Reserved		Reserved
[6]	PWM0_45CKEN	R/W	PWM45 clock enable 0: Clock disable 1: Clock enable
[5]	PWM0_23CKEN	R/W	PWM23 clock enable 0: Clock disable 1: Clock enable
[4]	PWM0_01CKEN	R/W	PWM01 clock enable 0: Clock disable 1: Clock enable
[3]	UART0CKEN	R/W	Uart0 clock enable 0: Clock disable 1: Clock enable
[2]	Reserved		Reserved
[1]	SPI0CKEN	R/W	SPI0 clock enable 0: Clock disable 1: Clock enable
[0]	I2C0CKEN	R/W	I2C0 clock enable 0: Clock disable 1: Clock enable

3.4 OTP Controller (OTPC)

3.4.1 Overview

OTP uses 1.2V/1.8V/2.5V/3.3V low leakage process. By applying high voltage pulses on the gate and substrate of thin oxide transistors, the oxide between the gate and substrate is decomposed to program the bit cells. The voltage required for OTP bit cell programming is approximately 6.5V, provided by an external VPP pin.

Before programming, all bits of a blank OTP array are read as "0". After programming, the OTP bit cell will be read as "1".

3.4.2 Features

- Power Supply
  - 1.2V DVDD (1.08V~1.32V)
  - 2.5V VBAT (1.8V~3.6V) for Read
  - VBAT (2.5V~3.6V) for program
- Program mode: VPP(6.25V~6.75V) from external pin
  Non-program mode: VPP floating
- Memory Density:
  - Main: 16KBnote
  - Row: 256B
- Redundant: Yes
- Bit Program Operation
- Word Read Operation
- Operation Temperature: -40 to 125°C
- Data Retention: >10 Years @125°C
- OTP Cell: 1.2V device
- Word Access Time: 60ns(max)
- Bit Program Time: 40us (typ)
- Operating Current:
  - IVDD_R: 0.1 mA (typ)
  - IVDD_P: 0.1 mA (typ)
  - IVCC_R: 1.6 mA (typ)
  - IVCC_P: 0.6 mA (typ)

- Deep Standby Current:
  - IVDD: 0.1 µA (typ)
  - IVCC: 0.1 µA (typ)

3.4.3 OTP Row Area CP FT Area Planning

This section describes the Row 256-byte area of the OTP, specifically the 64-word region and the specific content stored within it.

Row Addr.	Bit	Name	Description	Permissions: Before/After encryption	Hardware auto-load
0	[31:0]	DEBUG_KEY[63:32]	Debug decryption key[63:32]	RW/NA Encryption switch Debug_switch	Yes
1	[31:0]	DEBUG_KEY[31:0]	Debug decryption key[31:0]	RW/NA Encryption switch Debug_switch	Yes
2	[31:26]	PROT_CODE_OFFSET	First byte address for injection protection The code protection range is [256*offset ~ 256*offset + 256] For example, when prot_code_offset=0, the protection range is 0 to 255 bytes. When prot_code_offset=1, the protection range is 256 to 511 bytes.	RW/NA Encryption switch Inject_switch	Yes
	[25:0]	Reserved			
3	[31]	DEBUG_SWITCH	Debug Protection Switch 1’b0: Debug Protection Off 1’b1: Debug Protection On	Unprotected	Yes
	[30]	INJECT_SWITCH	Anti-Injection Protection Switch 1’b0: Anti-Injection Protection Off 1’b1: Anti-Injection Protection On	Unprotected	Yes
	[29:0]	Reserved		Unprotected	Yes

Note:

- *W: Writable*
- *NA: Can not read and can not write*
- *Unprotected: No encryption; can read and write under any circumstances.*

For example: RW/NA, readable and writable before encryption; unreadable and unwritable after encryption.

3.4.4 OTP Register Map

Register	Offset	R/W	Description	Reset Value
OTP_BA address: 0x4005_0000				
MODE_CTL	BASE+0x00	R/W		0x0000_0000
READ_PROG_CTL	BASE+0x04	R/W		0x0000_0000
BYTE_ADDR	BASE+0x08	R/W		0x0000_0000
ADDR_BIT	BASE+0x0c	R/W		0x0000_0000
TIMING_CTL	BASE+0x10	R/W		0x0000_0002
OPERATE_DATA_0	BASE+0x14	R/W		0x0000_0000
OPERATE_TRG	BASE+0x24	R/W		0x0000_0000

3.4.5 OTP Register Description

3.4.5.1 MODE_CTL

Register	Offset	R/W	Description	Reset Value
MODE_CTL	OTP_BA+0x00	R/W	Work Mode Control	0x0000_0000

Bits	Descriptions	
[31:2]	Reserved	Reserved
[1:0]	WORK_MODE	OTP Work Mode Configuration 2’b00: Read_Mode 2’b01: Prog_Mode (working in J-Link or RAM code mode) 2’b10: DMA_Mode (working in J-Link or RAM code mode) 2’b11: Idle_Mode (This mode is used when switching PTM between different analog states.) For example, *weak read* in the main region and *weak read* in the test row region can directly modify the PTM without considering timing, since their OTP analog states are consistent. The OTP only multiplexes at the PAD port. However, when performing a *weak read* in the main region, switching to *read* or *read2* in the main region, or switching to *read* or *read2* in the test row region, this involves a change in the OTP analog state, making it impossible to directly modify the PTM. It is necessary to first change the work_mode to idle before restarting the *read*, *prog*, *dma*, and other startup sequences. Default value: 2’b00

3.4.5.2 READ_PROG_CTL

Register	Offset	R/W	Description	Reset value
READ_PROG_CTL	OTP_BA+0x04	R/W	Working Configuration for OTP Read or Write Mode	0x0000_0000

<table>
<tr><th>Bits</th><th colspan="2">Descriptions</th></tr>
<tr><td>[31:17]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[16]</td><td>DMA_VPP_SAMP_FLAG</td><td>In DMA mode, the OTP VPP Pad can sample the test current flag. During the flag's set to high, OTP operations are disabled. Only current detection via the VPP pad is permitted. The flag is automatically set to low after set to high by 4µs (synchronized with TRG).
1’b0: VPP current cannot be sampled at this time
1’b1: VPP current can be sampled at this time</td></tr>
<tr><td>[15:13]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[12]</td><td>PROG_REDUNDANCY</td><td>When programming, should redundancy also be programmed? Both areas require separate programming.
1’b0: Not enable redundancy
1’b1: Enable redundancy
Default value: 1’b0</td></tr>
<tr><td>[11:9]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[8:4]</td><td>PTM</td><td>Work mode configuration of read: PTM is a complex combination of mode truth tables, requiring users to strictly adhere to the protocol during configuration.
Note: CPU fetch can only operate normally when PTM=5'b0_0000 or 5'b0_1000. Therefore, all cases where PTM ≠ 5'b0_0000 and PTM ≠ 5'b0_1000 must be implemented via SWD or ram_code.
<table>
<tr><th colspan="2">Mode</th><th>PTM[4:0] Value</th><th>Note</th></tr>
<tr><td rowspan="2">User Mode</td><td>Read</td><td>5’b0_0000</td><td rowspan="2">Equipped with ECC function.
ECC can be enabled or disabled.</td></tr>
<tr><td>Read2</td><td>5’b1_0000</td></tr>
</table>
Default value: 5’b0_0000</td></tr>
<tr><td>[3:1]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[0]</td><td>ECC_DISABLE</td><td>In read mode, disable or enable ECC. To ensure yield impact, ECC may only be disabled during testing.
1'b0: Do not disable ECC
1'b1: Disable ECC
Default value: 1’b0</td></tr>
</table>

3.4.5.3 BYTE_ADDR

Register	Offset	R/W	Description	Reset value
BYTE_ADDR	OTP_BA+0x08	R/W	The operating byte address of OTP in read, write, and DMA modes	0x0000_0000

Bits	Descriptions	
[31:14]	Reserved	Reserved.
[13:0]	OPERATE_ADDR	The operating byte address of OTP in read, write, and DMA modes. Users need to align the word. Note that although byte addresses are provided, OTP supports only word-aligned programming. The main area has an effective address range of [13:0], 16K bytes. The row area has an effective address range of [7:0], 256 bytes. The col area has an effective address range of [13:6], 256 bits. Default value: 14'b0

3.4.5.4 ADDR_BIT

Register	Offset	R/W	Description	Reset value
ADDR_BIT	OTP_BA+0x0C	R/W	Bit address in DMA mode	0x0000_0000

Bits	Descriptions	
[31:5]	Reserved	Reserved.
[4:0]	ADDR_BIT	This register is only active in DMA mode. DMA mode operates addresses by bit. Default value: 5'b0

3.4.5.5 TIMING_CTL

Register	Offset	R/W	Description	Reset value
TIMING_CTL	OTP_BA+0x10	R/W		0x0000_0002

Bits	Descriptions	
[31:4]	Reserved	Reserved.
[3:0]	PROG_TIMING_STEP	The programming time can be configured to 10us, 20us, 30us, 40us, 50us, 60us, 70us, 80us, 90us, 100us,110us, 120us, 130us, 140us, 150us, 160us and the default is 40us. 4'b0000: 10 us 4'b0001: 20 us 4'b0010: 30us(default) .... 4'b1111: 160 us Default value: 4'b0010

3.4.5.6 OPERATE_DATA_0

Register	Offset	R/W	Description	Reset value
OPERATE_DATA_0	OTP_BA+0x14	R/W	Read/write OTP data[31:0]	0x0000_0000

Bits	Descriptions	
[31:0]	DATA_0	Read/write OTP data[31:0] Default value: 32’b0

3.4.5.7 OPERATE_TRG

Register	Offset	R/W	Description	Reset value
OPERATE_TRG	OTP_BA+0x24	R/W	Indirect TRG of software in read, write, and DMA modes, and checking the operating status.	0x0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	TRG	Similar to the FMC, the software writes 1 to initiate indirect read, write, and DMA triggers. The hardware automatically clears the flag to 0 upon completion. When TRG is 1, it indicates the operation is busy and needs to wait until TRG is 0 before operating again. 1'b0: No software indirect operation is currently in progress or has completed. 1'b1: The current software indirect operation is busy. Default value: 1’b0 Note that in all illegal operations, TRG is prohibited from being pulled high by hardware

3.5 Firmware Encryption

3.5.1 Overview

In order to prevent a third party from maliciously copying the software code of a mature product on the market, directly omitting the cost of software development and forming an improper advantage, the PAN271x chip has an encryption function.

3.5.2 Features

- Support for anti-injection protection, with a protected area of 256 bytes.
  - Anti-injection protection, with its code_offset stored in non-volatile OTP info memory and automatically loaded at power-up.
  - Anti-injection protection, with its switch stored in OTP info and automatically loaded at power-up.
- The debug encryption key is stored in OTP info and automatically loaded at power-up.
- The debug encryption switch is stored in OTP info and automatically loaded at power-up.
- Before debug encryption, all address spaces are accessible.
- After anti-injection protection, the protected code area will never be accessible if it is unauthorized.
- After debug encryption:
  - If decryption fails, only the 0x4008_0000 to 0x4008_FFFF region is accessible.
  - If decryption succeeds, all address space is accessible (excluding encrypted areas).

3.5.3 Block Diagram

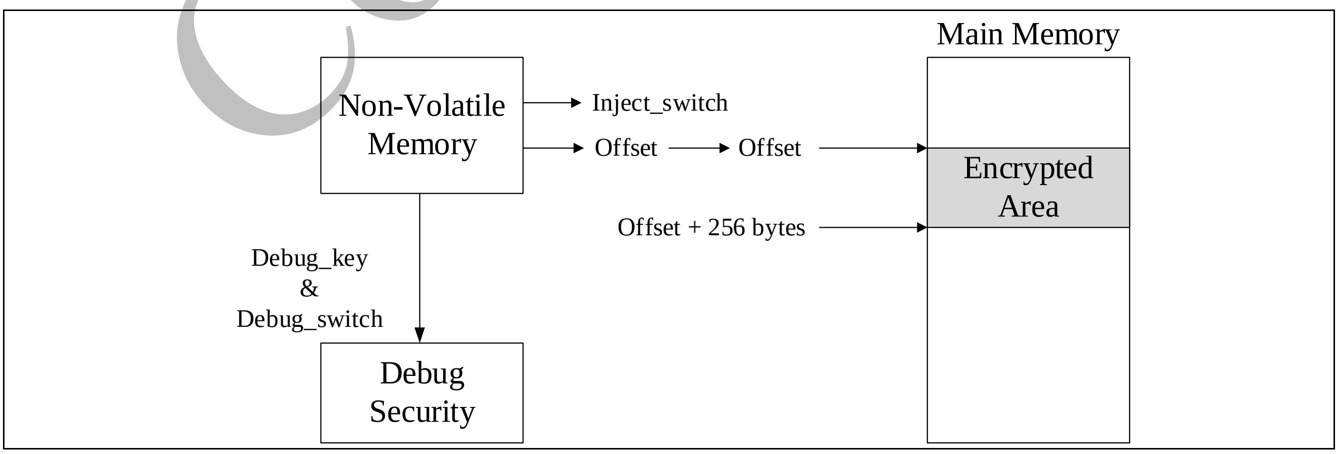


Figure 3-7 Encryption Block Diagram

The PAN271x provides two security measures: debug security and program anti-injection security. After power-on reset, the chip's security logic holds the CPU bus and automatically loads security information stored in non-volatile memory, such as keys, switches, and offsets. After the load is completed, release the bus. See below for details.

3.5.4 Program Anti-injection Security

The PAN271x main memory is an embedded OTP design, making it difficult for hackers to physically crack the encrypted area. Therefore, the method of theft involves injecting malicious programs into the RAM or OTP. This program can then read the encrypted code via the serial port, etc.

For example, suppose the encrypted area is A. The hacker embeds malicious program into another area B, then read the code within area A as data through area B. Therefore, to prevent malicious programs from illegally reading code, the following actions must be taken:

1. Design Principles for Design House:
    1) Hardware must ensure that programs in Area B (any non-encrypted area) cannot read Area A (the encrypted area) as data.
    2) To ensure the proper execution of legitimate applications, the following actions must be taken:
        ① Programs in Area B can access any data or instruction outside Area A.
        ② Programs in Area A can access data or instructions in any area.
2. Principles for User Program Development:

    The user's application must not provide complete print or serial port programs in Area A. Programs in Area A must be able to access data and instructions across all regions, otherwise the CPU will encounter an exception. If a complete print or serial port program exists in Area A, the hacker could simply execute the program in Area A to legitimately steal data.

3.5.5 Debug Security

Debug security applies to debug interfaces. Specific details are as follows:

1. Area A (the encrypted area) cannot be read by debug under any circumstances.
2. There are two situations for non-encrypted areas:
    1) If debug is unencrypted, the user can access all content outside the encrypted area by debug.

2) If debug is encrypted, two situations exist:

   ① Debug decryption succeeds, the user can use debug to access all content outside the encrypted area.

   ② Debug decryption fails, the user can only use debug to access the address space from 0x4008_0000 to 0x4008_FFFF for decryption. Accessing any other address will result in hardware tampering with the bus.

3.5.6 Chip_Security Register Map

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

Register	Offset	R/W	Description	Reset Value
CHIPS_BA address： 0x4008_0000 chip_security address				
VERIFI_DEBUG_KEY0	CHIP_S+0x00	R/W	Software verification debug ciphertext [31:0]	0x0000_0000
VERIFI_DEBUG_KEY1	CHIP_S+0x04	R/W	Software verification debug ciphertext [63:32]	0x0000_0000
VERIFI_DEBUG_KEY2	CHIP_S+0x08	R/W	Software verification debug ciphertext [95:64]	0x0000_0000
VERIFI_DEBUG_KEY3	CHIP_S+0x0c	R/W	Software verification debug ciphertext [127:96]	0x0000_0000

3.5.7 Chip_Security Register Description

3.5.7.1 VERIFY_DEBUG_KEY0

Register	Offset	R/W	Description	Reset Value
VERIFI_DEBUG_KEY0	CHIPS_BA+0x00	R/W	Software verification debug ciphertext	0x0000_0000

Bits	Descriptions	
[31:0]	VERIFY_DEBUG_KEY0	Software verification debug ciphertext [31:0] *Default value：32’b0*

3.5.7.2 VERIFY_DEBUG_KEY1

Register	Offset	R/W	Description	Reset Value
VERIFI_DEBUG_KEY1	CHIPS_BA+0x04	R/W	Software verification debug ciphertext	0x0000_0000

Bits	Descriptions	
[31:0]	VERIFY_DEBUG_KEY1	Software verification debug ciphertext [63:32] *Default value：32’b0*

3.5.7.3 VERIFY_DEBUG_KEY2

Register	Offset	R/W	Description	Reset Value
VERIFI_DEBUG_KEY2	CHIPS_BA+0x08	R/W	Software verification debug ciphertext	0x0000_0000

Bits	Descriptions	
[31:0]	VERIFY_DEBUG_KEY2	Software verification debug ciphertext [95:64] *Default value：32’b0*

3.5.7.4 VERIFY_DEBUG_KEY3

Register	Offset	R/W	Description	Reset Value
VERIFI_DEBUG_KEY3	CHIPS_BA+0x0c	R/W	Software verification debug ciphertext	0x0000_0000

Bits	Descriptions	
[31:0]	VERIFY_DEBUG_KEY3	Software verification debug ciphertext [127:96] *Default value：32’b0*

3.6 General Purpose I/O (GPIO)

3.6.1 Overview

The PAN271x series has up to 19 General Purpose I/O pins to be shared with other function pins depending on the chip configuration. These 19 pins are arranged in 3 ports named as P0, P1, P2. Each of the 19 pins is independent and has the corresponding register bits to control the pin mode function and data. Different package formats correspond to different GPIO numbers. The I/O type of each pin can be configured by software individually as Input, Push-pull output, Open-drain output, or Quasi-bidirectional mode. After the chip is reset, the I/O mode of all pins is stay in input mode and each port data register Px_DOUT[n] resets to 1. For Quasi-bidirectional mode, each I/O pin is equipped with a very weak individual pull-up resistor about 50 kΩ for VDD is from 1.8 V to 5.5 V.

3.6.2 Features

- Four I/O modes:
  - Quasi-bidirectional mode
  - Push-pull output
  - Open-drain output
  - Input-only with high impendence
- Quasi-bidirectional TTL/Schmitt trigger input mode selected by SYS_Px_MFP[23:16]
- I/O pin configured as interrupt source with edge/level setting
- I/O pin internal pull-up resistor enabled only in Quasi-bidirectional I/O mode
- Enabling the pin interrupt function will also enable the pin wake-up function
- High driver and high sink I/O mode suppor
- Support debounce function

3.6.3 Block Diagram

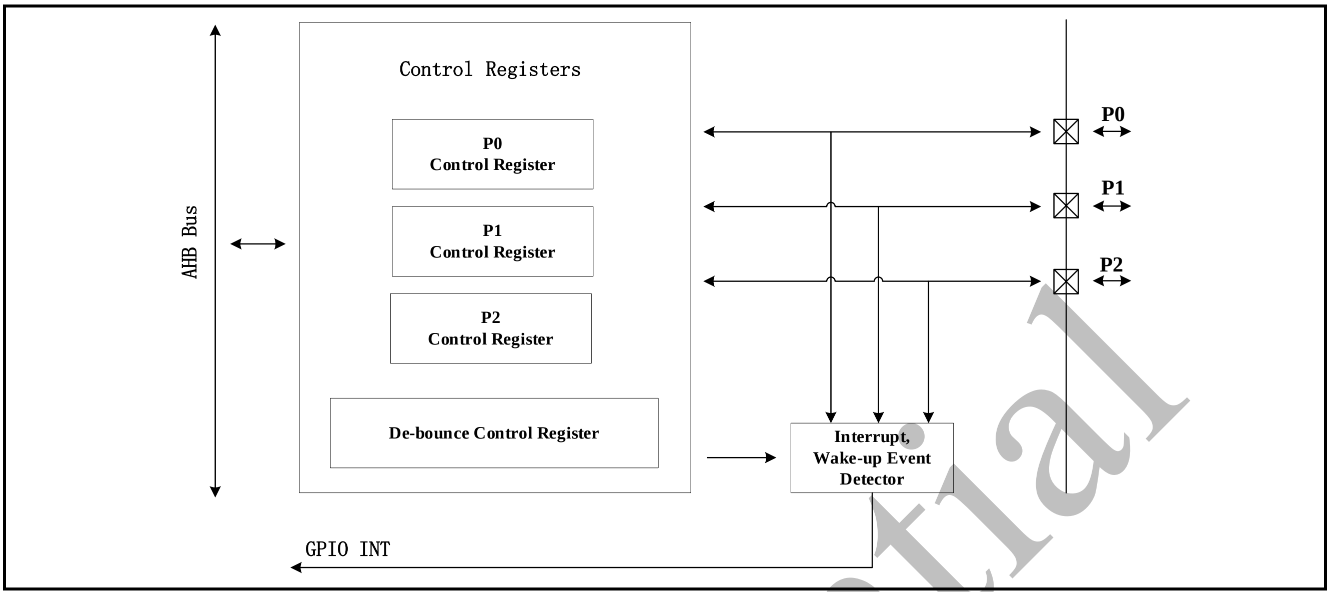


Figure 3-8 GPIO Controller Block Diagram

3.6.4 Basic Configuration

The GPIO pin functions are configured in *SYS_P0_MFP*, *SYS_P1_MFP*, *SYS_P2_MFP* registers.

3.6.5 Functional Description

The PAD diagram is introduced in Figure 3-9.

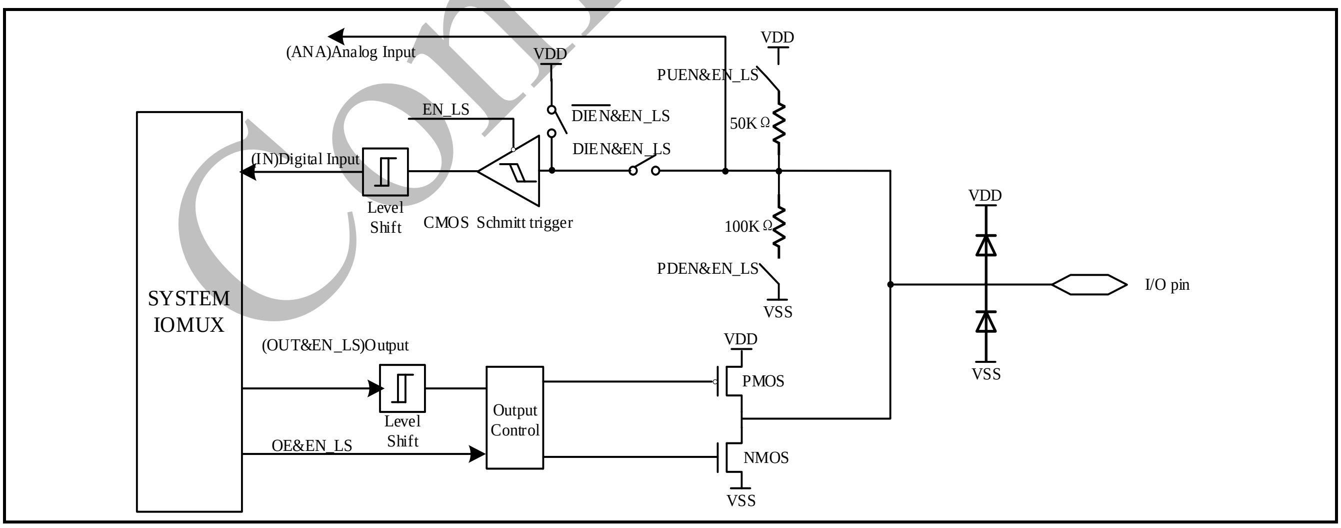


Figure 3-9 PAD Diagram

3.6.5.1 Input Mode

Set *MODEn* (Px_MODE[2n+1:2n]) to 00 as the Px.n pin is in Input mode and the I/O pin is in tri-state (high impedance) without output drive capability. The *PIN* (Px_PIN[n]) value reflects the status of the corresponding port pins.

3.6.5.2 Push-pull Output Mode

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

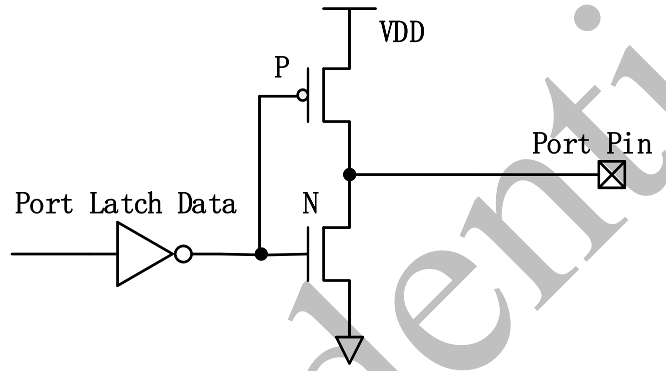


Figure 3-10 Push-Pull Output

3.6.5.3 Open-drain Output Mode

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

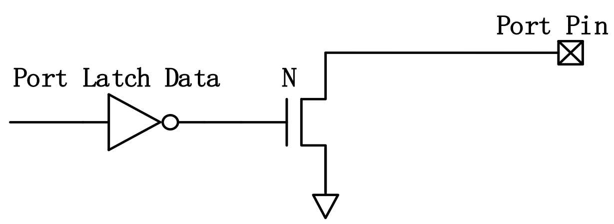


Figure 3-11 Open-Drain Output

3.6.5.4 Quasi-bidirectional Mode

Set *MODEn* (Px_MODE[2n+1:2n]) to 11 as the Px.n pin is in Quasi-bidirectional mode and the I/O pin supports digital output and input function at the same time but the source current is only up to hundreds uA. Before the digital input function is performed the corresponding *DOUT* (Px_DOUT[n]) bit must be set to 1. If the bit value in the corresponding *DOUT* (Px_DOUT[n]) bit is 0, the pin drive a low output on the pin. If the bit value in the corresponding *DOUT* (Px_DOUT[n]) bit is 1, the pin status is controlled by internal pull-up resistor.

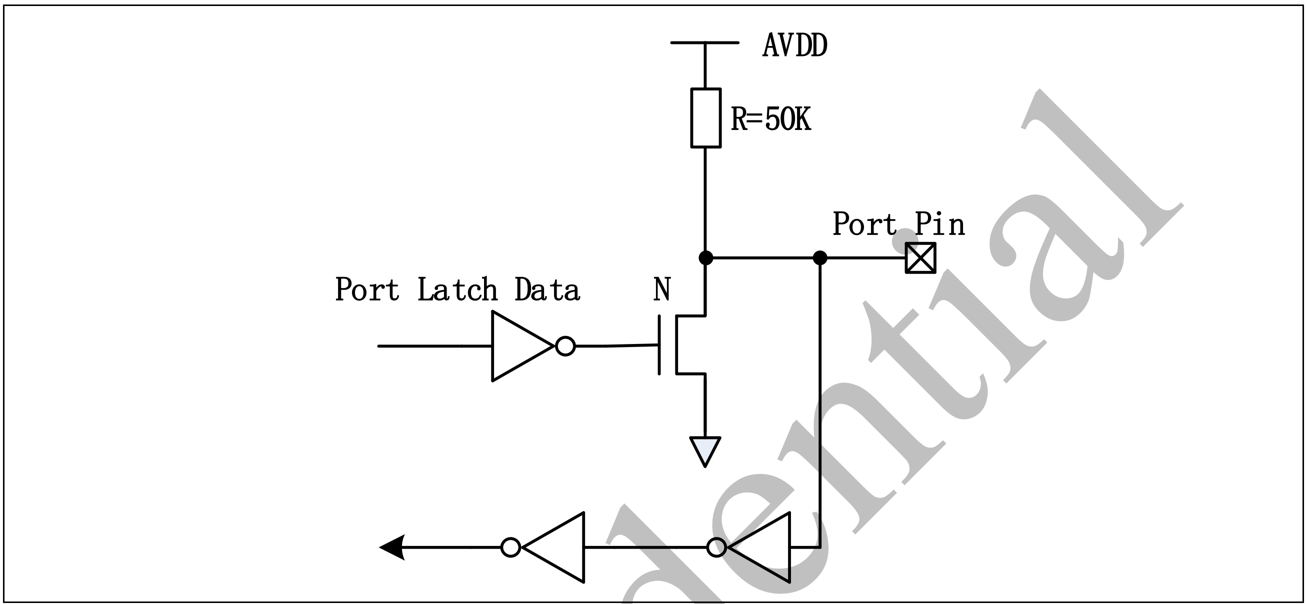


Figure 3-12 Quasi-Bidirectional I/O Mode

3.6.6 GPIO Interrupt and Wake-up Function

Each GPIO pin can be set as chip interrupt source by setting correlative RHIEN (*Px_INTEN*[n+16])/ FLIEN (*Px_INTEN*[n]) bit and TYPE (*Px_INTTYPE*[n]). There are five types of interrupt condition can be selected: low level trigger, high level trigger, falling edge trigger, rising edge trigger and both rising and falling edge trigger. For edge trigger condition, user can enable input signal de-bounce function to prevent unexpected interrupt happened which caused by noise. The de-bounce clock source and sampling cycle period can be set through DBCLKSRC (*GPIO_DBCTL*[4]) and DBCLKSEL (*GPIO_DBCTL*[3:0]) register.

The GPIO can also be the chip wake-up source when chip enters Idle/Power-down mode. The setting of wake-up trigger condition is the same as GPIO interrupt trigger.

3.6.7 GPIO Register Map

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

Register	Offset	R/W	Description	Reset Value
GPIO Base Address: GP_BA = 0x4002_0000				
P0_MODE	GP_BA+0x000	R/W	P0 I/O Mode Control	0x0000_0000
P0_DINOFF	GP_BA+0x004	R/W	P0 Digital Input Path Disable Control	0x00F4_000B
P0_DOUT	GP_BA+0x008	R/W	P0 Data Output Value	0x0000_00FF
P0_PIN	GP_BA+0x010	R	P0 Pin Value	0x0000_00XX
P0_DBEN	GP_BA+0x014	R/W	P0 De-bounce Enable Control	0x0000_0000
P0_INTTYPE	GP_BA+0x018	R/W	P0 Interrupt Mode Control	0x0000_0000
P0_INTEN	GP_BA+0x01C	R/W	P0 Interrupt Enable Control	0x0000_0000
P0_INTSRC	GP_BA+0x020	R/W	P0 Interrupt Source Flag	0x0000_0000
P1_MODE	GP_BA+0x040	R/W	P1 I/O Mode Control	0x0000_0000
P1_DINOFF	GP_BA+0x044	R/W	P1 Digital Input Path Disable Control	0x00FF_0000
P1_DOUT	GP_BA+0x048	R/W	P1 Data Output Value	0x0000_00FF
P1_PIN	GP_BA+0x050	R	P1 Pin Value	0x0000_00XX
P1_DBEN	GP_BA+0x054	R/W	P1 De-bounce Enable Control	0x0000_0000
P1_INTTYPE	GP_BA+0x058	R/W	P1 Interrupt Mode Control	0x0000_0000
P1_INTEN	GP_BA+0x05C	R/W	P1 Interrupt Enable Control	0x0000_0000
P1_INTSRC	GP_BA+0x060	R/W	P1 Interrupt Source Flag	0x0000_0000
P2_MODE	GP_BA+0x080	R/W	P2 I/O Mode Control	0x0000_0000
P2_DINOFF	GP_BA+0x084	R/W	P2 Digital Input Path Disable Control	0x0007_0000
P2_DOUT	GP_BA+0x088	R/W	P2 Data Output Value	0x0000_0007
P2_PIN	GP_BA+0x090	R	P2 Pin Value	0x0000_00XX
P2_DBEN	GP_BA+0x094	R/W	P2 De-bounce Enable Control	0x0000_0000
P2_INTTYPE	GP_BA+0x098	R/W	P2 Interrupt Mode Control	0x0000_0000
P2_INTEN	GP_BA+0x09C	R/W	P2 Interrupt Enable Control	0x0000_0000
P2_INTSRC	GP_BA+0x0A0	R/W	P2 Interrupt Source Flag	0x0000_0000
GPIO_DBCTL	GP_BA+0x180	R/W	De-bounce Cycle Control	0x0000_0700

3.6.8 GPIO Register Description

3.6.8.1 Port 0-2 I/O Mode Control (Px_MODE)

Register	Offset	R/W	Description	Reset Value
P0_MODE	GP_BA+0x000	R/W	P0 I/O Mode Control	0x0000_0000
P1_MODE	GP_BA+0x040	R/W	P1 I/O Mode Control	0x0000_0000
P2_MODE	GP_BA+0x080	R/W	P2 I/O Mode Control	0x0000_0000

Bits	Descriptions	
[31:16]	Reserved	Reserved.
[2n+1:2n] n=0,1..7	MODEn	Port 0-2 I/O Pin[N] Mode Control Determine each I/O mode of Px.n pins. 00: Px.n is in Input mode. 01: Px.n is in Push-pull Output mode. 10: Px.n is in Open-drain Output mode. 11: Px.n is in Quasi-bidirectional mode. **Note:** Max. n=7 for port 0. Max. n=7 for port 1. Max. n=2 for port 2.

3.6.8.2 Port 0-2 Digital Input Path Disable Control (Px_DINOFF)

Register	Offset	R/W	Description	Reset Value
P0_DINOFF	GP_BA+0x004	R/W	P0 Digital Input Path Disable Control	0x00F4_000B
P1_DINOFF	GP_BA+0x044	R/W	P1 Digital Input Path Disable Control	0x00FF_0000
P2_DINOFF	GP_BA+0x084	R/W	P2 Digital Input Path Disable Control	0x0007_0000

Bits	Descriptions	
[31:24] n=0,1..7	ANA_CHEN[n]	GPIO analog channel enable 0: Disable 1: Enable **Note:** Max. n=7 for port 0. Max. n=7 for port 1. Max. n=2 for port 2.
[n+16] n=0,1..7	DINOFF[n]	Port 0-2 Pin[N] Digital Input Path Disable Control Each of these bits is used to control if the digital input path of corresponding Px.n pin is disabled. If input is analog signal, users can disable Px.n digital input path to avoid input current leakage. 0: Px.n digital input path Enabled. 1: Px.n digital input path Disabled (digital input tied to low). **Note1:** Max. n=7 for port 0. Max. n=7 for port 1. Max. n=2 for port 2.
[15:8]	PDEN[n]	Port 0-2 Pin[N] Digital Pull Down Path Enable Control Each of these bits is used to control if the digital pull down path of corresponding Px.n pin is enabled. 0: Px.n digital pull down path Disabled. 1: Px.n digital pull down path Enabled.
[n] n=0,1..7	PUEN[n]	Port 0-2 Pin[N] Digital Pull Up Path Enable Control Each of these bits is used to control if the digital pull up path of corresponding Px.n pin is enabled. 0: Px.n digital pull up path Disabled. 1: Px.n digital pull up path Enabled.

3.6.8.3 Port 0-2 Data Output Value (Px_DOUT)

Register	Offset	R/W	Description	Reset Value
P0_DOUT	GP_BA+0x008	R/W	P0 Data Output Value	0x0000_00FF
P1_DOUT	GP_BA+0x048	R/W	P1 Data Output Value	0x0000_00FF
P2_DOUT	GP_BA+0x088	R/W	P2 Data Output Value	0x0000_0007

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

3.6.8.4 Port 0-2 Pin Value (Px_PIN)

Register	Offset	R/W	Description	Reset Value
P0_PIN	GP_BA+0x010	R	P0 Pin Value	0x0000_00XX
P1_PIN	GP_BA+0x050	R	P1 Pin Value	0x0000_00XX
P2_PIN	GP_BA+0x090	R	P2 Pin Value	0x0000_00XX

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[n] n=0,1..7	PIN[n]	Port 0-2 Pin[N] Pin Value Each bit of the register reflects the actual status of the respective Px.n pin. If the bit is 1, it indicates the corresponding pin status is high; else the pin status is low. **Note:** Max. n=7 for port 0. Max. n=7 for port 1. Max. n=2 for port 2.

3.6.8.5 Port 0-2 De-bounce Enable Control (Px_DBEN)

Register	Offset	R/W	Description	Reset Value
P0_DBEN	GP_BA+0x014	R/W	P0 De-bounce Enable Control	0x0000_0000
P1_DBEN	GP_BA+0x054	R/W	P1 De-bounce Enable Control	0x0000_0000
P2_DBEN	GP_BA+0x094	R/W	P2 De-bounce Enable Control	0x0000_0000

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

3.6.8.6 Port 0-2 Interrupt Mode Control (Px_INTTYPE)

Register	Offset	R/W	Description	Reset Value
P0_INTTYPE	GP_BA+0x018	R/W	P0 Interrupt Mode Control	0x0000_0000
P1_INTTYPE	GP_BA+0x058	R/W	P1 Interrupt Mode Control	0x0000_0000
P2_INTTYPE	GP_BA+0x098	R/W	P2 Interrupt Mode Control	0x0000_0000

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

3.6.8.7 Port 0-2 Interrupt Enable Control (Px_INTEN)

Register	Offset	R/W	Description	Reset Value
P0_INTEN	GP_BA+0x01C	R/W	P0 Interrupt Enable Control	0x0000_0000
P1_INTEN	GP_BA+0x05C	R/W	P1 Interrupt Enable Control	0x0000_0000
P2_INTEN	GP_BA+0x09C	R/W	P2 Interrupt Enable Control	0x0000_0000

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

3.6.8.8 Port 0-2 Interrupt Source Flag (Px_INTSRC)

Register	Offset	R/W	Description	Reset Value
P0_INTSRC	GP_BA+0x020	R/W	P0 Interrupt Source Flag	0x0000_0000
P1_INTSRC	GP_BA+0x060	R/W	P1 Interrupt Source Flag	0x0000_0000
P2_INTSRC	GP_BA+0x0A0	R/W	P2 Interrupt Source Flag	0x0000_0000

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

3.6.8.9 Interrupt De-bounce Cycle Control (GPIO_DBCTL)

Register	Offset	R/W	Description	Reset Value
GPIO_DBCTL	GP_BA+0x180	R/W	De-bounce Cycle Control	0x0000_0700

Bits	Descriptions	
[31:24]	P1_BDR_EN	Port1 p10~p17 infrared IO high-drive enable, apply to all IOs of the Port1. 0: Disable 1: Enable
[23:22]	Reserved	Reserved.
[21:20]	P2_DRIVE_SEL	Port2 GPIO Drive strength control, apply to p20, p21, p22 00: Minimum 01: Second minimum 10: Big 11: Maximum
[19:18]	P1_DRIVE_SEL	Port1 GPIO Drive strength control, apply to p10, p11, p12, p13, p14, p15, p16, p17 00: Minimum 01: Second minimum 10: Big 11: Maximum
[17:16]	P0_DRIVE_SEL	Port0 GPIO Drive strength control, apply to p00, p01, p02, p03, p04, p05, p06, p07 00: Minimum 01: Second minimum 10: Big 11: Maximum
[15:14]	Reserved	Reserved.
[n+8] n=0,1,2	Pn_EN	P0-P2 GPIO enable control 0: Pn port is disable 1: Pn port is enable **Note:** n=0 for p0 n=1 for p1 n=2 for p2
[7:5]	Reserved	Reserved.
[4]	DBCLKSRC	De-bounce Counter Clock Source Selection 0: De-bounce counter clock source is AHB_CLK. 1: De-bounce counter clock source is 32 KHz internal low speed RC oscillator (RCL).
[3:0]	DBCLKSEL	De-bounce Sampling Cycle Selection 0000: Sample interrupt input once per 1 clock. 0001: Sample interrupt input once per 2 clocks. 0010: Sample interrupt input once per 4 clocks. 0011: Sample interrupt input once per 8 clocks. 0100: Sample interrupt input once per 16 clocks. 0101: Sample interrupt input once per 32 clocks. 0110: Sample interrupt input once per 64 clocks. 0111: Sample interrupt input once per 128 clocks. 1000: Sample interrupt input once per 256 clocks. 1001: Sample interrupt input once per 2*256 clocks. 1010: Sample interrupt input once per 4*256 clocks. 1011: Sample interrupt input once per 8*256 clocks. 1100: Sample interrupt input once per 16*256 clocks. 1101: Sample interrupt input once per 32*256 clocks.

		1110: Sample interrupt input once per 64*256 clocks. 1111: Sample interrupt input once per 128*256 clocks.

3.7 Universal Serial Bus (USB)

3.7.1 Overview

The USB device controller is compatible with USB 2.0 Full-speed(12 Mbps) function.

3.7.2 Features

- Conforms to 1.1 and 2.0 revision of the USB specification.
- Conforms to 1.1 and 2.0 revision of the USB specification.
- Support 6 endpoints (include endpoint 0).
- EP0 FIFO: 32Bytes
- BULK Transactions: EP1/2/3/4 IN FIFO: 64 Bytes， EP1/2/3/4 OUT FIFO: 64Bytes
- ISOCHRONOUS Transactions: EP1/2/3 IN FIFO: 128 Bytes; EP1/2 OUT FIFO: 128 Bytes, EP3 OUT FIFO: 64 Bytes , EP4/5 OUT/IN FIFO: 64 Bytes
- Serial Interface Engine
- Support full speed devices.
- NRZI decoding/encoding
- Bit stuffing/stripping
- 32bytes CRC checking/generation
- On-chip pull-up resistor on DP(1.5kΩ) / DM(150kΩ)
- Support plug in interrupt and plug out interrupt.
- Data toggle synchronization mechanism
- Suspend and resume power management functions.

3.7.3 Block Diagram

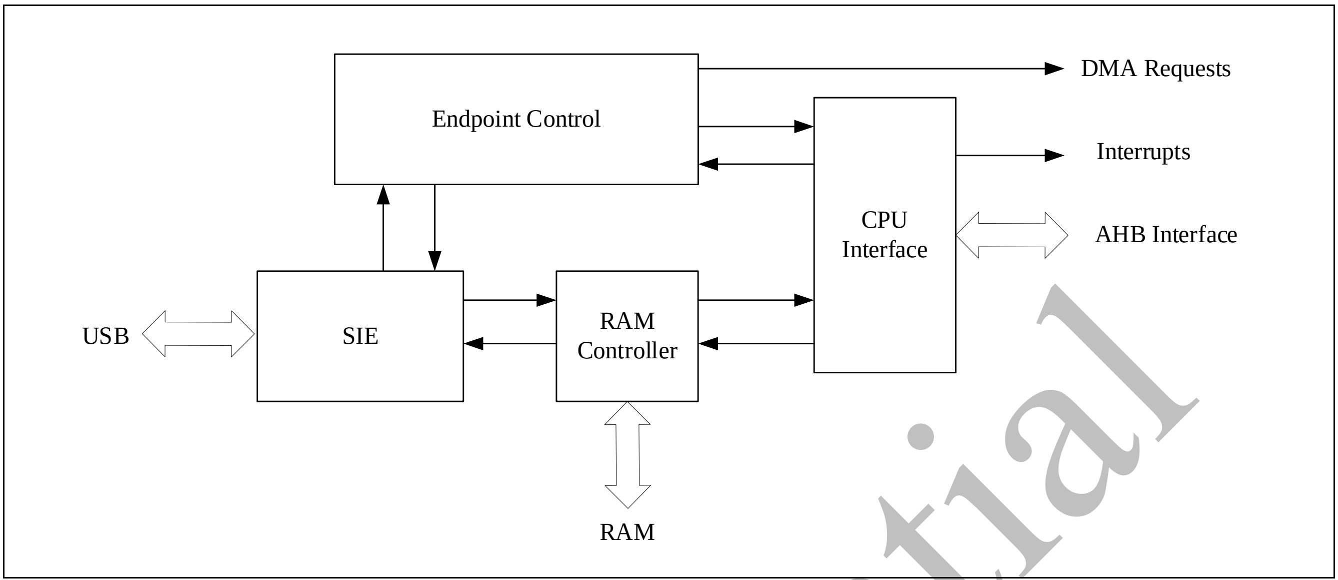


Figure 3-13 USB Block Diagram

3.7.4 Functional Description

3.7.4.1 Support BULK/ISOCHRONOUS transactions

This USB complies with the usb2.0 full_speed protocol and supports bulk and isochronous transmission.

3.7.4.2 Support plug in/out interrupt

Enable DM pull-up resistor (ANA_MISC[28]=1), When the USB is plugged in, {DP,DM}={1,0} will generate a PUPIF interrupt for more than 1ms. Reading the IntrUSB, if {DP,DM}={1,0}, it means that the USB is plugged in; If {DP,DM}={1,1} exceeds 1ms, PUPIF interrupt will be generated. Reading the IntrUSB, if {DP,DM}={1,1}, it means USB is plugged out.

3.7.5 USB Register Map

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

Register	Offset	R/W	Description	Reset Value
USB Base Address: USB_BA				
FADDR	USB_BA+0x00	R/W	Function address register	0x00
POWER	USB_BA+0x01	R/W	Power management register	0x00
INTRIN1	USB_BA+0x02	R	Interrupt register for Endpoint 0 plus IN Endpoints 1 to 4	0x00
INTRIN2	USB_BA+0x03	R	reserved	0x00
INTROUT1	USB_BA+0x04	R	Interrupt register for OUT Endpoints 1 to 4	0x00
INTROUT2	USB_BA+0x05	R	reserved	0x00
INTRUSB	USB_BA+0x06	R	Interrupt register for common USB interrupts	0x00
INTRIN1E	USB_BA+0x07	R/W	Interrupt enable register for IntrIn1	0x3F
INTRIN2E	USB_BA+0x08	R/W	reserved	0xFF
INTROUT1E	USB_BA+0x09	R/W	Interrupt enable register for IntrOut1	0x3E
INTROUT2E	USB_BA+0x0A	R/W	reserved	0xFF
INTRUSBE	USB_BA+0x0B	R/W	Interrupt enable register for IntrUSB	0x06
FRAME1	USB_BA+0x0C	R/W	Frame number bits 0 to 7	0x0000_0000
FRAME2	USB_BA+0x0D	R/W	Frame number bits 8 to 10	0x0000_0000
INDEX	USB_BA+0x0E	R/W	Index register for selecting the endpoint status and control registers	0x0000_0000
INMAXP	USB_BA+0x10	R/W	Maximum packet size for IN endpoint (Index register set to select Endpoints 1 – 4 only)	0x0000_0000
CSR0	USB_BA+0x11		Control Status register for Endpoint 0 (Index register set to select Endpoint 0)	
INCSR1		R/W	Control Status register 1 for IN endpoint (Index register set to select Endpoints 1 – 4)	0x0000_0000
INCSR2	USB_BA+0x12	R/W	Control Status register 2 for IN endpoint (Index register set to select Endpoints 1 – 4 only)	0x0000_0000
OUTMAXP	USB_BA+0x13	R/W	Maximum packet size for OUT endpoint (Index register set to select Endpoints 1 – 4 only)	0x0000_0000
OUTCSR1	USB_BA+0x14	R/W	Control Status register 1 for OUT endpoint (Index register set to select Endpoints 1 – 4 only)	0x00FF_0000
OUTCSR2	USB_BA+0x15	R/W	Control Status register 2 for OUT endpoint (Index register set to select Endpoints 1 – 4 only)	0x0000_0000
COUNT0	USB_BA+0x16	R/W	Number of received bytes in Endpoint 0 FIFO (Index register set to select Endpoint 0)	0x0000_0000
OUTCOUNT1			Number of bytes in OUT endpoint FIFO (lower byte) (Index register set to select Endpoints 1 – 4)	0x0000_0000
OUTCOUNT2	USB_BA+0x17	R	Number of bytes in OUT endpoint FIFO (upper byte) (Index register set to select Endpoints 1 – 4 only)	0x0000_0000
USB_DEBUG	USB_BA+0x1F	R/W	USB Debug control	0x0000_0000
FIFOx	USB_BA+0x20~2F	R/W	FIFOs for Endpoints 0 to 5	0x0000_0000

3.7.6 USB Register Description

3.7.6.1 FADDR

Register	Offset	R/W	Description	Reset Value
FADDR	USB_BA+0x00	R/W	Function address register	0x00

Bits	Descriptions	
[7]	UPDATE	Set when FADDR is written. Cleared when the new address takes effect (at the end of the current transfer).
[6:0]	FUNC ADDR	The function address.

3.7.6.2 POWER

Register	Offset	R/W	Description	Reset Value
POWER	USB_BA+0x01	R/W	Power management register.	0x00

Bits	Descriptions	
[7]	ISO UPDATE	When set by the CPU the USB will wait for an SOF token from the time InPktRdy is set before sending the packet. If an IN token is received before an SOF token, then a zero length data packet will be sent. This bit is only used by endpoints performing Isochronous transfers.
[6:4]	Reserved	Reserved.
[3]	RESET	This read only bit is set while Reset signaling is present on the bus.
[2]	RESUME	Set by the CPU to generate Resume signaling when the function is in Suspend mode. The CPU should clear this bit after 10 ms (a maximum of 15 ms) to end Resume signaling.
[1]	SUSPEND MODE	Set by the USB when Suspend mode is entered. Cleared when the CPU reads the interrupt register, or sets the Resume bit of this register.
[0]	ENABLE SUSPEND	Set by the CPU to enable entry into Suspend mode when Suspend signaling is received on the bus.

3.7.6.3 INTRIN1

Register	Offset	R/W	Description	Reset Value
INTRIN1	USB_BA+0x02	R	Interrupt register for Endpoint 0 plus IN Endpoints 1 to 3. **Note**: All active interrupts will be cleared when this register is read.	0x00

Bits	Descriptions	
[7:6]	Reserved	Reserved
[5]	EP5	IN Endpoint 5 interrupt.
[4]	EP4	IN Endpoint 4 interrupt.
[3]	EP3	IN Endpoint 3 interrupt.
[2]	EP2	IN Endpoint 2 interrupt.
[1]	EP1	IN Endpoint 1 interrupt.
[0]	EP0	Endpoint 0 interrupt.

3.7.6.4 INTRIN2

Register	Offset	R/W	Description	Reset Value
INTRIN2	USB_BA+0x03	R	Reserved	0x00

Bits	Descriptions	
[7:0]	Reserved	Reserved

3.7.6.5 INTROUT1

Register	Offset	R/W	Description	Reset Value
INTROUT1	USB_BA+0x04	R	Interrupt register for OUT Endpoints 1 to 3. **Note:** All active interrupts will be cleared when this register is read.	0x00

Bits	Descriptions	
[7:6]	Reserved	Reserved
[5]	EP5	OUT Endpoint 5 interrupt.
[4]	EP4	OUT Endpoint 4 interrupt.
[3]	EP3	OUT Endpoint 3 interrupt.
[2]	EP2	OUT Endpoint 2 interrupt.
[1]	EP1	OUT Endpoint 1 interrupt.
[0]	Reserved	Reserved

3.7.6.6 INTROUT2

Register	Offset	R/W	Description	Reset Value
INTROUT2	USB_BA+0x05	R	Reserved	0x00

Bits	Descriptions	
[7:0]	Reserved	Reserved

3.7.6.7 INTRUSB

Register	Offset	R/W	Description	Reset Value
INTRUSB	USB_BA+0x06	R	Interrupt register for common USB interrupts	0x00

Bits	Descriptions	
[7]	DP	D+ status
[6]	DM	D- status
[5]	Reserved	Reserved
[4]	PUPIF	Plug in or plug out interrupt happens.
[3]	SOF	Set at the start of each frame.
[2]	RESET	Set when Reset signaling is detected on the bus.
[1]	RESUME	Set when Resume signaling is detected on the bus while the USB is in Suspend mode.
[0]	SUSPEND	Set when Suspend signaling is detected on the bus.

3.7.6.8 INTRIN1E

Register	Offset	R/W	Description	Reset Value
INTRIN1E	USB_BA+0x07	R/W	Interrupt enable register for INTRIN1. On reset, the bits corresponding to Endpoint 0 and the IN endpoints included in the design are set to 1, while the remaining bits are set to 0.	0x3F

Bits	Descriptions	
[7:6]	Reserved	Reserved
[5]	EP5	IN Endpoint 5 interrupt enable bit.
[4]	EP4	IN Endpoint 4 interrupt enable bit.
[3]	EP3	IN Endpoint 3 interrupt enable bit.
[2]	EP2	IN Endpoint 2 interrupt enable bit.
[1]	EP1	IN Endpoint 1 interrupt enable bit.
[0]	EP0	IN Endpoint 0 interrupt enable bit.

3.7.6.9 INTRIN2E

Register	Offset	R/W	Description	Reset Value
INTRIN2E	USB_BA+0x08	R/W	Reserved	0xFF

Bits	Descriptions	
[7:0]	Reserved	Reserved

3.7.6.10 INTROUT1E

Register	Offset	R/W	Description	Reset Value
INTROUT1E	USB_BA+0x09	R/W	Interrupt enable register for IntrOut1.On reset, the bits corresponding to the OUT endpoints included in the design are set to 1, while the remaining bits are set to 0.	0x3E

Bits	Descriptions	
[7:6]	Reserved	Reserved
[5]	EP5	OUT Endpoint 5 interrupt enable bit.
[4]	EP4	OUT Endpoint 4 interrupt enable bit.
[3]	EP3	OUT Endpoint 3 interrupt enable bit.
[2]	EP2	OUT Endpoint 2 interrupt enable bit.
[1]	EP1	OUT Endpoint 1 interrupt enable bit.
[0]	Reserved	Reserved

3.7.6.11 INTROUT2E

Register	Offset	R/W	Description	Reset Value
INTROUT2E	USB_BA+0x0A	R/W	reserved	0xFF

Bits	Descriptions	
[7:0]	Reserved	Reserved

3.7.6.12 INTRUSBE

Register	Offset	R/W	Description	Reset Value
INTRUSBE	USB_BA+0x0B	R/W	Interrupt enable register for IntrUSB	0x26

Bits	Descriptions	
[7:6]	Reserved	Reserved.
[5]	SOF_MISS_DET_EN	Detect SOF token is missed at the start of frame enable bit.
[4]	PUPIF	Plug in or plug out interrupt enable bit.
[3]	SOF	Sof interrupt enable bit.
[2]	RESET	Reset interrupt enable bit.
[1]	RESUME	Resume interrupt enable bit.
[0]	SUSPEND	Suspend interrupt enable bit.

3.7.6.13 FRAME1

Register	Offset	R/W	Description	Reset Value
FRAME1	USB_BA+0x0C	R	Frame number register.	0x00

Bits	Descriptions	
[7:0]	FRAME1	Lower 8 bits of the last received frame number.

3.7.6.14 FRAME2

Register	Offset	R/W	Description	Reset Value
FRAME2	USB_BA+0x0D	R	Frame number register.	0x00

Bits	Descriptions	
[7:3]	Reserved	Reserved
[2:0]	FRAME2	Upper 3 bits of the last received frame number.

3.7.6.15 INDEX

Register	Offset	R/W	Description	Reset Value
INDEX	USB_BA+0x0E	R/W	Choose EndPoints.	0x00

Bits	Descriptions	
[7:4]	Reserved	Reserved
[3:0]	INDEX	Index is a 4-bit register that determines which endpoint control/status registers are accessed. Each IN endpoint and each OUT endpoint have their own set of control/status registers. Only one set of IN control/status and one set of OUT control/status registers appear in the memory map at any one time. Before accessing an endpoint's control/status registers, the endpoint number should be written to the Index register to ensure that the correct control/status registers appear in the memory map.

3.7.6.16 INMAXP

Register	Offset	R/W	Description	Reset Value
INMAXP	USB_BA+0x10	R/W	Maximum Packet Size/Transaction	0x00

Bits	Descriptions	
[7:0]	INMAXP	InMaxP is an 8-bit register that holds the maximum packet size for transactions through the currently-selected IN endpoint – in units of 8 bytes. The value of InMaxP should be less than 16 for EP1/2/3，because max FIFO size is 128。The value of InMaxP should be less than 8 for EP4/5，because max FIFO size is 64。There is an InMaxP register for each IN endpoint (except Endpoint 0).

3.7.6.17 CSR0

Register	Offset	R/W	Description	Reset Value
CSR0	USB_BA+0x11	R/W	CSR0 is an 8-bit register that provides control and status bits for Endpoint 0.	0x00

Bits	Descriptions	
[7]	ServicedSetupEnd	R/W. The CPU writes a 1 to this bit to clear the SetupEnd bit. It is cleared automatically.
[6]	ServicedOutPktRdy	R/W. The CPU writes a 1 to this bit to clear the OutPktRdy bit. It is cleared automatically.
[5]	SendStall	R/W. The CPU writes a 1 to this bit to terminate the current transaction. The STALL handshake will be transmitted and then this bit will be cleared automatically.
[4]	SetupEnd	This bit will be set when a control transaction ends before the DataEnd bit has been set. An interrupt will be generated and the FIFO flushed at this time. The bit is cleared by the CPU writing a 1 to the ServicedSetupEnd bit.
[3]	DataEnd	The CPU sets this bit: 1. When setting InPktRdy for the last data packet. 2. When clearing OutPktRdy after unloading the last data packet. 3. When setting InPktRdy for a zero length data packet. It is cleared automatically.
[2]	SentStall	R/CLEAR. This bit is set when a STALL handshake is transmitted. The CPU should clear this bit.
[1]	InPktRdy	R/W. The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when the data packet has been transmitted. An interrupt is generated when the bit is cleared. It is cleared automatically.
[0]	OutPktRdy	R. This bit is set when a data packet has been received. An interrupt is generated when this bit is set. The CPU clears this bit by setting the ServicedOutPktRdy bit.

3.7.6.18 COUNT0

Register	Offset	R/W	Description	Reset Value
COUNT0	USB_BA+0x16	R	It indicates the number of received data bytes in the Endpoint 0 FIFO	0x00

Bits	Descriptions	
[7:0]	Count0	EndPoint0 OUT Count.

3.7.6.19 INCSR1

Register	Offset	R/W	Description	Reset Value
INCSR1	USB_BA+0x11	R	InCSR1 is an 8-bit register that provides control and status bits for transfers through the currently-selected IN endpoint. There is an InCSR1 register for each IN endpoint (not including Endpoint 0).	0x00

Bits	Descriptions	
[7]	reserved	reserved
[6]	ClrDataTog	R/W. The CPU writes a 1 to this bit to reset the endpoint IN data toggle to 0.
[5]	SentStall	R/CLEAR. This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the InPktRdy bit is cleared (see below). The CPU should clear this bit by reading.
[4]	SendStall	R/W. The CPU writes a 1 to this bit to issue a STALL handshake to an IN token. The CPU clears this bit to terminate the stall condition. This bit has no effect if the IN endpoint is in ISO mode.
[3]	FlushFIFO	R/W. Self-clearing.The CPU writes a 1 to this bit to flush the next packet to be transmitted from the endpoint INFIFO. The FIFO pointer is reset and the InPktRdy bit (below) is cleared. Note: If the FIFO contains two packets, FlushFIFO will need to be set twice to completely clear the FIFO. It is cleared automatically.
[2]	UnderRun	R/CLEAR. In ISO mode, this bit is set when a zero length data packet is sent after receiving an IN token with the InPktRdy bit not set. In Bulk/Interrupt mode, this bit is set when a NAK is returned in response to an IN token. The CPU should clear this bit by reading.
[1]	FIFONotEmpty	R. This bit is set when there is at least 1 packet in the IN FIFO.
[0]	InPktRdy	The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared.

3.7.6.20 INCSR2

Register	Offset	R/W	Description	Reset Value
INCSR2	USB_BA+0x12	R/W	InCSR2 is an 8-bit register that provides further control bits for transfers through the currently-selected IN endpoint. There is an InCSR2 register for each IN endpoint (not including Endpoint 0).	0x00

Bits	Descriptions	
[7]	AutoSet	If the CPU sets this bit then InPktRdy will be automatically set when data of the maximum packet size (value in InMaxP) is loaded into the IN FIFO. If a packet of less than the maximum packet size is loaded, then InPktRdy will have to be set manually
[6]	ISO	The CPU sets this bit to enable the IN endpoint for isochronous transfers, and clears it to enable the IN endpoint for bulk or interrupt transfers.
[5]	Mode	The CPU sets this bit to enable the endpoint direction as IN, and clears it to enable the endpoint direction as OUT. Valid only where the same endpoint FIFO is used for both IN and OUT transactions.
[4]	Reserved	Reserved
[3]	FrcDataTog	The CPU sets this bit to force the endpoint IN data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received. This can be used by interrupt IN endpoints that are used to communicate rate feedback for isochronous endpoints.
[2:0]	Reserved	Reserved

3.7.6.21 OUTMAXP

Register	Offset	R/W	Description	Reset Value
OUTMAXP	USB_BA+0x13	R/W	Maximum Packet Size/transaction for EndPoint1/2/3	0x00

Bits	Descriptions	
[7:0]	OUTMAXP	OutMaxP is an 8-bit register that holds the maximum packet size for transactions through the currently-selected OUT endpoint – in units of 8 bytes(except EndPoint0). The value of OutMaxP should be less than 16 for EP1/2， because max FIFO size is 128。The value of OutMaxP should be less than 8 for EP3/4/5，because max FIFO size is 64。

3.7.6.22 OUTCSR1

Register	Offset	R/W	Description	Reset Value
OUTCSR1	USB_BA+0x14	R/W	It provides control and status bits for transfers through the currently-selected OUT endpoint.	0x00

Bits	Descriptions	
[7]	ClrDataTog	The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.
[6]	SentStall	R/CLEAR. This bit is set when a STALL handshake is transmitted. The CPU should clear this bit.
[5]	SendStall	R/W. The CPU writes a 1 to this bit to issue a STALL handshake. The CPU clears this bit to terminate the stall condition. This bit has no effect if the OUT endpoint is in ISO mode.
[4]	FlushFIFO	The CPU writes a 1 to this bit to flush the next packet to be read from the endpoint OUT FIFO. Note: If the FIFO contains two packets, FlushFIFO will need to be set twice to completely clear the FIFO. It is cleared automatically.
[3]	DataError	R. This bit is set when OutPktRdy is set if the data packet has a CRC or bit-stuff error. it is cleared when OutPktRdy is cleared. The bit is only valid in ISO mode.
[2]	OverRun	R/CLEAR. This bit is set if an OUT packet cannot be loaded into the OUT FIFO. The CPU should clear this bit. The bit is only valid in ISO mode.
[1]	FIFOFull	R. This bit is set when no more packets can be loaded into the OUT FIFO. It is cleared automatically.
[0]	OutPktRdy	R/CLEAR. This bit is set when a data packet has been received. The CPU should clear this bit when the packet has been unloaded from the OUT FIFO. An interrupt is generated when the bit is set.

3.7.6.23 OUTCSR2

Register	Offset	R/W	Description	Reset Value
OUTCSR2	USB_BA+0x16	R/W	OutCSR2 is an 8-bit register that provides further control bits for transfers through the currently-selected OUT endpoint.	0x00

Bits	Descriptions	
[7]	AutoClear	If the CPU sets this bit then the OutPktRdy bit will be automatically cleared when a packet of OutMaxP bytes has been unloaded from the OUT FIFO. When packets of less than the maximum packet size are unloaded, OutPktRdy will have to be cleared manually.
[6]	ISO	The CPU sets this bit to enable the OUT endpoint for isochronous transfers, and clears it to enable the OUT endpoint for bulk or interrupt transfers.
[5:0]	Reserved	Reserved

3.7.6.24 OUTCOUNT1

Register	Offset	R/W	Description	Reset Value
OUTCOUNT1	USB_BA+0x16	R	Endpoint OUT Count – lower 8 bits	0x00

Bits	Descriptions	
[7:0]	OUTCOUNT1	OutCount1 is a 8-bit read-only register that holds the lower 8 bits of the number of received data bytes in the packet in the FIFO associated with the currently-selected OUT endpoint. The value returned is valid while OutPktRdy (OutCSR1.D0) is set

3.7.6.25 OUTCOUNT2

Register	Offset	R/W	Description	Reset Value
OUTCOUNT2	USB_BA+0x17	R	Endpoint OUT Count – upper 3 bits	0x00

Bits	Descriptions	
[7:3]	Reserved	Reserved
[2:0]	OUTCOUNT2	OutCount2 is a 3-bit read-only register that holds the upper 3 bits of the number of received data bytes in the packet in the FIFO associated with the currently-selected OUT endpoint. The value returned is valid while OutPktRdy (OutCSR1.D0) is set.

3.7.6.26 USB_DEBUG

Register	Offset	R/W	Description	Reset Value
USB_DEBUG	USB_BA+0x1F	R/W	USB Debug control	0x00

Bits	Descriptions	
[7:6]	Reserved	Reserved
[5:4]	TEST_WIDTH_SEL	Select the duration of signal transmission
[3:1]	Reserved	Reserved
[0]	USB_DEBUG_EN	USB transmission signal test enable

3.7.6.27 FIFOx

Register	Offset	R/W	Description	Reset Value
FIFOx	USB_BA+0x20	R/W		0x00

FIFOx	Descriptions
FIFO5	FIFO5 for EndPoint5. Address: USB_BA+0x30, SIZE: 64byte
FIFO4	FIFO4 for EndPoint4. Address: USB_BA+0x30, SIZE: 64byte
FIFO3	FIFO3 for EndPoint3. Address: USB_BA+0x2C, IN SIZE:128byte, OUT SIZE: 64byte
FIFO2	FIFO2 for EndPoint2. Address: USB_BA+0x28, SIZE:128byte
FIFO1	FIFO1 for EndPoint1. Address: USB_BA+0x24, SIZE:128byte
FIFO0	FIFO0 for EndPoint0. Address: USB_BA+0x20, SIZE:64byte

3.8 Enhanced PWM Generator (PWM)

3.8.1 Overview

The PAN271x series has built in 1 PWM unit (named as PWM0) which is specially designed for motor driving control applications. Each PWM unit supports 6 PWM generators which can be configured as eight independent outputs, from CH0 to CH5 (CH as an abbreviation for channel). Also, 6 CHs can be acted as 3 complementary output pairs, (CH0, CH1), (CH2, CH3)and (CH4, CH5) with 3 programmable dead-time generators, or as 3 synchronous output pairs, (CH0, CH1), (CH2, CH3)and (CH4, CH5).

Every complementary PWM pairs share one 8-bit prescaler. There are eight clock dividers providing five divided frequencies (1, 1/2, 1/4, 1/8, 1/16) for each channel. Each PWM output has independent 16-bit counter for PWM period control, and 16-bit comparators for PWM duty control. The 6 PWM generators provide 12 independent PWM interrupt flags which are set by hardware when the corresponding PWM period counter comparison matched period and duty. Each PWM interrupt source with its corresponding enable bit can request PWM interrupt. The PWM generators works in Auto-reload mode to output PWM waveform continuously.

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

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

3.8.2 Features

- Support 1 sets of PWM， each set PWM has 6 channels which is described as below
- Six independent 16-bit PWM duty control units with maximum six port pins:
  - Six independent PWM outputs – CH0, CH1, CH2, CH3, CH4 and CH5
  - Three complementary PWM pairs, with each pin in a pair mutually complement to each other and capable of programmable dead-time insertion – (CH0, CH1), (CH2, CH3) and (CH4, CH5)
  - Three synchronous PWM pairs, with each pin in a pair in-phase – (CH0, CH1), (CH2, CH3)and (CH4, CH5)
- Group control bit – CH2 and CH4 are synchronized with CH0, CH3 and CH5 are synchronized with CH1
- Auto-reload mode PWM
- Up to 16-bit resolution
- Supports edge-aligned mode
- Programmable dead-time insertion between complementary paired PWMs
- Each pin of CH0 to CH5 has independent polarity setting control
- The PWM signals before polarity control stage are defined in the view of low logic. The PWM ports is active high or active low are controlled by polarity control register
- Supports independently PERIOD matching, falling CMP matching, PERIOD matching to trigger ADC conversion

3.8.3 Block Diagram

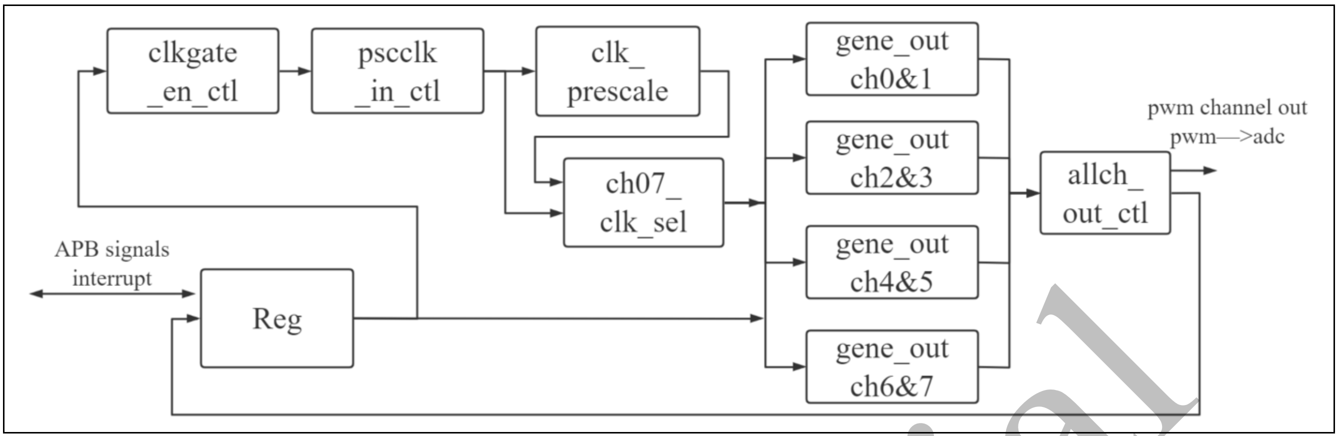


Figure 3-14 PWM0 Block Diagram

Figure 3-14 shows the architecture of PWM0 in pair.

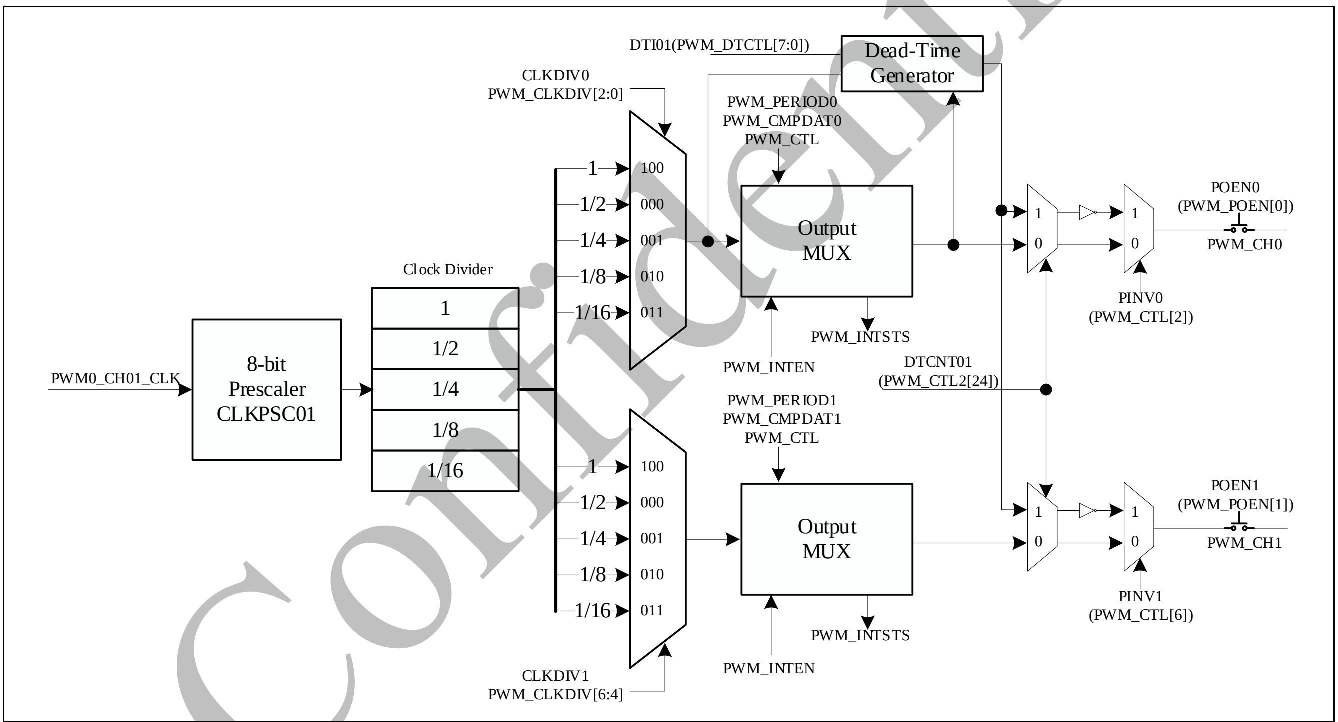


Figure 3-15 PWM0 Generator 0 Architecture Diagram

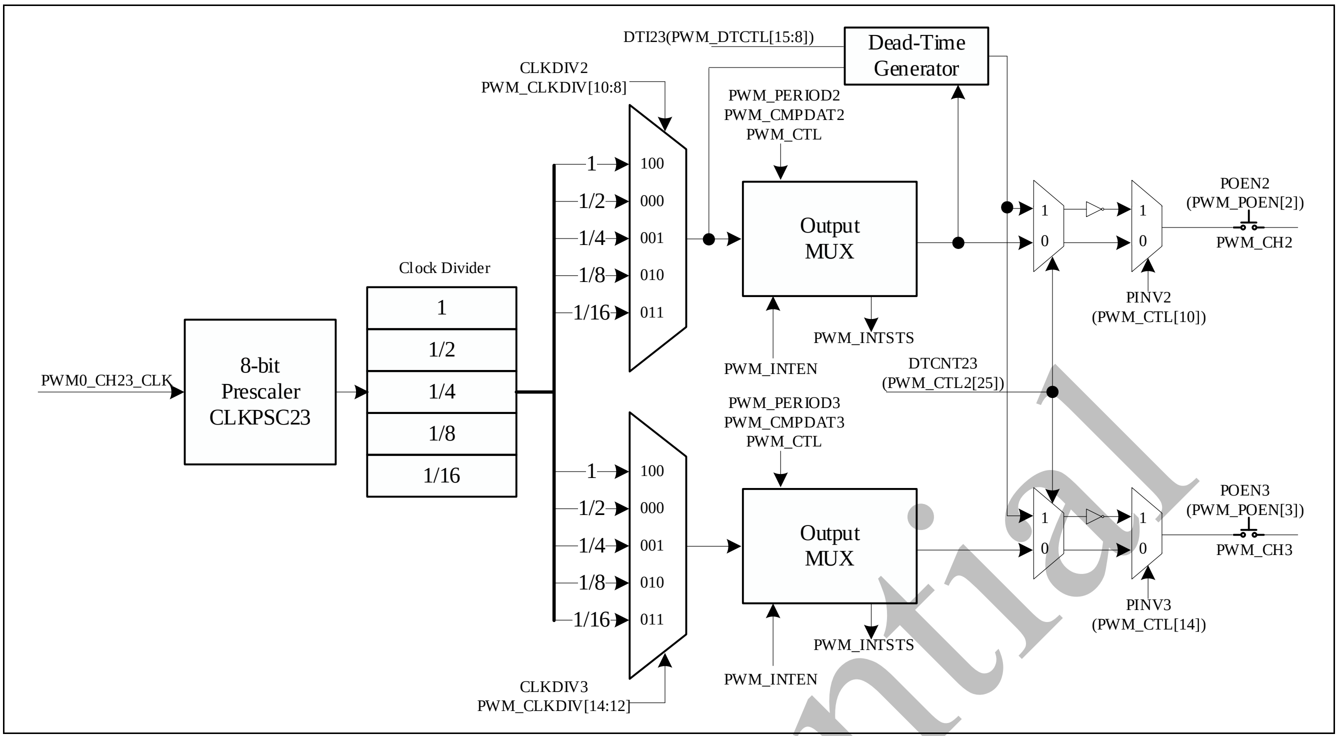


Figure 3-16 PWM0 Generator 2 Architecture Diagram

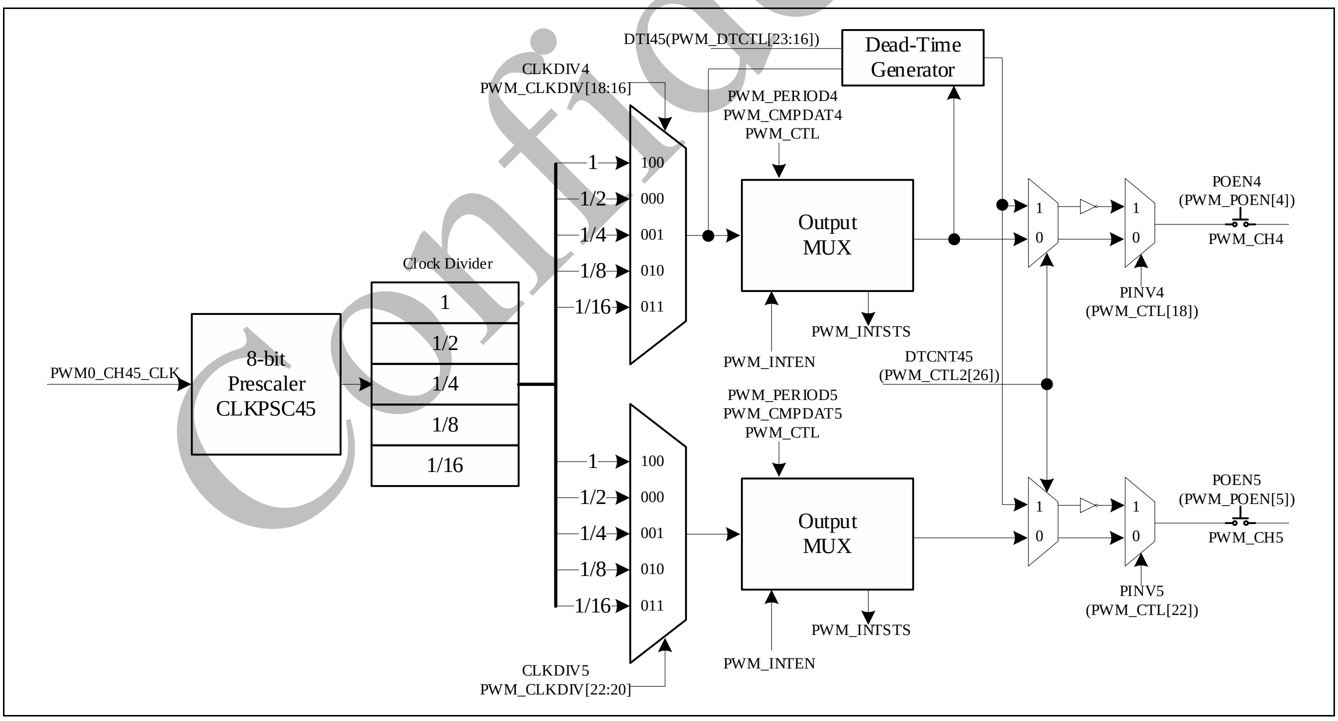


Figure 3-17 PWM0 Generator 4 Architecture Diagram

3.8.4 Functional Description

3.8.4.1 PWM Counter Type

This device supports 1 operation type: Edge-aligned.

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

Edge-aligned (Down counter):

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

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

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

Edge-aligned PWM (Down-counter)

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

Figure 3-18, Figure 3-19 and Figure 3-20 show the Edge-aligned PWM timing and operation flow.

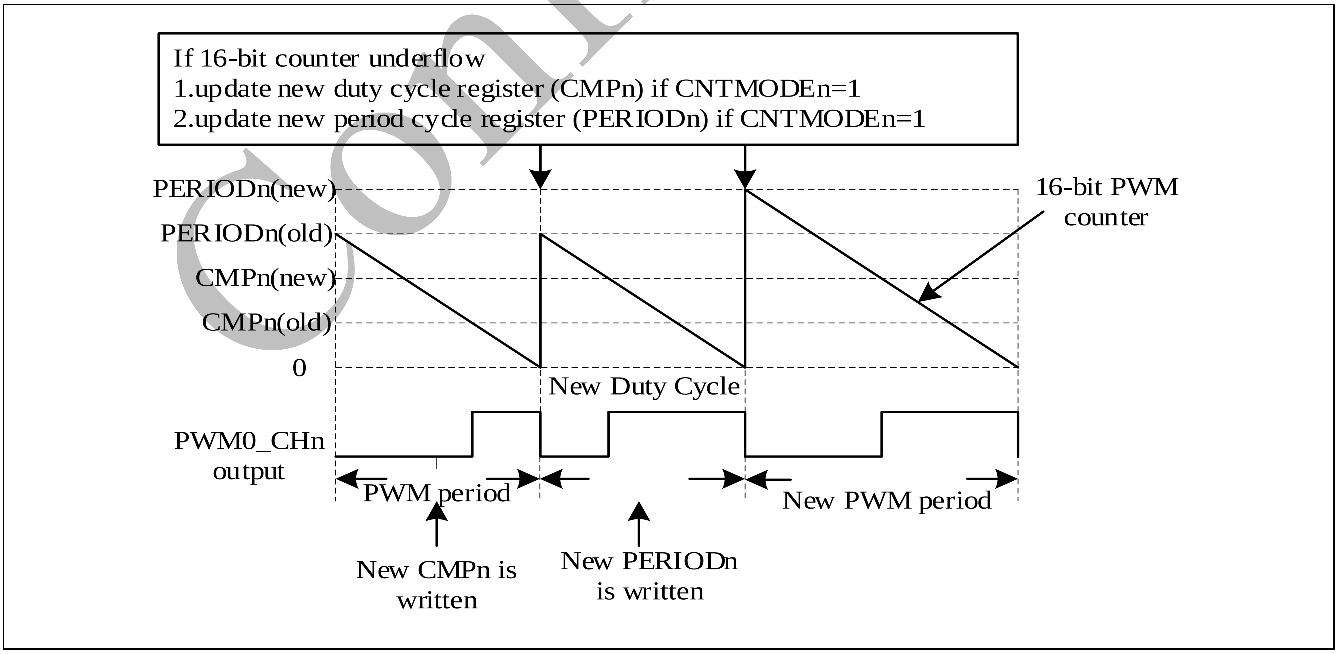


Figure 3-18 Edge-aligned Type PWM

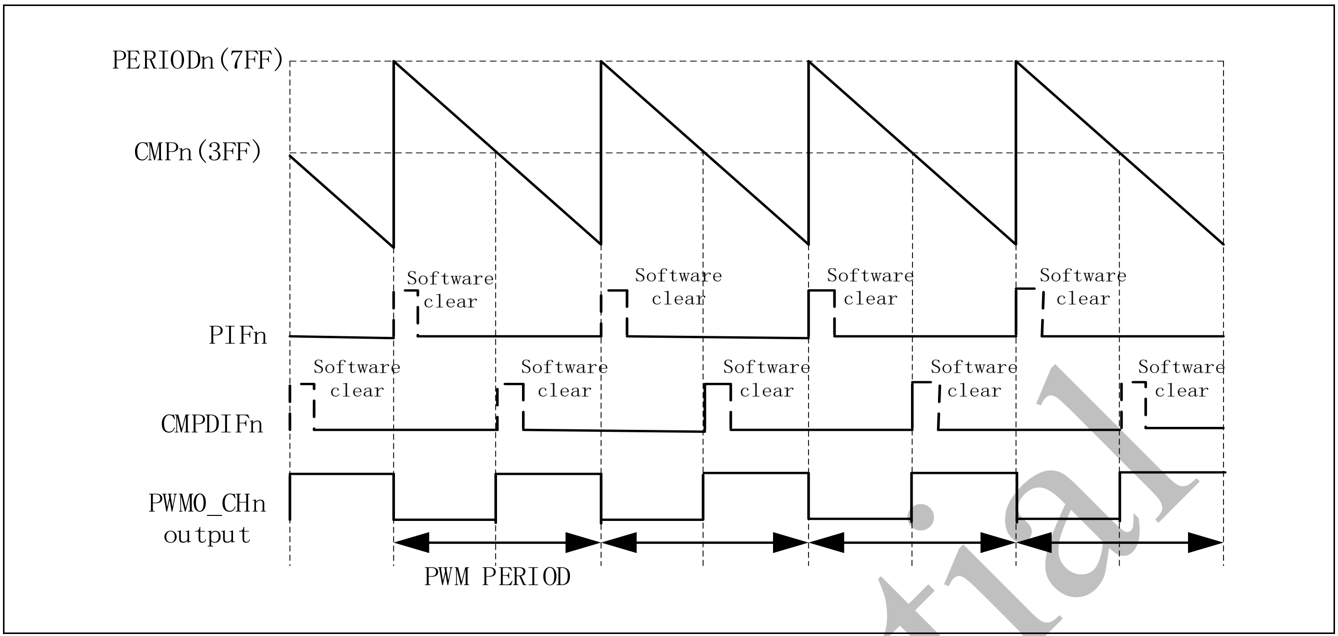


Figure 3-19 PWM Edge-aligned Waveform Timing Diagram

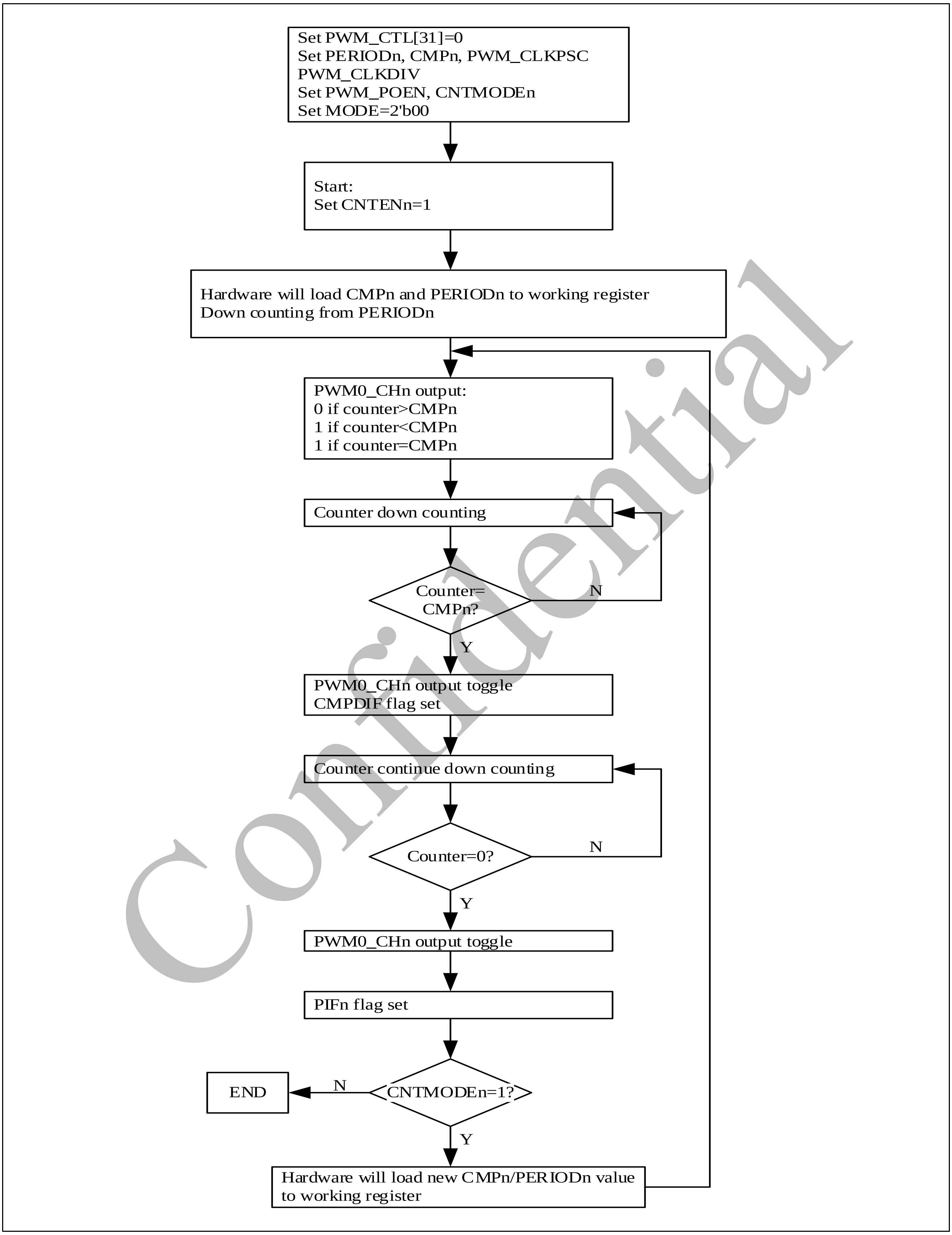


Figure 3-20 Edge-aligned Flow Diagram

The PWM period and duty control are decided by PWM down-counter period register *(PERIODn*) and PWM comparator register (*CMPn*). The PWM counter timing operation is shown in Figure 3-22. The pulse width modulation follows the formula below and the legend of PWM counter Comparator is shown in Figure 3-21. Note that the corresponding GPIO pins must be configured as PWM function (enable *PWM_POEN*) for the corresponding PWM channel.

PWM frequency = *APBCLK / ((CLKPSCnm + 1) * (clock divider)) / (PERIOD + 1)*; where nm, could be 01, 23 or 45 depending on the selected PWM channel, clock divider is configured by CLKDIV register.

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

PERIOD =0: PWM output is always low.

When PERIOD!=0, PWM output is as follow:

a. CMP ≥ PERIOD: PWM output is always high
b. CMP < PERIOD: PWM low width = (PERIOD - CMP) unit[1]; PWM high width = (CMP+1) unit
c. CMP = 0: PWM is always low

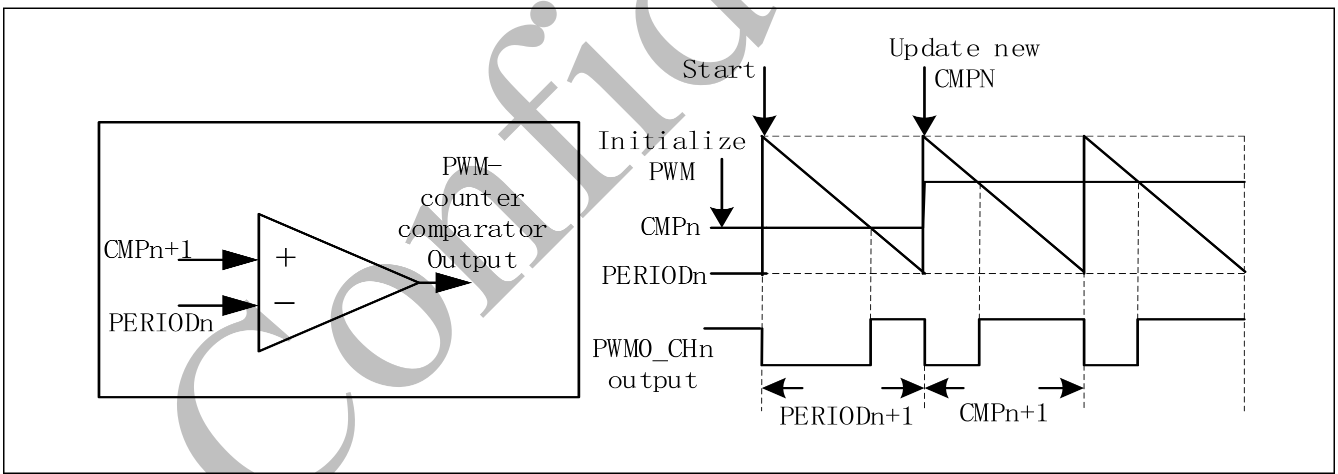


Figure 3-21 Legend of Internal Comparator Output of PWM Counter

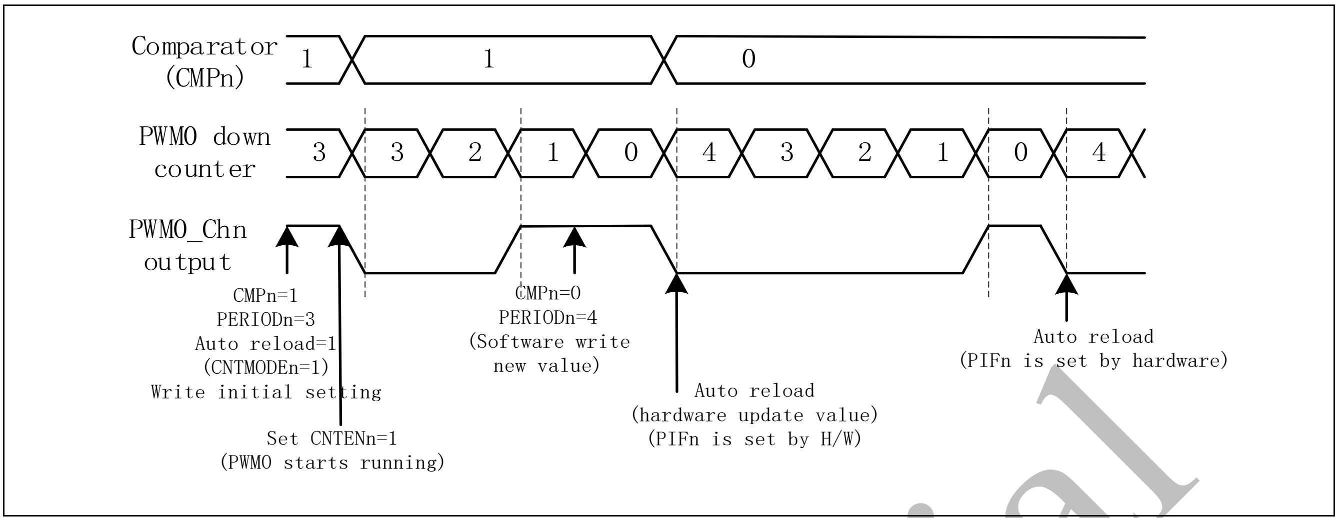


Figure 3-22 PWM Counter Operation Timing

3.8.4.2 PWM Double Buffering and Auto-reload Operation

The PAN271x series PWM have double buffering function, the reload value is updated at the start of next period without affecting current counter operation. The PWM counter value can be written into PERIODn.

CH0 will operate in Auto-reload mode if *CNTMODE0* bit is set to 1. It is recommended that switch CH0 operating mode before set *CNTEN0* bit to 1 to enable CH0 counter to start running because the content of PERIOD0 and CMP0 will be cleared to 0 to reset the CH0 period and duty setting when CH0 operating mode is changed. As CH0 operates at auto-reload mode, CMP0 and PERIOD0 should be written first and then set CNTEN0 bit to 1 to enable CH0 counter to start running. The PERIOD0 value will be reloaded to CH0 counter when the down counting reaches 0. If the PERIOD0 is set to 0, CH0 counter will be held. CH1~CH5 performs the same function as CH0.

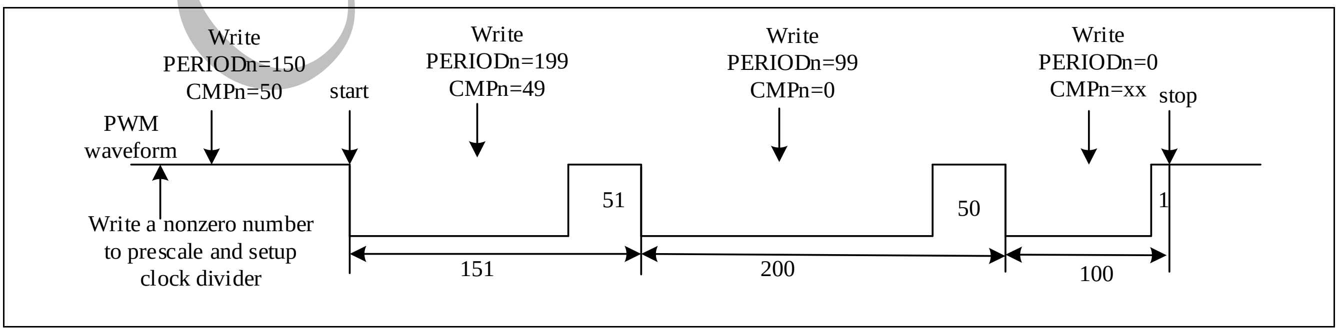


Figure 3-23 PWM Double Buffering Illustration

The double buffering function allows CMPn to be written at any point in the current cycle. The loaded value will take effect from the next cycle which can control output duty ratio easily.

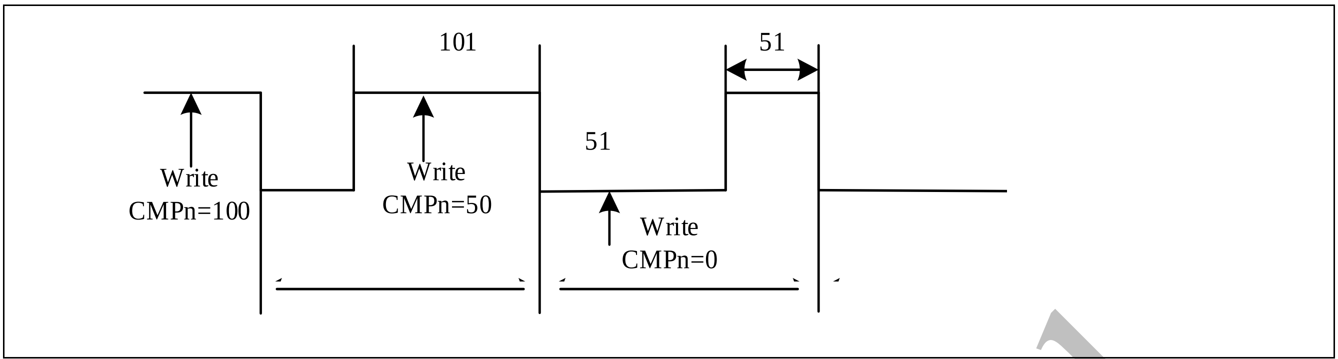


Figure 3-24 PWM Controller Output Duty Ratio

3.8.4.3 PWM Operation Modes

This powerful PWM unit supports independent mode which may be applied to DC or BLDC motor system, Complementary mode with dead-time insertion which may be used in the application of AC induction motor and synchronous motor, and Synchronous mode that makes both pins of each pair are in phase. Besides, the Group mode, which forces the CH2 and CH4 synchronous with CH0 generator, may simplify updating duty control in DC and BLDC motor applications, to generate interrupt timing.

Independent Mode

Independent mode is enabled when *MODE* (PWM_CTL2[29:28]) = 00.

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

Complementary Mode

Complementary mode is enabled when MODE (PWM_CTL2[29:28]) = 01.

In this module there are three duty-cycle generators utilized for complementary mode, with total of three PWM output pair pins in this module. The total six PWM outputs are grouped into output pairs of even and odd numbered outputs. In complimentary modes, the internal odd PWM signal CHn, always be the complement of the corresponding even PWM signal. For example, CH1 will be the complement of CH0. CH3 will be the complement of CH2, CH5 will be the complement of CH4. The time base for the PWM module is provided by its own 16-bit counter, which also incorporates selectable pre-scalar options.

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

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

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

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

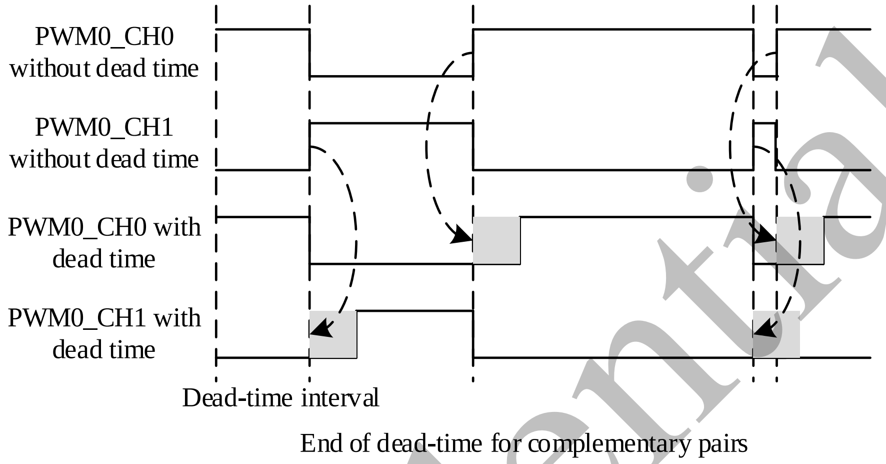


Figure 3-25 Dead-time Insertion

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

Synchronous Mode

Synchronous mode is enabled when MODE (PWM_CTL2[29:28]) = 10.

In the synchronization mode the PWM pair signals from PWM Generator are in-phase: CH1=CH0, CH3=CH2, CH5=CH4.

Group Mode

Group mode is enabled when GROUPEN (PWM_CTL2[30]) = 1.

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

If GROUPEN = 1, All CH2 and CH4 pairs will follow CH0 output, also CH3 and CH5 will follow CH1 output. If complementary mode is enabled when MODE (PWM_CTL2[29:28]) = 01, due to CH1 = invert(CH0), so other CHs will follow below formula:

CH4 = CH2 = CH0; CH5 = CH3 = CH1 = invert (CH0).

3.8.4.4 Polarity Control

Each PWM port from CH0 to CH5 has independent polarity control (PINV0~5) to configure the polarity of active state of PWM output which are described in the PINVn in *PWM Control Register* (PWM_CTL). By default, the PWM output is active high.

Figure 3-26 shows the initial state before PWM starts with different polarity settings.

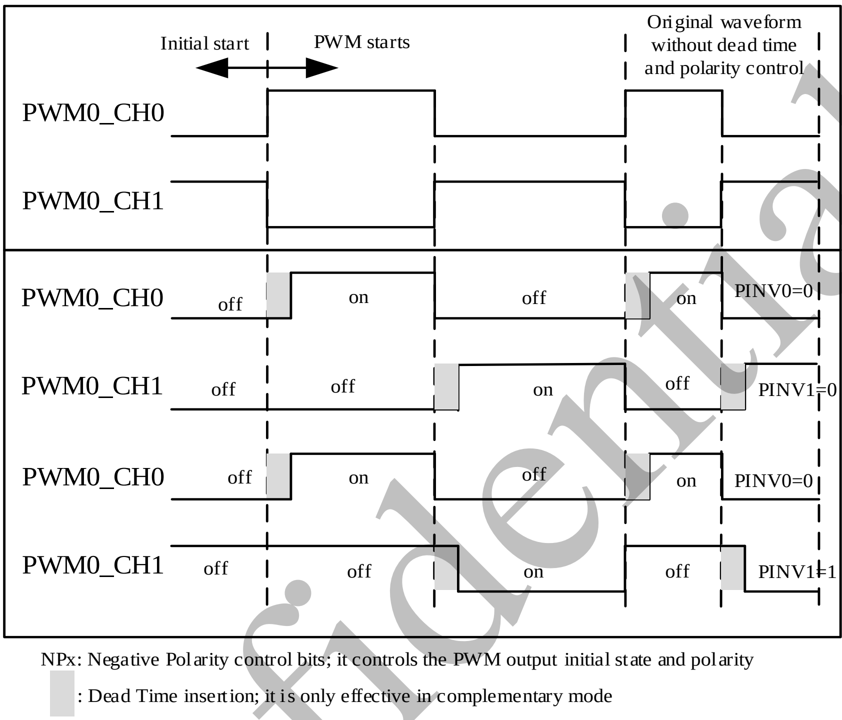


Figure 3-26 Initial State and Polarity Control with Rising Edge Dead-time Insertion

3.8.4.5 PWM Interrupt Architecture

There are two interrupt sources for PWM unit, which are

- PIFn PWM counter counts to period of edge-aligned type or counts to center of center-aligned type interrupt flag;
- CMPDIFn PWM counter down-counts to CMPn (PWM_CMPDATn[15:0]) interrupt flag.

The bits *ZIENn (PWM_INTEN)* control the ZIFn interrupt enable; the bits *PIENn (PWM_INTEN[21:16])* control the PIFn interrupt enable; and the bits *CMPDIENn (PWM_INTEN[13:8])* control the CMPDIFn interrupt enable. Note that all the interrupt flags are set by hardware and must be cleared by software.

Figure 3-27 shows the architecture of Motor Control PWM interrupts.

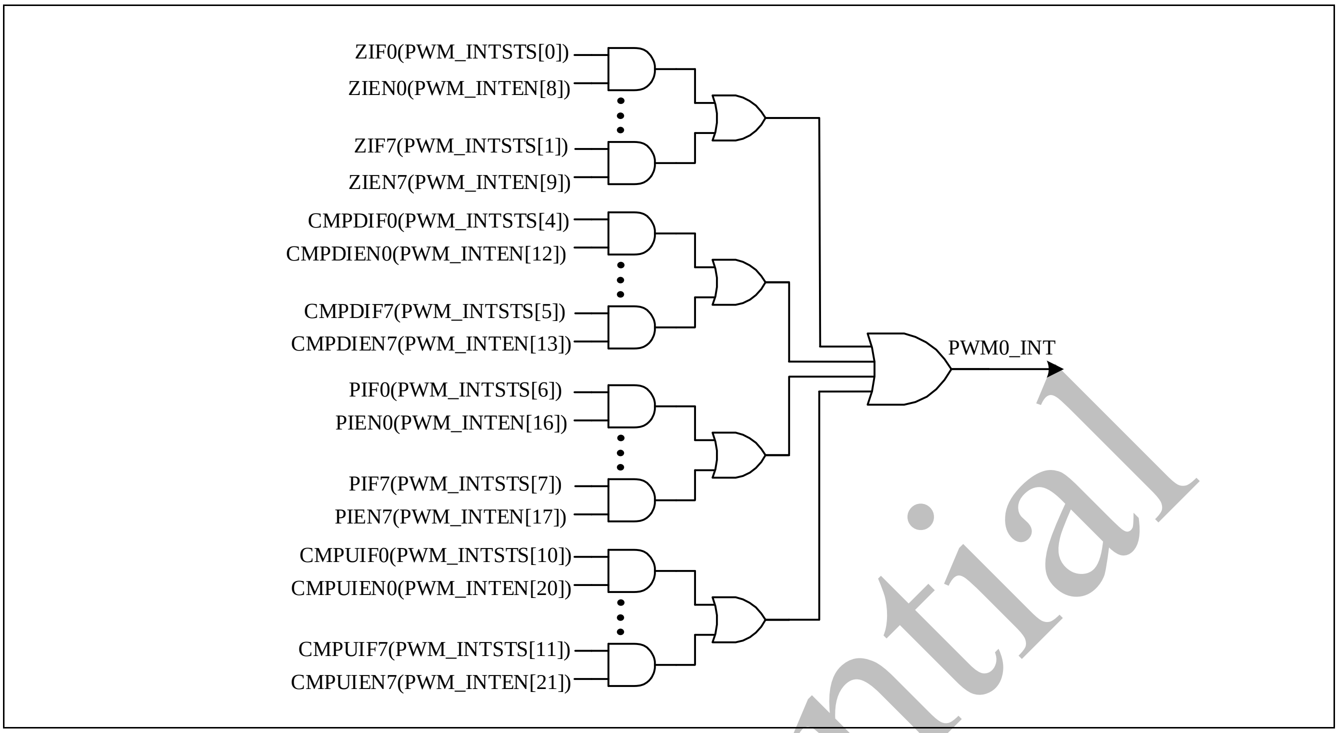


Figure 3-27 Motor Control PWM Interrupt Architecture

3.8.5 PWM Control Register Map

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

Register	Offset	R/W	Description	Reset Value
PWM Base Address: PWM_BA = 0x4000_4000				
PWM_CLKPSC_L	PWM_BA+0x00	R/W	PWM Clock Pre-scale Register	0x0000_0000
PWM_CLKDIV	PWM_BA+0x04	R/W	PWM Clock Select Register	0x0000_0000
PWM_CTL	PWM_BA+0x08	R/W	PWM Control Register	0x0000_0000
PWM_PERIOD0	PWM_BA+0x0C	R/W	PWM Counter Period Register 0	0x0000_0000
PWM_PERIOD1	PWM_BA+0x10	R/W	PWM Counter Period Register 1	0x0000_0000
PWM_PERIOD2	PWM_BA+0x14	R/W	PWM Counter Period Register 2	0x0000_0000
PWM_PERIOD3	PWM_BA+0x18	R/W	PWM Counter Period Register 3	0x0000_0000
PWM_PERIOD4	PWM_BA+0x1C	R/W	PWM Counter Period Register 4	0x0000_0000
PWM_PERIOD5	PWM_BA+0x20	R/W	PWM Counter Period Register 5	0x0000_0000
PWM_CMPDAT0	PWM_BA+0x2C	R/W	PWM Comparator Register 0	0x0000_0000
PWM_CMPDAT1	PWM_BA+0x30	R/W	PWM Comparator Register 1	0x0000_0000
PWM_CMPDAT2	PWM_BA+0x34	R/W	PWM Comparator Register 2	0x0000_0000
PWM_CMPDAT3	PWM_BA+0x38	R/W	PWM Comparator Register 3	0x0000_0000
PWM_CMPDAT4	PWM_BA+0x3C	R/W	PWM Comparator Register 4	0x0000_0000
PWM_CMPDAT5	PWM_BA+0x40	R/W	PWM Comparator Register 5	0x0000_0000
PWM_CTL2	PWM_BA+0x4C	R/W	PWM Control Register	0x0000_0000
PWM_FLAG	PWM_BA+0x50	R/W	PWM Status Register	0x0000_0000
PWM_INTEN	PWM_BA+0x54	R/W	PWM Interrupt Enable Register	0x0000_0000
PWM_INTSTS	PWM_BA+0x58	R/W	PWM Interrupt Status Register	0x0000_0000
PWM_POEN	PWM_BA+0x5C	R/W	PWM Output Enable Register	0x0000_0000
PWM_DTCTL	PWM_BA+0x64	R/W	PWM Dead-time Control Register	0x0000_0000
PWM_CLKPSC_H	PWM_BA+0x68	R/W	PWM Clock Pre-scale Register	0x0000_0000

3.8.6 PWM Control Register Description

3.8.6.1 PWM Pre-Scale Register (PWM_CLKPSC_L)

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

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	CLKPSC2	Clock Prescaler 2 for PWM Counter 2 Clock input is divided by (CLKPSC2 + 1) before it is fed to the corresponding PWM counter. If CLKPSC2= 0, the clock prescaler 2 output clock will be stopped. So the corresponding PWM counter will also be stopped.
[15:8]	CLKPSC1	Clock Prescaler 1 for PWM Counter 1 Clock input is divided by (CLKPSC1 + 1) before it is fed to the corresponding PWM counter. If CLKPSC1 = 0, the clock prescaler 1 output clock will be stopped. So the corresponding PWM counter will also be stopped.
[7:0]	CLKPSC0	Clock Prescaler 0 for PWM Counter 0 Clock input is divided by (CLKPSC0 + 1) before it is fed to the corresponding PWM counter. If CLKPSC0= 0, the clock prescaler 0 output clock will be stopped. So the corresponding PWM counter will also be stopped.

3.8.6.2 PWM Clock Selector Register (PWM_CLKDIV)

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

Bits	Descriptions	
[31:23]	Reserved	Reserved
[22:20]	CLKDIV5	Counter 5 Clock Divider Selection Select clock input for PWM counter. 000: Clock input / (CLKPSC45*2). 001: Clock input / (CLKPSC45*4). 010: Clock input / (CLKPSC45*8). 011: Clock input / (CLKPSC45*16). 100: Clock input / CLKPSC45. Others: Clock input.
[19]	Reserved	Reserved
[18:16]	CLKDIV4	Counter 4 Clock Divider Selection Select clock input for PWM counter. 000: Clock input / (CLKPSC45*2). 001: Clock input / (CLKPSC45*4). 010: Clock input / (CLKPSC45*8). 011: Clock input / (CLKPSC45*16). 100: Clock input / CLKPSC45. Others: Clock input.
[15]	Reserved	Reserved
[14:12]	CLKDIV3	Counter 3 Clock Divider Selection Select clock input for PWM counter. 000: Clock input / (CLKPSC23*2). 001: Clock input / (CLKPSC23*4). 010: Clock input / (CLKPSC23*8). 011: Clock input / (CLKPSC23*16). 100: Clock input / CLKPSC23. Others: Clock input.
[11]	Reserved	Reserved
[10:8]	CLKDIV2	Counter 2 Clock Divider Selection Select clock input for PWM counter. 000: Clock input / (CLKPSC23*2). 001: Clock input / (CLKPSC23*4). 010: Clock input / (CLKPSC23*8). 011: Clock input / (CLKPSC23*16). 100: Clock input / CLKPSC23. Others: Clock input.
[7]	Reserved	Reserved
[6:4]	CLKDIV1	Counter 1 Clock Divider Selection Select clock input for PWM counter. 000: Clock input / (CLKPSC01*2). 001: Clock input / (CLKPSC01*4). 010: Clock input / (CLKPSC01*8).

		011: Clock input / (CLKPSC01*16). 100: Clock input / CLKPSC01. Others: Clock input.
[3]	Reserved	Reserved
[2:0]	CLKDIV0	Counter 0 Clock Divider Selection Select clock input for PWM counter. 000: Clock input / (CLKPSC01*2). 001: Clock input / (CLKPSC01*4). 010: Clock input / (CLKPSC01*8). 011: Clock input / (CLKPSC01*16). 100: Clock input / CLKPSC01. Others: Clock input.

3.8.6.3 PWM Control Register (PWM_CTL)

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

Bits	Descriptions	
[31:23]	Reserved	Reserved
[22]	PINV5	PWM0_CH5 Output Inverter Enable Bit 0: PWM0_CH5 output inverter Disabled. 1: PWM0_CH5 output inverter Enabled.
[21]	Reserved	Reserved
[20]	CNTEN5	PWM Counter 5 Enable Start Run 0: Corresponding PWM counter running Stopped. 1: Corresponding PWM counter start run Enabled.
[19]	Reserved	Reserved
[18]	PINV4	PWM0_CH4 Output Inverter Enable Bit 0: PWM0_CH4 output inverter Disabled. 1: PWM0_CH4 output inverter Enabled.
[17]	Reserved	Reserved
[16]	CNTEN4	PWM Counter 4 Enable Start Run 0: Corresponding PWM counter running Stopped. 1: Corresponding PWM counter start run Enabled.
[15]	Reserved	Reserved
[14]	PINV3	PWM0_CH 3 Output Inverter Enable Bit 0: PWM0_CH3 output inverter Disabled. 1: PWM0_CH3 output inverter Enabled.
[13]	Reserved	Reserved
[12]	CNTEN3	PWM Counter 3 Enable Start Run 0: Corresponding PWM counter running Stopped. 1: Corresponding PWM counter start run Enabled.
[11]	Reserved	Reserved
[10]	PINV2	PWM0_CH2 Output Inverter Enable Bit 0: PWM0_CH2 output inverter Disabled. 1: PWM0_CH2 output inverter Enabled.
[9]	Reserved	Reserved
[8]	CNTEN2	PWM Counter 2 Enable Start Run 0: Corresponding PWM counter running Stopped. 1: Corresponding PWM counter start run Enabled.
[7]	Reserved	Reserved
[6]	PINV1	PWM0_CH1 Output Inverter Enable Bit 0: PWM0_CH1 output inverter Disable. 1: PWM0_CH1 output inverter Enable.
[5]	Reserved	Reserved
[4]	CNTEN1	PWM Counter 1 Enable/Disable Start Run 0: Corresponding PWM counter running Stopped. 1: Corresponding PWM counter start run Enabled.
[3]	Reserved	Reserved

[2]	PINV0	PWM0_CH0 Output Inverter Enable Bit 0: PWM0_CH0 output inverter Disabled. 1: PWM0_CH0 output inverter Enabled.
[1]	Reserved	Reserved
[0]	CNTEN0	PWM Counter 0 Enable Start Run 0: Corresponding PWM counter running Stopped. 1: Corresponding PWM counter start run Enabled.

3.8.6.4 PWM Counter Register 0-5 (PWM_PERIOD0-5)

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

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0] n=0,1..7	PERIODn	PWM Counter Period Value PERIODn determines the PWM counter period. **Edge-aligned type**: *PWM frequency = HCLK/((prescale+1)*(clock divider))/(PERIODn+1)*; where xy, could be 01, 23, 45, 67 depending on the selected PWM channel. *Duty ratio = (CMPn+1)/(PERIODn+1)*. PERIOD = 0: PWM is always low. When PERIOD!=0, PWM output is as follow: CMPn ≥ PERIODn: PWM output is always high. CMPn < PERIODn: PWM low width = (PERIODn-CMPn) unit; PWM high width = (CMP+1) unit. CMPn = 0: PWM is always low. **Note**: Any write to PERIODn will take effect in the next PWM cycle.

3.8.6.5 PWM Comparator Register 0-5 (PWM_CMPDAT0-5)

Register	Offset	R/W	Description	Reset Value
PWM_CMPDAT0	PWM_BA+0x2C	R/W	PWM Comparator Register 0	0x0000_0000
PWM_CMPDAT1	PWM_BA+0x30	R/W	PWM Comparator Register 1	0x0000_0000
PWM_CMPDAT2	PWM_BA+0x34	R/W	PWM Comparator Register 2	0x0000_0000
PWM_CMPDAT3	PWM_BA+0x38	R/W	PWM Comparator Register 3	0x0000_0000
PWM_CMPDAT4	PWM_BA+0x3C	R/W	PWM Comparator Register 4	0x0000_0000
PWM_CMPDAT5	PWM_BA+0x40	R/W	PWM Comparator Register 5	0x0000_0000

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0] n=0,1..7	CMPn	PWM Comparator Register CMP determines the PWM duty. **Edge-aligned type**: PWM frequency = HCLK/((CLKPSCnm+1)*(clock divider))/(PERIODn+1); where nm, could be 01, 23, 45, 67 depending on the selected PWM channel. *Duty ratio = (CMPn+1)/(PERIODn+1).* PERIOD = 0: PWM is always low. When PERIOD!=0, PWM output is as follow: CMPn ≥ PERIODn: PWM output is always high. CMPn < PERIODn: PWM low width = (PERIODn-CMPn) unit; PWM high width = (CMP+1) unit. CMPn = 0: PWM is always low. **Note**: Any write to CMPn will take effect in the next PWM cycle.

3.8.6.6 PWM Control Register2 (PWM_CTL2)

Register	Offset	R/W	Description	Reset Value
PWM_CTL2	PWM_BA+0x4C	R/W	PWM Control Register	0x0000_0000

Bits	Descriptions	
[31]	Reserved	Reserved
[30]	GROUPEN	Group Function Enable Bit 0: The signals timing of all PWM channels are independent. 1: Unify the signals timing of PWM0_CH0, PWM0_CH2, PWM0_CH4 in the same phase which is controlled by PWM0_CH0 and also unify the signals timing of PWM0_CH1, PWM0_CH3, PWM0_CH5 in the same phase which is controlled by PWM0_CH1.
[29:28]	MODE	PWM Operating Mode Select Bit 00: Independent mode. 01: Complementary mode. 10: Synchronized mode. 11: Reserved.
[27]	Reserved	Reserved
[26]	DTCNT45	Dead-time 4 Counter Enable Bit (PWM0_CH4 and PWM0_CH5 Pair for PWMC Group) 0: Dead-time 4 generator Disabled. 1: Dead-time 4 generator Enabled. **Note**: When the dead-time generator is enabled, the pair of PWM0_CH4 and PWM0_CH5 becomes a complementary pair for PWMC group.
[25]	DTCNT23	Dead-time 2 Counter Enable Bit (PWM0_CH2 and PWM0_CH3 Pair for PWMB Group) 0: Dead-time 2 generator Disabled. 1: Dead-time 2 generator Enabled. **Note**: When the dead-time generator is enabled, the pair of PWM0_CH2 and PWM0_CH3 becomes a complementary pair for PWMB group.
[24]	DTCNT01	Dead-time 0 Counter Enable Bit (PWM0_CH0 and PWM0_CH1 Pair for PWMA Group) 0: Dead-time 0 generator Disabled. 1: Dead-time 0 generator Enabled. **Note**: When the dead-time generator is enabled, the pair of PWM0_CH0 and PWM0_CH1 becomes a complementary pair for PWMA group.
[23:0]	Reserved	Reserved

3.8.6.7 PWM Flag Indication Register (PWM_FLAG)

Register	Offset	R/W	Description	Reset Value
PWM_FLAG	PWM_BA+0x50	R/W	PWM Status Flag Register	0x0000_0000

Bits	Descriptions	
[31:22]	Reserved	Reserved
[21:16] n=0,1..5	PFn	PWM Period Flag Flag is set by hardware when PWM0_CHn counter reaches PERIODn. **Note**: This bit can be cleared by software writing 1.
[15:14]	Reserved	Reserved
[13:8] n=0,1..5	CMPDFn	PWM Compare Down Flag Flag is set by hardware when PWMn counter down count reaches CMPn. **Note**: This bit can be cleared by software writing 1.
[7:0]	Reserved	Reserved

3.8.6.8 PWM Interrupt Enable Register (PWM_INTEN)

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

Bits	Descriptions	
[31:22]	Reserved	Reserved
[21:16] n=0,1..5	PIENn	PWM Period Interrupt Enable Bit 0: PWM0_CHn period interrupt Disabled. 1: PWM0_CHn period interrupt Enabled.
[15:14]	Reserved	Reserved
[13:8] n=0,1..5	CMPDIENn	PWM Compare Down Interrupt Enable Bit 0: PWM0_CHn compare down interrupt Disabled. 1: PWM0_CHn compare down interrupt Enabled.
[7:0]	Reserved	Reserved

3.8.6.9 PWM Interrupt Indication Register (PWM_INTSTS)

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

Bits	Descriptions	
[31:22]	Reserved	Reserved
[21:16] n=0,1..5	PIFn	PWM Period Interrupt Flag Flag is set by hardware when PWM0_CHn counter reaches PERIODn. **Note**: This bit can be cleared by software writing 1.
[15:14]	Reserved	Reserved
[13:8] n=0,1..5	CMPDIFn	PWM Compare Down Interrupt Flag Flag is set by hardware when PWMn counter down count reaches CMPn. **Note**: This bit can be cleared by software writing 1.
[7:0]	Reserved	Reserved

3.8.6.10 PWM Output Control Register (PWM_POEN)

Register	Offset	R/W	Description	Reset Value
PWM_POEN	PWM_BA+0x5C	R/W	PWM Output Enable Register	0x0000_0000

Bits	Descriptions	
[31:6]	Reserved	Reserved
[5:0] n=0,1..5	POENn	PWM Output Enable Bits 0: PWM channel n output to pin Disabled. 1: PWM channel n output to pin Enabled. **Note**: The corresponding GPIO pin must be switched to PWM function.

3.8.6.11 PWM Dead-time Interval Register (PWM_DTCTL)

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

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

3.8.6.12 PWM Pre-Scale Register (PWM_CLKPSC_H)

Register	Offset	R/W	Description	Reset Value
PWM_CLKPSC_H	PWM_BA+0x68	R/W	PWM Clock Pre-scale Register	0x0000_0000

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:16]	CLKPSC5	Clock Prescaler 5 for PWM Counter 5 Clock input is divided by (CLKPSC5 + 1) before it is fed to the corresponding PWM counter. If CLKPSC5= 0, the clock prescaler 5 output clock will be stopped. So the corresponding PWM counter will also be stopped.
[15:8]	CLKPSC4	Clock Prescaler 4 for PWM Counter 4 Clock input is divided by (CLKPSC4 + 1) before it is fed to the corresponding PWM counter. If CLKPSC4= 0, the clock prescaler 4 output clock will be stopped. So the corresponding PWM counter will also be stopped.
[7:0]	CLKPSC3	Clock Prescaler 3 for PWM Counter 3 Clock input is divided by (CLKPSC3 + 1) before it is fed to the corresponding PWM counter. If CLKPSC3= 0, the clock prescaler 3 output clock will be stopped. So the corresponding PWM counter will also be stopped.

3.8.7 Operation Steps

3.8.7.1 PWM Counter Start Procedure

The following procedure is recommended for PWM counter start:

1. Configure clock prescale register (*PWM_CLKPSC*).
2. Configure clock select register (*PWM_CLKDIV*) for setting clock source select (CLKDIVn).
3. Configure PWM control register (*PWM_CTL*) for setting auto-reload mode (CNTMODEn = 1), DISABLE PWM counter (CNTENn = 0).
4. Configure PWM control register (*PWM_CTL*) for setting inverter on/off (PINVn), and Dead-time generator on/off (DTCNTnm). (Optional)
5. Configure *PWM_DTCTL* register to set dead-time interval. (Optional)
6. Configure compare register (*PWM_CMPDATn*) for setting PWM duty (CMPn).
7. Configure PWM period counter register (*PWM_PERIODn*) .
8. Configure PWM interrupt enable register (*PWM_INTEN*) for setting PWM period interrupt type (INTTYPE), PWM period interrupt enable bit (PIENn), PWM compare down match interrupt enable bit (CMPDIENn). (Optional)
9. Configure PWM output enable register (*PWM_POEN*) to enable PWM output channel
10. Configure PWM control register (*PWM_CTL*) to enable PWM counter (CNTENn = 1)

3.8.7.2 PWM Counter Stop Procedure

Method 1:

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

Method 2:

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

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

3.9 Watchdog Timer (WDT)

3.9.1 Overview

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

3.9.2 Features

- 24-bit free running up counter for WDT time-out interval.
- Selectable time-out interval (2^4 ~ 2^24) WDT_CLK cycles and the time-out interval is 0.5 ms ~ 524.288s, if WDT_CLK = 32 kHz.
- Supports selectable WDT reset delay period, including 1026, 130, 18 or 3 WDT_CLK reset delay period.
- Supports WDT time-out wake-up function only if WDT clock source is selected as RCL or XTL.

3.9.3 Block Diagram

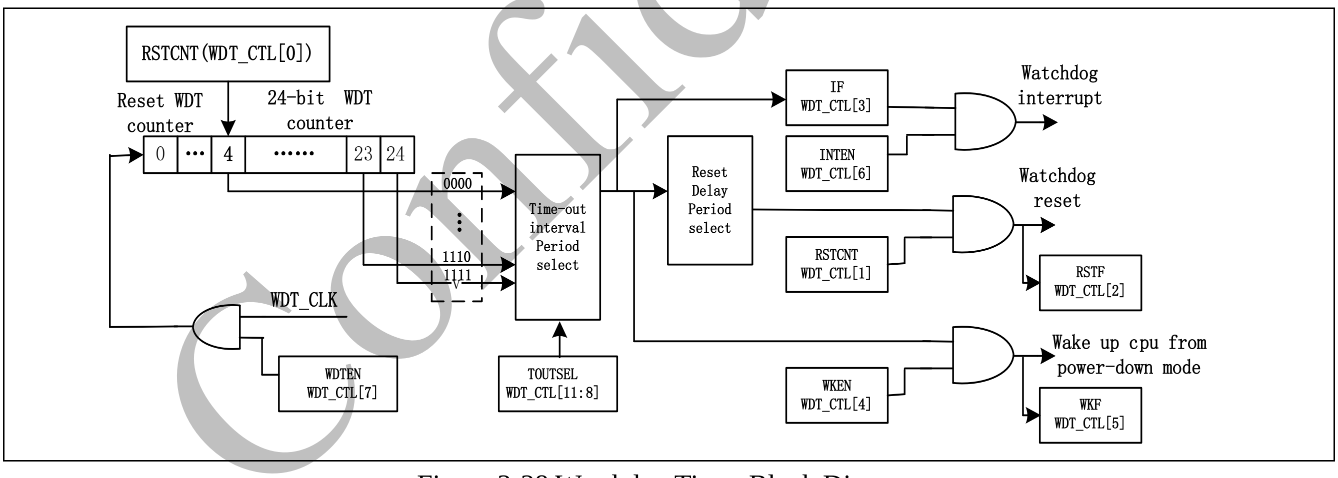


Figure 3-28 Watchdog Timer Block Diagram

3.9.4 Clock Control

The WDT clock control is shown in Figure 3-29.

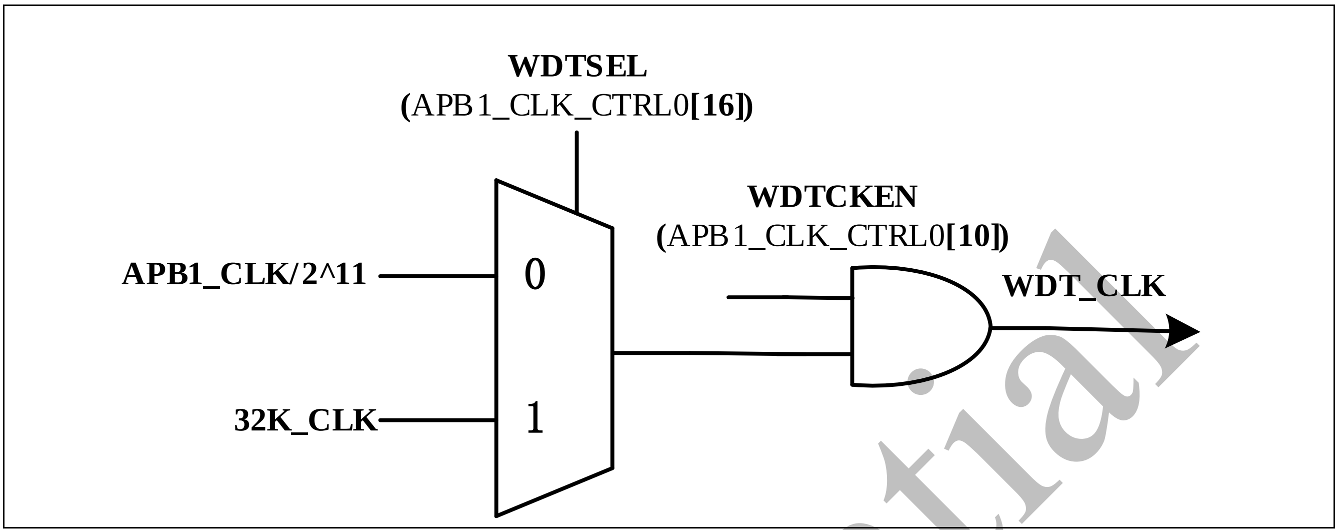


Figure 3-29 Watchdog Timer Clock Control

3.9.5 Basic Configuration

The WDT peripheral clock is enabled in *WDTCKEN* (APB1_CLK_CTRL0[10]) and clock source can be selected in *WDTSEL* (APB1_CLK_CTRL0[16]).

3.9.6 Functional Description

The WDT includes an 24-bit free running up counter with programmable time-out intervals. Table 3-7 shows the WDT time-out interval period selection and Figure 3-30 shows the WDT time-out interval and reset period timing.

3.9.6.1 WDT Time-out Flag

Setting *WDTEN* (WDT_CTL[7]) to 1 will enable the WDT function and the WDT counter to start counting up. There are eight time-out interval period can be selected by setting *TOUTSEL* (WDT_CTL[11:8]). When the WDT up counter reaches the *TOUTSEL* (WDT_CTL[11:8]) setting, the WDT time-out interrupt will occur and then WDT time-out flag TOF (WDT_CTL[16]) will be set to 1 immediately.

3.9.6.2 WDT Time-out Interrupt Flag

Setting WDTEN (WDT_CTL[7]) to 1 will enable the WDT function and the WDT counter to start counting up. There are eight time-out interval period can be selected by setting TOUTSEL (WDT_CTL[11:8]). When the WDT up counter reaches the TOUTSEL

(WDT_CTL[11:8]) setting, the WDT time-out interrupt will occur and then WDT time-out interrupt flag IF (WDT_CTL[3]) will be set to 1 immediately when INTEN (WDT_CTL[6]) is set to 1.

3.9.6.3 WDT Reset Delay Period and Reset System

A specified TRSTD reset delay period occurs when the IF (WDT_CTL[3]) is set to 1. User should set *RSTCNT* (WDT_CTL[0]) to reset the 24-bit WDT up counter value to avoid generating the WDT time-out reset signal before the TRSTD reset delay period expires. Moreover, user should set *RSTDSEL* (WDT_ALTCTL [1:0]) to select reset delay period to clear WDT counter. If the WDT up counter value has not been cleared after the specified TRSTD delay period expires, the WDT control will set RSTF (WDT_CTL[2]) to 1 if RSTEN (WDT_CTL[1]) bit is enabled, and then chip enters reset state immediately. Refer to Figure 3-30 Watchdog Timer Time-out Interval and Reset Period Timing. The TRST reset period will keep the last 63 WDT clocks and then chip restart executing program from reset vector (0x0000_0000). The RSTF (WDT_CTL[2]) will keep 1 after WDT time-out resets the chip. User can check RSTF (WDT_CTL[2]) via software to recognize if the system has been reset by WDT time-out reset or not.

Table 3-7 Watchdog Timer Time-out Interval Period Selection

TOUTSEL	**Time-Out Interval Period TTIS**	**Reset Delay Period TRSTD**
0000	2^{4} * TWDT	(3/18/130/1026) * TWDT
0001	2^{6} * TWDT	(3/18/130/1026) * TWDT
0010	2^{8} * TWDT	(3/18/130/1026) * TWDT
0011	2^{10} * TWDT	(3/18/130/1026) * TWDT
0100	2^{12} * TWDT	(3/18/130/1026) * TWDT
0101	2^{14} * TWDT	(3/18/130/1026) * TWDT
0110	2^{15} * TWDT	(3/18/130/1026) * TWDT
0111	2^{16} * TWDT	(3/18/130/1026) * TWDT
1000	2^{17} * TWDT	(3/18/130/1026) * TWDT
1001	2^{18} * TWDT	(3/18/130/1026) * TWDT
1010	2^{19} * TWDT	(3/18/130/1026) * TWDT
1011	2^{20} * TWDT	(3/18/130/1026) * TWDT
1100	2^{21} * TWDT	(3/18/130/1026) * TWDT
1101	2^{22} * TWDT	(3/18/130/1026) * TWDT
1110	2^{23} * TWDT	(3/18/130/1026) * TWDT
1111	2^{24} * TWDT	(3/18/130/1026) * TWDT

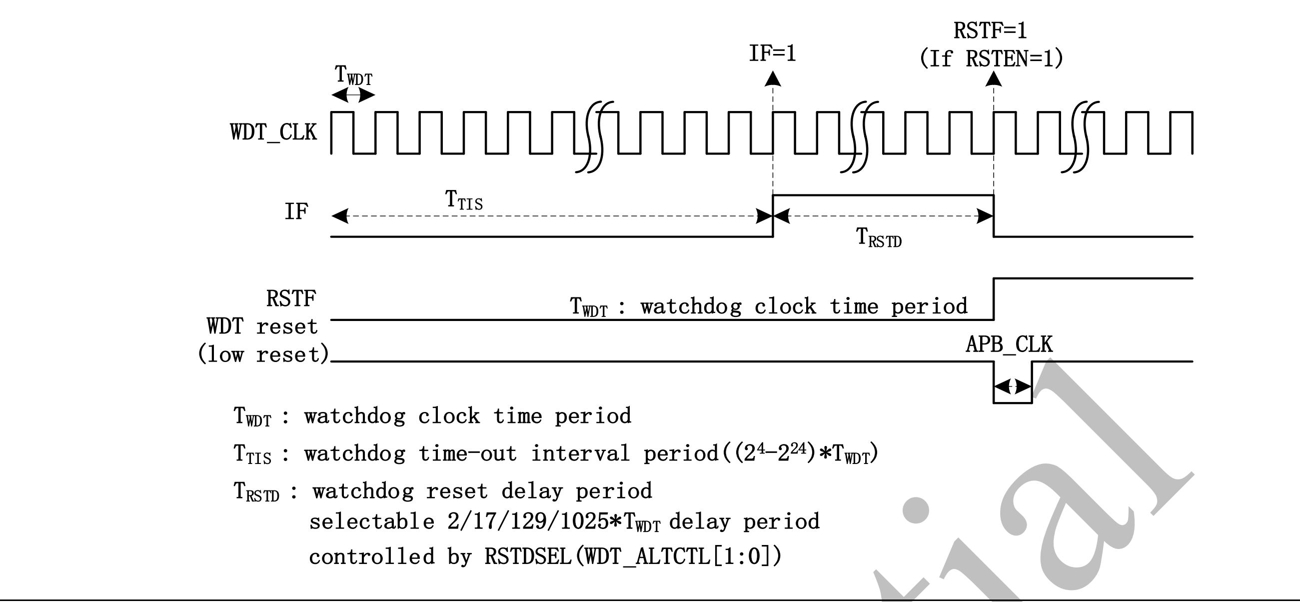


Figure 3-30 Watchdog Timer Time-out Interval and Reset Period Timing

3.9.6.4 WDT Wake-up

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

3.9.7 WDT Control Register Map

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

Register	Offset	R/W	Description	Reset Value
WDT Base Address: WDT_BA = 0x4000_4000				
WDT_CTL	WDT_BA+0x00	R/W	WDT Control Register	0x0000_0F00
WDT_ALTCTL	WDT_BA+0x04	R/W	WDT Alternative Control Register	0x0000_0000

3.9.8 WDT Register Description

3.9.8.1 WDT Control Register (WDT_CTL)

Register	Offset	R/W	Description	Reset Value
WDT_CTL	WDT_BA+0x00	R/W	WDT Control Register	0x0000_0F00

Bits	Descriptions	
[31]	ICEDEBUG	ICE Debug Mode Acknowledge Disable Bit (Write Protect) 0: ICE debug mode acknowledgment affects WDT counting. WDT up counter will be held while CPU is held by ICE. 1: ICE debug mode acknowledgment Disabled. WDT up counter will keep going no matter CPU is held by ICE or not. **Note**: This bit is write protected. Refer to the *SYS_REGLCTL* register.
[30:17]	Reserved	Reserved.
[16]	TOF	WDT Time-out Flag This bit will be set to 1 while WDT up counter value reaches the selected WDT time-out interval 0: WDT time-out interrupt did not occur. 1: WDT time-out interrupt occurred. **Note**: This bit is cleared by writing 1 to it.
[15:13]	Reserved	Reserved.
[12]	RST_REGION_SEL	Reset Region Select(Write Protect) 0: Just 1.2V region without anactrl & rcc module 1: All chip without work_mode(rom mode) module
[11:8]	TOUTSEL	WDT Time-out Interval Selection (Write Protect) These four bits select the time-out interval period for the WDT. 0000: 2^4 * WDT_CLK. 0001: 2^6 * WDT_CLK. 0010: 2^8 * WDT_CLK. 0011: 2^10 * WDT_CLK. 0100: 2^12 * WDT_CLK. 0101: 2^14 * WDT_CLK. 0110: 2^15 * WDT_CLK. 0111: 2^16 * WDT_CLK. 1000: 2^17 * WDT_CLK. 1001: 2^18 * WDT_CLK. 1010: 2^19 * WDT_CLK. 1011: 2^20 * WDT_CLK. 1100: 2^21 * WDT_CLK. 1101: 2^22 * WDT_CLK. 1110: 2^23 * WDT_CLK. 1111: 2^24 * WDT_CLK. **Note**: This bit is write protected. Refer to the SYS_REGLCTL register.
[7]	WDTEN	WDT Enable Bit (Write Protect) 0: WDT Disabled (This action will reset the internal up counter value). 1: WDT Enabled.

		Note: This bit is write protected. Refer to the SYS_REGLCTL register.
[6]	INTEN	WDT Time-out Interrupt Enable Bit (Write Protect) If this bit is enabled, the WDT time-out interrupt signal is generated and inform to CPU. 0: WDT time-out interrupt Disabled. 1: WDT time-out interrupt Enabled. **Note**: This bit is write protected. Refer to the SYS_REGLCTL register.
[5]	WKF	WDT Time-out Wake-up Flag (Write Protect) This bit indicates the interrupt wake-up flag status of WDT 0: WDT does not cause chip wake-up. 1: Chip wake-up from deepsleep mode if WDT time-out interrupt signal generated. **Note1**: This bit is write protected. Refer to the SYS_REGLCTL register. **Note2**: This bit is cleared by writing 1 to it.
[4]	WKEN	WDT Time-out Wake-up Function Control (Write Protect) If this bit is set to 1, while WDT time-out interrupt flag IF (WDT_CTL[3]) is generated to 1 and interrupt enable bit INTEN (WDT_CTL[6]) is enabled, the WDT time-out interrupt signal will generate a wake-up trigger event to chip. 0: Wake-up trigger event Disabled if WDT time-out interrupt signal generated. 1: Wake-up trigger event Enabled if WDT time-out interrupt signal generated. **Note**: This bit is write protected. Refer to the SYS_REGLCTL register.
[3]	IF	WDT Time-out Interrupt Flag This bit will be set to 1 while WDT up counter value reaches the selected WDT time-out interval 0: WDT time-out interrupt did not occur. 1: WDT time-out interrupt occurred. **Note**: This bit is cleared by writing 1 to it.
[2]	RSTF	WDT Time-out Reset Flag This bit indicates the system has been reset by WDT time-out reset or not. 0: WDT time-out reset did not occur. 1: WDT time-out reset occurred. **Note**: This bit is cleared by writing 1 to it.
[1]	RSTEN	WDT Time-out Reset Enable Bit (Write Protect) Setting this bit will enable the WDT time-out reset function If the WDT up counter value has not been cleared after the specific WDT reset delay period expires. 0: WDT time-out reset function Disabled. 1: WDT time-out reset function Enabled. **Note**: This bit is write-protected. Refer to the SYS_REGLCTL register.
[0]	RSTCNT	Reset WDT Up Counter (Write Protect) 0: No effect. 1: Reset the internal 24-bit WDT up counter value. **Note1**: This bit is write protected. Refer to the SYS_REGLCTL register. **Note2**: This bit will be automatically cleared by hardware.

3.9.8.2 WDT Alternative Control Register (WDT_ALTCTL)

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

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

3.10 I2C Serial Interface Controller (I2C)

3.10.1 Overview

The I2C master can transmit and receive data. The data rate can reach up to 533kbps.

3.10.2 Features

- Two operating modes: Master TX, Master RX
- Clock division, data rate up to 533kbps
- 7-bit address format
- Shared read/write buffer, 8-bit width, 1 byte depth
- SDA and SCL input signal filtering, capable of filtering glitches shorter than 3 clock cycles
- TX / RX interrupts

3.10.3 Block Diagram

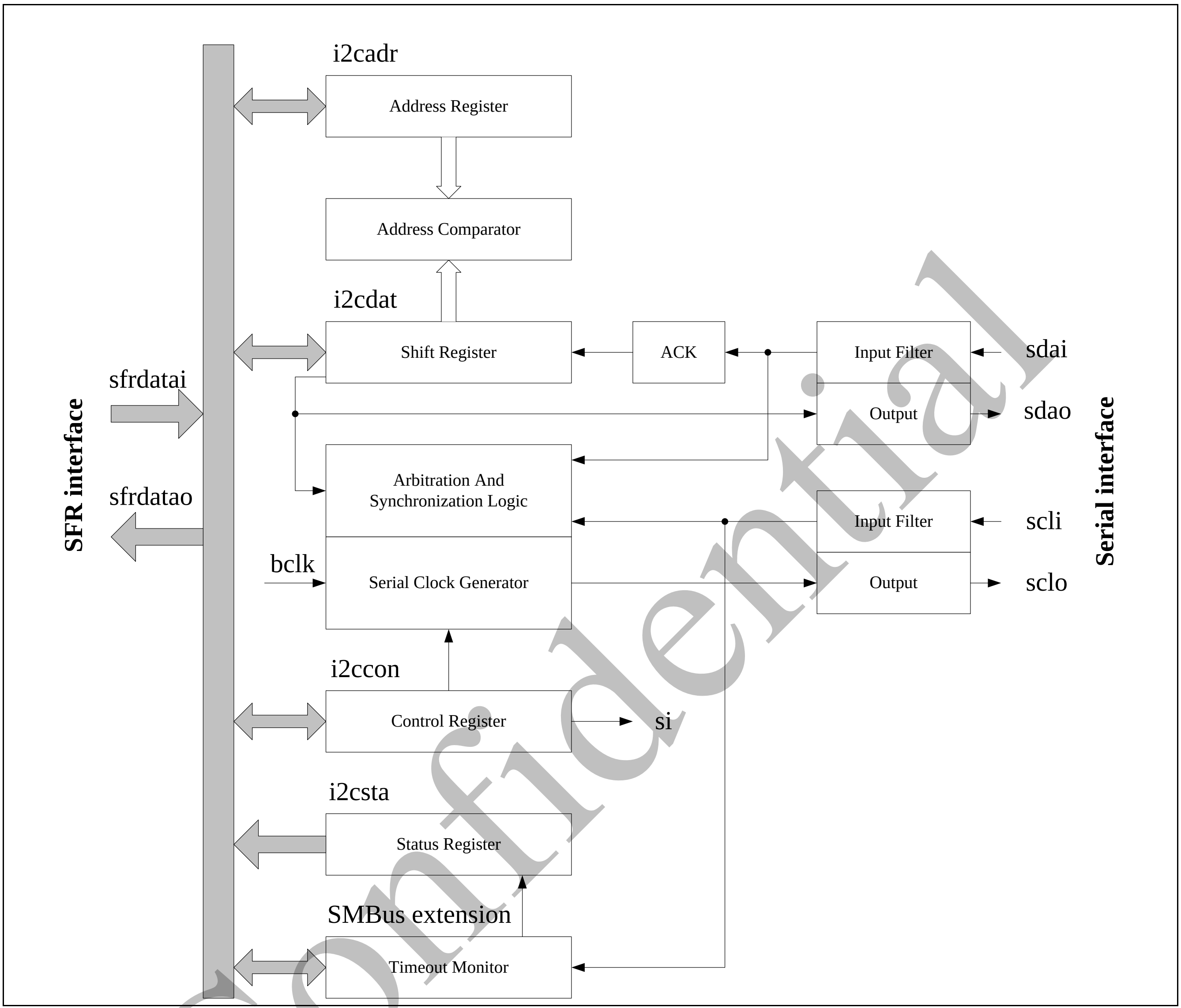


Figure 3-31 I2C Controller Block Diagram

3.10.4 Functional Description

I2C requires two wires for data transmission when connecting devices: "SCL" (Serial Clock Line) and "SDA" (Serial Data Line). Since I2C is a bidirectional port, it needs to use external open-drain buffers. Each device connected to the bus can be software-addressed via a unique address. I2C is a true multi-master bus. When two or more masters start data transmission simultaneously, it incorporates collision detection and arbitration to prevent data corruption. I2C also has filtering capabilities to eliminate glitches, ensuring data integrity and accuracy. The I2C data rate can reach up to 533kbps and operates in both master transmit and master receive modes.

3.10.4.1 Master Transmit Mode

In master transmit mode, the serial data is output through “SDA”, and the serial clock is output through “SCL”. The specific timing is shown in the figure below.

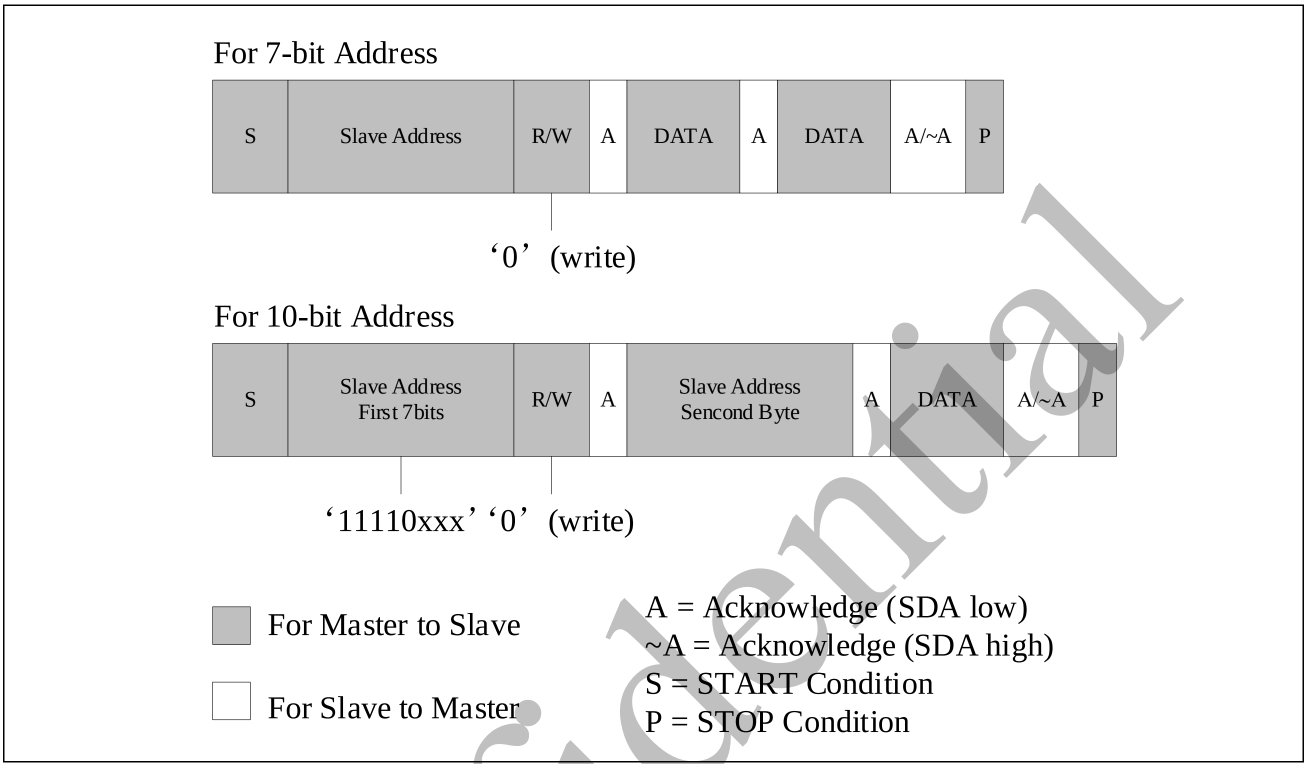


Figure 3-32 Master Transmit Diagram

3.10.4.2 Master Receive Mode

In master receive mode, the serial data is input through “SDA”, and the serial clock is output through “SCL”. The specific timing is shown in the figure below.

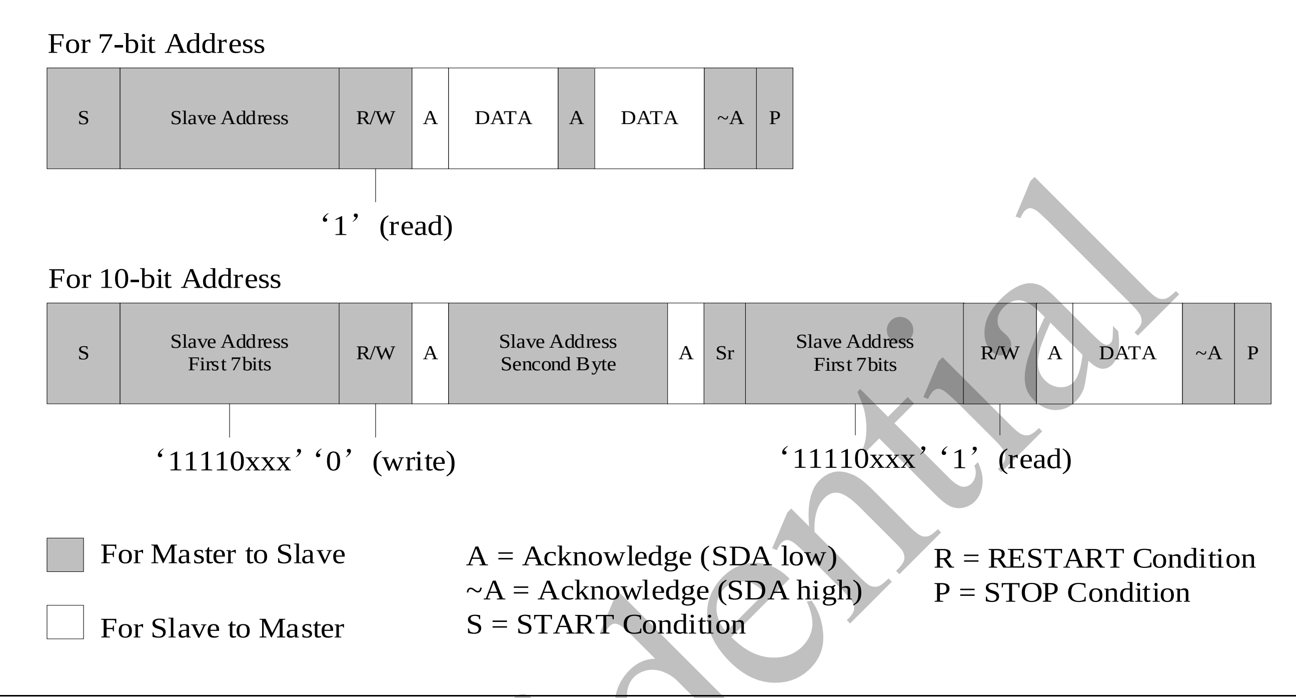


Figure 3-33 Master Receive Diagram

3.10.4.3 Start and Stop Signals

The start and stop signals during data transmission, as shown in the figure below.

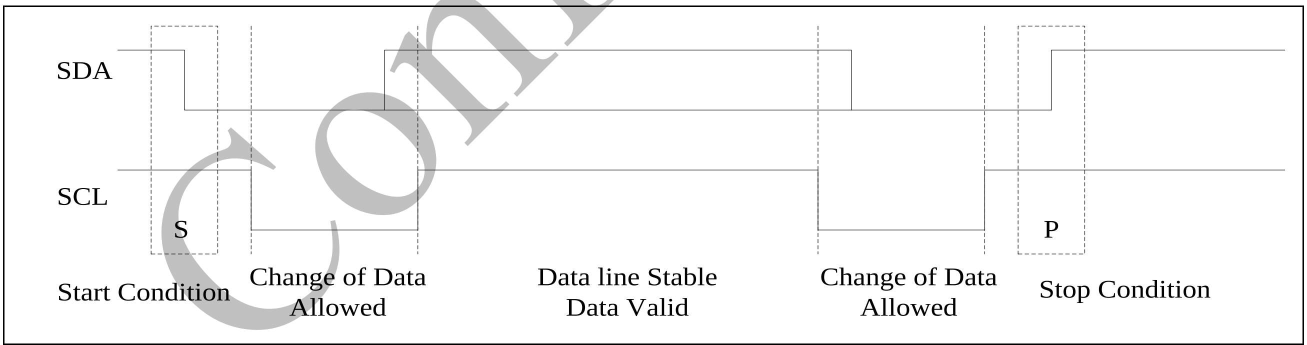


Figure 3-34 Start and Stop Diagram

3.10.4.4 Address Format

The following diagram shows the 7-bit address timing diagram during transmission.

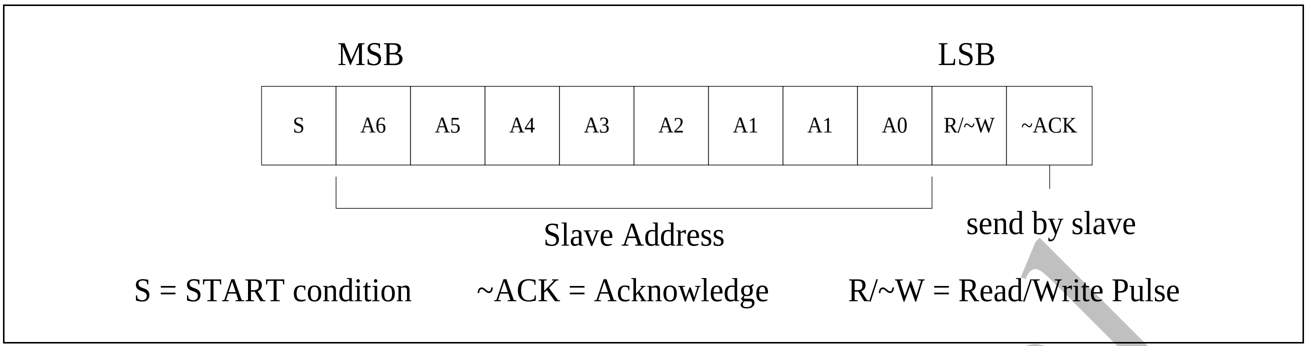


Figure 3-35 7-bit Address Timing Diagram

3.10.4.5 Clock Division

In master mode, the SCL clock is derived from the system clock through frequency division. The clock divider ratio is controlled by bits CR0, CR1, and CR2 of the I2C Control Register (I2CCON). The following table shows the CLKO rate in master mode. The BCLK input referenced is connected to the Timer 1 overflow frequency.

Table 3-8 Clock Division

SYS_CLK (MHz)	CR2	CR1	CR0	Bit Frequency (kbps)	CLK Divided by
8	0	0	0	31.25	256
8	0	0	1	35.7	224
8	0	1	0	41.67	192
8	0	1	1	150	160
8	1	0	0	8.3	960
8	1	0	1	66.67	120
8	1	1	0	133.3	60
8	1	1	1	Forbidden	
32	0	0	0	125	256
32	0	0	1	142.86	224
32	0	1	0	166.67	192
32	0	1	1	200	160
32	1	0	0	33.33	960
32	1	0	1	266.67	120
32	1	1	0	533.3	60
32	1	1	1	Forbidden	

When configuring the I2C system clock to 8 MHz (i.e., APB1_DIV = 2), dividing 8 MHz by 60 yields a SCL clock frequency of 133.3 kHz.

3.10.4.6 Input Filtering

The input signal is synchronized with the clock (“CLK”), and glitches shorter than three clock cycles are filtered out. Each filter consists of three flip-flops. The first flip-flop locks the input, while the other two form a shift register. When the second and third flip-flops are in the ‘11’ or “00” state, they are in the set or reset state.

3.10.4.7 Interrupt Generation

The I2C module has 15 states. When entering any of the 14 states, the “SI” bit in the I2C Control Register (I2CCON) is set by hardware and triggers an interrupt. The only state that does not trigger an interrupt or set is F8H, indicating that no I2C timing was detected. Within the interrupt service routine, the “SI” bit in the I2C Control Register (I2CCON) must be cleared by software writing ‘0’; writing “1” is invalid.

3.10.4.8 Polling Status

The MCU controls I2C through three function registers: the I2C Control Register (I2CCON), the I2C Data Register (I2CDAT), and the I2C Status Register (I2CSTA).

- I2C Control Register (I2CCON): contains the I2C enable bit “ENS1”, clock rate setting bits “CR0”, “CR1”, and “CR2”, start bit “STA” and stop bit “STO”, and the interrupt request flag “SI”, which requires monitoring the FSM during interrupts and clearing promptly.
- I2C Data Register (I2CDAT): contains bytes transmitted via the I2C bus or bytes received via the I2C bus. The MCU should only read it when an I2C interrupt occurs.
- I2C Status Register (I2CSTA): indicates the FSM state of the I2C module. The three lowest bits of this register are always set to zero. The I2C module has 15 possible states. Interrupts are generated when entering any of the 14 states. The only state that does not trigger an interrupt is F8H. This register records the I2C operational status. The software reads this status and performs corresponding actions.

The specific status are shown in the table below: “SLA” indicates the slave address; ‘R’ indicates R/W = 1, reading the slave address; “W” indicates R/W = 0, writing the slave address.

Table 3-9 I2C Status in Master Transmit Mode

Code	Status of Code	Software Response					I2C Hardware Next Action
		To/From I2CDAT	To I2CCON				
			STA	STO	SI	AA	
08H	The start signal has been initiated.	Load SLA+W	X	0	0	X	SLA+W will be sent. ACK will be received.
10H	The start signal is initiated once again.	Load SLA+W	X	0	0	X	SLA+W will be sent. ACK will be received.
		Load SLA+R	X	0	0	X	SLA+R will be sent. I2C will switch to MST/REC mode.
18H	SLA+W has been sent. ACK has been received.	Load data	0	0	0	X	Data will be sent. ACK will be received.
		No action	1	0	0	X	The start signal will be initiated once again.
		No action	0	1	0	X	The stop signal will be initiated. The STO flag will be reset.
		No action	1	1	0	X	The stop signal will be initiated, followed immediately by the start signal. The STO flag will be reset.
20H	SLA+W has been sent, but no ACK has been received.	Load data	0	0	0	X	Data will be sent. ACK will be received.
		No action	1	0	0	X	The start signal will be initiated once again.
		No action	0	1	0	X	The stop signal will be initiated. The STO flag will be reset.
		No action	1	1	0	X	The stop signal will be initiated, followed immediately by the start signal. The STO flag will be reset.
28H	Data has been sent. ACK has been received.	Load data	0	0	0	X	Data will be sent. ACK will be received.
		No action	1	0	0	X	The start signal will be initiated once again.
		No action	0	1	0	X	The stop signal will be initiated. The STO flag will be reset.
		No action	1	1	0	X	The stop signal will be initiated, followed immediately by the start signal. The STO flag will be reset.
30H	Data has been sent. but no ACK has been received.	Load data	0	0	0	X	Data will be sent. ACK will be received.
		No action	1	0	0	X	The start signal will be initiated once again.
		No action	0	1	0	X	The stop signal will be initiated. The STO flag will be reset.
		No action	1	1	0	X	The stop signal will be initiated, followed immediately by the start signal. The STO flag will be reset.

Table 3-10 I2C Status in Master Receive Mode

Code	Status of Code	Software Response					I2C Hardware Next Action
		To/From I2CDAT	To I2CCON				
			STA	STO	SI	AA	
08H	The start signal has been initiated.	Load SLA+R	X	0	0	X	SLA+R will be sent. ACK will be received.
10H	The start signal is initiated once again.	Load SLA+R	X	0	0	X	SLA+R will be sent. ACK will be received.
		Load SLA+W	X	0	0	X	SLA+W will be sent. I2C will switch to MST/ TRX mode.
40H	SLA+R has been sent. ACK has been received.	No action	0	0	0	0	Data will be received, but no ACK has been received.
		No action	0	0	0	1	Data will be received. ACK will be returned.
48H	SLA+R has been sent, but no ACK has been received.	No action	1	0	0	X	The start signal will be initiated once again.
		No action	0	1	0	X	The stop signal will be sent. The STO flag will be reset.
		No action	1	1	0	X	The stop signal will be initiated, followed immediately by the start signal. The STO flag will be reset.
50H	Data has been received. ACK has been returned.	Read data	0	0	0	0	Data will be received, but no ACK will be returned.
		Read data	0	0	0	1	Data will be received. ACK will be returned.
58H	Data has been received, but no ACK has been returned.	Read data	1	0	0	X	The start signal will be initiated once again.
		Read data	0	1	0	X	The stop signal will be initiated. The STO flag will be reset.
		Read data	1	1	0	X	The stop signal will be initiated, followed immediately by the start signal. The STO flag will be reset.

Table 3-11 Other States of I2C

Code	Status of Code	Software Response					I2C Hardware Next Action
		To/From I2CDAT	To I2CCON				
			STA	STO	SI	AA	
38H	Arbitration failed	No action	0	0	0	X	I2C will be released. The start signal will be initiated.
		No action	1	0	0	X	When in idle state (entering master mode)
F8H	No relevant valid information; SI=0	No action	No action				Wait or continue waiting for the current transmission
00H	Bus error during MST	No action	0	1	0	X	In MST mode, internal hardware will be affected. In all cases, the bus will be released. The STO flag will be reset.

3.10.5 I2C Register Map

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

Register	Offset	R/W	Description	Reset Value
I2C Base Address: 0×4000_0000				
I2C2CON	0x00	R/W	I2C Control Register	0x00
I2C2DAT	0x04	R/W	I2C Data Register	0x00
I2C2STA	0x08	R	I2C Status Register	0xF8

3.10.6 I2C Register Description

3.10.6.1 I2C Control Register (I2CCON)

The I2CCON register is used to configure the I2C module. The CPU can read or write to this register. Two bits within it are affected by the I2C hardware circuitry:

- The “SI” bit is set to 1 when an interrupt is triggered.
- The “STO” bit is cleared after a STOP command is transmitted.

Register	Offset	R/W	Description	Reset Value
I2CCON	0x00	R/W	I2C control register	0x00

Bits	Description	
[7]	CR2	Clock control
[6]	ENS1	I2C enable bit. When ENS1=0, the SDA and SCL signal lines are high. The I/O port is high-impedance state. When ENS1=1, the module is enabled.
[5]	STA	Start Flag When STA=1, I2C checks the bus status. If idle, it will initiate the START command.
[4]	STO	Stop Flag When STO=1, I2C sends the STOP command to the bus. After the command is sent, the hardware automatically resets.
[3]	SI	Interrupt Flag Whenever I2C enters any state (excluding the STATUS == 0xF8 state), an interrupt is triggered and this interrupt flag is pulled high. Writing 0 to reset the software, writing 1 has no effect.
[2]	AA	ACK Flag When AA=1, I2C returns ACK signal on the bus after receiving data. When AA=0, I2C returns NACK signal on the bus after receiving data.
[1:0]	CR1,CR0	Clock control

3.10.6.2 I2C Data Register (I2CDAT)

The I2CDAT register is a read/write buffer used to store data to be sent to the I2C bus or data received from the bus. The CPU can read from or write to this register.

Register	Offset	R/W	Description	Reset Value
I2C2DAT	0x04	R/W	I2C data register	0x00

Bits	Description	
[7:0]	DAT	Read/Write buffer

3.10.6.3 I2C Status Register (I2CSTA)

The I2CSTA register stores the current I2C status. The CPU can only read the value of this register.

Register	Offset	R/W	Description	Reset Value
I2CSTA	0x08	R	I2C status register	0xF8

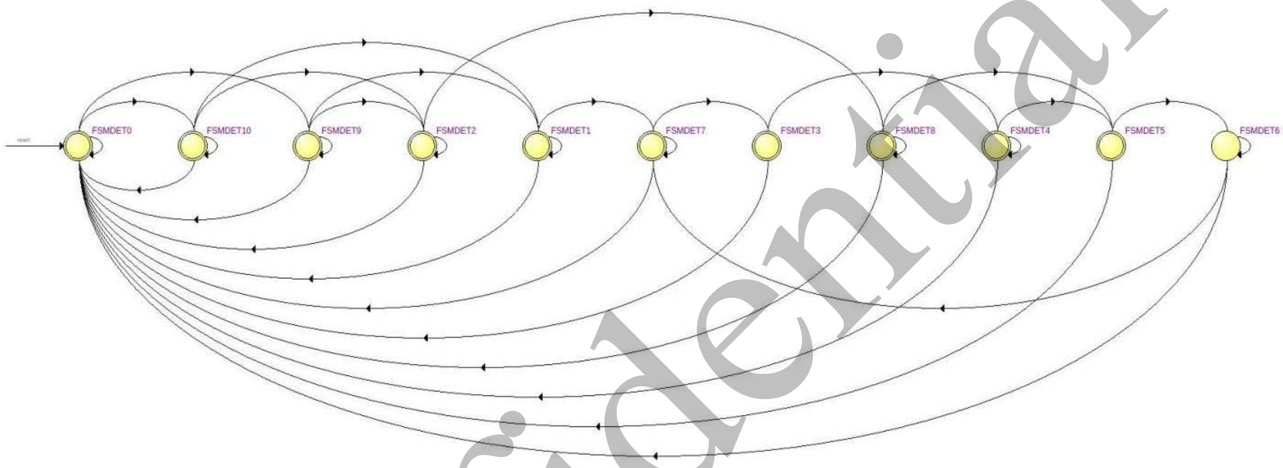
Bits	Descriptions	
[7:3]	STATUS	I2C status
[2:0]	Reserved	Reserved.

3.10.7 Hardware Finite State Machine

There are four finite state machines(FSM) inside the I2C module:

- FSMDET: Detect START/STOP command.
- FSMSYNC: Used for synchronizing clocks.
- FSMMOD: Detect working mode.
- FSMSTA: Control I2C operation.

3.10.7.1 FSMDET



FSMDET detects START/STOP commands. After detecting a START/STOP command, the state machine waits for 300ns. After detecting an SCL rising edge, the state machine waits for 200ns. In the 8051XC2 system design, the maximum clock frequency is 24MHz, with the parameters controlling the wait time set as SDA_SAMPLE_COUNT=5 and SDA_SWITCH_COUNT=8.

FSMDET0: SCL = 0, SDA undefined. When SCL is low, the state machine may lock up in this state.

FSMDET1: Bus idle, SCL=1, SDA=1. Detect START command.

FSMDET2: SCL=1, SDA=0. Detect STOP command.

FSMDET3: SCL=1, SDA=0. Detect START command.

FSMDET4: SCL=1, SDA=0. Detect STOP command.

FSMDET5: SCL=1, SDA=1. Detect STOP command, and wait for bus idle;

FSMDET6: Bus idle

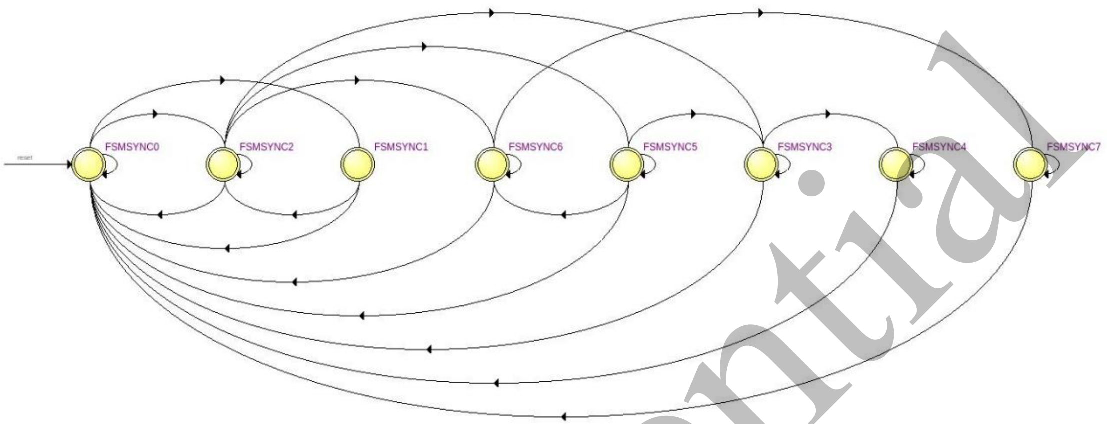
FSMDET7: SCL=1, SDA=0, START command detected, and wait 300ns.

FSMDET8: SCL=1, SDA=1, STOP command detected, and wait 300ns.

FSMDET9: SCL=1, SDA=1. Enter this state after detecting SCL pull up, and wait for at least 200ns;

FSMDET10: SCL=1, SDA=0. Enter this state after detecting SCL pull up, and wait for at least 200ns.

3.10.7.2 FSMSYNC



FSMSYNC is used to control SCL generation and clock synchronization.

FSMSYNC0: Reset state. SCL is high. Wait for high-level counting to complete.

FSMSYNC1: SCL is low. Initiat low-level count.

FSMSYNC2: SCL is low. Wait for low-level counting to complete.

FSMSYNC3: Confirm if SCL is high.

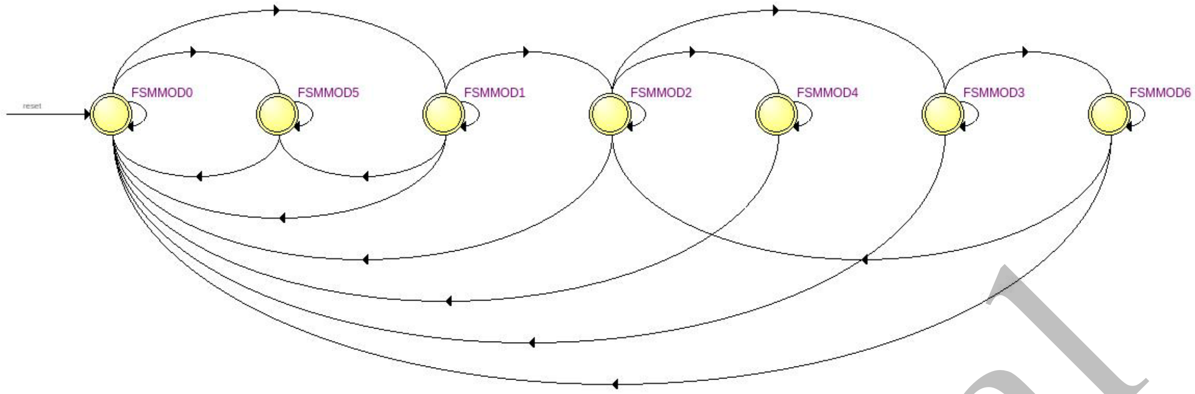
FSMSYNC4: SCL low level extended.

FSMSYNC5: SCL is low. Wait for interrupt processing to complete.

FSMSYNC6: SCL is low. Wait for SDA to be at low level.

FSMSYNC7: Wait for next transmission.

3.10.7.3 FSMMOD



FSMMOD is used to detect I2C working mode.

FSMMOD0: Default state. Waiti for software startup.

FSMMOD1: I2C sends START command and waits for SCL to pull down.

FSMMOD2: I2C sends address or data, waits for STOP command or RESTART command, and detects BUS ERROR.

FSMMOD3: Wait for RESTART command transmission to complete.

FSMMOD4: Wait for STOP command transmission to complete.

FSMMOD5: Send two SCL clock signals to resolve I2C bus hangs (SDA is pulled down by the slave, and the master cannot send START command).

FSMMOD6: Wait for RESTART command transmission to complete.

3.11 Analog-to-Digital Converter (ADC)

3.11.1 Overview

ADC contains one successive approximation analog-to-digital converters (SAR A/D converter) with multiple input channels. The A/D converters can be started by software, external pin or PWM trigger.

3.11.2 Features

- Analog external channel input voltage range: 0 ~ AVDD
- Analog internal channel input voltage range: 0 ~ VBG
- Support 12-bit resolution
- Support averaging function(optional). Maximum of 128 times, minimum of 1 time.
- Multiple input channels are optional
  - Multiple GPIO analog external input channels
  - Internal bandgap inpu
  - Temperature input
  - 1/4 VDD input
- An A/D conversion can be started by:
  - Software trigger
  - External pin trigger
  - PWM trigger
  - Internally configurable trigger sources at 8kHz/16kHz/32kHz/64kHz (suitable for audio applications)
- Each Conversion result is held in data register with valid and overrun indicators
- Support FIFO mode, FIFO depth is 8.
- Support multiple sampling rates (256K, 500K, 1M, 2M), with a default sampling rate of 1M.
- ADC specifications:

Symbol	Conditions	Parameter			Unit	Remark
		Min	Type	Max		
System clock		-	32	-	MHz	
Power AVDD		1.8	3.3	3.6	V	
Power DVDDL		1.1	1.2	1.3	V	
Data Rate		0.256	1	2	Msps	

Input Voltage		0	-	VREFP	V	
Internal Reference Voltage		VBG	AVDD	AVDD	V	
Input Capacitor		-	3	-	pF	
Operating Temperature		-40	27	125	℃	
Operating Current	1Msps	-	350	-	uA	Differential Mode
	1Msps	-	650	-	uA	Single-ended Mode
Settling Time	Buffer off	-	-	5	us	
	Buffer on	-	-	2	us	
INL		-1.5	-	1.5	LSB12	After averaging 128 times internally
DNL		-1	-	1	LSB12	
ENOB	20K sine in	-	11	-	bits	VDD Differential Mode
SFDR		-	80	-	dB	
SNR		-	69	-	dB	

3.11.3 Block Diagram

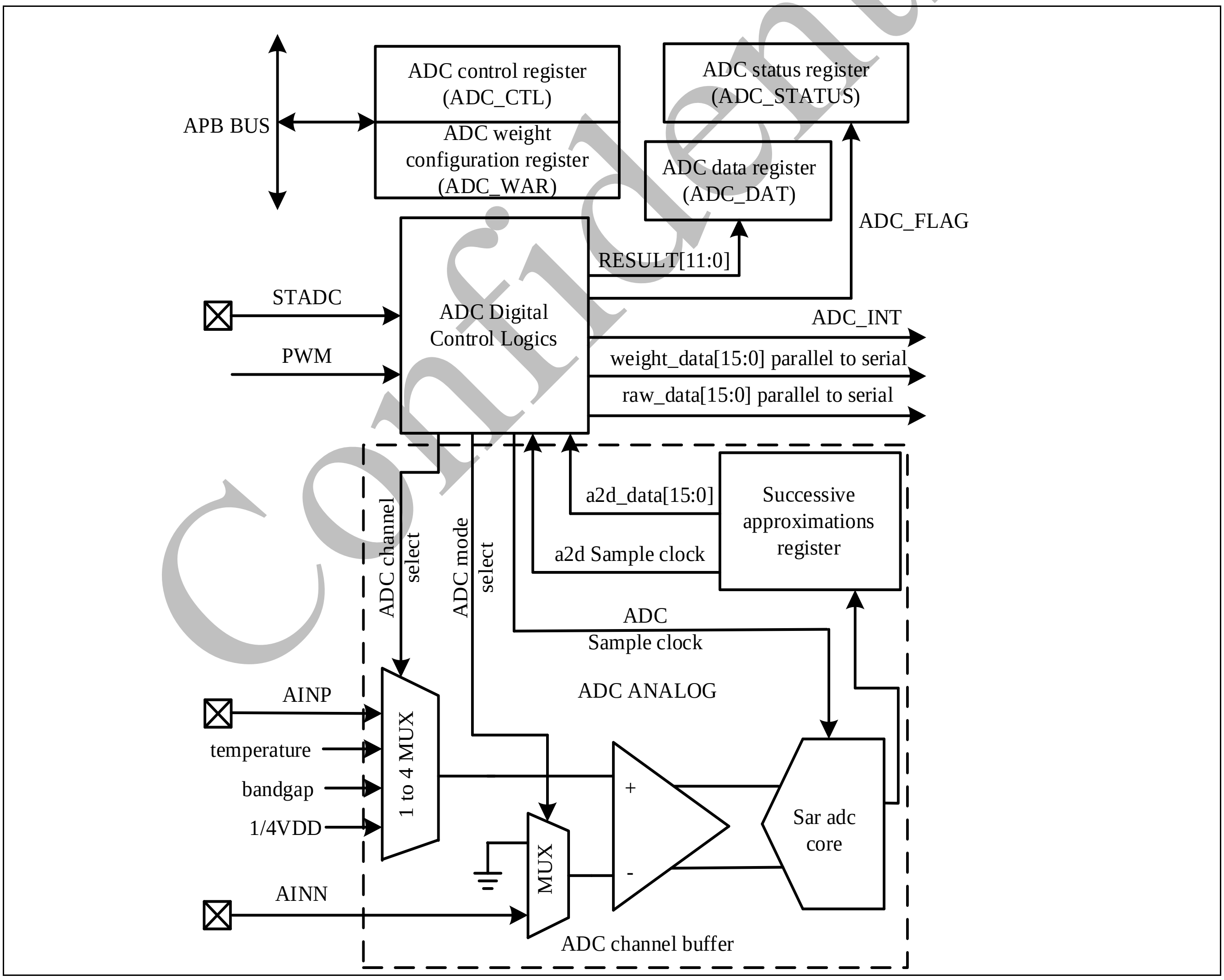


Figure 3-36 ADC Control System Block Diagram

3.11.3.1 ADC Data Path

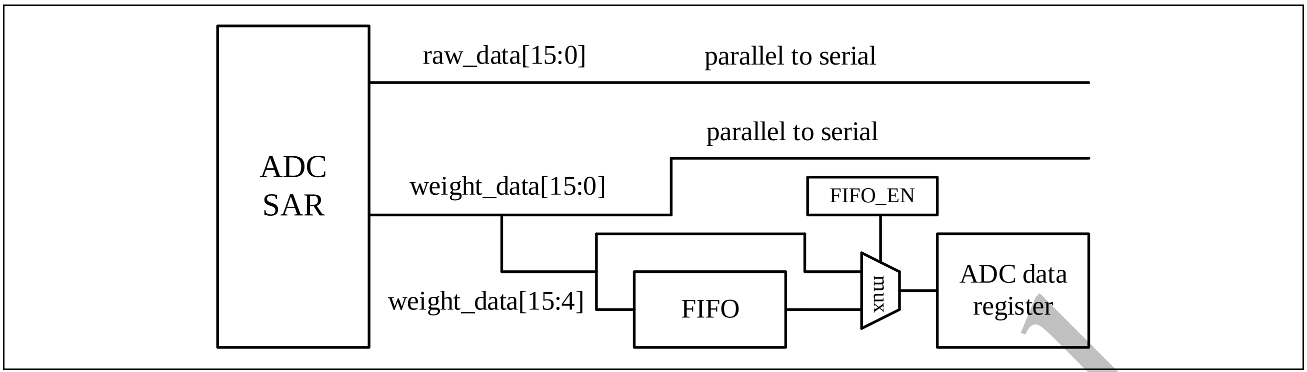


Figure 3-37 ADC Data Path Diagram

3.11.4 Functional Description

3.11.4.1 ADC Operation

The operation is as follows:

1. Confirm the ADC sampling rate and the clock for receiving data in the digital part.
2. If the values in the weight registers ADC_WDR1/2/3/4/5 need to be changed, make the changes in advance.
3. Confirm the trigger mode to use: Software trigger, hardware trigger, and HW TMR trigger for audio applications are available. Hardware trigger is further divided into external PIN trigger and PWM trigger. The register defaults to software trigger. Each trigger becomes effective only after reading the previous data completion FLAG or VALID signal (regardless of trigger mode). The trigger signal must be enabled after the functional mode configuration (e.g., software/hardware trigger selection) is completed. After enabling HW TMR via the HW_TME_EN bit, HW TMR trigger takes the highest priority, directly overriding other software/hardware triggers. The trigger frequency can be selected via HW_TMR_SEL. High trigger frequencies require sufficient MCU core processing speed to prevent data loss.
4. Confirm whether the trigger is continuous (check if the CON_SAMPLE register is enabled). Continuous means that once triggered, the ADC continuously samples data (outputting multiple data). To disable sampling in continuous mode, disable the continuous enable signal (After the last sampling, the hardware automatically pulls the trigger signal low upon detection of the VALID signal), or directly disable the ADC enable signal. Otherwise, data sampling will continue indefinitely. Non-continuous

trigger will only output a single data.

5. Use the averaging function. The averaging function calculates the average by removing the minimum and maximum values after multiple continuous sampling (without needing to enable the CON_SAMPLE register), outputting only a single data. The average frequency is optional. The averaging function can also be set to continuous operation (needing to enable the CON_SAMPLE register), outputting multiple data.
6. Data may be discarded optionally, with the specific quantity configurable. The default value is 1. The reason for discarding is that there is a data setup time when the ADC first starts working, causing errors in the previous data. If ANA_WORK = 0, it is recommended to discard the first data whenever sampling is retriggered.
7. After data sampling completes and the VALID signal goes high, a FLAG is generated. If interrupt is enabled, an interrupt will occur.
8. After the sampling is complete, the data results will be stored in either the FIFO or the register. If the FIFO is enabled, the data will be stored in the FIFO; otherwise, it will be stored in the register.

ANA_WORK=0 indicates that the analog circuit will not operate when the trigger signal is disabled. ANA_WORK=1 indicates that the analog circuit will continue to operate after the trigger signal is disabled. EOC defaults to high.

Whenever the sampling is triggered for the first time, discard the first data to ensure sampling reliability. Afterwards, if you want to directly enable the trigger signal to ensure correct data sampling, you can disable the analog sampling clock. Note: When the ANA_WORK register is set to high, wait for about 2us (after the analog data setup time has ended) before triggering (whether it is the first trigger or not). Under these conditions, data does not need to be discarded. Otherwise, it is recommended to discard the data.

After the trigger signal is enabled, the hardware enters internal shutdown mode. To disable sampling in continuous mode, simply disable the continuous enable signal or directly disable the ADC enable signal.

The ADC design is a fully differential structure. To enhance the accuracy of the ADC sampling output, the sampled data needs to be weighted.

The weight calculation formula is as follows:

$$F_{out} = s(16) + \sum_{i=1}^{15} \{2 * b[i] - 1\} * s(i) + \{b[0] - 1\} * s(0)$$

s[i] is a variable weight value (configured by the register), while s(16) is a fixed value. The default weight values for s[i] are shown in the table below:

s[i]	Weight Value
s16	2048
s15	816
s14	480
s13	328
s12	176
s11	100
s10	64
s9	36
s8	22
s7	12
s6	6
s5	4
s4	2
s3	1
s2	0.5
s1	0.5
s0	0.5

Weight values are not always constant; different chips have different weight values, which are obtained through FT calibration. However, from the perspective of overall weight values, the difference between each weight value change and the original weight value is not significant. To perform averaging, multiple results must be continuously sampled and weighted before averaging. The maximum and minimum values are then removed, and the average of the cumulative sampled results is calculated to obtain a new ADC data. To perform average sampling, first delete DISC_NUM data, then continuously sample the DC_TIMES data, remove the maximum and minimum values, calculate the average value, and finally output.

3.11.4.2 ADC Interrupts and FLAG

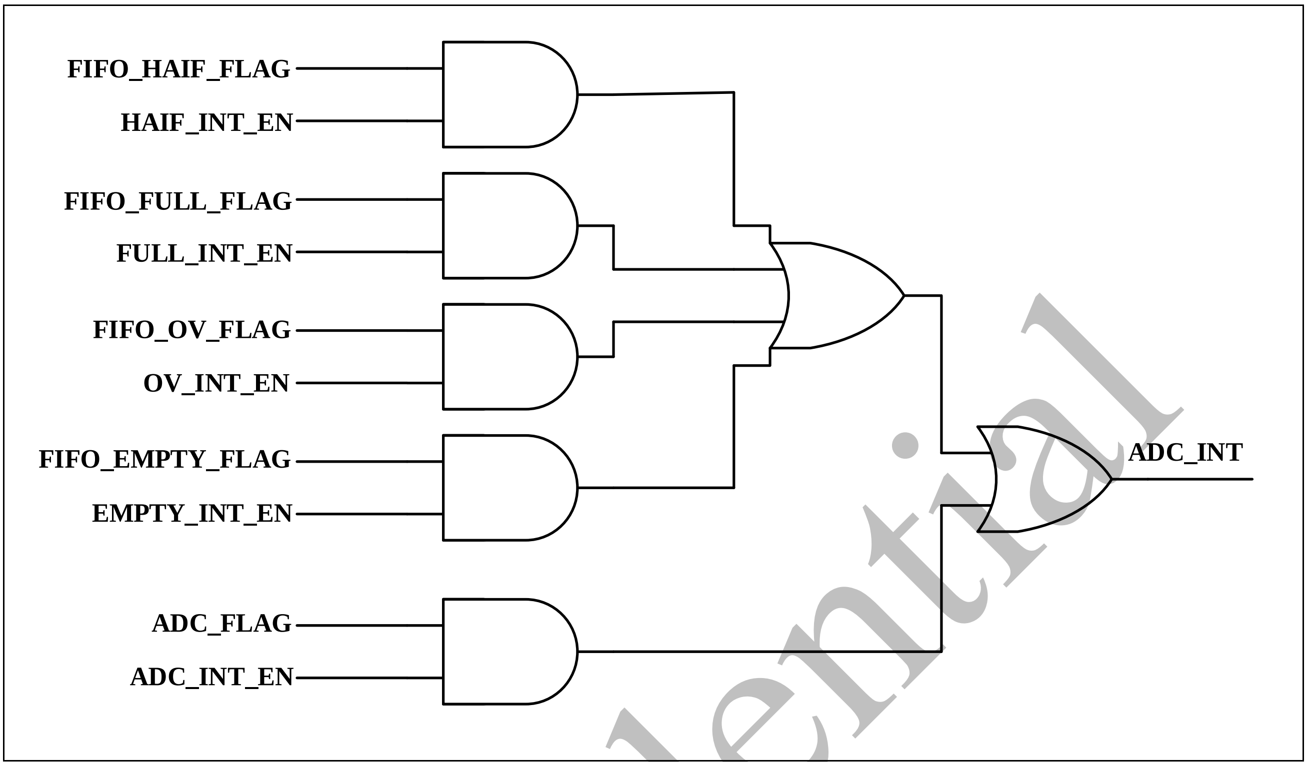


Figure 3-38 ADC Interrupt

As shown in Figure 3-38, there are two types of FLAGs: one is the FLAG generated by the ADC completing one sampling, and the other is the internal state FLAG of the FIFO. The FIFO internal status FLAG is read-only and automatically cleared by the hardware. The interrupts it generates are also automatically cleared by the hardware. If interrupts are not required, disable the interrupt enable for each FLAG. When ADC_INT or ADC_FLAG is pulled high, write 1 to ADC_FLAG to clear it, thereby also clearing the interrupt.

Precautions for use:

1. During data sampling, if you want to change the sampling mode, you must first disable ADC_EN.
2. If the value obtained after calculating the weight is negative, the data output will be 0; if it exceeds 16'hFFFF, the data output will be 16'hFFFF.
3. Configure the weight registers ADC_WDR1/2/3/4/5 first. Do not change these weight registers during data sampling, otherwise the output data will have errors.

3.11.5 ADC Register Map

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

Register	Offset	R/W	Description	Reset Value
ADC_BA: 0x40005000				
ADC_DAT	ADC_BA+0x00	R	ADC Data Register	0x0000_0000
ADC_CTL	ADC_BA+0x04	R/W	ADC Control Register	0x0000_0000
ADC_STATUS	ADC_BA+0x08	R/W	ADC Status Register	0x0000_0000
ADC_CTL2	ADC_BA+0x0C	R/W	ADC Control Register 2	0x8080_1310
ADC_WDR1	ADC_BA+0x10	R/W	ADC Weight Difference Register 1	0x4000_C808
ADC_WDR2	ADC_BA+0x14	R/W	ADC Weight Difference Register 2	0xCC02_4060
ADC_WDR3	ADC_BA+0x18	R/W	ADC Weight Difference Register 3	0xF002_C0C0
ADC_WDR4	ADC_BA+0x1C	R/W	ADC Weight Difference Register 4	0xA405_8040
ADC_WDR5	ADC_BA+0x20	R/W	ADC Weight Difference Register 5	0x8008_1020
ADC_TEMP	ADC_BA+0x24	R/W	ADC Temperature Sensor	0x0200_0000

3.11.6 ADC Register Description

3.11.6.1 ADC Data Register (ADC_DAT)

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

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	RESULT	A/D Conversion Result FIFO_EN=0，RESULT is the data that does not pass through the FIFO. FIFO_EN=1，RESULT is the data that passed through the FIFO. When DWIDTH_SEL=1 and FIFO_EN=0, RESULT is 16-bit data; otherwise, it is 12-bit data.

3.11.6.2 ADC Control Register (ADC_CTL)

Register	Offset	R/W	Description	Reset Value
ADC_CTL	ADC_BA+0x04	R/W	A/D Control Register	0x0000_0000

Bits	Description	
[31:16]	Reserved	Reserved
[15]	DWIDTH_SEL	Read Data Bit Width Selection 0: ADC_DAT: 12-bit data 1: ADC_DAT: 16-bit data (Effective only when FIFO_EN=0)
[14:13]	HW_TMR_SEL	The compare value of hardware timer select 00: 8kHz 01: 16kHz 10: 32kHz 11: 64kHz
[12]	HW_TMR_EN	Hardware Timer Enable HW_TMR trigger priority is higher than other software/hardware triggers. 0: timer disable 1: timer enable Default: 1’b0
[11]	HALF_INT_EN	FIFO_HALF Interrupt Enable 0: FIFO_HALF interrupt disable 1: FIFO_HALF interrupt enable Default: 1’b0
[10]	OV_INT_EN	FIFO_OV Interrupt Enable 0: FIFO_OV interrupt disable 1: FIFO_OV interrupt enable Default: 1’b0
[9]	EMPTY_INT_EN	FIFO_EMPTY Interrupt Enable 0: FIFO_EMPTY interrupt disable 1: FIFO_EMPTY interrupt enable Default: 1’b0
[8]	FULL_INT_EN	FIFO_FULL Interrupt Enable 0: FIFO_FULL interrupt disable 1: FIFO_FULL interrupt enable Default: 1’b0
[7]	PWM_CH_SEL	PWM Trigger Source Selection (Fixed to PWM Cycle Mode) 0: PWM trigger source is PWM_CH0 FLAG. 1: PWM trigger source is PWM_CH1 FLAG. Default: 1’b0
[6]	SWTRG	Software Trigger Enable Hardware automatically clears after each trigger completion. Default: 1’b0
[5]	FIFO_EN	ADC FIFO Enable 0: ADC FIFO disable 1: ADC FIFO enable Default: 1’b0

[4]	HWTRGEN	Hardware Trigger Enable 0: Hardware trigger disable 1: Hardware trigger enable Default: 1'b0
[3]	HWTRGCOND	External pin trigger edge function selection. At least 4 PCLKs must remain stable in both high and low voltage states to effectively trigger. 0: Falling edge trigger 1: Rising edge trigger Default: 1'b0
[2]	HWTRGSEL	Hardware Trigger Source Selection 0: External STADC PIN trigger 1: PWM trigger Default: 1'b0
[1]	ADCIEN	ADC Interrupt Enable 0: ADC interrupt disable 1: ADC interrupt enable Default: 1'b0
[0]	ADCEN	ADC Sampling Enable 0: ADC not operating 1:ADC operating normally Default: 1'b0

3.11.6.3 ADC Status Register (ADC_STATUS)

Register	Offset	R/W	Description	Reset Value
ADC_STATUS	ADC_BA+0x08	R/W	ADC Status Register	0x0000_0000

Bits	Description	
[31:9]	Reserved	Reserved
[8]	ADC_FLAG	ADC Conversion End Flag Note: This bit can be cleared to 0 by writing a 1 in software. Default: 1'b0
[7]	Reserved	Reserved.
[6]	BUSY	The ADC is currently sampling data. The internal trigger signal has not yet been automatically pulled low by the hardware.
[5]	FIFO_HALF_FALG	FIFO Half-Full Flag
[4]	FIFO_OV_FALG	FIFO Overflow Flag
[3]	FIFO_EMPTY_FALG	FIFO Empty Flag
[2]	FIFO_FULL_FALG	FIFO Full Flag
[1]	OV	Data Overflow Flag, active only when FIFO_EN=0 OV is set to 1 if the conversion data in RESULT has not been read before loading a new conversion result into this register. It is cleared by hardware after reading the ADC_DAT register. 0: Data in RESULT is the latest conversion result. 1: Data in RESULT is overwritten.
[0]	VALID	Valid Flag, active only when FIFO_EN=0 This bit is set to 1 when the ADC conversion completes and is cleared by hardware after reading the ADC_DAT register. 0: Data in the RESULT bit is invalid. 1: Data in the RESULT bit is valid.

3.11.6.4 ADC Control Register 2 (ADC_CTL2)

Register	Offset	R/W	Description	Reset Value
ADC_CTL2	ADC_BA+0x0C	R/W	A/D Control Register 2	0x8080_1310

Bits	Description	
[31:29]	BIT_TCTRL	Bit-By-Bit Conversion Timing Control Total time: 440 ns to 580 ns, 20 ns per step Default:3’h4
[28:27]	BUF_CHOP	Internal buffer chop frequency 00: 0. 01: 1. 10: Sampling clock 8 division. 11: Sampling clock 32 division. Default:2’h0
[26]	BUFFER_EN	ADC channel BUF enable for single ended DC mode 0: Buffer disable. 1: Buffer enable. Default:1’b0
[25:24]	Reserved	Reserved.
[23:18]	GPADC_VBG_TRIM	GPADC 1.2 V Reference Voltage Trim
[17:16]	GPADC_CHANNEL_SEL	ADC Channel Selection 00: External Channel 01: VTEMP Channel 10: VBG Channel 11: 1/4 VDD Channel
[15:13]	CLK_DIV	ADC Digital Clock Division ADC Digital Processing Clock Frequency = ADC Digital Clock / (ADC_DIV + 1). Note: The condition “ADC Digital Processing Clock Frequency / Sampling Clock Frequency ≥ 16” must be satisfied. Default:3’b0
[12:11]	CLK_SEL	Sample Rate Clock Division Selection 00: 125 01: 64 10: 32 11: 16 Default:2’h2
[10:8]	DC_TIMES	Sampling frequency N (only supports averaging function) N(0~7): 2^N+2 times Default: 3’h3
[7:4]	DISC_NUM	Number of data items to be discarded during sampling. Default:4’h1
[3]	ANA_WORK	Does ADC analog operation depend on the trigger signal? 0: Analog operation depends on the trigger signal state 1: Analog operation is independent of the trigger signal Default: 1’b0
[2]	CON_SAMPLE	Continuous Sampling Enable

		0: Data is not sampled continuously. 1: Data is sampled continuously. Default: 1’b0
[1]	AVERAGE	Sample Data Averaging Enable 0: Data is not averaged. 1: Data requires averaging. Default: 1’b0
[0]	Reserved	Reserved.

3.11.6.5 ADC Weight difference Register 1 (ADC_WDR1)

Register	Offset	R/W	Description	Reset Value
ADC_WDR1	ADC_BA+0x10	R/W	A/D Weight Difference Register 1	0x4000_C808

Bits	Description	
[31:27]	S1	Bit31: Integer bit Bit30~Bit27: Decimal bit Default: 5’d8
[26:16]	Reserved	Reserved.
[15:5]	S11	Bit15~Bit9: Integer bit Bit8~Bit5: Decimal bit Default: 11’d1600
[4:0]	S0	Bit4: Integer bit Bit3~Bit0: Decimal bit Default: 5’d8

3.11.6.6 ADC Weight difference Register 2 (ADC_WDR2)

Register	Offset	R/W	Description	Reset Value
ADC_WDR2	ADC_BA+0x14	R/W	A/D Weight Difference Register 2	0xCC02_4060

Bits	Description	
[31:18]	S15	Bit30~Bit22: Integer bit Bit21~Bit18: Decimal bit Default: 14’d13056
[17:8]	S9	Bit17~Bit12: Integer bit Bit11~Bit8: Decimal bit Default: 10’d576
[7:0]	S6	Bit7~Bit4: Integer bit Bit3~Bit0: Decimal bit Default: 8’d96

3.11.6.7 ADC Weight difference Register 3 (ADC_WDR3)

Register	Offset	R/W	Description	Reset Value
ADC_WDR3	ADC_BA+0x18	R/W	A/D Weight Difference Register 3	0xF002_C0C0

Bits	Description	
[31:19]	S14	Bit31~Bit23: Integer bit Bit22~Bit19: Decimal bit Default: 13’d7680
[18:9]	S8	Bit18~Bit13: Integer bit Bit12~Bit9: Decimal bit Default: 10’d352
[8:0]	S7	Bit8~Bit4: Integer bit Bit3~Bit0: Decimal bit Default: 9’d192

3.11.6.8 ADC Weight difference Register 4 (ADC_WDR4)

Register	Offset	R/W	Description	Reset Value
ADC_WDR4	ADC_BA+0x1C	R/W	A/D Weight Difference Register 4	0xA405_8040

Bits	Description	
[31:19]	S13	Bit31~Bit23: Integer bit Bit22~Bit19: Decimal bit Default: 13’d5248
[18:7]	S12	Bit18~Bit11: Integer bit Bit10~Bit7: Decimal bit Default: 12’d2816
[6:0]	S5	Bit6~Bit4: Integer bit Bit3~Bit0: Decimal bit Default: 7’d64

3.11.6.9 ADC Weight difference Register 5 (ADC_WDR5)

Register	Offset	R/W	Description	Reset Value
ADC_WDR5	ADC_BA+0x20	R/W	A/D Weight Difference Register 5	0x8008_1020

Bits	Description	
[31:21]	S10	Bit31~Bit25: Integer bit Bit24~Bit21: Decimal bit Default: 11’d1024
[20:14]	S4	Bit20~Bit18: Integer bit Bit17~Bit14: Decimal bit Default: 7’d32
[13:8]	S3	Bit13~Bit12: Integer bit Bit11~Bit8: Decimal bit Default: 6’d16
[7:2]	S2	Bit7~Bit6: Integer bit Bit5~Bit2: Decimal bit Default: 6’d8
[1:0]	Reserved	Reserved.

3.11.6.10 ADC Temperature Sensor (ADC_TEMP)

Register	Offset	R/W	Description	Reset Value
ADC_TEMP	ADC_BA+0x24	R/W	A/D Temperature Sensor	0x0200_0000

Bits	Description	
[31]	GPADC_CAL_CAP_EN	ADC Automatic Calibration Capacitor Enable Default: 1'b0
[30]	GPADC_CAL_CMP_EN	ADC Automatic Calibration Comparator Enable Default: 1'b0
[29:27]	GPADC_CALCAP_SEL	ADC Automatic Calibration Capacitor Selection (3'b000 to 3'b111 correspond to weighting capacitor 1 to 8) Default: 3'b000
[26]	GPADC_CALCAP_SIDE_SEL	ADC Automatic Calibration Capacitor Selection - N/P side Default: 1'b0
[25:20]	GPADC_CALCMP_VOS	VOS Trim for ADC Auto-Calibration Comparators Default: 6'b100000
[19]	Reserved	Reserved.
[18]	GPADC_VREFP_BYP_EN	ADC vrefp bypass Enable 1: vrefp=VBAT 0: Can be bypassed Default: 1'b0
[17]	GPADC_VREFP_VBG_EN	ADC vrefp VBG Enable 1: vrefp=VBG 0: other modes Default: 1'b0
[16]	GPADC_DIFFMODE	ADC Channel Differential Mode Enable 0: Single-ended mode 1: Differential mode
[15:13]	GPADC_PGA_GAIN	ADC PGA Gain Control
[12]	GPADC_VBOOST_BYP_EN	Internal Boost Switching Power Supply for ADC 1: VBAT 0: 2V
[11:3]	Reserved	Reserved.
[2:1]	GPADC_VBOOST_TRIM	GPADC BOOST Voltage Regulation
[0]	GPADC_VREF_TEST_EN	GPADC VREF Voltage Test Enable 1: On 0: Off

3.12 Serial Peripheral Interface (SPI)

3.12.1 Overview

The SSP is a master or slave interface for synchronous serial communication with peripheral devices that have either Motorola SPI, National Semiconductor Microwire or Texas Instruments synchronous serial interfaces.

The SSP performs serial-to-parallel conversion on data received from a peripheral device. The CPU accesses data, control, and status information through the AMBA APB interface. The transmit and receive paths are buffered with internal FIFO memories allowing up to eight 8-bit values to be stored independently in both transmit and receive modes. Serial data is transmitted on SSPTXD and received on SSPRXD.

The SSP includes a programmable bit rate clock divider and prescaler to generate the serial output clock SSPCLKOUT from the input clock SSPCLK. Bit rates are supported to 2MHz and higher, subject to choice of frequency for SSPCLK and the maximum bit rate is determined by peripheral devices.

The SSP operating mode, frame format, and size are programmed through the control registers SSPCR0 and SSPCR1.

Four individually maskable interrupt outputs are generated:

- SSPTXINTR requests servicing of the transmit buffer
- SSPRXINTR requests servicing of the receive buffer
- SSPRORINTR indicates an overrun condition in the receive FIFO
- SSPRTINTR indicates that a timeout period expired while data was present in the receive FIFO

A single combined interrupt, SSPINTR output, is asserted if any of the individual interrupts are asserted and unmasked.

Depending on the operating mode selected, the SSPFSSOUT output operates as an active HIGH frame synchronization output for Texas Instruments synchronous serial frame format or an active LOW slave select for SPI and Microwire.

3.12.2 Features

The PrimeCell SSP has the following features:

- Compliance to the AMBA Specification (Rev 2.0)
- Master or slave operation
- Programmable clock bit rate and prescale
- Separate transmit and receive first-in, first-out memory buffers, 8 bits wide, 8 locations deep
- Programmable choice of interface operation, SPI
- Programmable data frame size from 4 to 8bits
- Independent masking of transmit FIFO, receive FIFO, and receive overrun interrupts

The features of the Motorola SPI-compatible interface are:

- Full duplex, four-wire synchronous transfers
- Programmable clock polarity and phase

3.12.3 Block Diagram

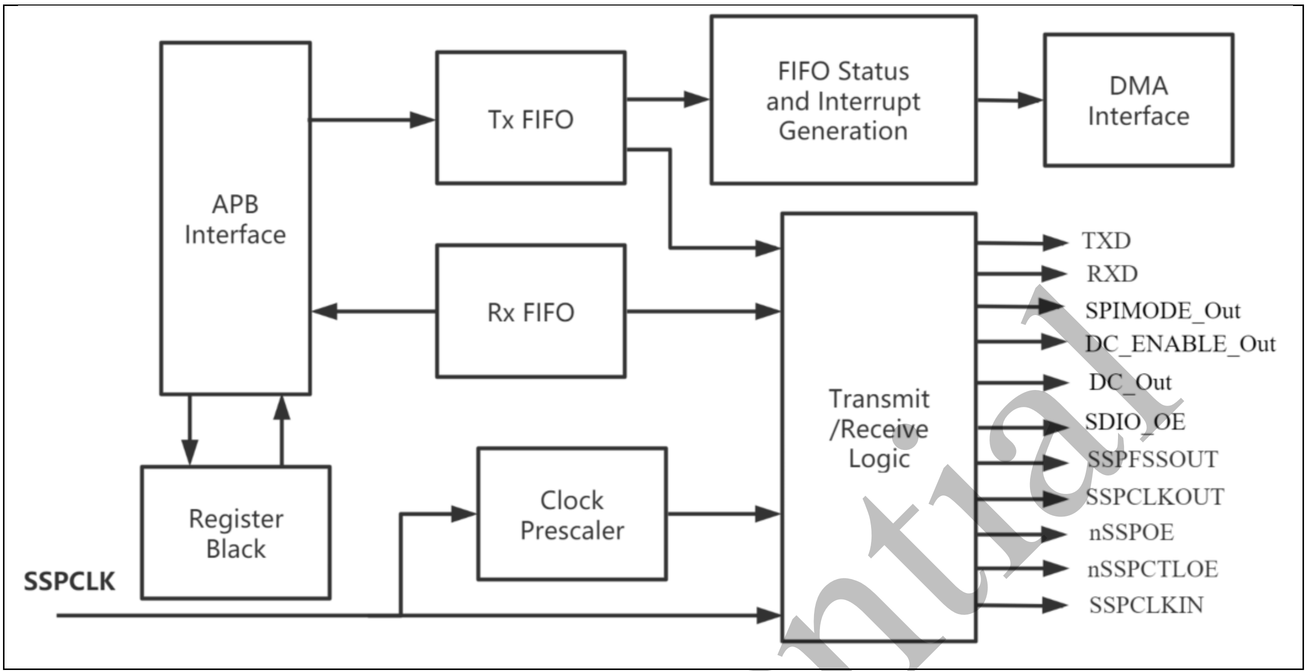


Figure 3-39 SPI Block Diagram

APB interface

It generates read and write decodes for accesses to status and control registers, and transmit and receive FIFO memories.

Register block

It stores data written, or to be read, across the APB interface.

Clock prescaler

It provides the serial output clock SSPCLKOUT with divide SSPCLK by a factor of 2-254.

TX/RX FIFO

TX/RX FIFO is a 8-bit wide, 8-locations deep, First-In, First-Out (FIFO) memory buffer.

Transmit and receive logic

When working as a master, the transmit logic reads the data from TX FIFO successively, it converts parallel input signals into serial output signals. Then the data stream output to the external slave by SSPTXD pin. Moreover, the master can receive the data from slave though SSPRXD pin and convert serial signals to parallel signals. The parallel signals will be extracted and stored into the RX FIFO.

When working as a slave, the SSPCLKIN clock is provided by an attached master and used to time its transmission and reception sequences.

FIFO state and interrupt generation logic

3.12.4 Functional Description

This chapter describes the functional behavior of SSP Synchronous Serial Port in detail.

The functions of the SSP are described in the following sections:

- AMBA APB interface
- Register block
- Clock prescaler
- Transmit FIFO
- Receive FIFO
- Transmit and receive logic
- Interrupt generation logic
- Synchronizing registers and logic

3.12.4.1 AMBA APB interface

The AMBA APB interface generates read and write decodes for accesses to status and control registers, and transmit and receive FIFO memories.

The AMBA APB is a local secondary bus that provides a low-power extension to the higher bandwidth AMBA Advanced High-performance Bus (AHB) within the AMBA system hierarchy. The AMBA APB groups narrow-bus peripherals to avoid loading the system bus and provides an interface using memory-mapped registers, which are accessed under programmed control.

3.12.4.2 Register block

The register block stores data written or to be read across the AMBA APB interface.

3.12.4.3 Clock prescaler

When configured as a master, an internal prescaler, comprising two free-running reloadable serially linked counters, is used to provide the serial output clock SSPCLKOUT.

You can program the clock prescaler, through the SSPCPSR register, to divide SSPCLK by a factor of 2 to 254 in steps of two. By not utilizing the least significant bit of the SSPCPSR register, division by an odd number is not possible and this ensures a symmetrical (equal mark space ratio) clock is generated.

The output of the prescaler is further divided by a factor of 1 to 256, through the programming of the SSPCR0 control register, to give the final master output clock SSPCLKOUT.

3.12.4.4 Transmit FIFO

The common transmit FIFO is a 16-bit wide, 8-locations deep, first-in, first-out memory buffer. CPU data written across the AMBA APB interface are stored in the buffer until read out by the transmit logic.

When configured as a master or a slave parallel data is written into the transmit FIFO prior to serial conversion and transmission to the attached slave or master respectively, through the SSPTXD pin.

SSP supports converting transmit FIFO data item to be transmitted in an MSB first order and compatible with dss(data size select) for both SSP master and slave.

3.12.4.5 Receive FIFO

The common receive FIFO is a 8-bit wide, 8-locations deep, first-in, first-out memory buffer. Received data from the serial interface are stored in the buffer until read out by the CPU across the AMBA APB interface.

When configured as a master or slave, serial data received through the SSPRXD pin is registered prior to parallel loading into the attached slave or master receive FIFO respectively.

SSP supports converting receive FIFO data item to be read in an MSB first order as and compatible with dss for both SSP master and slave.

3.12.4.6 Transmit and receive logic

When configured as a master :the clock to the attached slaves is derived from a divided down version of SSPCLK through the prescaler operations described previously. The master transmit logic successively reads a value from its transmit FIFO and performs parallel to serial conversion on it. In standard full-duplex 4-wire mode, the serial data stream and frame control signal, synchronized to SSPCLKOUT, are output through the SSPTXD pin to the attached slaves. The master receive logic performs serial to parallel conversion on the incoming synchronous SSPRXD data stream, extracting and storing values into its receive FIFO, for subsequent reading through the APB interface.

When configured as a slave, the SSPCLKIN clock is provided by an attached master and used to time its transmission and reception sequences. The slave transmit logic, under control of the master clock, successively reads a value from its transmit FIFO, performs parallel to serial conversion, then output the serial data stream and frame control signal through the slave SSPTXD pin. The slave receive logic performs serial to parallel conversion on the incoming

SSPRXD data stream, extracting and storing values into its receive FIFO, for subsequent reading through the APB interface.

3.12.4.7 Interrupt generation logic

Four individual maskable, active HIGH interrupts are generated by the SSP. A combined interrupt output is also generated as an OR function of the individual interrupt requests.

You can use the single combined interrupt with a system interrupt controller that provides another level of masking on a per-peripheral basis. This allows use of modular device drivers that always know where to find the interrupt source control register bits.

The individual interrupt requests could also be used with a system interrupt controller that provides masking for the outputs of each peripheral. In this way, a global interrupt controller service routine would be able to read the entire set of sources from one wide register in the system interrupt controller. This is attractive where the time to read from the peripheral registers is significant compared to the CPU clock speed in a real-time system.

The peripheral supports both the above methods.

The transmit and receive dynamic data-flow interrupts, SSPTXINTR and SSPRXINTR, are separated from the status interrupts so that data can be read or written in response to the FIFO trigger levels.

3.12.4.8 Synchronizing registers and logic

The SSP supports both asynchronous and synchronous operation of the clocks, PCLK and SSPCLK. Synchronization registers and hand shaking logic have been implemented, and are active at all times. This has a minimal impact on performance or area. Synchronization of control signals is performed on both directions of data flow, that is from the PCLK to the SSPCLK domain and from the SSPCLK to the PCLK domain.

3.12.5 SSP Operation

The operation of the SSP is described in the following sections:

- Interface reset
- Configuring the SSP
- Enable SSP operation
- Clock ratios
- Programming the SSPCR0 Control Register
- Programming the SSPCR1 Control Register

- Frame format
  - Motorola SPI frame format
- Examples of master and slave configurations

3.12.5.1 Interface reset

The SSP is reset by the global reset signal PRESETn and a block-specific reset signal nSSPRST. An external reset controller must use PRESETn to assert nSSPRST asynchronously and negate it synchronously to SSPCLK. PRESETn must be asserted LOW for a period long enough to reset the slowest block in the on-chip system, and then taken HIGH again. The SSP requires PRESETn to be asserted LOW for at least one period of PCLK.

3.12.5.2 Configuring the SSP

Following reset, the SSP logic is disabled and must be configured when in this state. Control registers SSPCR0 and SSPCR1 need to be programmed to configure the peripheral as a master or slave operating under one of the following protocols:

- Motorola SPI
- 4-wire SPI

The bit rate, derived from the external SSPCLK, requires the programming of the clock prescale register SSPCPSR.

3.12.5.3 Enable SSP operation

You can either prime the transmit FIFO, by writing up to eight 8-bit values when the PrimeCell SSP is disabled, or allow the transmit FIFO service request to interrupt the CPU. Once enabled, transmission or reception of data begins on the transmit (SSPTXD) and receive (SSPRXD) pins.

3.12.5.4 Clock ratios

There is a constraint on the ratio of the frequencies of PCLK to SSPCLK. The frequency of SSPCLK must be less than or equal to that of PCLK. This ensures that control signals from the SSPCLK domain to the PCLK domain are certain to get synchronized before one frame duration:

FSSPCLK <= FPCLK.

In the slave mode of operation, the SSPCLKIN signal from the external master is double synchronized and then delayed to detect an edge. It takes three SSPCLKs to detect an edge on SSPCLKIN. SSPTXD has less setup time to the falling edge of SSPCLKIN on which the master is sampling the line. The setup and hold times on SSPRXD with reference to SSPCLKIN must be more conservative to ensure that it is at the right value when the actual sampling occurs within the SSPMS. To ensure correct device operation, SSPCLK must be at least 8 times faster than the maximum expected frequency of SSPCLKIN.

The frequency selected for SSPCLK must accommodate the desired range of bit clock rates. The ratio of minimum SSPCLK frequency to SSPCLKOUT maximum frequency in the case of the slave mode is 8 and for the master mode it is 2.

To generate a maximum bit rate of 1.8432Mbps in the Master mode, the frequency of SSPCLK must be at least 3.6864MHz. With an SSPCLK frequency of 3.6864MHz, the SSPCPSR register has to be programmed with a value of 2 and the SCR[7:0] field in the SSPCR0 register needs to be programmed as zero.

To work with a maximum bit rate of 1.8432Mbps in the slave mode, the frequency of SSPCLK must be at least 14.75MHz. With an SSPCLK frequency of 14.75MHz, the SSPCPSR register can be programmed with a value of 2 and the SCR[7:0] field in the SSPCR0 register can be programmed as 3. Similarly the ratio of SSPCLK maximum frequency to SSPCLKOUT minimum frequency is 254 x 256.

The minimum frequency of SSPCLK is governed by the following equations, both of which have to be satisfied:

FSSPCLK(min) >= 2 x FSSPCLKOUT(max) [for master mode]

FSSPCLK(min) >= 8 x FSSPCLKIN(max) [for slave mode]

The maximum frequency of SSPCLK is governed by the following equations, both of which have to be satisfied:

FSSPCLK(max) <= 254 x 256 x FSSPCLKOUT(min) [for master mode]

FSSPCLK(max) <= 254 x 256 x FSSPCLKIN(min) [for slave mode]

3.12.5.5 Programming the SSPCR0 Control Register

The SSPCR0 register is used to:

- Program the serial clock rate
- Select the data word size (where applicable)

The Serial Clock Rate (SCR) value, in conjunction with the SSPCPSR clock prescale divisor value (CPSDVSR), is used to derive the SSP transmit and receive bit rate from the external

SSPCLK.

The frame format is programmed through the FRF bits and the data word size through the DSS bits.

Bit phase and polarity, applicable to Motorola SPI format only, are programmed through the SPH and SPO bits.

3.12.5.6 Programming the SSPCR1 Control Register

The SSPCR1 register is used to:

- Select master or slave mode
- Enable a loop back test feature
- Enable the SSP peripheral
- MSB/ LSB of Tx FIFO data item transmit first configuration
- MSB/ LSB of Rx FIFO data item receive first configuration

To configure the SSP as a master, clear the SSPCR1 register master or slave selection bit (MS) to 0, which is the default value on reset.

Setting the SSPCR1 register MS bit to 1 configures the SSP as a slave. When configured as a slave, enabling or disabling of the SSP SSPTXD signal is provided through the SSPCR1 slave mode SSPTXD output disable bit (SOD). This can be used in some multislave environments where masters might parallel broadcast.

To enable the operation of the PrimeCell SSP set the Synchronous Serial Port Enable(SSE)bit to 1.

3.12.5.7 Bit rate generation

The serial bit rate is derived by dividing down the input clock SSPCLK. The clock is first divided by an even prescale value CPSDVSR from 2 to 254, which is programmed in SSPCPSR. The clock is further divided by a value from 1 to 256, which is 1 + SCR, where SCR is the value programmed in SSPCR0.

The frequency of the output signal bit clock SSPCLKOUT is defined below:

$$F_{SSPCLKOUT}=F_{SSPCLK}/(CPSDVR \times (1+SCR))$$

For example, if SSPCLK is 3.6864MHz, and CPSDVSR = 2, then SSPCLKOUT has a frequency range from 7.2kHz to 1.8432MHz

3.12.5.8 Frame format

Each data frame is between 4 and 8 bits long depending on the size of data programmed, and is transmitted starting with the MSB.

SSP only support Motorola SPI format.

The serial clock (SSPCLKOUT) is held inactive while the SSP is idle, and transitions at the programmed frequency only during active transmission or reception of data. The idle state of SSPCLKOUT is utilized to provide a receive timeout indication that occurs when the receive FIFO still contains data after a timeout period.

For Motorola SPI , the serial frame (SSPFSSOUT) pin is active LOW, and is asserted (pulled down) during the entire transmission of the frame.

3.12.5.9 Motorola SPI frame format

The Motorola SPI interface is a four-wire interface where the SSPFSSOUT signal behaves as a slave select. The main feature of the Motorola SPI format is that the inactive state and phase of the SSPCLKOUT signal are programmable through the SPO and SPH bits within the SSPSCR0 control register.

SPO, clock polarity

When the SPO clock polarity control bit is LOW, it produces a steady state low value on the SSPCLKOUT pin. If the SPO clock polarity control bit is HIGH, a steady state high value is placed on the SSPCLKOUT pin when data is not being transferred.

SPH, clock phase

The SPH control bit selects the clock edge that captures data and allows it to change state. It has the most impact on the first bit transmitted by either allowing or not allowing a clock transition before the first data capture edge.

When the SPH phase control bit is LOW, data is captured on the first clock edge transition. If the SPH clock phase control bit is HIGH, data is captured on the second clock edge transition. For continuous back-to-back SPI transactions, if SPH=0, SSPSSFOUT turns high for one cycle of SSPCLKOUT for each SPI transaction; if SPH=1, SSPSSFOUT keeps low during continuous transmission. If master transmit fifo get empty, then SSPSSFOUT turns high before next master transmit fifo tansaction com.

If SPO^SPH=0, master and slave both sample SSPRXD on posedge of SSPCLKOUT/SSPCLKIN, and drive SSPTXD on negedge of SSPCLKOUT/SSPCLKIN.

If SPO^SPH=1, master and slave both sample SSPRXD on negedge of SSPCLKOUT/SSPCLKIN, and drive SSPTXD on posedge of SSPCLKOUT/SSPCLKIN.

3.12.5.10 Motorola SPI Format with SPO=0, SPH=0

Continuous transmission signal sequences for Motorola SPI format with SPO=0, SPH=0, tx_lsb=0 and rx_lsb=0 are shown in Figure 3-40.

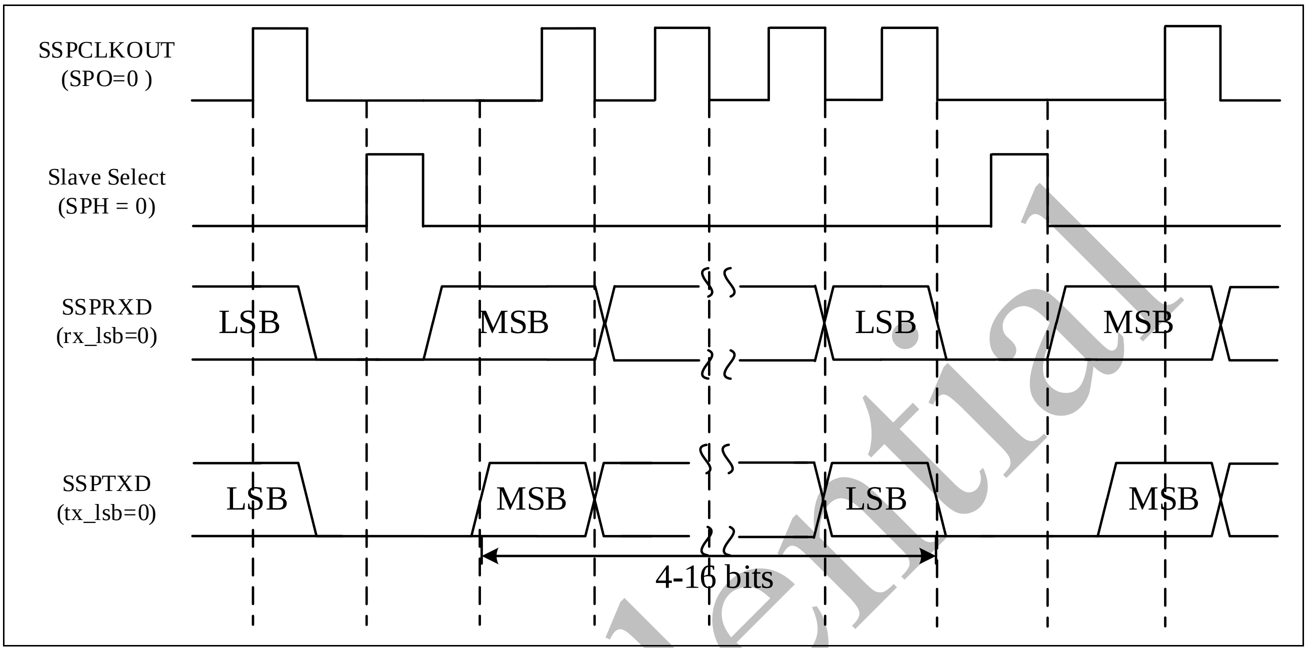


Figure 3-40 Motorola SPI frame format (continuous transfer) with SPO=0 and SPH=0, tx_lsb=0, rx_lsb=0

In this configuration, during idle periods:

- The SSPCLKOUT signal is forced LOW
- SSPFSSOUT is forced HIGH
- The transmit data line SSPTXD is arbitrarily forced LOW
- The nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- When the SSP is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad (active LOW enable)
- When the SSP is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad (active LOW enable)

If the SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. This causes slave data to be enabled onto the SSPRXD input line of the master. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad.

One half SSPCLKOUT period later, valid master data is transferred to the SSPTXD pin. Now that both the master and slave data have been set, the SSPCLKOUT master clock pin goes HIGH after one further half SSPCLKOUT period.

The data is now captured on the rising and propagated on the falling edges of the SSPCLKOUT signal.

In the case of a single word transmission, after all bits of the data word have been transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSPFSSOUT signal must be pulsed HIGH between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not allow it to be altered if the SPH bit is logic zero. Therefore the master device must raise the SSPFSSIN pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SSPFSSOUT pin is returned to its idle state one SSPCLKOUT period after the last bit has been captured.

3.12.5.11 Motorola SPI Format with SPO=0, SPH=1

The transfer signal sequence for Motorola SPI format with SPO=0, SPH=1, tx_lsb=0 and rx_lsb=1 is shown in Figure 3-41.

In this configuration, during idle periods:

- The SSPCLKOUT signal is forced LOW
- SSPFSSOUT is forced HIGH
- The transmit data line SSPTXD is arbitrarily forced LOW
- The nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- When the SSP is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad (active LOW enable)
- When the SSP is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad (active LOW enable)

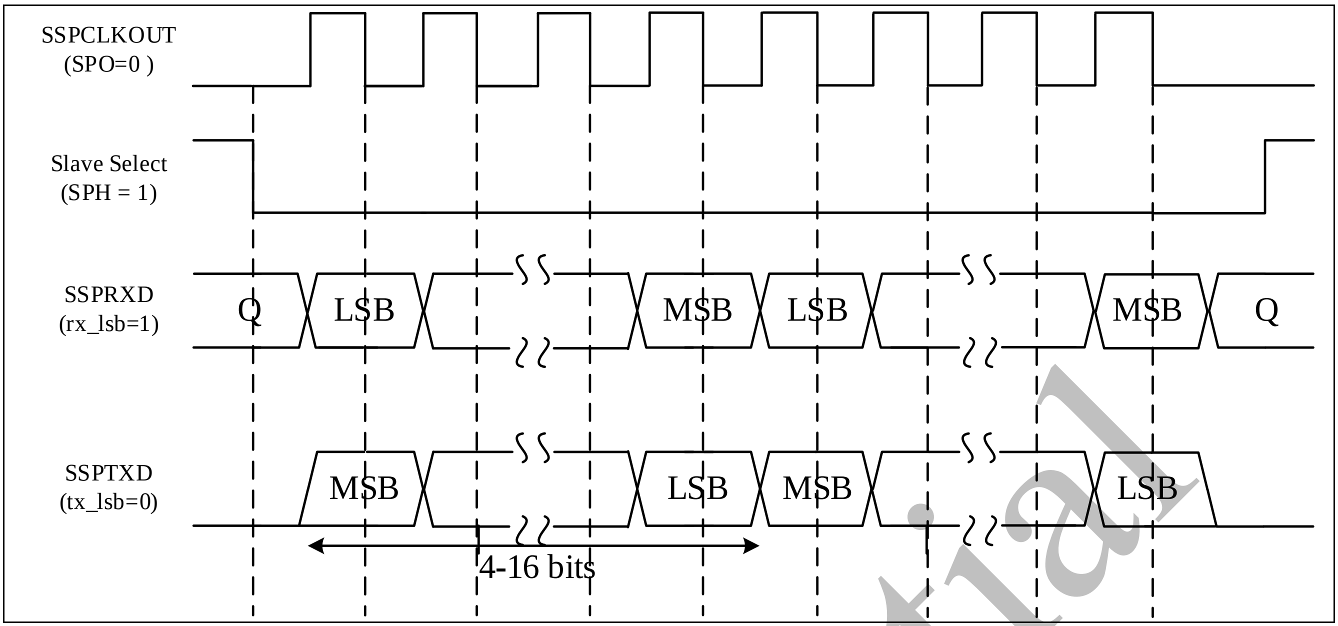


Figure 3-41 Motorola SPI frame format with SPO=0 and SPH=1, tx_lsb=0, rx_lsb=1

If the SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad. After a further one half SSPCLKOUT period, both master and slave valid data is enabled onto their respective transmission lines. At the same time, the SSPCLKOUT is enabled with a rising edge transition.

Data is then captured on the falling edges and propagated on the rising edges of the SSPCLKOUT signal.

In the case of a single word transfer, after all bits have been transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured. For continuous back-to-back transfers, the SSPFSSOUT pin is held LOW between successive data words and termination is the same as that of the single word transfer.

3.12.5.12 Motorola SPI Format with SPO=1, SPH=0

The transmission signal sequences for Motorola SPI format with SPO=1, SPH=0, tx_lsb=1 and rx_lsb=1 are shown in Figure 3-42.

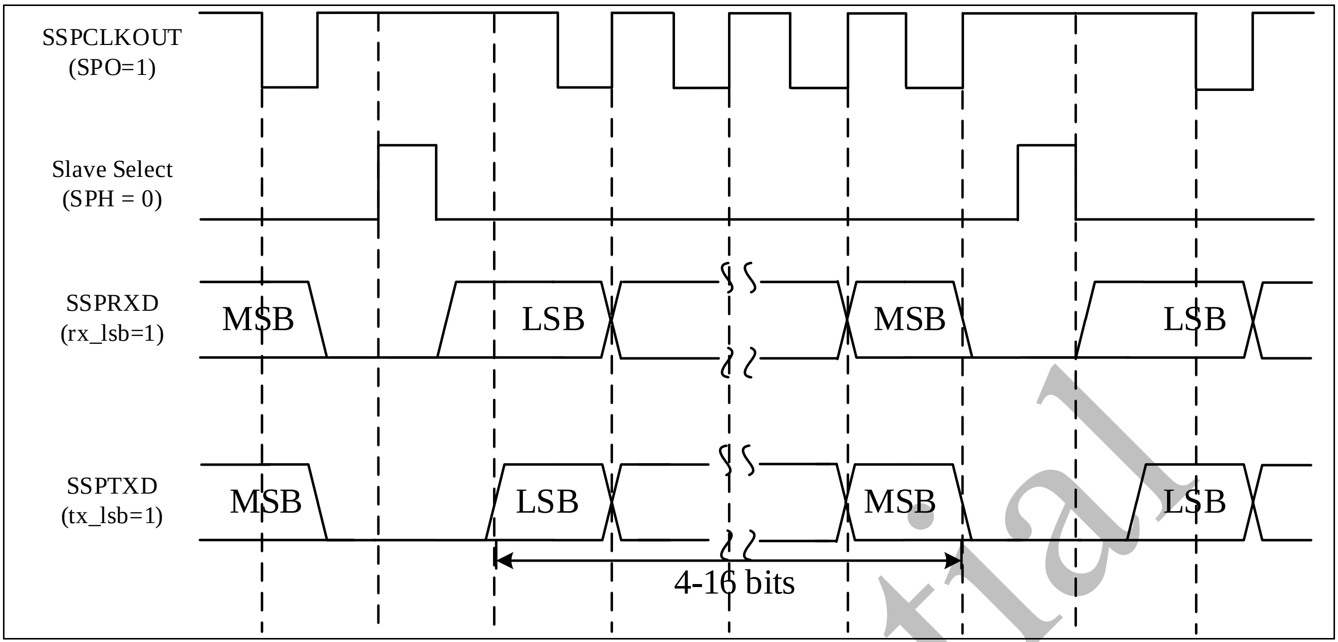


Figure 3-42 Motorola SPI frame format with SPO=1 and SPH=0, tx_lsb=1, rx_lsb=1

In this configuration, during idle periods

- The SSPCLKOUT signal is forced HIGH
- SSPFSSOUT is forced HIGH
- The transmit data line SSPTXD is arbitrarily forced LOW
- The nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- When the SSP is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad (active LOW enable)
- When the SSP is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad (active LOW enable)

If the SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW, which causes slave data to be immediately transferred onto the SSPRXD line of the master. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad.

One half period later, valid master data is transferred to the SSPTXD line. Now that both the master and slave data have been set, the SSPCLKOUT master clock pin becomes LOW after one further half SSPCLKOUT period. This means that data is captured on the falling edges and be propagated on the rising edges of the SSPCLKOUT signal.

In the case of a single word transmission, after all bits of the data word are transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSPFSSOUT signal must be pulsed HIGH between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not allow it to be altered if the SPH bit is logic zero. Therefore the master device must raise the SSPFSSIN pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SSPFSSOUT pin is returned to its idle state one SSPCLKOUT period after the last bit has been captured.

3.12.5.13 Motorola SPI Format with SPO=1, SPH=1

The transfer signal sequence for Motorola SPI format with SPO=1, SPH=1, tx_lsb=1 and rx_lsb=0 is shown in Figure 3-43.

In this configuration, during idle periods:

- The SSPCLKOUT signal is forced HIGH
- SSPFSSOUT is forced HIGH
- The transmit data line SSPTXD is arbitrarily forced LOW
- The nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- When the SSP is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad (active LOW enable)
- When the SSP is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad (active LOW enable)

If the SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad. After a further one half SSPCLKOUT period, both master and slave data are enabled onto their respective transmission lines. At the same time, the SSPCLKOUT is enabled with a falling edge transition. Data is then captured on the rising edges and propagated on the falling edges of the SSPCLKOUT signal.

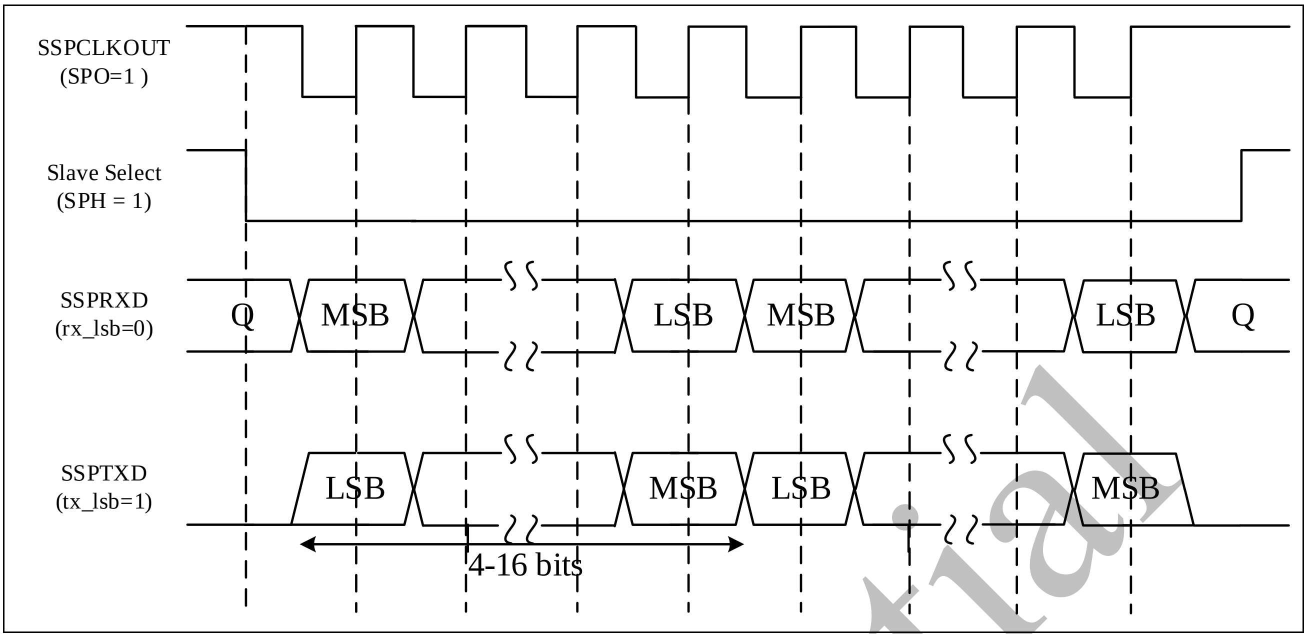


Figure 3-43 Motorola SPI frame format with SPO=1 and SPH=1, tx_lsb=1, rx_lsb=0

After all bits have been transferred, in the case of a single word transmission, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

For continuous back-to-back transmissions, the SSPFSSOUT pins remains in its active LOW state, until the final bit of the last word has been captured, and then returns to its idle state as described above.

For continuous back-to-back transfers, the SSPFSSOUT pin is held LOW between successive data words and termination is the same as that of the single word transfer.

3.12.5.14 Examples of master and slave configurations

Figure 3-44, Figure 3-45 and Figure 3-46 show how the SSP peripheral can be connected to other synchronous serial peripherals, when it is configured as a master or slave.

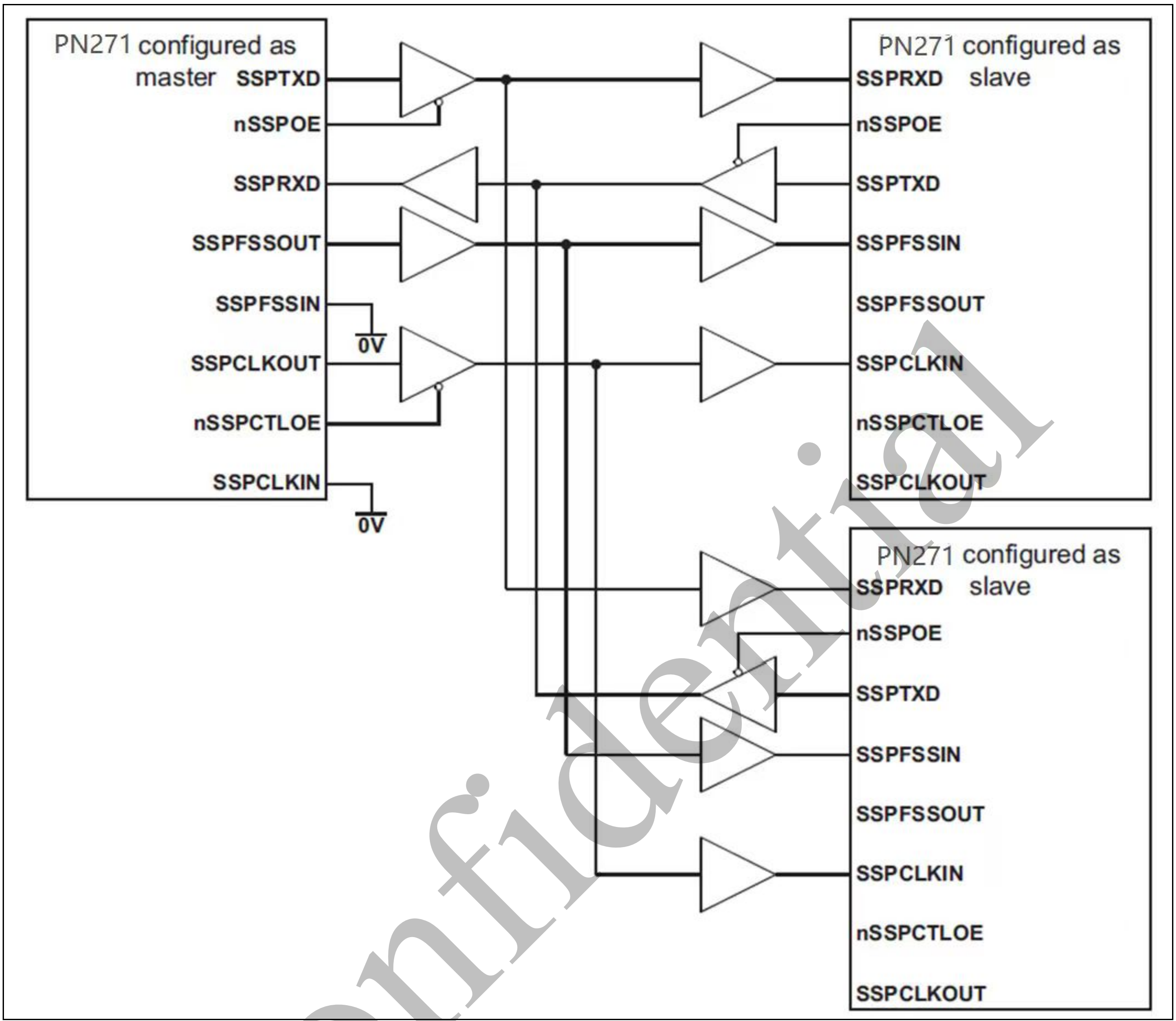


Figure 3-44 SSP master coupled to two slaves

Figure 3-44 shows the SSP instanced three times, as a single master and two slaves. The master can broadcast to the two slaves through the master SSPTXD line. In response, only one slave drives its nSSPOE signal HIGH, thereby enabling its SSPTXD data onto the SSPRXD line of the master.

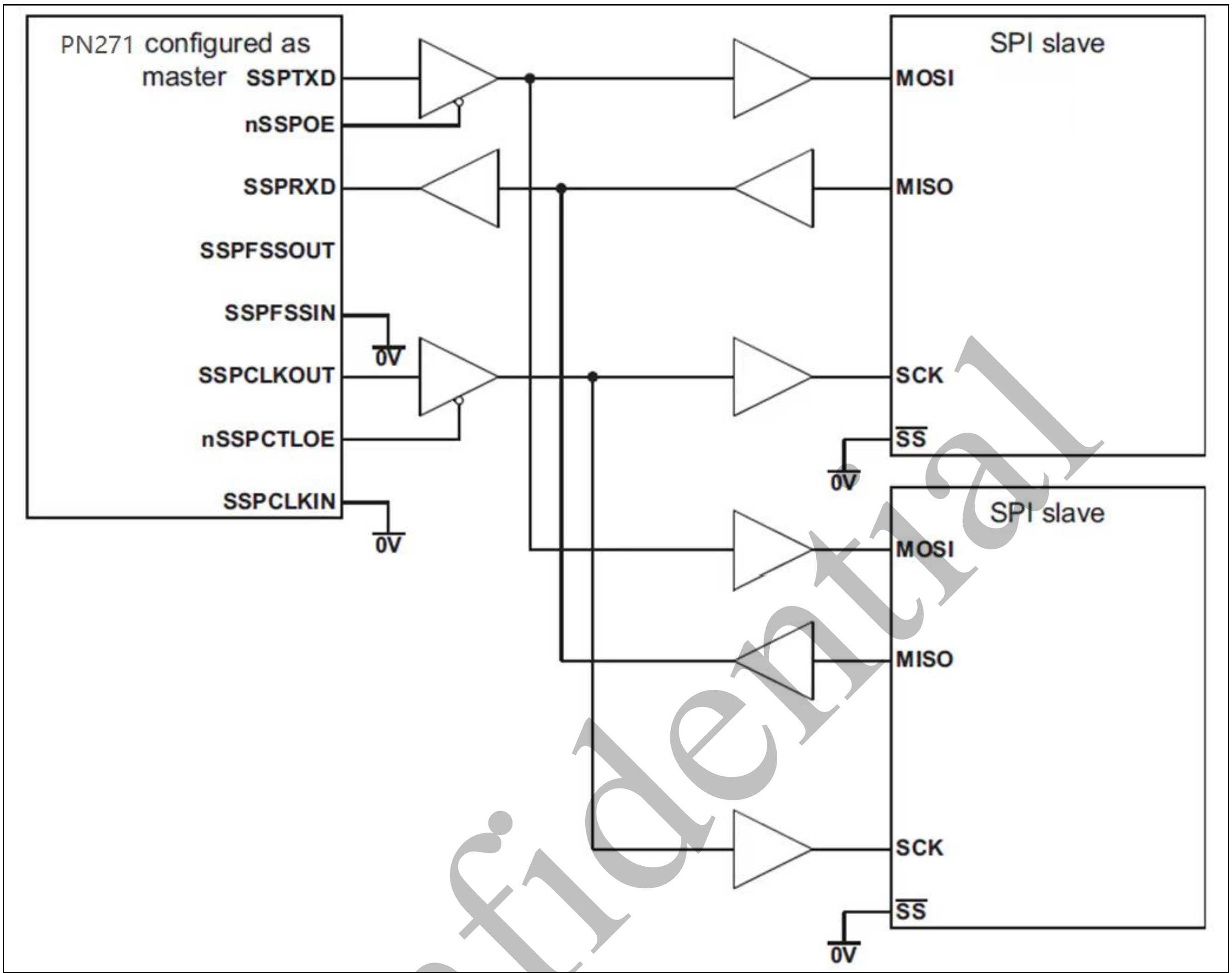


Figure 3-45 SSP master coupled to two other slaves

Figure 3-45 shows how an SSP, configured as master, interfaces to two Motorola SPI slaves. Each SPI Slave Select (SS) signal is permanently tied LOW and configures them as slaves. Similar to the above operation, the master can broadcast to the two slaves through the master SSP SSPTXD line. In response, only one slave drives its SPI MISO port onto the SSPRXD line of the master.

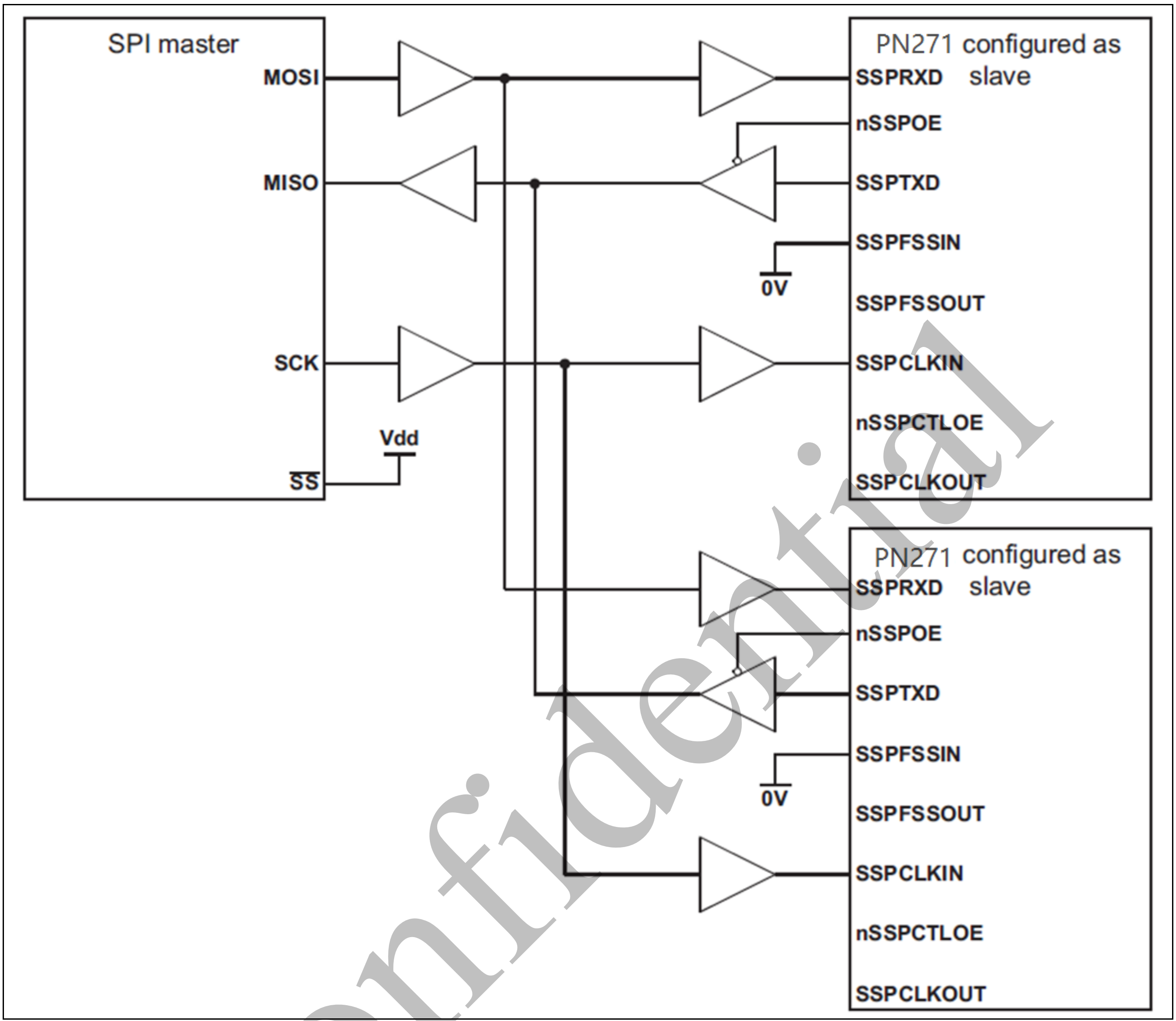


Figure 3-46 SPI master coupled to two SSP slaves

Figure 3-46 shows a Motorola SPI configured as a master and interfaced to two instances of SSP configured as slaves. In this case the slave Select Signal (SS) is permanently tied HIGH and configures it as a master. The master can broadcast to the two slaves through the master SPI MOSI line and in response, only one slave drives its nSSPOE signal LOW. This enables its SSPTXD data onto the MISO line of the master.

3.12.5.15 Interrupts

There are five interrupts generated by SSP. The specific interrupt signals are described in Table 3-8. Four of these are individual, maskable, active-HIGH interrupts as follows:

- SSPRXINTR
- SSPTXINTR

- SSPRORINTR
- SSPRTINTR
- SSPINTR

You can mask each of the four individual maskable interrupts by setting the appropriate bits in the SSPIMSC register. Setting the appropriate mask bit HIGH enables the interrupt.

Provision of the individual outputs in addition to a combined interrupt output, enables the use of either a global interrupt service routine, or modular device drivers to handle interrupts.

The transmit and receive dynamic dataflow interrupts SSPTXINTR and SSPRXINTR have been separated from the status interrupts, so that data can be read or written in response to only the FIFO trigger levels.

The status of the individual interrupt sources can be read from SSPRIS and SSPMIS registers.

Table 3-12 Interrupt Signals

Port Name	Description	Asserted condition	Notes
SSPRXINTR	SPI receive FIFO service interrupt request	≥ 4 entries in RX FIFO	-
SSPTXINTR	SPI transmit FIFO service interrupt request	≤ 4 entries in TX FIFO	This enables operation in either of the following ways: Data can be written to the transmit FIFO prior to enabling the PrimeCell SSP and the interrupts The PrimeCell SSP and interrupts can be enabled so that data can be written to the transmit FIFO by an interrupt service routine.
SSPRORINTR	SPI receive overrun interrupt request	RX FIFO is already full and an additional data frame is received	Data is overwritten in the receive shift register, but not the FIFO.
SSPRTINTR	SPI time out interrupt request	RX FIFO is not empty and the SPI has remained idle for a fixed 32 bit period	This interrupt is deasserted if the receive FIFO becomes empty by subsequent reads, or if new data is received on SSPRXD. It can also be cleared by writing to the RTIC bit in the SSPICR register.
SSPINTR	Combined single interrupt	Any of the four individual interrupts above are asserted and enabled	An OR function of the individual masked sources

3.12.6 SSP Register Map

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

Register	Offset	R/W	Description	Reset Value
SSP Base Address: SSP_BA =0x4000_1000				
SSPCR0	SSP_BA+0x00	R/W	Control Register 0	0x0000_0000
SSPCR1	SSP_BA+0x04	R/W	Control Register 1	0x0000_0080
SSPDR	SSP_BA+0x08	R/W	Receive FIFO(read) and Transmit FIFO Data Register	Configurable
SSPSR	SSP_BA+0x0C	R	Status Register	0x0000_0003
SSPCPSR	SSP_BA+0x10	R/W	Clock Prescale Register	0x0000_0000
SSPIMSC	SSP_BA+0x14	R/W	Interrupt Mask Set and Clear Register	0x0000_0000
SSPRIS	SSP_BA+0x18	R	Raw Interrupt Status Register	0x0000_0008
SSPMIS	SSP_BA+0x1C	R	Masked Interrupt Status Register	0x0000_0000
SSPICR	SSP_BA+0x20	W	Interrupt Clear Register	0x0000_0000
SSPDMACR	SSP_BA+0x24	R/W	DMA Control Register	0x0000_0000

3.12.7 SSP Register Description

3.12.7.1 Control Register 0 (SSPCR0)

Register	Offset	R/W	Description	Reset Value
SSPCR0	SSP_BA+0x00	R/W	Control Register 0	0x0000_0007

Bits	Description	
[31:16]	Reserved	Reserved
[15:8]	SCR	Serial Clock Rate. The value SCR is used to generate the transmit and receive bit rate of the SSP. The bit rate is: $\frac{f_{ssp\_apb\_clk}}{CPSDVSR \times (1 + SCR)}$ Where CPSDVSR is an even value from 2-254, programmed through the SSPCPSR register and SCR is a value from 0-255.
[7]	SPH	Serial Clock Phase. For Motorola SPI frame format only 0: Data are captured on the first edge of the serial clock. Slave Select pulse high between each spi bus data item. 1: Data are captured on the second edge of the serial clock. Slave Select keep low for continuous transmission.
[6]	SPO	Serial Clock Polarity. For Motorola SPI frame format only 0: Inactive state of serial clock is low 1: Inactive state of serial clock is high
[5:4]	Reserved	Reserved
[3:0]	DSS	Data Size Select: 0000 ~ 0010: Reserved. 0011: 4-bit data. 0100: 5-bit data. 0101: 6-bit data. 0110: 7-bit data. 0111: 8-bit data. 1000: 9-bit data (not support) 1001: 10-bit data (not support) 1010: 11-bit data. (not support) 1011: 12-bit data. (not support) 1100: 13-bit data. (not support) 1101: 14-bit data. (not support) 1110: 15-bit data. (not support) 1111: 16-bit data. (not support)

3.12.7.2 Control Register 1 (SSPCR1)

Register	Offset	R/W	Description	Reset Value
SSPCR1	SSP_BA+0x04	R/W	Control Register 1	0x0000_0000

Bits	Description	
[31:4]	Reserved	Reserved
[3]	SOD	Slave-mode Output Disable Bit. This bit is relevant only in the slave mode, MS=1. In multiple-slave systems, it is possible for a SPI master to broadcast a message to all slaves in the system while ensuring that only one slave drives data onto its serial output line. In such systems the RXD lines from multiple slaves could be tied together.To operate in such systems, the SOD bit can be set if the SPI slave is not supposed to drive the SSPTXD line: 0: SPI can drive the SSPTXD output in slave mode. 1: SPI must not drive the SSPTXD output in slave mode.
[2]	MS	Master or Slave Mode Select. This bit can be modified only when the SPI is disabled, SSE=0: 0: Device configured as master, default. 1: Device configured as slave.
[1]	SSE	Synchronous Serial Port Enable: 0: SPI operation disabled. 1: SPI operation enabled
[0]	LBM	Loop Back Mode: 0: Normal serial port operation enabled. 1: Output of transmit serial shifter is connected to input of receive serial shifter internally.

3.12.7.3 Data Register (SSPDR)

Register	Offset	R/W	Description	Reset Value
SSPDR	SSP_BA+0x08	R/W	Receive FIFO(read) and Transmit FIFO Data Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved
[7:0]	DATA	Transmit/Receive FIFO: Read: Receive FIFO. Write: Transmit FIFO. You must right-justify data when the SSP is programmed for a data size that is less than 16 bits. Unused bits at the top are ignored by transmit logic. The receive logic automatically right-justifies.

3.12.7.4 Status Register (SSPSR)

Register	Offset	R/W	Description	Reset Value
SSPSR	SSP_BA+0x0C	R	Status Register	0x0000_0003

Bits	Description	
[31:5]	Reserved	Reserved
[4]	BSY	SPI Busy Flag: 0: SPI is idle. 1: SPI is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.
[3]	RFF	Receive FIFO Full: 0: Receive FIFO is not full. 1: Receive FIFO is full.
[2]	RNE	Receive FIFO Not Empty: 0: Receive FIFO is empty. 1: Receive FIFO is not empty
[1]	TNF	Transmit FIFO Not Full
[0]	TNE	Transmit FIFO Empty 0: Transmit FIFO is not empty. 1: Transmit FIFO is empty

3.12.7.5 Clock Prescale Register (SSPCPSR)

Register	Offset	R/W	Description	Reset Value
SSPCPSR	SSP_BA+0x10	R/W	Clock Prescale Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved
[7:0]	CPSDVSR	Clock Prescale Divisor. Must be an even number from 2-254, depending on the frequency of SSPCLK. The least significant bit always returns zero on reads.

3.12.7.6 Interrupt Mask Set and Clear Register (SSPIMSC)

In this register, on a read it gives the current value of the mask on the relevant interrupt. A write of 1 to the particular bit sets the mask, enabling the interrupt to be read. A write of 0 clears the corresponding mask.

Register	Offset	R/W	Description	Reset Value
SSPIMSC	SSP_BA+0x14	R/W	Interrupt Mask Set and Clear Register	0x0000_0000

Bits	Description	
[31:4]	Reserved	Reserved
[3]	TXIM	Transmit FIFO interrupt mask: TX FIFO Under-Half-Full Interrupt Enable 0: Transmit FIFO half empty or less condition interrupt is masked. (Interrupt not enabled) 1: Transmit FIFO half empty or less condition interrupt is not masked.
[2]	RXIM	Receive FIFO interrupt mask: RX FIFO Half-Full or Above Interrupt Enable 0: Receive FIFO half full or less condition interrupt is masked. (Interrupt not enabled) 1: Receive FIFO half full or less condition interrupt is not masked.
[1]	RTIM	Receive timeout interrupt mask: RX FIFO Timeout Interrupt Enable 0: Receive FIFO not empty and no read prior to timeout period interrupt is masked. (Interrupt not enabled) 1: Receive FIFO not empty and no read prior to timeout period interrupt is not masked.
[0]	RORIM	Receive overrun interrupt mask: RX FIFO Full Interrupt Enable 0: Receive FIFO written to while full condition interrupt is masked. (Interrupt not enabled) 1: Receive FIFO written to while full condition interrupt is not masked.

3.12.7.7 Raw Interrupt Status Register (SSPRIS)

Register	Offset	R/W	Description	Reset Value
SSPRIS	SSP_BA+0x18	R/W	Raw Interrupt Status Register	0x0000_0008

Bits	Description	
[31:4]	Reserved	Reserved
[3]	TXRIS	Gives the raw interrupt state, prior to masking, of the *SSPTXINTR* interrupt TX FIFO Under-Half-Full Flag
[2]	RXRIS	Gives the raw interrupt state, prior to masking, of the *SSPRXINTR* interrupt RX FIFO Half-Full or Above Flag
[1]	RTRIS	Gives the raw interrupt state, prior to masking, of the *SSPRTINTR* interrupt RX FIFO Timeout Flag
[0]	RORRIS	Gives the raw interrupt state, prior to masking, of the *SSPRORINTR* interrupt RX FIFO Full Flag

3.12.7.8 Masked Interrupt Status Register (SSPMIS)

Register	Offset	R/W	Description	Reset Value
SSPMIS	SSP_BA+0x1C	R	Masked Interrupt Status Register	0x0000_0000

Bits	Description	
[31:4]	Reserved	Reserved
[3]	TXMIS	Gives the transmit FIFO masked interrupt state, after masking, of the *SSPTXINTR* interrupt TX FIFO Under-Half-Full Interrupt Flag
[2]	RXMIS	Gives the receive FIFO masked interrupt state, after masking, of the *SSPRXINTR* interrupt RX FIFO Half-Full or Above Interrupt Flag
[1]	RTMIS	Gives the receive timeout masked interrupt state, after masking, of the *SSPRTINTR* interrupt RX FIFO Timeout Interrupt Flag
[0]	RORMIS	Gives the receive over run masked interrupt status, after masking, of the *SSPRORINTR* interrupt RX FIFO Full Interrupt Flag

3.12.7.9 Interrupt Clear Register (SSPICR)

Register	Offset	R/W	Description	Reset Value
SSPICR	SSP_BA+0x20	R/W	Interrupt Clear Register	0x0000_0000

Bits	Description	
[31:2]	Reserved	Reserved
[1]	RTIC	Clears the *SSPRTINTR* interrupt Write 1 to clear the RX FIFO timeout flag and timeout interrupt flag.
[0]	RORIC	Clears the *SSPRORINTR* interrupt Write 1 to clear the RX FIFO full flag and full interrupt flag.

3.13 UART Controller (UART)

3.13.1 Overview

UART (Universal Asynchronous Receiver/Transmitter) implements serial communication functionality within SOCs. Specifically, UART converts parallel data into serial data for transmission over serial interfaces, while also converting received serial data back into parallel data. UART is commonly used to connect processors with external devices such as sensors, displays, debug interfaces, and more.

3.13.2 Features

- Baud rate: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 921600
- 8-bit data width
- Full-duplex operation mode
- Stop bit: 1 bit
- No parity checking, no flow control, no DMA transfer functionality.
- FIFO depth: 4
- Clock: RCH/XTH (32 MHz)
- Interrupts
- RX timeout mechanism

3.13.3 Block Diagram

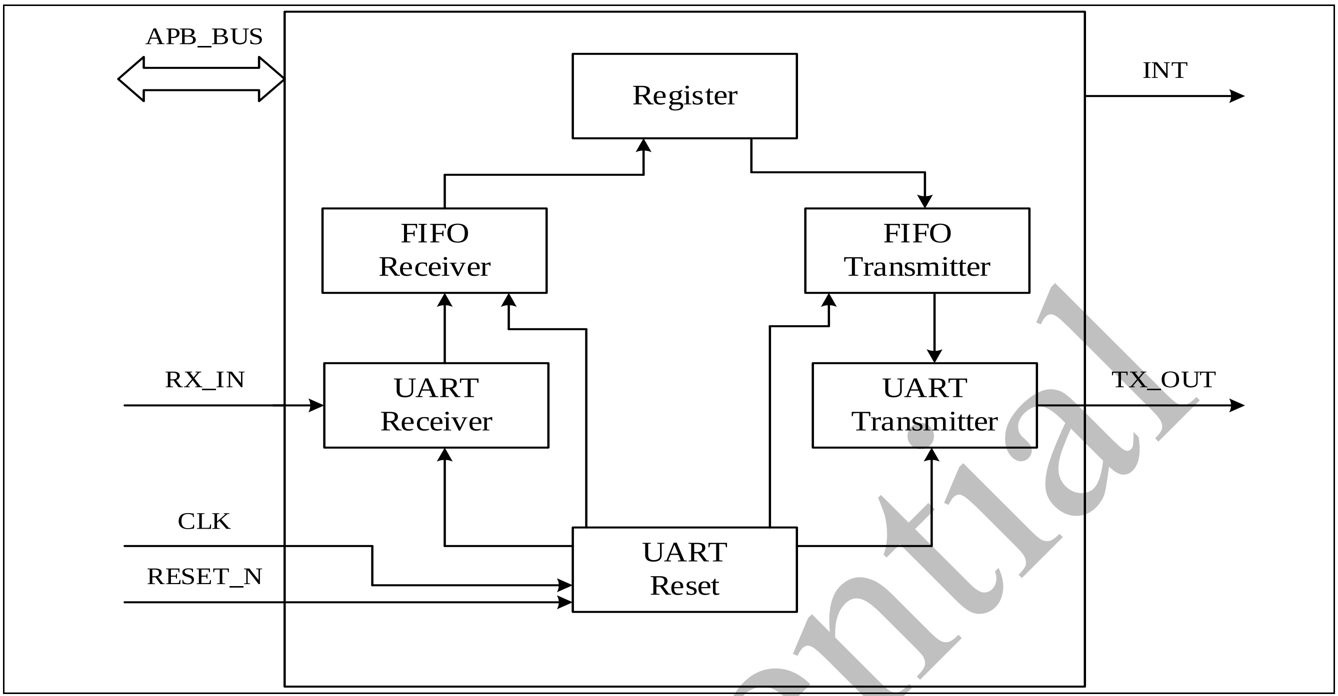


Figure 3-47 UART Controller Block Diagram

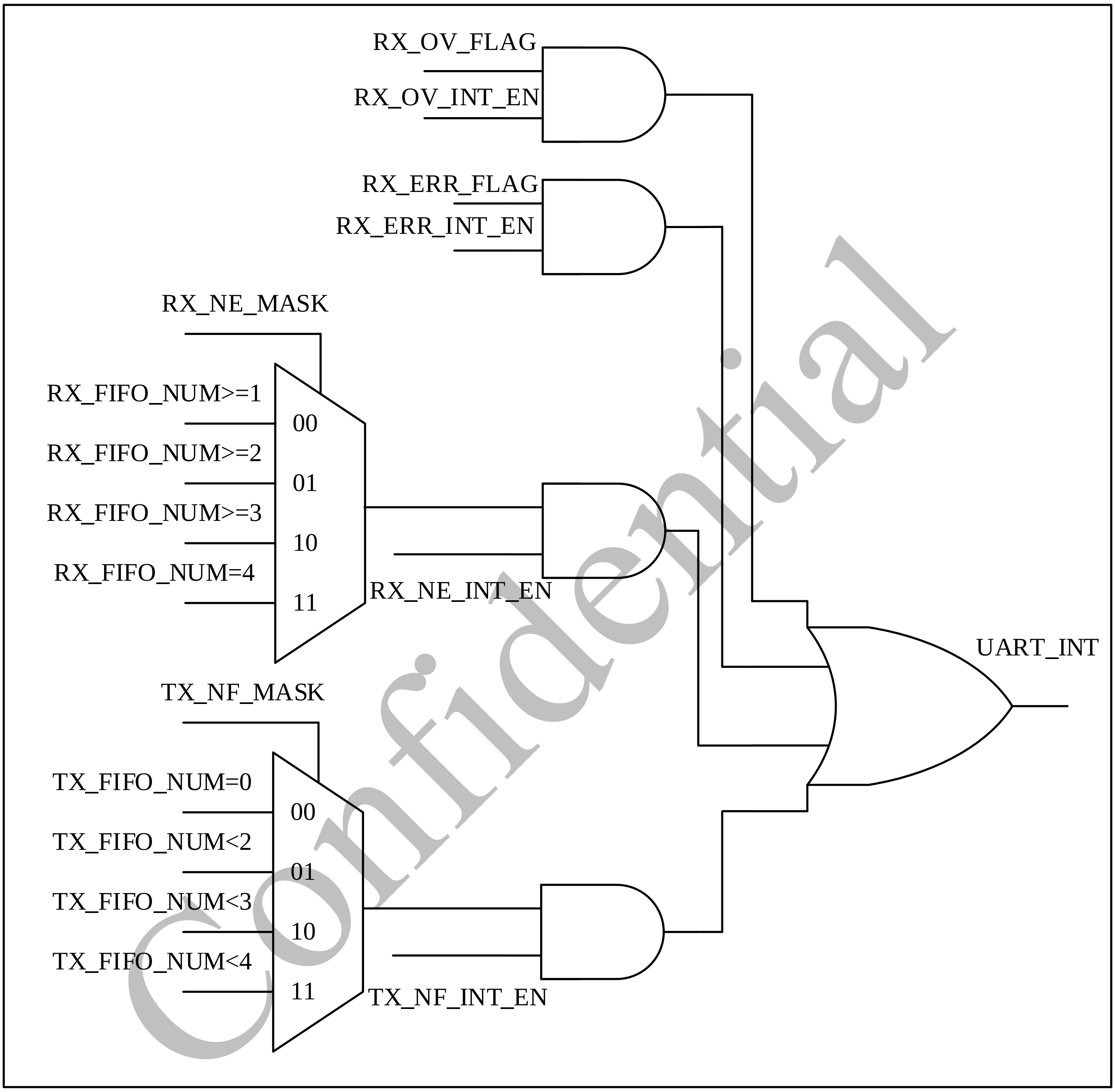
3.13.4 Functional Description

3.13.4.1 UART Operation Flow

The UART configuration process is as follows:

1. Confirm the baud rate BAUD and configure BCNT.
2. Enable interrupts.
3. Enable TX or RX (both can be enabled simultaneously for full-duplex operation). Once enabled, do not change the baud rate.
4. Prepare data for transmission or reception by writing to TX_DATA or reading from RX_DATA.
5. If UART clock deviation requires adjusting BCNT, disable both TX and RX enable signals beforehand. TX and RX share the same BCNT (i.e., identical baud rate).
6. When UART input clock deviation occurs, fine-tune BCNT to maintain stable baud rate.
7. After RX Timeout is enabled, counting begins when RXFIFO is non-empty. An RX Timeout interrupt is triggered when the count reaches TIMEOUT_CNT_TOTAL. While The count is reset to zero if new RX data is received or the MCU reads RXFIFO before reaching TIMEOUT_CNT_TOTAL.

3.13.4.2 UART Interrupts and FLAG



In the diagram above, only RX_ERR_FLAG and interrupts require software operations to read and clear. All others are cleared automatically by the hardware internally. The diagram shows RX_FIFO_NUM >= 1, where 1 represents one-fourth of the FIFO depth (the FIFO depth in this design is 4).

3.13.5 UART Register Map

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

Register	Offset	R/W	Description	Reset Value
UART Base Address: UART0_BA = 0x4000_3000 UART1_BA = 0x4000_A000				
UART_RXDATA	UARTx_BA+0x00	R	UART RXDATA Register	0x0000_0000
UART_TXDATA	UARTx_BA+0x04	R/W	UART TXDATA Register	0x0000_0000
UART_CTL	UARTx_BA+0x08	R/W	UART Control Register	0x0000_1004
UART_STATUS	UARTx_BA+0x0C	R	UART Status Register	0x0000_0000
UART_BCNT	UARTx_BA+0x10	R/W	UART BAUD Configuration Register	0x0000_0D06
TIMEOUT_CTL	UARTx_BA+0x14	R/W	Timeout Control Register	0x0000_0000

3.13.6 UART Register Description

3.13.6.1 UART RXDATA Register (UART_RXDATA)

Register	Offset	R/W	Description	Reset Value
UART_RXDATA	UARTx_BA+0x00	R	UART RXDATA Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	RX_DATA	8-bit data received via UART. Default: 8’h0

3.13.6.2 UART TXDATA Register (UART_TXDATA)

Register	Offset	R/W	Description	Reset Value
UART_TXDATA	UARTx_BA+0x04	R/W	UART TXDATA Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	TX_DATA	8-bit data to be transmitted via UART. Default: 8’h0

3.13.6.3 UART Control Register (UART_CTL)

Register	Offset	R/W	Description	Reset Value
UART_CTL	UARTx_BA+0x08	R/W	UART Control Register	0x0000_1004

Bits	Description	
[31:17]	Reserved	Reserved.
[16]	RX_TIMOUT_INT_EN	1’b0: rx fifo timout interrupt disabled 1’b1: rx fifo timout interrupt enabled
[15:14]	Reserved	Reserved.
[13:12]	RX_NE_MASK	RX FIFO Non-Empty Interrupt MASK 2’b00: Interrupt MASK ≥ 1/4 FIFO depth 2’b01: Interrupt MASK ≥ 2/4 FIFO depth 2’b10: Interrupt MASK ≥ 3/4 FIFO depth 2’b11: Interrupt MASK ≥ 4/4 FIFO depth For example: If the 4-depth RX FIFO contains two or more data units, an interrupt will be generated if the non-empty interrupt is enabled and RX_NE_MASK is set to 2'b01. Default: 2’b01
[11]	RX_NE_INT_EN	RX FIFO Non-Empty Interrupt Enable 0: Non-empty interrupt disabled 1: Non-empty interrupt enabled Default: 1’b0
[10]	RX_OV_INT_EN	Receive Overflow Interrupt Enable 0: Overflow interrupt disabled. 1: Overflow interrupt enabled. Default: 1’b0
[9]	RX_ERR_INT_EN	Receive Error Interrupt Enable 0: Receive error interrupt disabled. 1: Receive error interrupt enabled. Default: 1’b0
[8]	RX_EN	UART RX Enable (including UART Receive and RX FIFO) 0: UART RX disabled. 1: UART RX enabled. Default: 1’b0
[7:4]	Reserved	Reserved.
[3:2]	TX_NF_MASK	TX FIFO Non-Full Interrupt MASK 2’b00: FIFO Empty Interrupt MASK 2’b01: Interrupt MASK < 2/4 FIFO depth 2’b10: MASK Interrupt MASK < 3/4 FIFO depth 2’b11: MASK Interrupt MASK < 4/4 FIFO depth For example: If the 4-depth TX FIFO contains fewer than two data units, an interrupt will be generated if the non-full interrupt is enabled and TX_NF_MASK is set to 2’b01. Default: 2’b01
[1]	TX_NF_INT_EN	TX FIFO Non-Full Interrupt Enable 0: Non-full interrupt disabled

		1: Non-full interrupt enabled Default: 1'b0
[0]	TX_EN	UART TX Enable (including UART Transmit and TX FIFO) 0: UART TX disabled. 1: UART TX enabled. Default: 1'b0

3.13.6.4 UART Status Register (UART_STATUS)

Register	Offset	R/W	Description	Reset Value
UART_STATUS	UARTx_BA+0x0C	R	UART Status Register	0x0000_0000

Bits	Description	
[31:23]	Reserved	Reserved.
[22]	TX_NF_FLAG	When the value of TX_FIFO_NUM satisfies the configuration of TX_NF_MASK, TX_NF_FLAG is set to 1. If TX_NF_INT_EN is then enabled, an interrupt is generated.
[21:19]	TX_FIFO_NUM	The quantity of data inside the TX FIFO. For example, if the depth of the TX FIFO is 4, there are 5 possibilities: 0, 1, 2, 3, and 4. 0 indicates that the TX FIFO is empty. Default: 3’h0
[18]	TX_FULL_FLAG	TX FIFO Full Flag TX_FIFO_NUM = Maximum Depth Default: 1’b0
[17]	TX_EMPTY_FLAG	TX FIFO Empty Flag TX_FIFO_NUM = 0 Default: 1’b0
[16]	TX_BUSY	UART is transmitting data. Default: 1’b0
[15:10]	Reserved	Reserved.
[9]	rx_timout_flag	1’b0: RX FIFO timneout flag Invalid 1’b1: RX FIFO timneout flag Effective
[8]	RX_NE_FLAG	When the value of RX_FIFO_NUM satisfies the configuration of RX_NE_MASK, the RX_NE_FLAG is set to 1. If RX_NE_INT_EN is then enabled, an interrupt is generated.
[7:5]	RX_FIFO_NUM	The quantity of data inside the RX FIFO. For example, if the depth of the RX FIFO is 4, there are 5 possibilities: 0, 1, 2, 3, and 4. 0 indicates that the RX FIFO is empty. Default: 3’h0
[4]	RX_OV_FLAG	When the RX FIFO is full and a new frame of data is received, the overflow flag RX_OV_FLAG will be set to 1. If the data is not read out immediately before the next frame arrives, this frame of data will be lost. Note: The RX_OV_FLAG bit is automatically cleared by hardware based on the FIFO status. Default: 1’b0
[3]	RX_ERR_FLAG	During RX data reception, if a stop bit error occurs (i.e., a logic 0 is received during the stop bit state), the RX_ERR_FLAG will be set to 1. Note: This flag will be cleared when read. Default: 1’b0
[2]	RX_FULL_FLAG	RX FIFO Full Flag RX_FIFO_NUM = Maximum Depth Default: 1’b0
[1]	RX_EMPTY_FLAG	RX FIFO Empty Flag RX_FIFO_NUM = 0

		Default: 1’b0
[0]	RX_BUSY	UART is receiving data. Default: 1’b0

3.13.6.5 UART BAUD Configuration Register (UART_BCNT)

Register	Offset	R/W	Description	Reset Value
UART_BCNT	UARTx_BA+0x10	R/W	UART BAUD Configuration Register	0x0000_0D06

Bits	Description	
[31:16]	Reserved	Reserved.
[15:0]	BCNT	Baud Rate Count Configuration Register (BCNT refers to the number of UART clock cycles) Note: The BCNT register only configures the baud rate count value based on the UART clock (not the baud rate value itself). Configuration must be completed before enabling RX or TX. (BCNT must not be written as 0). BCNT = UART Clock / UART Baud Rate Default: 16’d3334 (Clock: 32MHz, Baud Rate: 9600)

3.13.6.6 UART TIMEOUT CTL Register (TIMEOUT_CTL)

Register	Offset	R/W	Description	Reset Value
TIMEOUT_CTL	UARTx_BA+0x14	R/W	Timeout control	0x0000_0000

Bits	Description	
[31:21]	Reserved	Reserved.
[20]	TIMEOUT_EN	Timeout Enable Control 1’b0: Disabled 1’b1: Enabled
[19:0]	TIMEOUT_CNT_TOTAL	Configure the counter for triggering the timeout.

3.14 Timer Controller (TMR)

3.14.1 Overview

Timers are indispensable peripherals in SoC design, providing precise time management and time-based control functions for the system. Through flexible configuration and multi-mode operation, timers can adapt to various application scenarios, ranging from simple time measurement to complex real-time control systems.

3.14.2 Features

- Supports periodic mode and continuous mode
- Supports 25-bit counter
- APB-compatible clock source
- Supports rising edge, falling edge, and both-edge capture functions

3.14.3 Block Diagram

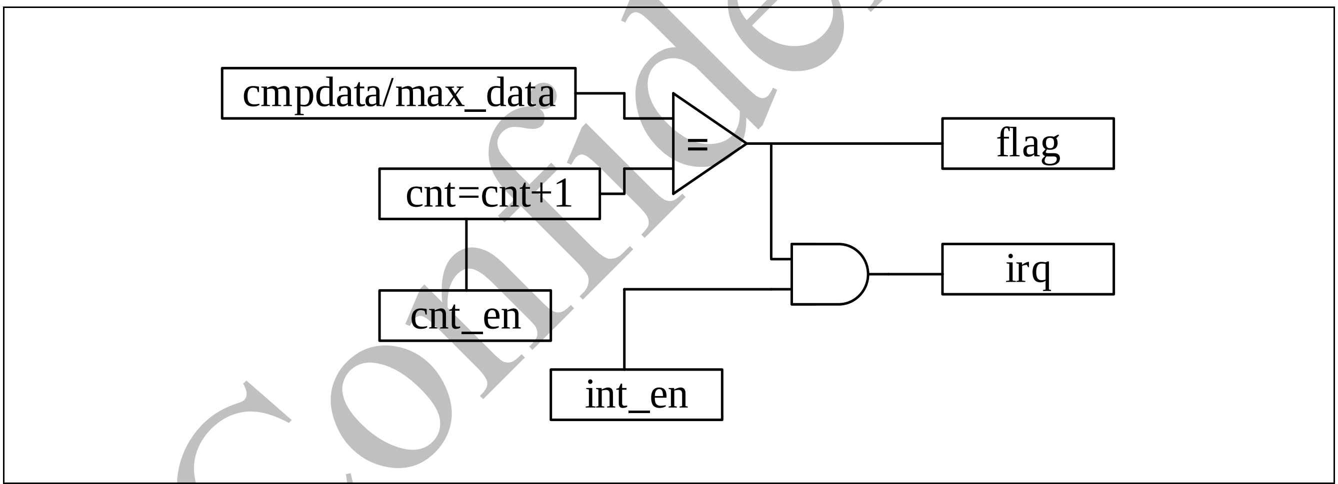


Figure 3-48 Timer Controller Block Diagram

3.14.4 Clock and Reset

The Timer uses the system APB bus clock, with clock and reset controlled by the RCC module.

3.14.5 Functional Description

The top-level module of Timer is shown in Figure 3-49:

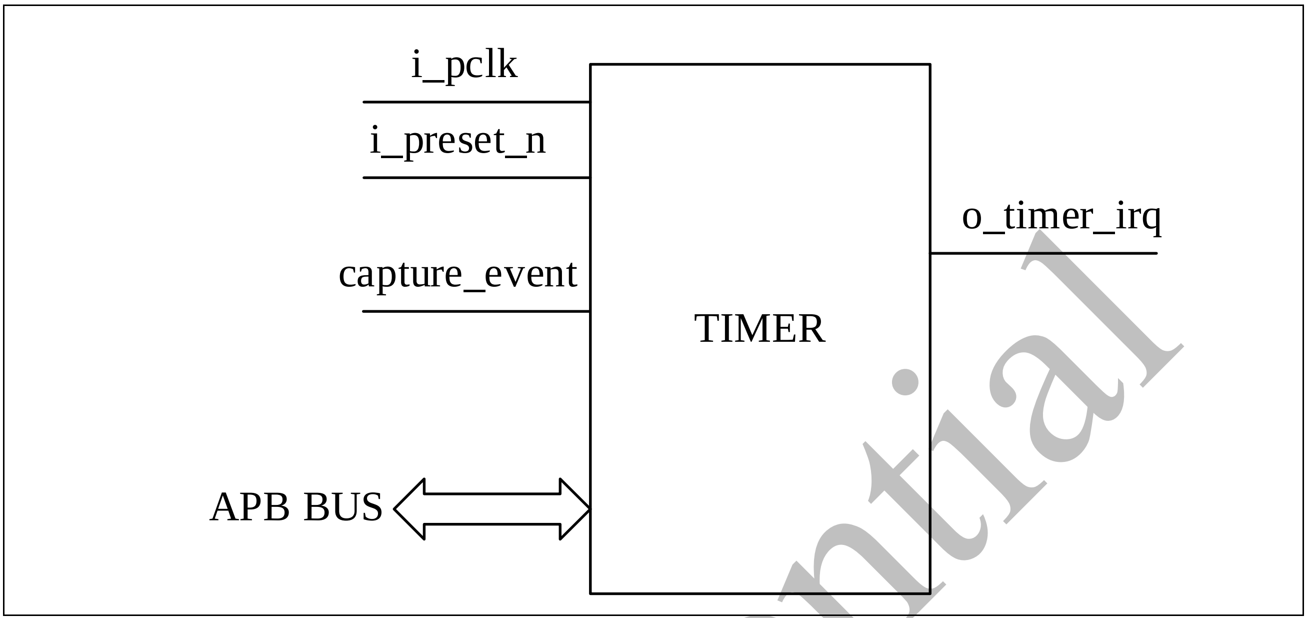


Figure 3-49 Timer Module Diagram

The timer operates in two modes: periodic mode and continuous mode.

3.14.5.1 Periodic Mode Operation Procedure

1. Configure the operating mode: TIMER_CTL → WORK_MODE=0
2. Configure the compare value: TIMER_CTL → CMPDATA; enable the interrupt: TIMER_CTL → INT_EN
3. Enable TIMER_CTL → CNT_EN

When the counter enable bit CNT_EN is set to 1, the timer begins counting. If CNT = CMPDATA, a TIMER_FLAG is generated. Both the TIMER_FLAG and interrupt require software clearing by writing 1. Regardless of whether the TIMER_FLAG and interrupt are cleared, the CNT value will be reset and restart counting. If the timer function is no longer needed, set CNT_EN to 0. Note: When CNT_EN is set to 1, do not configure CMPDATA to 0.

3.14.5.2 Continuous Mode Operation Procedure

1. Configure the operating mode: TIMER_CTL → WORK_MODE=1
2. Configure the compare value: TIMER_CTL → CMPDATA; enable the interrupt: TIMER_CTL → INT_EN

3. Enable TIMER_CTL → CNT_EN

When the counter enable bit CNT_EN is set to 1, the timer begins counting. If the count value equals the compare value, a TIMER_FLAG is generated. Both the TIMER_FLAG and interrupt require software clearing by writing 1.

Unlike the periodic mode, when CNT = CMPDATA, CNT does not reset to zero. Instead, it continues accumulating up to max, resets, and then starts accumulating again.

Regardless of whether the TIMER_FLAG and interrupt are cleared, the CNT value will be reset and restart counting. If the timer function is no longer needed, set CNT_EN to 0. Note: When CNT_EN is set to 1, do not configure CMPDATA to 0.

3.14.5.3 Capture Function Operation Procedure

1. Configure the operating mode: TIMER_CTL → WORK_MODE=1
2. Configure the compare value: TIMER_CTL → CMPDATA; enable the interrupt: TIMER_CTL → INT_EN
3. Configure capture edges: rising edge, falling edge, or both edges. TIMER_CAPTURE → CAPTURE_SEL
4. Enable the capture function: TIMER_CAPTURE → CAPTURE_EN
5. Enable the CNT: TIMER_CTL → CNT_EN
6. Query the captured CNT value: TIMER_CAPTURE → CAP_CNT

3.14.6 TMR Register Map

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

Register	Offset	R/W	Description	Reset Value
TMR Base Address: TMR_BA = 0x4000_8000				
TIMER_CTL	TMR_BA+0x00	R/W	Timer Control Register	0x0000_0000
TIMER_CAPTURE	TMR_BA+0x04	R/W	Timer Capture Control Register	0x0000_0000

3.14.7 TMR Register Description

3.14.7.1 Timer Control Register (TIMER_CTL)

Register	Offset	R/W	Description	Reset Value
TIMER_CTL	TMR_BA+0x00	R/W	Timer Control Register	0x0000_0000

Bits	Descriptions	
[31]	INT_EN	Timer Count = CMPDATA Interrupt Enable 0: Timer interrupt disabled 1: Timer interrupt enabled Default: 1’b0
[30]	WORK_MODE	1’b0: Periodic mode, counter increments until CMP, then resets. 1’b1: Continuous mode, counter increments until all CMP bits are 1, then resets.
[29]	CNT_EN	Timer Count Enable 0: Timer count disabled 1: Timer count enabled When CNT_EN is set to 1, do not configure CMPDATA to 0. Default: 1’b0
[28]	TIMER_FLAG	Timer CNT=CMPDATA Count Flag When the FLAG bit is set to high, software must write 1 to clear it. Note: Writing 1 to clear the flag will also clear the Timer interrupt. Default: 1’b0
[27]	CAPTURE_INT_EN	Timer Capture Interrupt Enable 0: Timer capture interrupt disabled 1: Timer capture interrupt enabled Default: 1’b0
[26]	CAPTURE_FLAG	Timer Capture Flag When the FLAG bit is set to high, software must write 1 to clear it. Note: Writing 1 to clear the flag will also clear the Timer interrupt. Default: 1’b0
[25]	SOFTWARE_CNT_CLEAR	Timer CNT Clear Enable 1’b0: Timer CNT not cleared 1’b1: Timer CNT cleared Software must be manually configured to 1 or 0.
[24:0]	CMPDATA	Timer Count Comparison Value Default: 25’h0

3.14.7.2 Timer Capture Register (TIMERx_CAP)

Register	Offset	R/W	Description	Reset Value
TIMER_CAPTURE	TMR_BA+0x04	R/W	Timer Capture Control Register	0x0000_0000

Bits	Descriptions	
[31:28]	Reserved	Reserved.
[27]	CAPTURE_EN	Capture Function Enable 1’b1: Function enable
[26:25]	CAPTURE_SEL	2’b00: Rising edge capture 2’b01: Falling edge capture 2’b1x: Both edges capture
[24:0]	CAP_CNT	Timer Count Capture Value Default: 25’h0

3.15 CLK Measurement

3.15.1 Overview

The clock frequency measurement module is usually used to measure the operating frequency of the processor or other key components. This is very important for performance analysis, power management and system optimization. By accurately measuring the clock frequency, the system can better monitor and optimize its operating status to ensure a balance between performance requirements and power constraints.

3.15.2 Features

- Supports internal/external clock selection
- Supports high/low level and frequency measurement (range 1Hz to 16MHz)
- The clock source is the system crystal oscillator XTH (default 32M)
- Supports a 26-bit counter
- Supports multiple consecutive samplings (sampling times can be selected as 1, 2, 4, 8, 16, 32, 64, 128)

3.15.3 Block Diagram

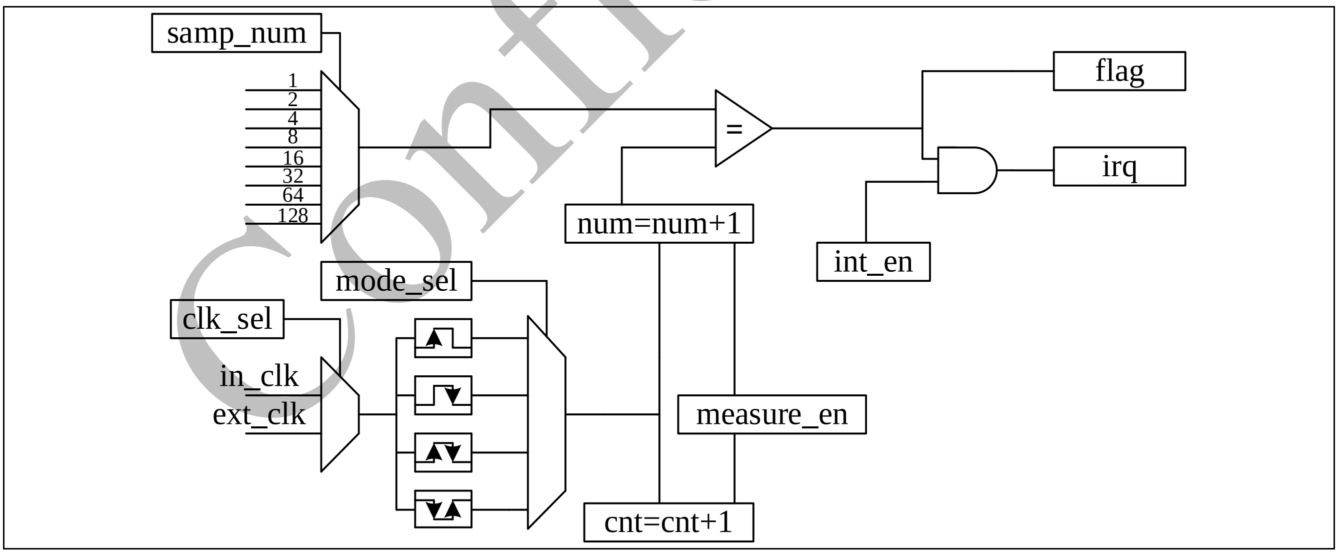


Figure 3-50 CLK Measure Block Diagram

3.15.4 Functional Description

3.15.4.1 Measure Principle

The measurement uses a stable clock (REF_CLK) to count the clock to be measured (CAL_CLK). The frequency of the clock to be measured is Fcal, and the frequency of the reference clock is Fref. Within M cycles of the clock to be measured, the reference clock has a total of N cycles. Then the frequency of the clock to be measured is:

$$M / Fcal = N / Fref$$

$$Fcal = Fref * M / N$$

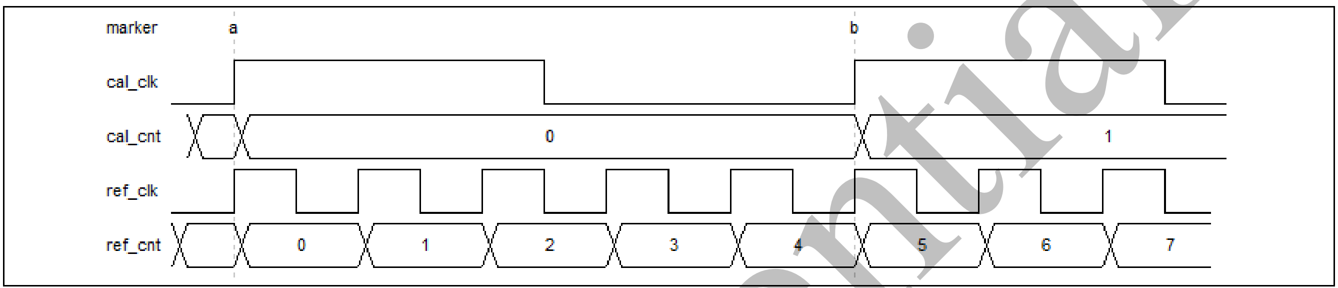


Figure 3-51 Measure Principle

The measurement method introduces two kinds of errors. One is the error caused by the jitter of the reference clock. The more stable the reference clock, the smaller the absolute error of the clock to be measured; the second is the error caused by signal synchronization. After entering the module, the clock to be measured will pass through a two-stage synchronizer. If the clock period to be measured is not an integer multiple of the stable clock period, the measurement will produce an absolute error less than or equal to one stable clock period. The following proves that the measurement error is less than or equal to a stable clock cycle:

- For terminal a, the resulting error is [-T, 0].
- For terminal b, the resulting error is [0, T].

The total error is the sum of the errors at the a and b terminals, and the total error range is [-T, T]. Therefore, the absolute error <=T.

For this solution, the accuracy of using 32M to measure a 32K cycle is 1/1000. The accuracy of measuring ten 32K cycles is 1/10000. To achieve the accuracy requirement of 500ppm, at least 20 clock cycles need to be measured.

As shown in the figure below, the first RC32K clock cycle is synchronized to 4 stable clock cycles, and the second RC32K clock cycle is synchronized to 3 stable clock cycles. The error introduced by the measurement method will not be eliminated.

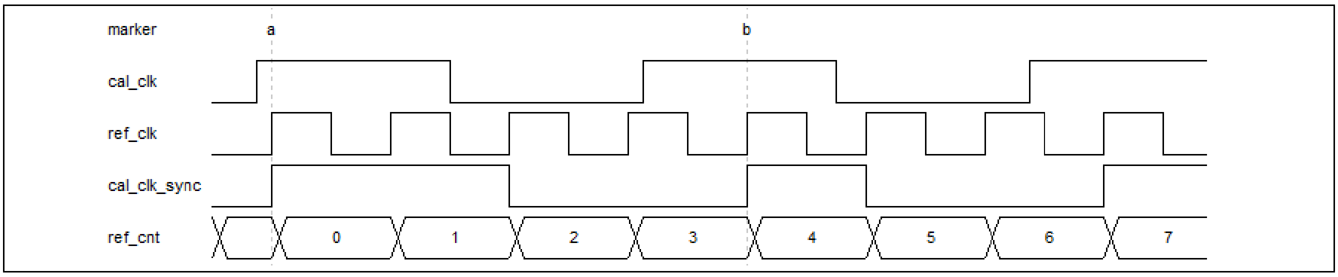


Figure 3-52 Measure Principle

The clock to be measured can also cause errors, which can be reduced. If the clock is unstable, multiple measurements will get different results. The longer the measurement period, the smaller the difference between multiple measurements and the more accurate the results obtained.

3.15.5 CLK Measure Register Map

Register	offset	R/W	Description	Reset Value
CLKMEAS_BASE addr：0x4000_7000				
ClkMeasCtlReg	CLKMEAS_BASE+0x00	R/W	Control Register	0x0000_0080
ClkMeasCntReg	CLKMEAS_BASE+0x04	R	Count Register	0x0000_0000

3.15.6 CLK Measurement Register Description

3.15.6.1 ClkMeasCtlReg

Register	Offset	R/W	Description	Reset Value
ClkMeasCtlReg	CLKMEAS_BASE +0x00	R/W	Control register	0x0000_0080

Bits	Description	
[31:9]	Reserved	Reserved
[8]	MEASURE_FLAG	Clock measurement end flag. 1: Clock measurement is end 0: Clock measurement is not end When the flag bit is asserted (pulled high), the software needs to write a 1 to clear it. Note: Writing a 1 to clear the flag bit to 0 will also clear the clock measurement interrupt. Reset_value:{1b0}
[7:5]	SAMP_NUM	Consecutive Sampling Count 3’h0: sample once only 3’h1: sample 2 times consecutively 3’h2: sample 4 times consecutively 3’h3: sample 8 times consecutively 3’h4: sample 16 times consecutively 3’h5: sample 32 times consecutively 3’h6: sample 64 times consecutively 3’h7: sample 128 times consecutively Reset_value:{3h4}
[4:3]	MODE_SEL	Edge Mode Selection for the Clock to be Measured 2’h0: Rising edge, measure the frequency of the clock 2’h1: Falling edge, measure the frequency of the clock 2’h2: Rising edge followed by falling edge, measure the high-level pulse width (only sample once) 2’h3: Falling edge followed by rising edge, measure the low-level pulse width (only sample once) Reset_value:{2b00}
[2]	CLK_SEL	Clock to be tested selection 0: Internal clock as the source to be detected 1: External clock as the source to be detected Reset_value:{1b0}
[1]	INT_EN	Clock Measurement Interrupt Enable Bit 0: Clock measurement interrupt is disabled 1: Clock measurement interrupt is enabled Reset_value:{1b0}
[0]	MEASURE_EN	Clock Measurement Enable 0: Clock measurement is disabled. 1: Clock measurement is enabled. Reset_value:{1b0} Note: After a continuous sampling ends, the hardware will automatically turn off.

		If you need to turn it on again, you must first disable it and then re-enable it, otherwise, it will be ineffective.

3.15.6.2 ClkMeasCntReg

Register	offset	R/W	Description	Reset Value
ClkMeasCntReg	CLKMEAS_BASE +0x04	R		0x0000_0000

Bits	Description	
[31:26]	Reserved	Reserved
[25:0]	MEASURE_CNT	The total count value of clock measurement, which will be cleared by hardware each time the MEASURE_EN is enabled to ensure the accuracy of the subsequent measurement values. Reset_value:{26’h0000000}

3.15.7 Software flow

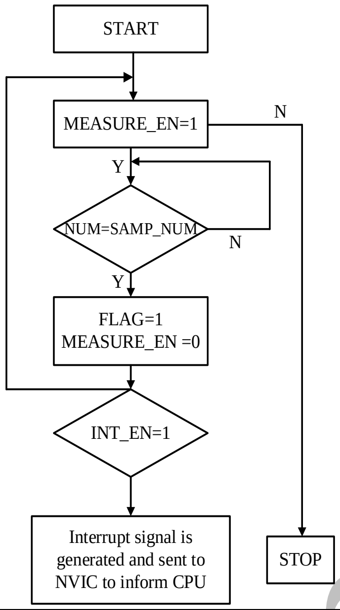


Figure 3-53 Measure flow

1. Select the clock to be tested and the reference clock, wait for the clock to stabilize.
2. Configure SAMP_NUM and MODE_SEL. Set the Sampling Count and Edge Mode for measurement.
3. Configure INT_EN, select interrupt mode or polling mode.
4. Enable MEASURE_EN.
5. In interrupt mode. After the interrupt is triggered, check the interrupt status. If MEAS_STOP_INT is set to 1, read the value of MEASURE_CNT. Calculate the true frequency of CAL_CLK through the formula in 3.15.4.1.
6. In the polling mode. After the software detects that MEASURE_FLAG is set to 1, reads MEASURE_CNT and calculates the CAL_CLK frequency.
7. Clear status flag, interrupt flag.

3.16 KEYSCAN

3.16.1 Overview

Keyscan supports key matrix with up to 6 rows by 6 columns. Each individual rows or columns can be enabled or disabled through register settings. GPIO pins can be configured to be used for keyscan. A few key scan parameters can be set through registers, including polarity (low or high indicating key pressed); scan interval; row interval and de-bounce time.

The keyscan has a polling mode and an interrupt mode. Keyscan will automatically scan the output/input pins, and store the row/column info corresponding to the key pressed into read only registers. For polling mode, software need to read register bit periodictity. When keyscan interrupt is enabled, a valid key press or release both can trigger an interrupt. After a keyscan interrupt is serviced, software can read related register bit to acquire the key state.

3.16.2 Features

- The size of the button can be configured, m*n (m=1-6, n=1-6), and the maximum support is 6*6.
- Support key debounce.
- The scan period can be configured.
- The button polarity is configurable, high-level active or low-level active.
- Two flag bits, the button state change flag bit, the wake-up flag bit under low power consumption.
- The read-only register keeps the key state.

3.16.3 Block Diagram

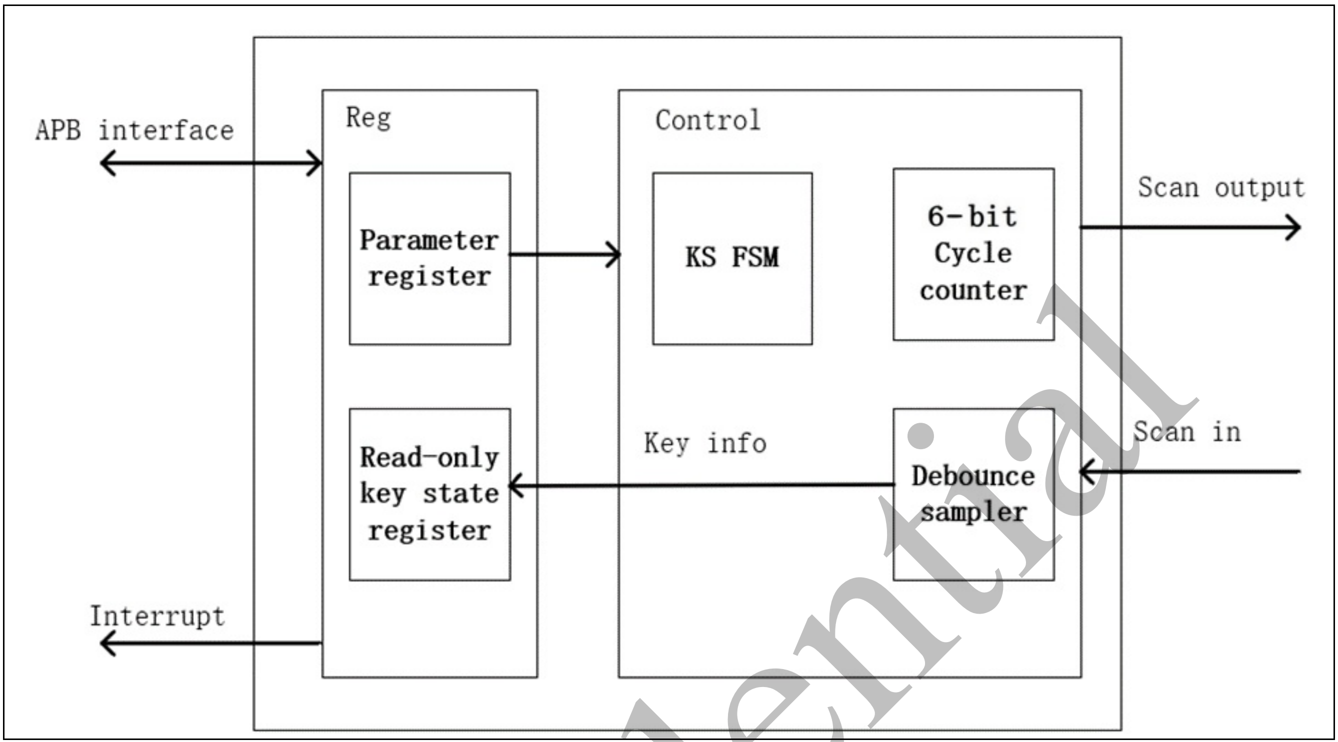


Figure 3-54 Keyscan Block Diagram

Reg module works in APB clock domain and has the following functions:

- Scan the parameter configuration and pass the parameters to the Control module.
- Save the button information sampled by the Control module to the read-only register.
- Generate an interrupt signal.

The Control module works in the sclk (sclk, scan clk) scan clock domain and has the following functions:

- Output the row scan signal.
- Sample the column input signal and pass the sampled key information to the Reg module.

3.16.3.1 Pins Description

System interface

Table 3-13 System Interface Description

Pins	I/O	Description
PCLK	I	APB clock
PRESETN	I	APB clock domain reset signal
KSCLK	I	CTL module working clock
KS_RESETN	I	Ksclk clock domain reset signal
KS_INT	O	Keyscan module interrupt signal

APB bus interface

Table 3-14 APB Bus Interface Description

Pins	I/O	Description
PADDR[31:0]	I	APB address
PWRITE	I	APB read and write enable signal 1: Write 0: Read
PSEL	I	APB select signal
PENABLE	I	APB enable signal
PWDATA[31:0]	I	APB write data line
PRDATA[31:0]	O	APB read data line

GPIO interface

Table 3-15 GPIO Interface Description

Pins	I/O	Description
KSI[5:0]	I	GPIO column input signal
KSO[5:0]	O	GPIO row output signal
KSOE[5:0]	O	GPIO row output enable signal 1: GPIO output enable 0: GPIO output is invalid

3.16.4 Functional Description

3.16.4.1 Scanning Principle

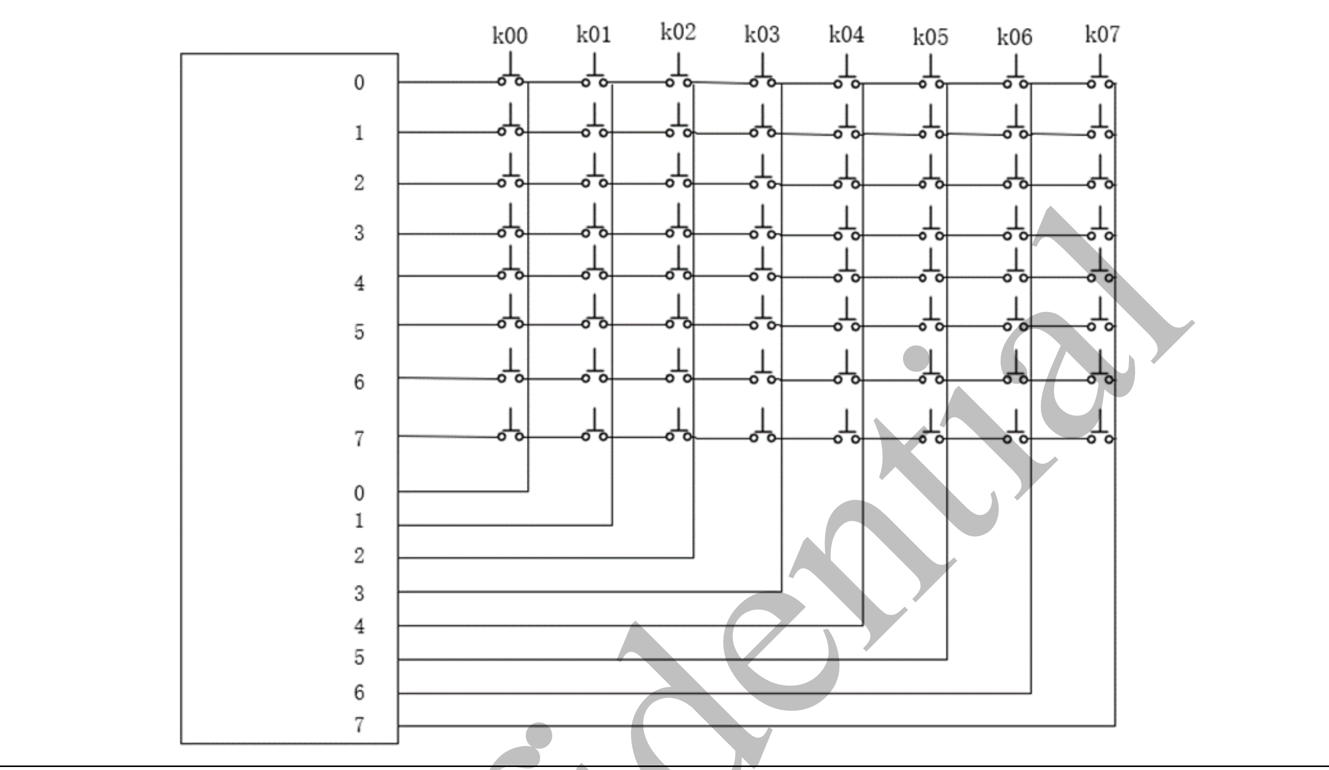


Figure 3-55 Scanning Principle

Take the 8*8 key matrix as an example, the scan signal is active at low level, and the column input defaults to high level.

When scanning row 0, row 0 outputs low level, and rows 1-7 output high level. When a key in row 0 is pressed, the corresponding column input signal changes from high to low, and the other column input signals remain high. For example, when k00 is on, the input signal in column 0 is low level, and the input signal in columns 1-7 is high level. Keyscan reads the column input signal, and saves the key state at this time to the corresponding register.

When scanning the first row, the first row outputs low level, and the remaining rows output high level. Keyscan reads the column input signal, and saves the key state at this time to the corresponding register.

By analogy, until all rows are scanned, the 8*8 key status is saved in the corresponding register.

The Keyscan module supports up to 6*6 key matrix, each key corresponds to a read-only register. You can scan only part of the row and part of the column through the configuration register.

The column input signal is read by keyscan after debounce, and the debounce time can be configured by register.

After scanning one row, the time from scanning to the next row can be configured by register, this time is called row interval.

Scan all rows once, which is called a frame. The time from the end of scanning one frame to scanning the next frame can be configured by register, and this time is called frame interval. The user can flexibly configure the row interval and frame interval according to the actual situation to achieve different scanning frequencies.

State Machine

There are five states in Keyscan: IDLE, INTERVAL, STEP, DEB, WAIT.

- **IDLE**. The keyscan_en signal is not turned on, and the keyscan is in the IDLE state. The software configuration register needs to be maintained in this state. After the keyscan_en signal is turned on, the state machine transitions from IDLE to INTERVA state.
- **INTERVAL**, frame interval status (scan interval). After scanning all rows once, it will enter the INTERVAL state. The next scan will be started after waiting for the scan interval.
- **STEP**, valid state of the row. In this state, the keyscan sets the output signal of a certain row to the valid state, and the output signals of the other rows are in the invalid state. And it detects the column input signal.
- **DEB**, debounce state. It happens after detecting a change in the button state. After the button state is maintained for the debounce time, the state information is written into the register and PAN271 series enters the WAIT state.
- **WAIT**, row interval status (row interval). Scan the next row after waiting for the row interval time.

Scan Mode

There are two ways to use the keyscan module: polling mode and interrupt mode.

In polling mode, after keyscan is enabled, the key status will be reflected in the register in real time. The software reads the value of the register regularly to check the button status.

In interrupt mode, the interrupt enable bit needs to be turned on. After scanning all the rows, if a change in the button state (button pressed or released) is detected, an interrupt is triggered and the enable signal is reset. The software needs to reset the enable signal and then start scanning again.

Keyscan supports up to 6*6 keys. No matter which scanning mode is used, and no matter how many keys are implemented, the hardware will scan 6 lines once, which is called a frame. The scanning order is from row KSO[0] to row KSO[5].

Parameter Configuration

The keyscan module has two clock domains, the read and write registers are in the APB clock domain, and the scan control is in the sclk clock domain. The sclk clock is obtained by dividing the frequency of the RCC module by the abp clock. The frequency division coefficient range is [0, 512].

$$f\_sclk = f\_ apbclk / rcc\_clkdiv$$

The hardware scans all 6 rows every time, even if the realized key matrix is 1*1. Scanning all 6 lines is called a frame. The time for scanning a frame is called a frame period, and the reciprocal is called the frame frequency. The frame frequency should be at least twice the frequency of key presses. For example, the key press frequency is 25HZ, that is, the key is pressed 25 times in 1 second. Then the frame frequency is at least 50HZ, which means scanning 50 times per second.

The frame frequency is related to the sclk clock frequency, scan line interval, frame interval, debounce period, and how many rows of button states are changed within a frame. In the worst case, all 6 rows of keys change within a frame. At this time, the frame period can be roughly calculated with the following formula:

$Frame\ period = ((line\ interval + debounce\ period) * 6 + frame\ interval) * sclk\ clock\ period.$

The frame frequency should meet:

$$Frame\ frequency = 1/frame\ period > 2 * f\_key\_press$$

It can be seen from the above formula that if the sclk clock frequency is relatively low, configure a small line interval, debounce period, and frame interval. If the sclk clock frequency is relatively high, the line interval, debounce period, and frame interval should be configured larger.

GPIO Allocation

GPIO has built-in pull-up input, pull-down input function.

KSO[5:0] corresponds to the row, which needs to configure the GPIO port mode as output mode.

KSI[5:0] corresponds to the column, which need to configure the GPIO port as pull-up input or pull-down input.

For example, to realize an 4*4 button matrix, you can use KSO[3:0] as the row and KSI[3:0] as the column by configuring the IoCfgReg register. You can also use KSO[1:0] KSO[5:4] as the row. That is, any 4 rows can be selected from KSO[5:0].

GPIO		GPIO	
P02	Ks_o[1]	P13	Ks_o[3]
P03	Ks_i[1]	P14	Ks_o[4]
P04	Ks_i[2]/Ks_o[3]	P15	Ks_o[5]
P05	Ks_i[3]	P16	Ks_i[1]
P06	Ks_i[4]	P17	Ks_o[4]
P07	Ks_i[5]	P20	Ks_i[2]
P10	Ks_o[0]	P21	Ks_o[0]
P11	Ks_i[0]/Ks_o[1]	P22	Ks_i[0]
P12	Ks_o[2]		

3.16.4.2 Configuration Process

IOMUX Configuration

According to the size of the key matrix to be implemented, the user can configure GPIO as the input and output of the keyscan.

For the GPIO that implements the keyscan output function, PUEN is disabled.

For the GPIO that implements the keyscan input function, enable PUEN.

Register configuration

IoCfgReg register (4000_9004)

Configure KSOE and KSIE, select row output and column output port

IntCfgReg register (4000_9008)

Disable or enable INT_EN, select polling mode or interrupt mode

KsCfgReg register (4000_900C)

Since the power consumption is related to the clock frequency, if the formula in section 2.6.1 is satisfied, the sclk clock frequency should be as low as possible. The following are the default values of some registers:

DEB_VALUE =0100. Represents 16 sclk clock cycles

RINTVAL_VALUE = 0000, which means 1 sclk clock cycle.

SINTVAL_VALUE=1000, which means 256 sclk clock cycles

Assuming the button state change frequency is 25HZ, and the above default configuration is used, at this time, the sclk clock cycle should meet:

$$T < 20ms /((16+1)*6+256) = 56us$$

In other words, the sclk clock frequency should be greater than 18KHz. In actual use, the above registers should be flexibly configured according to the button jitter time and usage scenarios.

KsEnReg register (4000_9000)

Set KS_EN to 1, turn on keyboard scan.

Example: 2*2 Button Matrix

Turn on the APB clock and keyscan clock.

Configure IOMUX as follow:

1. IoCfgReg register (4000_9004), configured as 0x0000_00C3
2. IntCfgReg register (4000_9008) is configured as 0x0000_0001
3. KsEnReg register (4000_9000), configured as 0x0000_0001

Example: Wake Up in Low Power Consumption Mode

1. Configure row interval, frame interval, debounce time and other parameters.
2. Enable ks_en and wk_en.
3. Write 1 to clear the wkf register and wait for the system to enter the low-power mode.
4. Any button state change will wake up the system and trigger an interrupt after the system wakes up.
5. Execute the interrupt program and clear the wkf register.

3.16.5 Keyscan Regiser Map

Register	offset	R/W	Description	Reset Value
KeyScan base addr:0x4000_9000-0x4000_9FFF				
KSENREG	KS_BA+0x00	R/W	Keyscan enable register	0x0000_0000
IOCFGREG	KS_BA+0x04	R/W	Configurate keyscan IO register	0x0000_0000
INTCFGREG	KS_BA+0x08	R/W	Configurate interrupt register	0x0000_0000
KSCFGREG	KS_BA+0x0C	R/W	Configurate keyscan register	0x0000_0804
KSINFOREG0	KS_BA+0x10	R	Key information register0	0x0000_0000
KSINFOREG1	KS_BA+0x14	R	Key information register1	0x0000_0000

3.16.6 Keyscan Register Description

3.16.6.1 KsEnReg

Register	offset	R/W	Description	Reset Value
KSENREG	KS_BA+0x00	R/W	Keyscan enable register	0x0000_0000

Bits	Description	
[31:2]	Reserved	Reserved
[1]	WK_EN	1: Wake up function is enabled 0: Wake up function is disabled Reset value: 0x0
[0]	KS_EN	1: Keyscan is enabled 0: Keyscan is disabled If int_en is set, ks_en will be cleared while detecting interrupt. If wk_en is set, ks_en can not be cleared by hardware. Reset value: 0x0

3.16.6.2 IoCfgReg

Register	offset	R/W	Description	Reset Value
IOCFGREG	KS_BA+0x04	R/W	Configurate keyscan IO register	0x0000_0000

Bits	Description	
[31:12]	Reserved	Reserved
[11:6]	KSOE	Keyscan output configuration 1: KSO enabled 0: KSO disabled, keyscan does not scan this row. Reset_value: 0x0
[5:0]	KSIE	Keyscan input configuration 1: KSI enabled 0: KSI disabled, keyscan does not check this column. Reset_value: 0x0

3.16.6.3 IntCfgReg

Register	offset	R/W	Description	Reset Value
INTCFGREG	KS_BA+0x08	R/W	Configurate interrupt register	0x0000_0000

Bits	Description	
[31:4]	Reserved	Reserved
[3]	WKF	Keyscan wakeup flag This bit is cleared by writing 1 to it. This bit is valid only when WK_EN is set. Reset_value: 0x0
[2]	KSIF	Keyscan key_change interrupt Flag. 1: Key_change occurred 0: No change This bit is cleared by writing 1 to it. This bit is valid only when INT_EN is set. Reset_value: 0x0
[1]	KSF	Keyscan key_change flag 1: Key_change occurred 0: Key_change did not occur This bit is cleared by writing 1 to it. Reset_value: 0x0
[0]	INT_EN	Keyscan key_change interrupt Enable bit 1: Key_change Interrupt enabled 0: Key_change Interrupt disabled INT_EN =1, KSIF=KSF; INT_EN = 0, KSIF=0 Reset_value: 0x0

3.16.6.4 KsCfgReg

Register	offset	R/W	Description	Reset Value
KSCFGREG	KS_BA+0x0C	R/W	Configurate keyscan register	0x0000_0804

Bits	Description	
[31:14]	Reserved	Reserved
[13]	POL	Configure the valid level 1: High valid 0: Low valid Reset value : 0x0
[12]	CLR_KEY	Clear all key status register 1: Clear enabled 0: No change This bit will be cleared automatically after writing 1 to it. Reset value: 0x0
[11:8]	SINTVAL_VALUE	Determine the interval time between the last row and the first row of next scan. Period = 2 ^ SINTVAL_VALUE + 1(sclk) Eg. SINTVAL_VALUE = 1000, period = 2 ^ 8 + 1 =257(sclk) Reset value: {4b1000}
[7:4]	RINTVAL_VALUE	Determine period between the current row and the next row. Period = RINTVAL_VALUE + 1(sclk) Eg. RINTVAL_VALUE = 0000, period = 1 (sclk) Reset value: {4b0000}
[3:0]	DEB_VALUE	Debounce time. Only the keys state hold larger or equal than debounce time can they be written to registers. Debounce time = 2 ^ DEB_VALUE - 1(sclk) (DEB_VALUE >=1) If DEB_VALUE is set to 0,it means debounce time is 2 sclk. Eg. DEB_VALUE = 0100, debounce time = 2 ^ 4 - 1= 15 (sclk) Reset value: {4b0100}

3.16.6.5 KsInfoReg0

Register	offset	R/W	Description	Reset Value
KSINFOREG0	KS_BA+0x10	R	Key information register0	0x0000_0000

Bits	Description	
[31:30]	Reserved	Reserved
[29:24]	ROW3	ROW3[x]-ROW3_COLx x=0-5 1: Key is pressed 0: Key is released
[23:22]	Reserved	Reserved
[21:16]	ROW2	ROW2[x]-ROW2_COLx x=0-5 1: Key is pressed 0: Key is released
[15:14]	Reserved	Reserved
[13:8]	ROW1	ROW1[x]-ROW1_COLx x=0-5 1: Key is pressed 0: Key is released
[7:6]	Reserved	Reserved
[5:0]	ROW0	ROW0[x]-ROW0_COLx x=0-5 1: Key is pressed 0: Key is released

3.16.6.6 KsInfoReg1

Register	offset	R/W	Description	Reset Value
KSINFOREG1	KS_BA+0x14	R	Key information register1	0x0000_0000

Bits	Description	
[31:14]	Reserved	Reserved
[13:8]	ROW5	ROW5[x]-ROW5_COLx x=0-5 1: Key is pressed 0: Key is released
[7:6]	Reserved	Reserved
[5:0]	ROW4	ROW4[x]-ROW4_COLx x=0-5 1: Key is pressed 0: Key is released

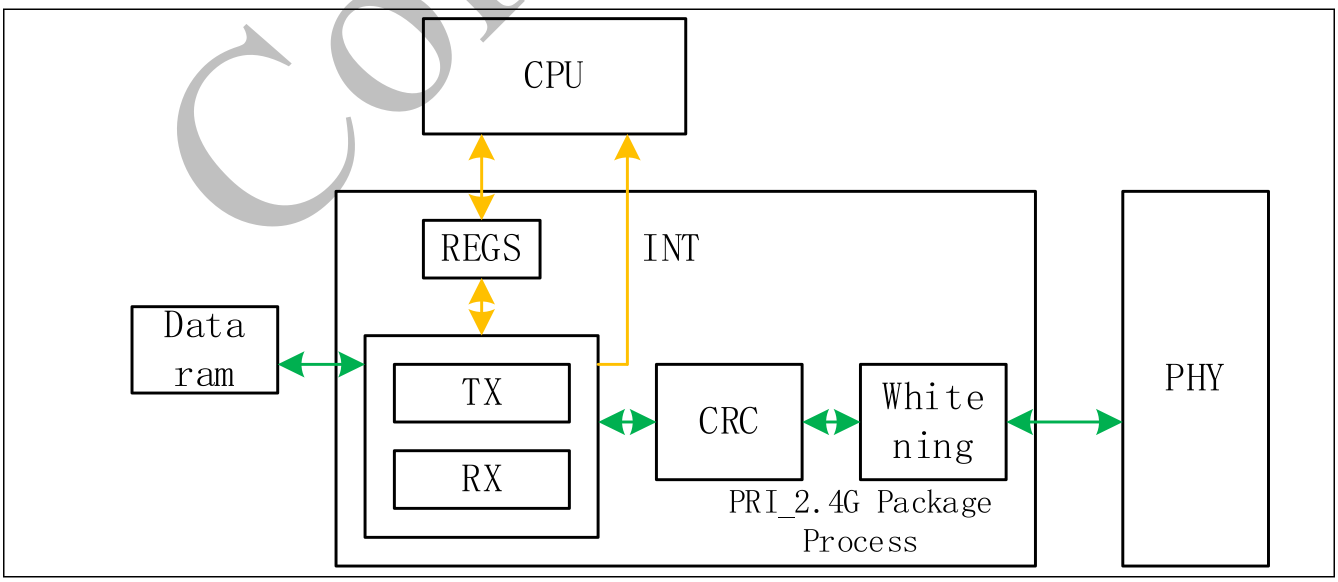
4 2.4 GHz proprietary protocols

4.1 System features

- Support for 250K, 1M and 2M PHY
- XN297L, PAN1026, Nordic Transceiver protocol compliant
- Support for No Acknowledge, Acknowledge and Acknowledge with payload
- Support for CRC8, CRC16 and CRC24
- Support for the whitening
- Compatible with Bluetooth frame structure, can simulate Bluetooth broadcast and scanning
- Compatible with Bluetooth CODED PHY S2/S8
- Compatible with 250K CODED PHY S2/S8
- Support for 2-byte address
- Support for the same spread spectrum function as the BLE protocol
- Support for hardware retransmission (enhanced and normal)
- Continuous reception (PLL always on)
- Multi-channel reception (3-pipe)
- Address matching error tolerance
- BLE mode supports whitelist filtering

4.2 Block diagram

The overall structure of the proprietary is as follows:



The proprietary protocols implementation is mainly a packet processing module, including register module, TX state control, RX state control, CRC, whitening.

4.3 Frame structure for proprietary protocols

4.3.1 XN297 frame structure

The XN297 frame structure has three modes: Normal, Normal_m1 and Enhanced.

4.3.1.1 Normal mode

Preamble (3Bytes)	Addr (2-5Bytes)	Payload (1-64Bytes)	Crc24/crc16/crc8 (3/2/1Bytes)

The normal mode frame structure consists of Preamble, Addr, Payload and CRC, which the address and length can be configured, and the CRC can choose crc24, crc16 and then crc8.

4.3.1.2 Normal_m1 mode

The frame structure is the same as the normal mode frame structure, in this mode:
After TX ends, it automatically enters RX mode after waiting for the latency (hardware latency).
After RX ends, it automatically enters TX mode after waiting for the latency (hardware latency).

4.3.1.3 Enhanced mode

Preamble (3Bytes)	Addr (2-5Bytes)	Signal (10bits)	Payload (0-64Bytes)	Crc24/crc16/crc8 (3/2/1Bytes)

Where signal includes:

Length (7bits)	PID (2bits)	NOACK (1bit)

Preamble: 3bytes total, 0x710F55, sent from the high bit.
Addr: 2, 3, 4, 5bytes address length configurable, configured by register, sent from the high bit.
Signal: In enhanced mode, signal is made up of {length[6:0], tx_pid[1:0], tx_no_ack}. It is sent from the high bit, and is not available in normal mode. Pid can be maintained by the software at each transmit.
Payload: Send from the low bit.
Crc24/crc16/crc8: Sent from the high bit.
Whitenning: Addr (Optional), Signal(Normal mode without this segment), Payload and Crc24. Initial values are configurable.
CRC: Addr(Optional), Signal (Normal mode without this segment), Payload.

4.3.2 NRF frame structure

The NRF frame structure has three modes: Normal, Normal_m1 and Enhanced.

4.3.2.1 Normal mode

Preamble (1~24Bytes)	Addr (2-5Bytes)	Header1 (0-1Bytes)	Header0 (0-1Bytes)	Length (0-1Bytes)	Payload (0-255Bytes)	Crc24/crc16/crc8 (3/2/1Bytes)

The normal mode frame structure consists of Preamble, Addr, Header1, Header0, Length, Payload and CRC, which the length of Preamble, the Addr and Length can be configured, and the CRC can choose crc24, crc16 and then crc8. Header1, Header0 and Length are optional fields, determined by register pri_04[8] and register pri_04[10:9] together.

4.3.2.2 Normal_m1 mode

The frame structure is the same as the normal mode frame structure, in this mode:

After TX ends, it automatically enters RX mode after waiting for the interval timing which is consists of hardware timing and setting timing.

After RX ends, it automatically enters TX mode after waiting for the interval timing which is consists of hardware timing and setting timing.

4.3.2.3 Enhanced mode

Preamble (1Byte)	Addr (2-5Bytes)	Signal (9/11bits)	Payload (0-255Bytes)	Crc24/crc16/crc8 (3/2/1Bytes)

There are two types of enhanced mode, the difference is the structure of the signal, the length of the enhanced-1 is 6bits, and the length of the enhanced-2 is 8bits, as follows:

Length (6bits)	PID (2bits)	NOACK (1bit)

Length (8bits)	PID (2bit)	NOACK (1bit)

Preamble: 1byte in total. If the highest bit of the Addr is 1, send 10101010, otherwise send 01010101. It is sent from the high bit. Length is configurable, which supports 1, 2, 3, 4, 6, 8, 12, 16, 24B.

Addr: Configured by the register, variable length (2-5bytes), sent from the high bit.

Signal: In enhanced mode, signal is made up of {length[5/7:0], tx_pid[1:0], tx_no_ack}. It is not available in normal mode. Pid can be maintained by the software at each transmit.

Payload: Sent from the high bit.

Crc24/crc16/crc8: The value after the calculation is completed, sent from the high bit.

Whitening: Addr (Optional), Signal(Normal mode without this segment), Payload and Crc24. Initial values are configurable.

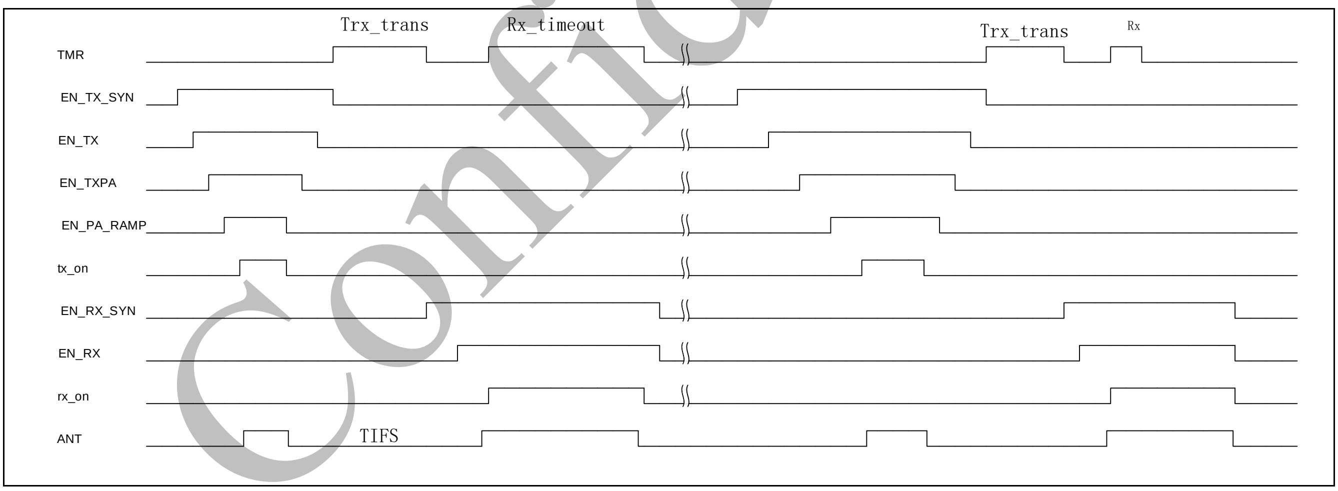
CRC: Addr(Optional), Signal (Normal mode without this segment), Payload.

4.3.3 250k frame structure

The 250k rate mode supports XN297, nordic and CC2510FX/CC2511FX communication protocols. For CC2510FX/CC2511FX frame structure, use nordic normal mode frame structure adaptation, and enable CRC and whitening function. It supports the same spread spectrum mode as the BLE protocol.

4.4 Timing Description

The switching timing of PTX is shown in the figure below:



The timing of PRX is similar with PTX. For single TX or single RX timing, see the TX or RX section in the figure above. For conversion time calculations, see the following table:

State	Enable Digital-to-Analog Interface Signal	Time Description	Min	Max	Time Accuracy	Remark
TX	TX_RAM_Ready → EN_TX_SYN	Time required for state transition	2us	33us	1us	Reg_ldo_ana_setup_time is configured to cfg+2us
	EN_TX_SYN → EN_TX		1us	16us	1us	Reg_tx_rfpll_setup_tim is configured to cfg+1

	EN_TX→EN_TXPA	RFPLL and TX Link Stabilization Time	1us	256us	1us	Reg_tx_ana_setup_time is configured to cfg+1
	EN_TXPA→ EN_PA_RAMP	PA Preparation Time	1us	64us	1us	Configure reg_ramp_dly_time_up
	EN_PA_RAMP→ PA_SETUP_DONE	PA RAMP Time	4us	32us	4us	Pa_ram_up_dly_sel, cfg*4+4
	PA_SETUP_DONE→ TX_EN_DIG	Time required for state transition	2us	2us		TX-FSM control
	Launch complete→ EN_PA_RAMP Lower	Beat Delay	3us	3us		Beat Delay
	EN_PA_RAMP→ EN_TXPA	RAMP DOWN Time	4us	32us	4us	Ramp_down_cfg, cfg*4+4
	EN_TXPA→EN_TX	Time required after disabling EN_TXPA	5us	130us	2us	Padly_cfg
	EN_TX→EN_TX_SYN	Time required after disabling EN_TX	1us	8us	1us	Rf_pll_close_cfg,cfg+1
	EN_TX_SYN→ TX_CLOSE_DONE	Time required for state transition	1us	8us	1us	Tx_ana_close_time, Notify mac tx to close
	EN_LDO_ANA_AON	Software Control				Software Control

State	Enable Digital-to-Analog Interface Signal	Time Description	Min	Max	Time Accuracy	Remark
RX	RX Start → EN_LDO_ANA_AON	Software Control				Software Control
	RX_RAM_Ready→ EN_RX_SYN	Time required for state transition	2us	33us	1us	Reg_ldo_ana_setup_time is configured to cfg+2
	EN_RX_SYN→EN_RX	EN_RX_SYN stabilization time	1us	253us	4us	Reg_rx_rfpll_setup_time is configured to cfg*4+1
	EN_RX→DIG_RX_Period	EN_RX stabilization time	1us	32us	1us	Reg_rx_ana_setup_time is configured to cfg+1
	DIG_RX_Period					Digital Receive Processing
	DIG_RX_Period→EN_RX					RX-FSM Control
	EN_RX→EN_RX_SYN	Time required after disabling EN_RX	1us	8us	1us	Ana_close_time,cfg+1
	EN_RX_SYN→ RX_ANA_CLOSE_DONE	Time required after disabling EN_RX_SYN	1us	8us	1us	rfpll_cose_time+LDO_close, cfg+1
	EN_LDO_ANA_AON→ Exit RX mode	Time required after disabling EN_LDO_RF				Software Control

4.5 PID

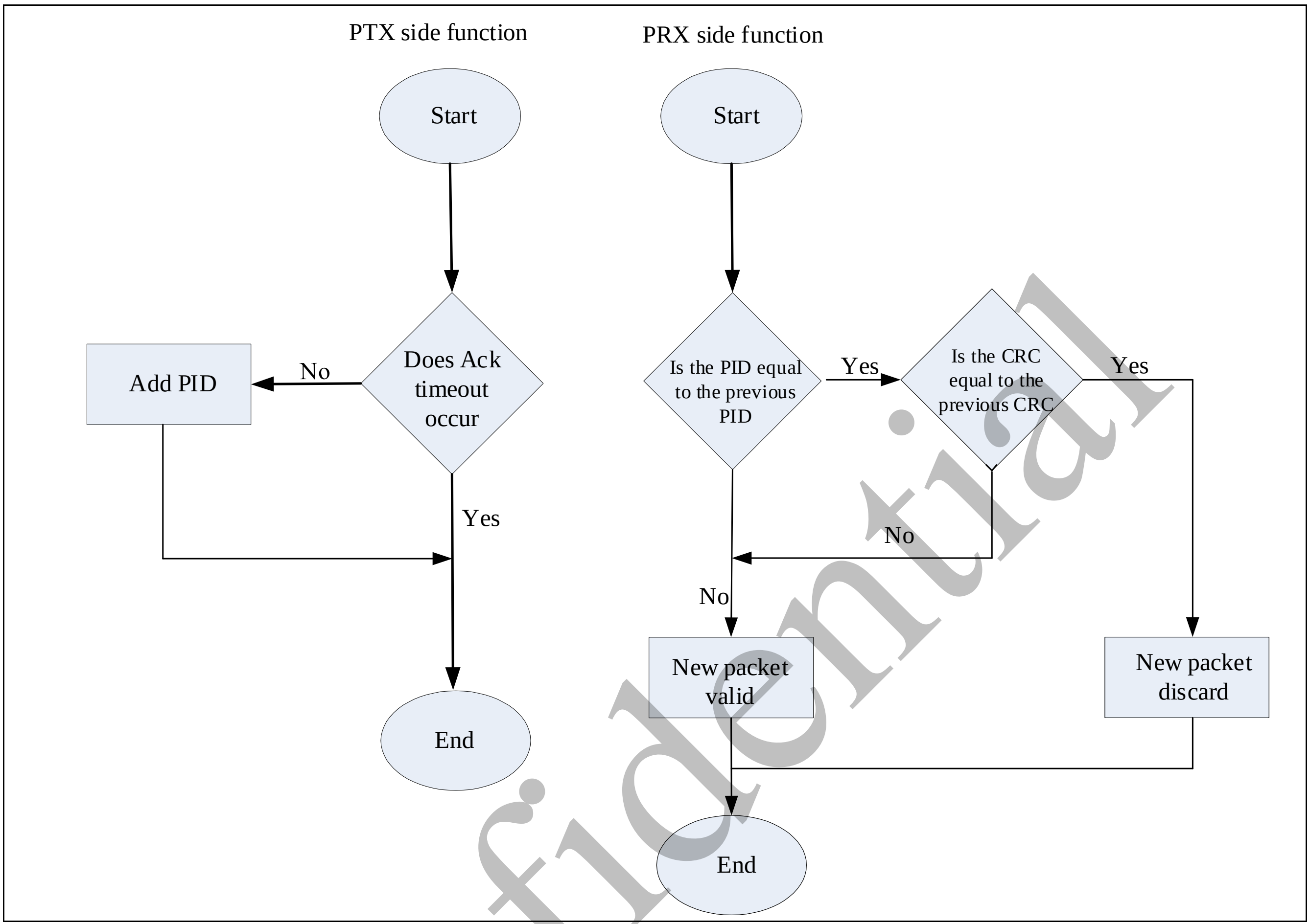


Figure 4-1 PID generation and detection

Each packet of data includes a two-bit PID (Packet Identifier Digital) to help the receiving side identify whether the data is a new packet or a resent packet, preventing multiple deposits of the same packet, and the PID is generated and detected as shown in Figure 4-1. The PID value is incremented by one when no timeout occurs for the ack packet at the sending side.PID can be maintained by the software at each transmit.

4.6 Register Description

4.6.1 PRI_R00

Register	Offset	R/W	Description	Reset Value
pri_r00	0x448	R/W		0x0

Bits	Descriptions	
[31:24]	TX_PAYLOAD_LENGTH	Send payload length.
[23:16]	RX_PAYLOAD_LENGTH	Receive payload length.
[15]	ACCADDR_CRC_DIS	0: CRC will include access address, header, payload 1: CRC will just include header, payload
[14]	CRC_SEL24	1: CRC24 0: Determined by the register crc_sel16
[13:12]	ADDR_BYTE_LENGTH	RX/TX address width, if the address width is set below 5 bytes, the address uses a low byte. 00: 2 bytes 01: 3 bytes 10: 4 bytes 11: 5 bytes
[11]	DPY_EN	In enhanced mode: By configuring this bit, the hardware can independently determine the length of the receiving packet, and there is no need to configure the receiving payload length rx_payload_length information on the software side. Normal mode is not supported. Note: Enhanced PTX devices must enable this bit when receiving ack with payload.
[10]	CRC_ENABLE	CRC enable bit 1: CRC enable 0: CRC disable
[9]	CRC_SEL16	CRC_SEL24 equal to 0, this bit is valid. 1: CRC16 0: CRC8
[8]	SCR_ENABLE	Scrambling function is enabled or not. 1: Whitening enable 0: Whitening disable
[7]	NRF_ENHANCE	0: NRF_ENH0 1: NRF_ENH1
[6]	ENHANCE	Enhanced Mode Configuration
[5]	BW_MODE	0: 1Mbps 1: 2Mbps
[4:3]	CHIP_MODE	Operating mode selection（In BLE mode, pri module clock gate） 0x: RSV 10: 297 11: NRF
[2]	Reserved	Reserved
[1]	TX_NOACK_EN	If configured to 1, the no-ack bit is 1 in enhanced mode, and no rx reply to ACK is required.
[0]	PRI_RX	RX/TX control bit

		1: PRX 0: PTX

4.6.2 PRI_R01

Register	Offset	R/W	Description	Reset Value
pri_r01	0x44c	R/W		0x00000000

Bits	Descriptions	
[31]	OM_FRAME_EN	Set to 1 to enable the 6200 frame format. Guard={addr0,addr0}
[30:29]	RX_PID	Read only The rx pid of current package
[28:27]	TX_PID	Read only The tx pid of current package
[26]	PRI_RX_LNGTH_ERR_FLAG	Read only Received packet length exceeds set value, error flag
[25]	PRI_ACC_ADDR_ERR_FLAG	Read only Acc addr error flag in spread spectrum mode
[24]	OM_SCR_MODE_EN	OM whitening mode enabled
[23]	ACC_ADDR_ERR_MASK	Acc addr err interrupt mask in spread spectrum mode
[22]	PRI_RX_LNGTH_ERR_IRQ_MASK	Maximum received packet length err irq mask
[21]	ENDIAN	Active @nrf mode 0: Little endian 1: Big endian Note: when nrf_enh0 and xn297 mode should set endian to 1
[20]	TX_DONE_IER	Configured to 0, no tx_done_irq is generated
[19]	RX_DONE_IER	Configured to 0, no rx_done_irq is generated
[18]	RX_GOON	When crc error, rx will go on when this bit is set 0: RX will stop when crc error 1: RX will go on when crc error
[17]	PRI_RST	Pri softwre reset control 0: Do not reset pri module 1: Reset pri module
[16]	PRI_EXIT_RX	Software exit rx mode control 0: Software does not exit rx mode 1: Software exit rx mode Flag bit: rx_timeout_irq_flag
[15]	TX_DONE_IRQ_FLAG	Tx end flag bit RO (end of master state machine when valid)
[14]	RX_DONE_IRQ_FLAG	Rx end flag bit RO (end of master state machine when valid)
[13]	RX_PID_ERR_IRQ_FLAG	Rx pid error interrupt flag bit RO
[12]	RX_CRC_ERR_IRQ_FLAG	Rx crc error interrupt flag bit RO
[11]	RX_TIMEOUT_IRQ_FLAG	Rx timeout flag bit (including rx mode timeout, ack timeout, software exit rx) RO
[10]	RETRANS_FLAG	Retransmit Flag RO
[9]	RX_IRQ_FLAG	Rx interrupt flag bit RO (when valid, sub-state machine ends, main state machine does not end)
[8]	IRQ_CLEAR_EN	Interrupt clear flag, when configured to 1, clear all current interrupt information

[7]	RX_CRC_ERR_IRQ_MASK	Configured to 1, the total interrupt pri_irq will not have that interrupt message
[6]	RX_PID_ERR_IRQ_MASK	Configure to 1, no rx_pid_err_irq will be generated
[5]	TX_RXACK_OUTTIME_IRQ_MASK	Configured to 1, this interrupt will not be generated
[4]	ACC_MATCH_IRQ_MASK	Acc_addr_irq mask configured to 1, this interrupt will not be generated
[3]	RX_IRQ_MASK	Configured to 1, this interrupt will not be generated
[2]	OM_FRAME_EN	Set to 1 to enable the 6200 frame format. Guard={~addr0,addr0}
[1:0]	MULTI_RX_ACC_ADDR	The pipe number of current receive packet

4.6.3 PRI_R02

Register	Offset	R/W	Description	Reset Value
pri_r02	0x450	R/W		0x0

Bits	Descriptions	
[31:16]	RX_WAIT_TIME	The maximum time to wait for ACK after PTX is converted to RX mode, beyond which the transmission is considered to have failed, an interrupt is generated, and the software resends. Normal mode: The waiting time after entering the rx state
[15]	RX_WAIT_TIME_OUT_EN	Configured to 1, normal mode rx wait timeout enable
[14:0]	TRX_TRANS_WAIT_TIME	Enhanced mode: the waiting time of PTX to RX, or RX to TX

4.6.4 PRI_R03

Register	Offset	R/W	Description	Reset Value
pri_r03	0x454	R/W		0x0

Bits	Descriptions	
[31:0]	RAR[31:0]	rx_addr LSB 32bits

4.6.5 PRI_R04

Register	Offset	R/W	Description	Reset Value
pri_r04	0x458	R/W		0x0

Bits	Descriptions	
[31:24]	RX_HEADER1	The value of rx header, active when hdr_len_exist is 1, read only When HDR_LEN_NUMB: 11: RX_header1 is active Others: No use
[23:16]	RX_HEADER0	The value of rx header, active when hdr_len_exist is 1, read only When Hdr_len_numb: 00: No use 01: No use 10: RX_header0 is active 11: RX_header0 is active Bit[0]: demod_ack in enhance frame
[15]	PRI_RX_FEC	Enable RX spread spectrum
[14]	PRI_TX_FEC	Enable TX spread spectrum
[13:12]	PRI_CI_MODE	Select spread spectrum mode 2’b00-S8 2’b01-S2 2’b1x-reserved
[11]	NORMAL_M1	Normal_m1 mode, cannot be used in conjunction with enhance mode 0: Disable 1: Enable, no need to judge other flag bits when trx switching, just meet the switching time
[10:9]	HDR_LEN_NUMB	Payload length’s locate of frame, active when pld_len_exist is 1 00: No use, must not set this value 01: Length is exist and is the first byte of frame; no header 10: Length is exist and is the second byte of frame; header is the first byte of frame 11: Length is exist and is the third byte of frame; header is the first two bytes of frame
[8]	HDR_LEN_EXIST	Header and Length are exist 0: Disable, payload length is decided by pri_r00[23:16] 1: Exist, header and length are exist at frame
[7:0]	RAR[39:32]	RX_addr MSB 8bits

4.6.6 PRI_R05

Register	Offset	R/W	Description	Reset Value
pri_r05	0x45c	R/W		0x0

Bits	Descriptions	
[31:0]	TAR[31:0]	tx_addr LSB 32bits

4.6.7 PRI_R06

Register	Offset	R/W	Description	Reset Value
pri_r06	0x460	R/W		0x00007f00

Bits	Descriptions	
[31:24]	TX_HEADER1	The value of rx header, active when hdr_len_exist is 1, read only When Hdr_len_numb: 00: No use 01: No use 10: No use 11: TX_header1 is active
[23:16]	TX_HEADER0	The value of rx header, active when hdr_len_exist is 1, read only When Hdr_len_numb: 00: No use 01: No use 10: TX_header0 is active 11: TX_header0 is active
[15]	PRI_RX_MAX_CTRL_EN	The max length of rx paylad control 1: Enable 0: Disable Valid in dynamic payload length mode. When enabled, the receive packet length threshold is controlled by the rx_payload_length. The received payload length ≤ rx_payload_length: Can be received normally. The received payload length > rx_payload_length: Only rx_payload_length length of data is received and a length_error interrupt is generated.
[14:8]	WHITENING_INIT	The initial value of whitening(default: 0x7f)
[7:0]	TAR[39:32]	TX_addr MSB 8bits

4.6.8 PRI_R07

Register	Offset	R/W	Description	Reset Value
pri_r07	0x464	R/W		0x40_0000

Bits	Descriptions	
[31]	Reserved	Reserved
[30:23]	Rcvd_payload_length	RX received packet data length, read only
[22:17]	RX_RAM_START_ADDR	RX data ram start address: 0x400
[16]	RX_RAM_READY	RX data ram ready,must be 1 before rx receive
[15:7]	Reserved	Reserved
[6:1]	TX_RAM_START_ADDR	TX data ram start address: 0x000
[0]	TX_RAM_READY	TX data ram ready,must be 1 before tx send

4.6.9 PRI_R08

Register	Offset	R/W	Description	Reset Value
pri_r08	0x468	R/W		0x0

Bits	Descriptions	
[31:0]	ADDR1[31:0]	Multi-channel RX mode addr[39:8]

4.6.10 PRI_R09

Register	Offset	R/W	Description	Reset Value
pri_r09	0x46c	R/W		0x8810000

Bits	Descriptions	
[31:28]	Reserved	Reserved
[27]	Fec_dec_timeout_rst_en	S2s8 mode mis-triggered timeout automatically reset PHY enable
[26:23]	Pri_tst_select_1	Debug signal 1 position, valid value: 0–9
[22:19]	Pri_tst_select_0	Debug signal 0 position, valid value: 0–9
[18]	ADDR_EN[2]	Enable multi-channel RX mode addr2
[17]	ADDR_EN[1]	Enable multi-channel RX mode addr1
[16]	ADDR_EN[0]	Enable multi-channel RX mode addr0, enabled by default
[15:8]	P2_Addr_lsb[7:0]	Multi-channel RX mode addr2 [7:0] LSB
[7:0]	P1_Addr_lsb[7:0]	Multi-channel RX mode addr1 [7:0] LSB

4.6.11 PRI_R10

Register	Offset	R/W	Description	Reset Value
pri_r10	0x470	R/W		0x2010_0080

Bits	Descriptions	
[31]	Ndc_pid_order	Control order of pid & noack 0: Pid first, then noack 1: Noack first, then pid
[30]	ndc_ack_sel	0: 1-ack, 0-noack 1: 1-noack, 0-ack
[29:26]	Tx_on_dly_symb	Tx_on delay time, symbol count, valid value: 0~10
[25]	Rx_contious_mode	Continuous reception mode, without disabling the PLL Require enabling rx goon
[24]	RX_DATA_REVERSE_EN	Enable RX data inversion
[23]	PRE_SYNC_12B_EN	Enable 12bit presync
[22]	PRE_SYNC_8B_EN	Enable 8bit presync
[21]	PRE_SYNC_4B_EN	Enable 4bit presync
[20]	PRE_SYNC_ENABLE	Enable Pre sync output
[19]	250K_MODE_EN	Enable 250k rate mode
[18]	BOE_PRE_SEL	Boe preamble switch 0: 0x55 1: 0xAA
[17]	LQI_EN_FOR_SYNC	LQI sync function enable control 0: Disable 1: Enable Note: just enable when addr is 2 byte
[16:15]	RX_PID_MANUAL	Rx pid configured externally by user
[14:13]	TX_PID_MANUAL	Tx pid configured externally by user
[12]	PRE_2BYTE_MODE	preamble extended control 0: Disable 1: Enable Note: when enable 297-preamble repeat twice. When 250K mode is used, it must set to 1.
[11]	TX_DATA_REVERSE_EN	Enable TX data bitstream inversion
[10:9]	SCR_INI[8:7]	Scr init initial value [8:7]
[8:6]	PRE_LEN[2:0]	Preamble length control 0:2B 1:3B 2:4B 3:6B 4:8B 5:12B 6:16B 7:24B Need to set 0x504[12] to 1, Pre_2byte_mode

[5]	BOE_MODE_EN	Enable Boe crc scr polynomial
[4:2]	ADDR_ERR_THR[2:0]	The allowed number of error bit during access addr matching 0: Exactly matched
[1]	PID_MANUAL_EN	Configured to 1, allow the tx_pid_manual and rx_pid_manual configured externally by user.
[0]	ADDR_SCR_DIS	The whitening of addr control 0: Enabled, whitening includes addr, header, payload 1: Disabled, whitening includes header, payload

4.6.12 PRI_R11

Register	Offset	R/W	Description	Reset Value
pri_r11	0x474	R/W		0xF800_0000

Bits	Descriptions	
[31]	Ack_addr_auto_set_en	Auto set ack TX addr in enhance mode prx device, enable multi channel
[30]	Acc_addr_err_retrans_en	Acc addr err retrans enable
[29]	Length_err_retrans_en	Length err retrans enable
[28]	Crc_err_retrans_en	Crc err retrans enable
[27]	Timeout_retrans_en	Timeout err retrans enable
[26:24]	Retrains_max_times	Maximum retransmission count: 0 indicates no retransmission; at least one bit (bit 26–29) must be set for retransmission
[23:9]	Retans_wait_time	Retransmission wait time, in microseconds
[8]	Acc_match_irq_flag	Acc match interrupt flag RO
[7]	Pri_rx_irq_flag	RX interrupt flag RO (When active, sub-state machine terminates; main state machine remains active)
[6]	Pri_pid_err_irq_flag	RX PID error interrupt flag RO
[5]	Pri_timeout_irq_flag	Rx timeout flag (includes RX mode timeout, ACK timeout, software exit from Rx) RO
[4]	Pri_crc_err_irq_flag	RX CRC error interrupt flag RO
[3]	Pri_rx_done_irq_flag	RX end flag RO (When active, the main state machine terminates.)
[2]	Pri_tx_done_irq_flag	TX end flag RO (When active, the main state machine terminates.)
[1]	Pri_length_err_irq_flag	RO Received packet length exceeds set value, error flag signal
[0]	Pri_acc_addr_err_irq_flag	RO Spread spectrum mode ACC address error flag signal

4.6.13 PRI_R12

Register	Offset	R/W	Description	Reset Value
pri_r12	0x478	R/W		0x0

Bits	Descriptions	
[31:0]	Pri_rx_wl_adva_lsb	BLE RX Mode Whitelist Configuration: Low 32-bit. This whitelist corresponds to a segment within the Payload (configurable) and operates in conjunction with the pld_start_byte and wl_match_mode registers.

4.6.14 PRI_R13

Register	Offset	R/W	Description	Reset Value
pri_r13	0x47C	R/W		0x0

Bits	Descriptions	
[31:27]	Reserved	Reserved
[26:21]	Pri_rx_pld_start_byte	BLE RX mode is based on the filtering mechanism of BLE payload. Configure the starting byte for whitelist filtering. The payload range for broadcast packets is 9 to 39 bytes, with the first 8 bytes comprising the Header (2 bytes) and AdvA (6 bytes). This feature operates in conjunction with the wl_match_mode and wl_adva registers. The RX_GOON must be enabled.
[20:18]	Pri_rx_wl_match_mode	BLE RX Mode Payload-Based Whitelist Filtering Mode Selection 000: No filtering, report all 001: Report only if wl_adva[7:0] matches 010: Report only if wl_adva[15:0] matches 011: Report only if wl_adva[23:0] matches 100: Report only if wl_adva[31:0] matches 101: Report only if wl_adva[39:0] matches 110: Report only if all wl_adva[47:0] bits match 111: Same as 000, no filtering, report all
[17:16]	Pri_rx_blelen_match_mode	BLE RX Mode Filtering Mechanism Based on Header.Length 00: No filtering on length 01: Only continue receiving packets if the received length equals payload_length 10: Only continue receiving packets if the received length is greater than payload_length 11: Only continue receiving packets if the received length is less than payload_length
[15:0]	Pri_rx_wl_adva_msb	BLE RX Mode Whitelist Configuration High 16-bit This whitelist corresponds to a segment within the Payload (configurable) and operates in conjunction with the pld_start_byte and wl_match_mode registers.

4.7 Instructions for use

4.7.1 Normal mode

4.7.1.1 PTX mode

1. **Initialization**
   a. Clock initialization, select XTH as clock source, enable BLE_32M/BLE_32K clock.
   b. Interrupt enabled, configure interrupt service function.
   c. Memory initialized, emngr_init()；
   d. RF LDO enabled, ana_init()；
   e. LL common initialized, llhwc_cmn_init()；
   f. Phy configuration：PHY2_PHY_DRV_SEQ_TX_STRT/END_ADDR；
      PHY2_PHY_DRV_SEQ_RX_STRT/END_ADDR
2. **TX initialization：**
   a. Prepare tx data , address: 0x50028000+PRI_R07_TX_RAM_START_ADDR
   b. Configure PRI_R00_CHIP_MODE: 2, XN297L; 3,NRF
   c. Configure PRI_R00_ENHANCE: 0, Normal mode
   d. Configure operating bandwidth PRI_R00_BW_MODE: 0, 1M; 1, 2M
   e. Configure the device address width PRI_R00_ADDR_BYTE_LEN: 0,2Bytes; 1,3bytes; 2, 4bytes; 3 5bytes
   f. Configure the device address high byte PRI_R06_TAR_MSB, only used in 5bytes address width
   g. Configure the device address low bytes PRI_R05_TAR_LSB
   h. Configure the TX packet start relative address PRI_R07_TX_RAM_START_ADDR, base address 0x50028000
   i. Configure the TX packet length PRI_R00_TX_PAYLOAD_LEN, package length refers to the protocol requirements
   j. Configure whether do the TX packet whitening PRI_R00_SCR_EN: 0, De-whitening;1, Whitening
   k. Configure whether to enable CRC PRI_R00_CRC_EN, if CRC is enabled, configure the length PRI_R00_CRC_SEL16: 0, 8bits; 1,16bits
3. **TX starting**
   a. Configure tx mode PRI_R00_PRI_RX: 0

b. Configure tx active,RAM PRI_R07_TX_RAM_READY

4.7.1.2 PRX mode

1. **Initialization：same as 4.7.1.1**
2. **RX initialization：Note that NRF mode should not configure PRI_R00_SCR_EN**
   a. Configure PRI_R00_CHIP_MODE: 2, XN297L; 3,NRF
   b. Configure PRI_R00_ENHANCE: 0, Normal mode
   c. Configure operating bandwidth PRI_R00_BW_MODE: 0, 1M； 1, 2M
   d. Configure the device address width PRI_R00_ADDR_BYTE_LEN: 0,2Bytes; 1,3bytes;2, 4bytes; 3 5bytes
   e. Configure the device address high byte PRI_R04_RAR_MSB, only used in 5bytes address width
   f. Configure the device address low bytes PRI_R03_RAR_LSB
   g. Configure the TX packet start relative address PRI_R07_RX_RAM_START_ADDR, base address 0x50028000
   h. Configure the TX packet length PRI_R00_RX_PAYLOAD_LEN, the packet length same as the PTX
   i. Configure whether do the TX packet whitening PRI_R00_SCR_EN: same as the PTX
   j. Configure whether to enable CRC PRI_R00_CRC_EN, if CRC is enabled, configure the length PRI_R00_CRC_SEL16: 0, 8bits; 1,16bits
3. **RX starting**
   a. Configure rx mode, PRI_R00_PRI_RX: 1
   b. Packet receive timeout enable PRI_R02_RX_WAIT_TIME_OUT_EN: 0, Enable; 1, Disable. If enabled, PRI_R02_RX_WAIT_TIME configures the timeout(Unit: us).
   c. Configure rx active, RAM PRI_R07_RX_RAM_READY

4.7.2 Enhanced mode

4.7.2.1 PTX mode

1. **Initialization: same as 4.7.1.1**
2. **TX initialization：Note that NRF mode should not configure PRI_R00_SCR_EN**
   a~k: Configured as 4.7.1.1(except for step c PRI_R00_ENHANCE configured to 1)
   l. NRF Mode Model Selection Register PRI_R00[7]: 0,nrf_enh0;1,nrf_enh1
   XN297L mode does not need this configuration.

m. Configure the peer device address high byte PRI_R04_RAR_MSB, only used in 5bytes address.
n. Configure the peer device address low bytes PRI_R03_RAR_LSB.
o. Configure whether to receive ACK, PRI_R00_TX_NOACK_EN: 0, Do not receive；1, Receive. If received, perform the following steps p~t.
p. Configure RX packet SRAM ready PRI_R07_RX_RAM_READY.
q. Configure dynamic parsing of RX packet length PRI_R00_DPY_EN, must be enabled.
r. Configure the RX packet start address PRI_R07_RX_RAM_START_ADDR
s. Configure the TRX transition time PRI_R02_TRX_TRANS_WAIT_TIME
t. Configure timeout enable PRI_R02_RX_WAIT_TIME_OUT_EN and RX packet timeout PRI_R02_RX_WAIT_TIME

3. TX starting

a. Configure tx mode PRI_R00_PRI_RX: 0
b. Configure tx active,RAM PRI_R07_TX_RAM_READY

4.7.2.2 PRX mode

1. Initialization：same as 4.7.1.1

2. RX initialization：Note that NRF mode should not configure PRI_R00_SCR_EN

a~j: Configured as 4.7.1.2 (except for step b PRI_R00[7] configured to 1)

k. NRF enhanced mode selection PRI_R00[7], 0,nrf_enh0; 1,nrf_enh1
XN297L mode does not need this configuration.
l. Configure the TRX transition time PRI_R02_TRX_TRANS_WAIT_TIME, the sum of this time and the PHY startup shutdown time is the actual TRX transition time.
m. Configure the peer device address high byte PRI_R06_TAR_MSB, only used in 5bytes address.
n. Configure the peer device address low bytes PRI_R05_TAR_LSB
o. If ACK with payload, enable PRI_R00_RX_ACK_PAYLOAD_EN, and perform the following step p. Additional note: Step p can be configured before receiving the packet or in the Interrupt Service Program after receiving the packet.
p. Fill TX SRAM TX packet data, configure the start address PRI_R07_TX_RAM_START_ADDR and the length PRI_R00_TX_PAYLOAD_LEN.

3. RX starting

a. Configure rx mode, PRI_R00_PRI_RX: 1

b. Packet receive timeout enable PRI_R02_RX_WAIT_TIME_OUT_EN: 0, Enable; 1, Disable. If enabled, PRI_R02_RX_WAIT_TIME configures the timeout(Unit: us).

c. Configure rx active, RAM PRI_R07_RX_RAM_READY

4.7.3 Normal_m1 mode

4.7.3.1 PTX mode

1. **Initialization: same as 4.7.1.1**
2. **TX initialization**

   a~k: Configured as 4.7.1.1(except for step c PRI_R01_NORMAL_M1 configured to 1)

   l. Configure the peer device address high byte PRI_R04_RAR_MSB, only used in 5bytes address

   m. Configure the peer device address low bytes PRI_R03_RAR_LSB

   n. Configure RX packet SRAM ready PRI_R07_RX_RAM_READY

   o. Configure the RX packet start address PRI_R07_RX_RAM_START_ADDR

   p. Configure the TRX transition time PRI_R02_TRX_TRANS_WAIT_TIME

   q. Configure timeout enable PRI_R02_RX_WAIT_TIME_OUT_EN and RX packet timeout PRI_R02_RX_WAIT_TIME
3. **TX starting**

   a. Configure tx mode PRI_R00_PRI_RX: 0

   b. Configure tx active, RAM PRI_R07_TX_RAM_READY

4.7.3.2 PRX mode

1. **Initialization：same as 4.7.1.1**

   a~j. Configured as 4.7.1.2 (except for step b PRI_R01_NORMAL_M1 configured to 1 also)

   k. Configure the TRX transition time PRI_R02_TRX_TRANS_WAIT_TIME, the sum of this time and the PHY startup shutdown time is the actual TRX transition time.

   l. Configure the peer device address high byte PRI_R06_TAR_MSB, only used in 5bytes address

   m. Configure the peer device address low bytes PRI_R05_TAR_LSB

   n. Fill TX SRAM TX packet data, configure the start address PRI_R07_TX_RAM_START_ADDR and the length PRI_R00_TX_PAYLOAD_LEN
2. **RX starting**

   a. Configure rx mode, PRI_R00_PRI_RX: 1

b. Packet receive timeout enable PRI_R02_RX_WAIT_TIME_OUT_EN: 0, Enable; 1, Disable. If enabled, PRI_R02_RX_WAIT_TIME configures the timeout(Unit: us).
c. Configure rx active, RAM PRI_R07_RX_RAM_READY

4.7.4 250k mode

4.7.4.1 PTX mode

1. **Initialization: same as 4.7.1.1**
2. **TX initialization**

a. Prepare tx data , address: 0x50028000+PRI_R07_TX_RAM_START_ADDR
b. Configure PRI_R00_CHIP_MODE: 3,NRF
c. Configure PRI_R00_ENHANCE: 0, Normal mode
d. Configure operating bandwidth PRI_R00_BW_MODE：0, 1M
e. SRAM populate data, address: 0x50028000+PRI_R07_TX_RAM_START_ADDR
f. Configure PRI_R00_CHIP_MODE: 3,NRF
g. Configure PRI_R00_ENHANCE: 0, Normal mode
h. Configure operating bandwidth PRI_R00_BW_MODE：0, 1M；
i. Configure the device address width PRI_R00_ADDR_BYTE_LEN：0,2Bytes, 1,3bytes;2, 4bytes; 3 5bytes
j. Configure the device address high byte PRI_R06_TAR_MSB, only used in 5bytes address width
k. Configure the device address low bytes PRI_R05_TAR_LSB
l. Configure the rate mode PRI_R11_250K_MODE_EN: 1, enable 250k rate mode
m. Configure the TX packet start relative address PRI_R07_TX_RAM_START_ADDR, base address 0x50028000
n. Configure the TX packet length PRI_R00_TX_PAYLOAD_LEN, package length refers to the protocol requirements
o. Configure whether do the TX packet whitening PRI_R00_SCR_EN：0, De-whitening;1, Whitening
p. Configure whether to enable CRC PRI_R00_CRC_EN, if CRC is enabled, configure the length PRI_R00_CRC_SEL16: 0, 8bits; 1,16bits

4.7.4.2 PRX mode

1. **Initialization: same as 4.7.1.1**

2. **RX initialization：**
   a. Configure PRI_R00_CHIP_MODE: 3, NRF
   b. Configure PRI_R00_ENHANCE: 0, Normal mode
   c. Configure operating bandwidth PRI_R00_BW_MODE: 0, 1M；
   d. Configure the device address width PRI_R00_ADDR_BYTE_LEN: 0,2Bytes, 1,3bytes;2, 4bytes; 3 5bytes
   e. Configure the device address high byte PRI_R04_RAR_MSB, only used in 5bytes address width
   f. Configure the device address low bytes PRI_R03_RAR_LSB
   g. Configure the rate mode PRI_R11_250K_MODE_EN: 1, enable 250k rate mode
   h. Configure the TX packet start relative address PRI_R07_RX_RAM_START_ADDR, base address 0x50028000
   i. Configure the TX packet length PRI_R00_RX_PAYLOAD_LEN, the packet length same as the PTX
   j. Configure whether do the TX packet whitening PRI_R00_SCR_EN: same as the PTX
   k. Configure whether to enable CRC PRI_R00_CRC_EN, if CRC is enabled, configure the length PRI_R00_CRC_SEL16: 0, 8bits; 1,16bits
3. **RX starting**
   a. Configure rx mode, PRI_R00_PRI_RX: 1
   b. Packet receive timeout enable PRI_R02_RX_WAIT_TIME_OUT_EN: 0, Enable; 1, Disable. If enabled, PRI_R02_RX_WAIT_TIME configures the timeout(Unit: us).
   c. Configure rx active, RAM PRI_R07_RX_RAM_READY

4.7.5 Interrupt Service Program

The time point of interrupt reporting comes at the same time as the hardware post_tx and post_rx, so after entering the interrupt, the software is prohibited from reading or writing the registers of the PHY module during Tpost_txs, Tpost_rx. The above time can be calculated by software.

4.7.5.1 PTX mode

After clearing the corresponding interrupt, you need to pull down PRI_R07_TX_RAM_READY, For Enhanced mode, you also need to pull down PRI_R07_RX_RAM_READY.

4.7.5.2 PRX mode

After clearing the corresponding interrupt, you need to pull down PRI_R07_RX_RAM_READY.

Abbreviation

A	
APB	Advanced Peripheral Bus
ADC	Analog-to-Digital Converter
ALT	Alternate Function Select
ANAC	Analog Control
APROM	Application ROM
ATT	Attribute Protocol
B	
BAUDR	Baud Rate Select Register
BLDC	Brushless Direct Current Motor
Bluetooth LE	Bluetooth Low Energy
BOD	Brown-out Detector
BOM	Bill of Materials
C	
CFGx	Configuration Register for Channel x
ChEnReg	DMA Channel Enable Register
CMPDAT	Compare value
CPU	Central Processing Unit
CRC-32	Cyclic Redundancy Check
CTLx	Control Register for Channel x
CTRLR0	Control Register 0
CTRLR1	Control Register 1
D	
DARx	Destination Address Register for Channel x
Desiccant	A material for adsorbing moisture while remaining dry
DFBA	Data Flash Base Address Register
DLF	Divisor Latch Fraction Register
DLH	Divisor Latch High
DLL	Divisor Latch Low
DR	Data Register
DSTATARx	Destination Status Address Register for Channel x
DSTATx	Destination Status Register for Channel x
E	
ESD	Electro-Static discharge
F	
FCR	FIFO Control Register
FIFO	First Input First Output
FMC	Flash Memory Controller
Floor life	The longest time that the HiSilicon product is allowed to remain in the workshop (environ-ment <30 °C / 60% RH, before unpacking the moisture-proof packaging to reflow).
G	
GAP	Generic Access Profile
GATT	Generic Attribute Profile
GPIO	General-purpose I/O
H	
HCLK	Tstem Clock
HIC	Humidity Indicator Card
HID	Human Interface Device
HIRC	32MHz internal high speed osc-illator
HTX	Halt TX
HXT	32MHz external high speed crystal oscillator
I	
I2C	Inter-Integrated Circuit
IAP	In-Application-Programming
ICE	In-Circuit-Emulator
ICP	In-Circuit Programming
ICR	Interrupt Clear Register
IDR	Identification Register
IER	Interrupt Enable Register
IIR	Interrupt Identity Register
IMR	Interrupt Mask Register
IRSR	Interrupt Raw Status Registers
ISB	Instruction Synchronization Barrier
ISP	In-System Programming
ISR	Interrupt Service Routine
L	
L2CAP	Logical Link Control and Adaptation Protocol
LCR	Line Control Register
LCR_EXT	Line Extended Control Register
LDO	Low dropout regulator
LDROM	Loader ROM
LIRC	32 kHz internal low speed RC oscillator
LNA	Low Noise Amplifier
LSB	Least significant bit
LSR	Line Status Register
LstDstReg	Last Destination Transaction Request Register
LstSrcReg	Last Source Transaction Request Register

LVR	Low Voltage Reset
LXT	32.768 kHz external low speed crystal oscillator
M	
MBB	Moisture Barrier Bag
MCR	Modem Control Register
MCU	Micro Control Unit
MDM	Mobile Device Management
MFP	Multiple Function Port
MISO	Master input slave output
MOSI	Master output slave input
MSB	Most Significant Bit
MSL	Moisture sensitivity level, this product is on level 3
MSR	Modem Status Register
MSTICR	Multi-Master Interrupt Clear Register
N	
NMI	Non Maskable Interrupt
NVIC	Nested Vectored Interrupt Controller
P	
PA	Power Amplifier
PLL	Phase Locked Loop
POR	Power-on Reset
PWM	Pulse Width Modulation
R	
RAR	Receive Address Register
RBR	Receive Buffe Register
ReqDstReg	Destination Software Transaction Request Register
ReqSrcReg	Source Software Transaction Request Register
RF	Radio frequency
RFL	Receive FIFO Level
RISR	Raw Interrupt Status Register
ROM	Read-Only Memory
RSSI	Received Signal Strength Indication
RTOS	Real Time Operating System
RXFLR	Receive FIFO Level Register
RXFTLR	Receive FIFO Threshold Level Register
RXOICR	Receive FIFO Overflow Interrupt Clear Register
RXUICR	Receive FIFO Underflow Interrupt Clear Register
S	
SARx	Source Address Register for Channel x Register
SCB	System Control Block Registers
SCB	System Control Block Registers
SCR	Scratchpad Register
SER	Slave Enable Register
SglReqDstReg	Single Destination Transaction Request Register
SglReqSrcReg	Single Source Transaction Request Register
Shelf Life	Normal storage time after moisture-proof packaging
SM	Security Manager
SoC	System on chip
SPI	Serial Peripheral Interface
SPROM	Security protection ROM
SR	Status Register
SRAM	Static random access memory
SSTATARx	Source Status Address Register for Channel x
SSTATx	Source Status Register for Channel x
Statusint	Combined Interrupt Status Register
SWD	Serial Wire Debug
SysTick	System Timer
T	
TAR	Transmit Address Register
TFL	Transmit FIFO Level
THR	Transmit Holding Register
THRE	Transmitter Holding Register Empty
TMR	Timer Controller
TXFLR	Transmit FIFO Level Register
TXFTLR	Transmit FIFO Threshold Level Register
TXOICR	Transmit FIFO Overflow Interrupt Clear Register
U	
UART	Universal Asynchronous Receiver/Transmitters
USR	UART Status Register
W	
WDT	Watchdog Timer

Revision History

Version	Date	Content
1.0	Feb. 2026	Initial
1.1	Mar. 2026	Add SSOP24 package.

USING THIS DOCUMENT

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure the accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

TRADEMARKS

Other names mentioned in this document are trademarks/registered trademarks of their respective owners.

DISCLAIMER

All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

Contact Us

Shanghai Panchip Microelectronics Co., Ltd.

The 302 Room of Building D, No. 666 Shengxia Road

Zhangjiang Hi-Tech Park, Shanghai

People's Republic of China

021-50802371

http://www.panchip.com