Panchip Microelectronics Co., Ltd.

PAN1020

User Manual

BLE SoC Transceiver

Version: 1.8

Release date: Jun. 2022

Shanghai Panchip Microelectronics Co., Ltd.

Address: The Room 302 of Building D, No. 666 Shengxia Road
Zhangjiang Hi-Tech Park, Shanghai
People's Republic of China

Tel: 021-50802371

Website: http://www.panchip.com

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 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.

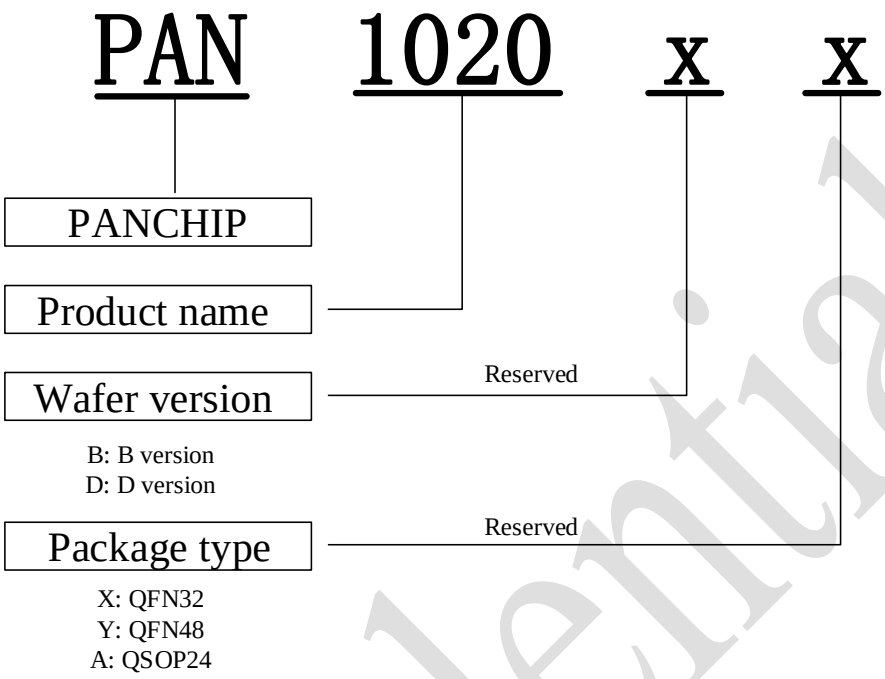
DISCLAIMER

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.

REVISION HISTORY

Version	Date	Content	Reference
1.0	Nov.2017	Initial	《PAN163CX Datasheet_V1.2-EN》
1.1	Jul. 2018	Add the pin description of QFN 48-PIN Complement the pakage dimension of the QFN 48-PIN	-
1.2	Dec.2018	Refresh parts of the parameter of "ELECTRICAL CHARACTERISTICS" Modify the description of "Feature" Add the charpter of "PRECAUTIONS", "STORAGE CONDITIONS" and "CONTACT US".	-
1.3	June.2019	Add the package QSOP24	-
1.4	Oct. 2019	Update the QSOP24 application reference circuit	-
1.5	Dec.2019	Update: 1. The interface "ADC_CLK_DIV_CFG_I<2:0>" 2. The electrical characteristics of ADC.	-
1.6	Jul.2020	Add "4.5 Digital Input/Output Characteristics".	-
1.7	Oct.2020	1) Add "Naming Rules" and "Product Series" 2) P1.1 of PAN1020DY has no actual function, P1.1 of PAN1020BY has actual function.	-
1.8	Jun.2022	Delete the "HIRC". Update the address. Update operating frequency up to 26MHz	-

Naming Rules



Product Series

Product series	Wafer version	Package	GPIO	UART	SPI	I2C	PWM	ADC
PAN1020BX	B	QFN32	25	2	3	2	One 8-channel	One 8-channel
PAN1020BY	B	QFN48	41	2	3	2	One 8-channel	One 8-channel
PAN1020BA	B	QSOP24	15	2	3	2	One 4-channel	One 4-channel
PAN1020DX	D	QFN32	25	2	3	2	One 8-channel	One 8-channel
PAN1020DY	D	QFN48	40	2	3	2	One 8-channel	One 8-channel
PAN1020DA	D	QSOP24	15	2	3	2	One 4-channel	One 4-channel

Contents

List of Figures

List of Tables

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
BLE	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
DmaCfgReg	DMA Configuration Register
DMACR	DMA Control Register
DmaIdReg	DMA ID Register
DMARDLR	DMA Receive Data Level Register
DMASA	DMA Software Acknowledge
DMATDLR	DMA Transmit Data Level Register
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 (environment <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
HTX	Halt TX
HXT	16 MHz 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
WWDT	Window Watchdog Timer

1 General Description

The PAN1020 integrated circuit has a fully integrated radio transceiver and baseband processor for Bluetooth Low Energy. It can be used as an application processor as well as a data pump in fully hosted systems.

The PAN1020 contains an embedded Flash memory for storing Bluetooth profiles as well as custom application code. The qualified BLE protocol stack, stored in a dedicated Flash area, as well as the customer application software run on the embedded MCU processor. Low leakage Retention RAM is used to store all the sensitive data and connection information while in Deep Sleep mode.

The BLE firmware includes the L2CAP service layer protocols, Security Manager (SM), Attribute Protocol (ATT), the Generic Attribute Profile (GATT) and the Generic Access Profile (GAP). Furthermore, application profiles such as Proximity, Health Thermometer, Heart Rate, Blood Pressure, Glucose and Human Interface Device (HID) are supported.

The MCU part of PAN1020 is the 32-bit microcontroller. It supports a wide range of applications from low-end, price sensitive designs to computing-intensive ones and provides advanced high-end features in economical products.

The PAN1020 has many high-performance peripheral functions, such as general purpose I/O port(25 GPIOs for QFN32 package, 40 GPIOs for PAN1020DY, 41 GPIOs for PAN1020BY and 15 GPIOs for QSOP24 package), three 32-bit timers, two UARTs, two group SPI interfaces, two I2C interfaces, one 16-bit PWM generators providing eight channels, an 8-channel 12-bit ADC, Watchdog Timer, Window Watchdog Timer, and a Brown-out Detector. All these peripherals have been incorporated into the PAN1020 to reduce component count, board space and system cost.

Additionally, the PAN1020 is equipped with ISP (In-System Programming) and ICP (In-Circuit Programming) functions, which allow the user to update the program memory without removing the chip from the actual end product. PAN1020 also supports In-Application-Programming (IAP) function, user switches the code executing without the chip reset after the embedded flash updated.

The PAN1020 can run up to 26 MHz and operate at a wide voltage range of 2.2V ~ 3.6V and temperature range of -40℃ ~ +85℃. For PAN1020, the embedded FLASH size up to 256 Kbytes and SRAM up to 16 Kbytes. It also offers size configurable Data Flash (shared with program flash), and configurable flash size for the ISP.

1.1 Key Features

- **RF**
  - 2.4GHz RF transceiver(Compatible with BLE4.2)
  - RX sensitivity: -90 dBm@1Mbps
  - Maximum received signal: 0 dBm
  - Programmable TX output power: 13 dBm(Maximum), 8 dBm(Typical)
  - Single wire antenna: no RF matching or RX/TX switching required
- **Core**
  - MCU core running up to 26 MHz
  - One 24-bit system timer

- Supports low power Idle mode
- A single-cycle 32-bit hardware multiplier
- Supports Serial Wire Debug (SWD) interface and two watchpoints/four breakpoints

- **Memory**
  - 256 KB Flash memory for program memory
  - 16 KB SRAM
- **Peripheral**
  - QFN32 package
    - 25 GPIOs
    - Two UARTs
    - Three SPIs
    - Two I2Cs
    - One 8-channel ADC
    - One 8-channel PWM0
  - QFN48 package
    - 41 GPIOs for PAN1020BY, 40 GPIOs for PAN1020DY
    - Two UARTs
    - Three SPIs
    - Two I2Cs
    - One 8-channel ADC
    - One 8-channel PWM0
  - QSOP24 package
    - 15 GPIOs
    - Two UARTs
    - Three SPIs
    - Two I2Cs
    - One 4-channel ADC
    - One 4-channel PWM0
  - Three channel 32-bit Timers (one 8-bit pre-scaler counter with 24-bit up-timer for each timer)
  - DMA up to 3 channels (one per source and destination pair)
  - Two UART devices with DMA
  - Two group SPI master and slave devices with DMA
  - Two I2C master and slaver devices with DMA
  - 12-Bit ADC with Eight Channels
  - One built-in 16-bit PWM generators with eight channels
  - One WDT with 18-bit up counter
  - One WWDT with 6-bit down counter value (CNTDAT) and 6-bit compare value (CMP-DAT)
- **Special features**
  - ISP (In-System Programming), ICP (In-Circuit Programming), and IAP (In Application Programming)
  - BOD (Brown-out Detector) threshold levels: 2.87V/2.72V/2.34V/2.06V

- 96-bit unique ID
- LVR (Low Voltage Reset) threshold voltage level: 1.7±0.1V

- **Package**
  - QFN32 package, 5 × 5 mm
  - QFN48 package, 6 × 6 mm
  - QSOP24 package, pin pitch = 0.635mm
- **DC/AC Charactaristics**
  - Operating Temperature: -40℃~85℃
  - Operating voltage: 2.2~3.6V
  - Reliability: ESD HBM pass ±2KV
  - Built-in LDO for wide operating voltage: 2.2V to 3.6V
    - ~2uA @ deep sleep mode, wake up by internal 32K oscillator

1.2 Typical Applications

- TV and STB remote control
- Wireless mouse and keyboard
- Wireless gamepads
- Smart home automation

2 Block Diagram

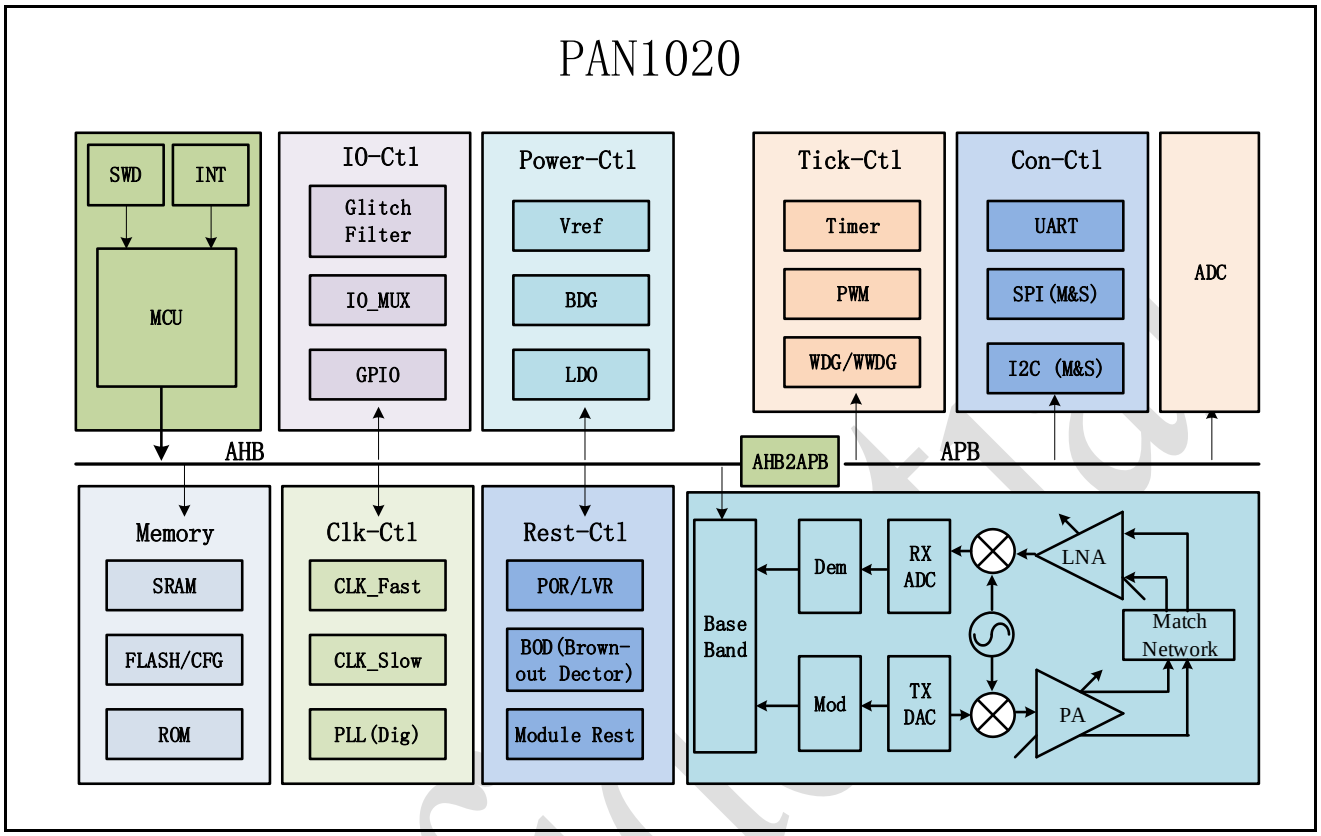


Figure 2-1 PAN1020 Block Diagram

3 Pin Information

3.1 QFN 32-PIN Diagram

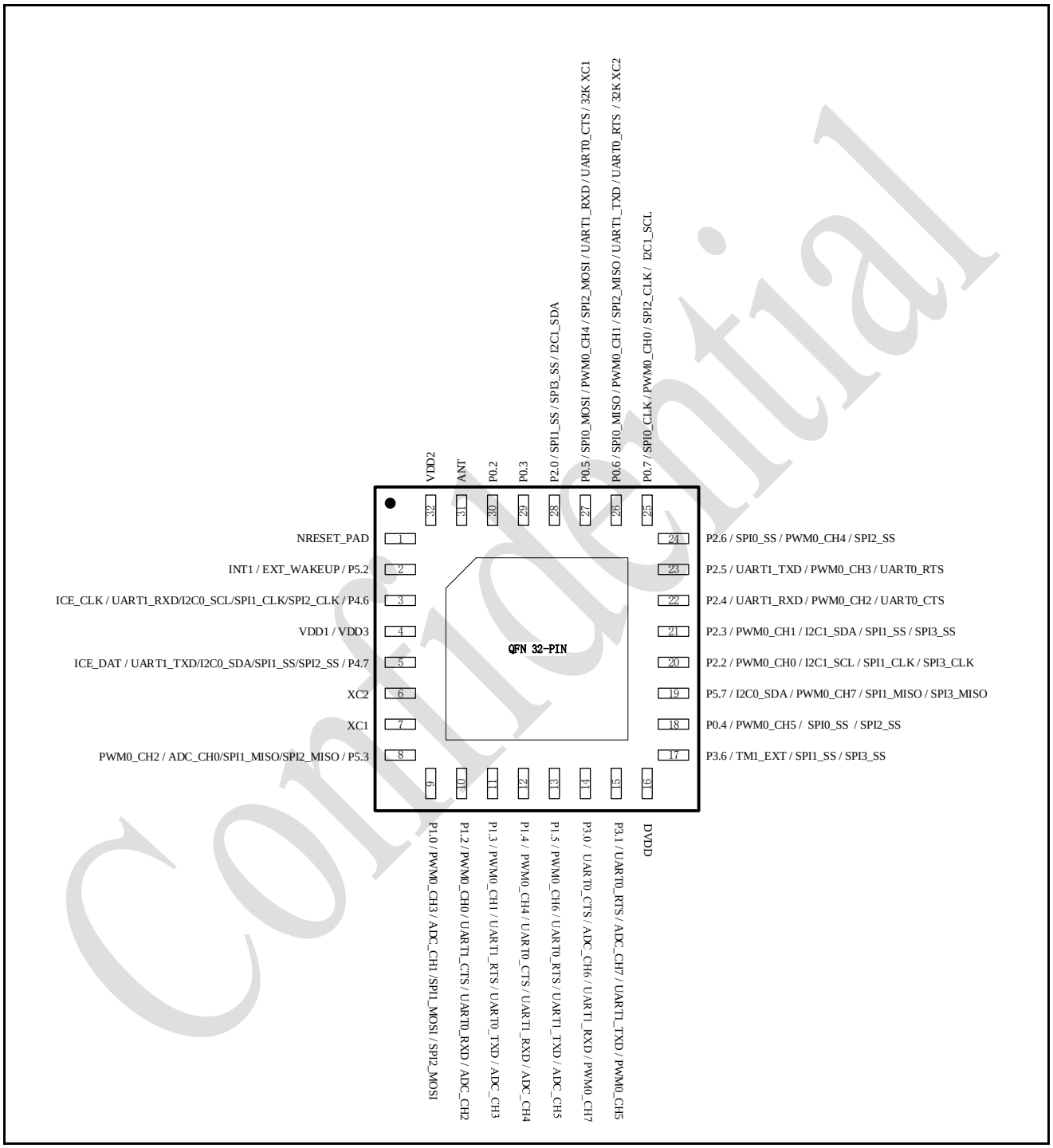


Figure 3-1 PAN1020 QFN 32-PIN Diagram

3.2 QFN 48-PIN Diagram

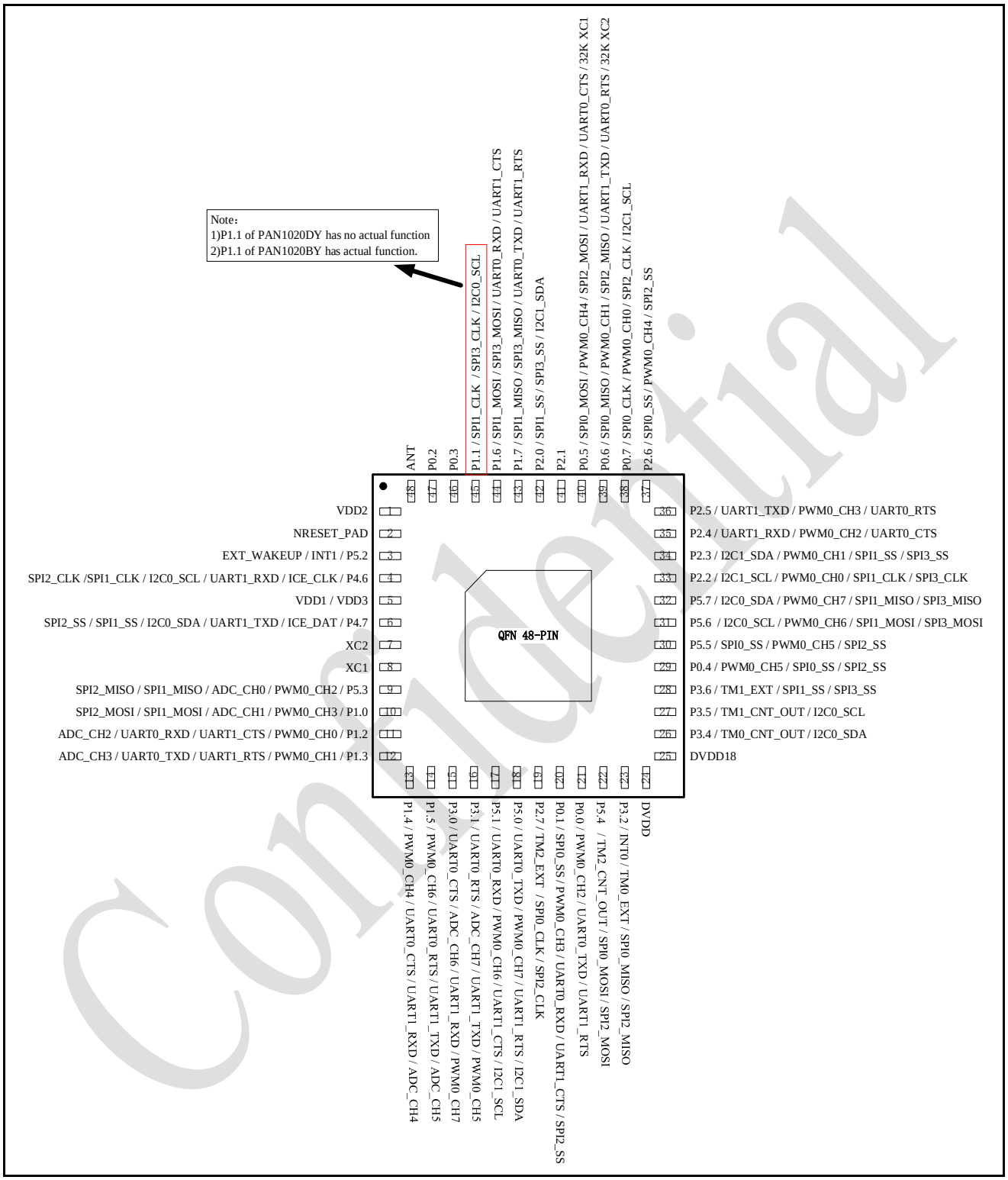


Figure 3-2 PAN1020 QFN 48-PIN Diagram

3.3 QSOP24-PIN Diagram

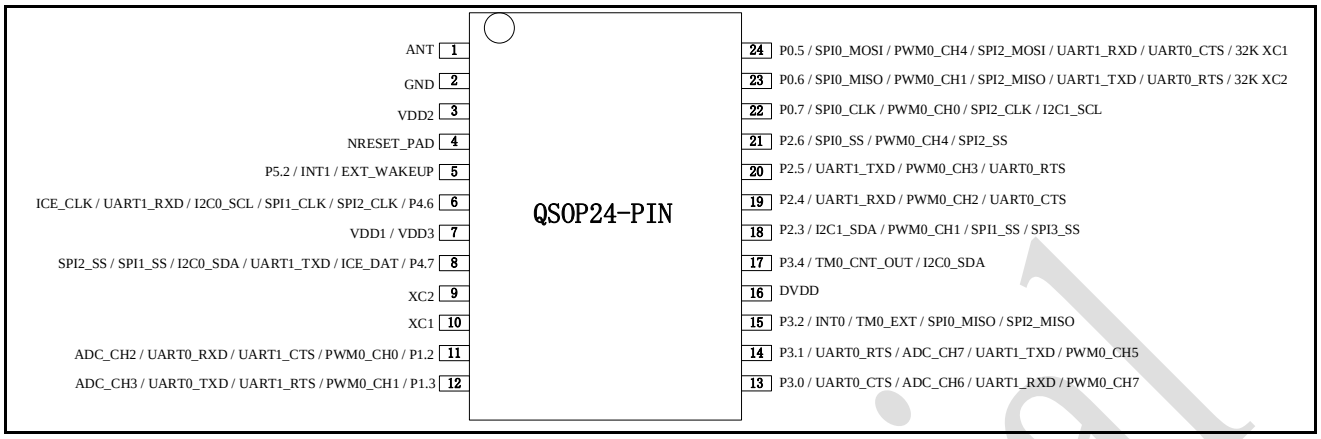


Figure 3-3 PAN1020 QSOP24-PIN Diagram

3.4 Pin Descriptions

Detail pin descriptions see Table 3-1.

IO is in high-impedance state by default. Pull-up is not enabled by default.

Table 3-1 PAN1020 Pin Descriptions

Pin Number			Pin Name	Pin Type	Description
QFN 32	QFN 48	QSOP24			
1	2	4	NRESET_PAD	I	Reset pin
2	3	5	P5.2	I/O	General purpose digital I/O pin
			INT1	I	External interrupt pin
			EXT_WAKEUP	I	External wake-up pin
3	4	6	P4.6	I/O	General purpose digital I/O pin
			ICE_CLK	I	ICE clk input pin
			UART1_RXD	I	UART1 RX pin
			I2C0_SCL	I/O	I2C0 CLK pin
			SPI1_CLK	O	SPI1 CLK pin
			SPI2_CLK	I	SPI2 CLK pin
4	5	7	VDD1	P	SoC power supply VDD1 pin
			VDD3	P	SoC power supply VDD3 pin
5	6	8	P4.7	I/O	General purpose digital I/O pin
			ICE_DAT	I	Debug and program data pin
			UART1_TXD	O	UART1 TX pin
			I2C0_SDA	I/O	I2C0 data pin

			SPI1_SS	O	SPI1 SS pin
			SPI2_SS	I	SPI2 SS pin
6	7	9	XC2	AO	Crystal pin2
7	8	10	XC1	AI	Crystal pin1
8	9	-	P5.3	I/O	General purpose digital I/O pin
			PWM0_CH2	O	PWM0 channel2 output pin
			ADC_CH0	AI	ADC channel0 analog input pin
			SPI1_MISO	I	SPI1 MISO pin
			SPI2_MISO	O	SPI2 MISO pin
9	10	-	P1.0	I/O	General purpose digital I/O pin
			PWM0_CH3	O	PWM0 channel3 output pin
			ADC_CH1	AI	ADC channel1 analog input pin
			SPI1_MOSI	O	SPI1 MOSI pin
			SPI2_MOSI	I	SPI2 MOSI pin
10	11	11	P1.2	I/O	General purpose digital I/O pin
			PWM0_CH0	O	PWM0 channel0 output pin
			UART1_CTS	I	UART1 CTS pin
			UART0_RXD	I	UART0 RX pin
			ADC_CH2	AI	ADC channel2 analog input pin
11	12	12	P1.3	I/O	General purpose digital I/O pin
			PWM0_CH1	O	PWM0 channel1 output pin
			UART1_RTS	O	UART1 RTS pin
			UART0_TXD	O	UART0 TX pin
			ADC_CH3	AI	ADC channel3 analog input pin
12	13	-	P1.4	I/O	General purpose digital I/O pin
			PWM0_CH4	O	PWM0 channel4 output pin
			UART0_CTS	I	UART0 CTS pin
			UART1_RXD	I	UART1 RX pin
			ADC_CH4	AI	ADC channel4 analog input pin
13	14	-	P1.5	I/O	General purpose digital I/O pin
			PWM0_CH6	O	PWM0 channel6 output pin
			UART0_RTS	O	UART0 RTS pin
			UART1_TXD	O	UART1 TX pin
			ADC_CH5	AI	ADC channel5 analog input pin

14	15	13	P3.0	I/O	General purpose digital I/O pin
			UART0_CTS	I	UART0 CTS pin
			ADC_CH6	AI	ADC channel6 analog input pin
			UART1_RXD	I	UART1 RX pin
			PWM0_CH7	O	PWM0 channel7 output pin
15	16	14	P3.1	I/O	General purpose digital I/O pin
			UART0_RTS	O	UART0 RTS pin
			ADC_CH7	AI	ADC channel7 analog input pin
			UART1_TXD	O	UART1 TX pin
			PWM0_CH5	O	PWM0 channel5 output pin
-	17	-	P5.1	I/O	General purpose digital I/O pin
			PWM0_CH6	O	PWM0 channel output pin
			UART0_RXD	I	UART0 RX pin
			UART1_CTS	I	UART1 CTS pin
			I2C1_SCL	I/O	I2C1 CLK pin
-	18	-	P5.0	I/O	General purpose digital I/O pin
			UART0_TXD	O	UART0 RX pin
			PWM0_CH7	O	PWM0 channel7 output pin
			UART1_RTS	O	UART1 RTS pin
			I2C1_SDA	I/O	I2C1 data pin
-	19	-	P2.7	I/O	General purpose digital I/O pin
			TM2_EXT	I	Timer2 external input pin
			SPI0_CLK	O	SPI0 CLK pin
			SPI2_CLK	I	SPI2 CLK pin
-	20	-	P0.1	I/O	General purpose digital I/O pin
			SPI0_SS	O	SPI0 SS pin
			PWM0_CH3	O	PWM0 channel3 output pin
			UART0_RXD	I	UART0 RX pin
			UART1_CTS	I	UART1 CTS pin
			SPI2_SS	I	SPI2 SS pin
-	21	-	P0.0	I/O	General purpose digital I/O pin
			PWM0_CH2	O	PWM0 channel2 output pin
			UART0_TXD	O	UART0 TX pin
			UART1_RTS	O	UART1 RTS pin

-	22	-	P5.4	I/O	General purpose digital I/O pin
			TM2_CNT_OUT	O	TM2_CNT output pin
			SPI0_MOSI	O	SPI0 MOSI pin
			SPI2_MOSI	I	SPI2 MOSI pin
-	23	15	P3.2	I/O	General purpose digital I/O pin
			INT0	I/O	External interrupt0
			TM0_EXT	I	Timer0 external input pin
			SPI0_MISO	I	SPI0 MISO pin
			SPI2_MISO	O	SPI2 MISO pin
16	24	16	DVDD	P	Core power supply, generaged by internal LDO
-	25	-	DVDD18	P	-
-	26	17	P3.4	I/O	General purpose digital I/O pin
			TM0_CNT_OUT	O	TM0_CNT output pin
			I2C0_SDA	I/O	I2C0 data pin
-	27	-	P3.5	I/O	General purpose digital I/O pin
			TM1_CNT_OUT	O	TM1_CNT output pin
			I2C0_SCL	I/O	I2C0 CLK pin
17	28	-	P3.6	I/O	General purpose digital I/O pin
			TM1_EXT	I	Timer1 external input pin
			SPI1_SS	O	SPI1 SS pin
			SPI3_SS	I	SPI3 SS pin
18	29	-	P0.4	I/O	General purpose digital I/O pin
			PWM0_CH5	O	PWM0 channel5 output pin
			SPI0_SS	O	SPI0 SS pin
			SPI2_SS	I	SPI2 SS pin
-	30	-	P5.5	I/O	General purpose digital I/O pin
			SPI0_SS	O	SPI0 SS pin
			PWM0_CH5	O	PWM0 channel5 output pin
			SPI2_SS	I	SPI2 SS pin
-	31	-	P5.6	I/O	General purpose digital I/O pin
			I2C0_SCL	I/O	I2C0 CLK pin
			PWM0_CH6	O	PWM0 channel6 output pin
			SPI1_MOSI	O	SPI1 MOSI pin
			SPI3_MOSI	I	SPI3 MOSI pin

19	32	-	P5.7	I/O	General purpose digital I/O pin
			I2C0_SDA	I/O	I2C0 data pin
			PWM0_CH7	O	PWM0 channel7 output pin
			SPI1_MISO	I	SPI1 MISO pin
			SPI3_MISO	O	SPI3 MISO pin
20	33	-	P2.2	I/O	General purpose digital I/O pin
			I2C1_SCL	I/O	I2C1 CLK pin
			PWM0_CH0	O	PWM0 channel0 output pin
			SPI1_CLK	O	SPI1 CLK pin
			SPI3_CLK	I	SPI3 CLK pin
21	34	18	P2.3	I/O	General purpose digital I/O pin
			I2C1_SDA	I/O	I2C1 data pin
			PWM0_CH1	O	PWM0 channel1 output pin
			SPI1_SS	O	SPI1 SS pin
			SPI3_SS	I	SPI3 SS pin
22	35	19	P2.4	I/O	General purpose digital I/O pin
			UART1_RXD	I	UART1 RX pin
			PWM0_CH2	O	PWM0 channel2 output pin
			UART0_CTS	I	UART0 CTS pin
23	36	20	P2.5	I/O	General purpose digital I/O pin
			UART1_TXD	O	UART1 TX pin
			PWM0_CH3	O	PWM0 channel3 output pin
			UART0_RTS	O	UART0 RTS pin
24	37	21	P2.6	I/O	General purpose digital I/O pin
			SPI0_SS	O	SPI0 SS pin
			PWM0_CH4	O	PWM0 channel4 output pin
			SPI2_SS	I	SPI2 SS pin
25	38	22	P0.7	I/O	General purpose digital I/O pin
			SPI0_CLK	O	SPI0 CLK pin
			PWM0_CH0	O	PWM0 channel0 output pin
			SPI2_CLK	I	SPI2 CLK pin
			I2C1_SCL	I/O	I2C1 CLK pin
26	39	23	P0.6	I/O	General purpose digital I/O pin
			SPI0_MISO	I	SPI0 MISO pin

			PWM0_CH1	O	PWM0 channel1 output pin
			SPI2_MISO	O	SPI2 MISO pin
			UART1_TXD	O	UART1 TX pin
			UART0_RTS	O	UART0 RTS pin
			32K XC2	AO	32K Crystal pin2
27	40	24	P0.5	I/O	General purpose digital I/O pin
			SPI0_MOSI	O	SPI0 MOSI pin
			PWM0_CH4	O	PWM0 channel4 output pin
			SPI2_MOSI	I	SPI2 MOSI pin
			UART1_RXD	I	UART1 RX pin
			UART0_CTS	I	UART0 CTS pin
			32K XC1	AI	32K Crystal pin1
-	41	-	P2.1	I/O	General purpose digital I/O pin
28	42	-	P2.0	I/O	General purpose digital I/O pin
			SPI1_SS	O	SPI1 SS pin
			SPI3_SS	I	SPI3 SS pin
			I2C1_SDA	I/O	I2C1 data pin
-	43	-	P1.7	I/O	General purpose digital I/O pin
			SPI1_MISO	I	SPI1 MISO pin
			SPI3_MISO	O	SPI3 MISO pin
			UART0_TXD	O	UART0 TX pin
			UART1_RTS	O	UART1 RTS pin
-	44	-	P1.6	I/O	General purpose digital I/O pin
			SPI1_MOSI	O	SPI1 MOSI pin
			SPI3_MOSI	I	SPI3 MOSI pin
			UART0_RXD	I	UART0 RX pin
			UART1_CTS	I	UART1 CTS pin
-	45[Note]	-	P1.1	I/O	General purpose digital I/O pin (PAN1020DY does not have this function.)
			SPI1_CLK	O	SPI1 CLK pin (PAN1020DY does not have this function.)
			SPI3_CLK	I	SPI3 CLK pin (PAN1020DY does not have this function.)
			I2C0_SCL	I/O	I2C0 CLK pin (PAN1020DY does not have this function.)

29	46	-	P0.3	I/O	General purpose digital I/O pin
30	47	-	P0.2	I/O	General purpose digital I/O pin
31	48	1	ANT	AIO	Antenna pin
32	1	3	VDD2	P	RF power supply VDD2 pin
33	49	2	GND	P	Ground pin

Note: P1.1 of PAN1020DY has no actual function, P1.1 of PAN1020BY has actual function.

4 Function Description

4.1 System Manager

4.1.1 Overview

System management includes the following sections:

- System Reset
- System Power Architecture
- System Memory Map
- Chip reset and on-chip controllers reset, and multi-functional pin control
- System Timer (SysTick)
- System Control registers

4.1.2 System Reset

The system reset can be issued by one of the events listed below. These reset event flags can be read from SYS_RSTSTS register to determine the reset source. Hardware reset can reset chip through peripheral reset signals. Software reset can trigger reset through control registers.

- Hardware Reset Sources
  - Power-on Reset (POR)
  - Low level on the nRESET pin
  - Watchdog Time-out Reset and Window Watchdog Reset (WDT/WWDT Reset)
  - Low Voltage Reset (LVR)
  - Brown-out Detector Reset (BOD Reset)
  - CPU Lockup Reset
- Software Reset Sources
  - CHIP Reset will reset whole chip by writing 1 to CHIPRST (SYS_IPRST0[0])
  - System Reset to reboot but keeping the booting setting from APROM or LDROM by writing 1 to SYSRESETREQ (SCS_AIRCR[2])
  - CPU Reset for PAN1020 core only by writing 1 to CPURST (SYS_IPRST0[1])

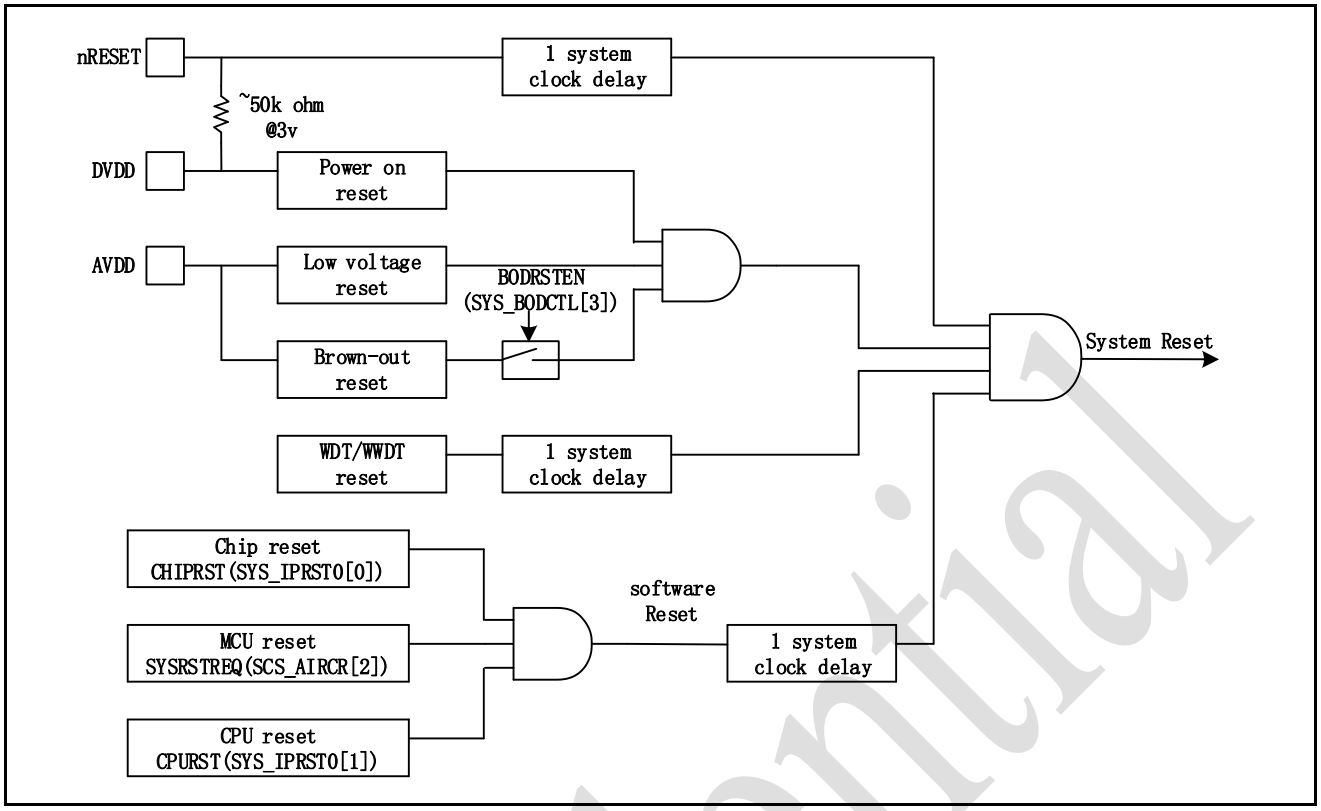


Figure 4-1 System Reset Resources

There are a total of 9 reset sources in the PAN1020. In general, CPU reset is used to reset M6_core only; the other reset sources will reset M6_core and all peripherals. However, there are small differences between each reset source and they are listed in Table 4-1.

Table 4-1 Reset Value of Registers

Reset Sources Register	POR	nRESET	WDT	LVR	BOD	CHIP	MCU	CPU
SYS_RSTSTS	Bit 1 = 1	Bit 1 = 1	Bit 2 = 1	Bit 1 = 1	Bit 4 = 1	Bit 0 = 1	Bit 5 = 1	Bit 7 = 1
CHIPRST (SYS_IPRST0[0])	0x0	0x0	0x0	0x0	0x0	0x0	0x0	-
BS (FMC_ISPCTL[1]) ISPEN (FMC_ISPCTL[0])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	-
FMC_DFBA	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	Reload from CONFIG1	-	-
BSC (FMC_ISPSTS[2:1])	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	Reload from CONFIG0	-	-

VECMAP (FMC_ISPSTS[20:9])	Reload base on CON-FIG0	Reload base on CON-FIG0	Reload base on CON-FIG0	Reload base on CON-FIG0	Reload base on CON-FIG0	Reload base on CON-FIG0	-	-
Other Peripheral Regis-ters	Reset Value							-
FMC Registers	Reset Value							
Note: '-' means that the value of register keeps original setting.								

4.1.2.1 nRESET Reset

The nRESET reset means to generate a reset signal by pulling low nRESET pin, which is an asynchronous reset input pin and can be used to reset system at any time. When the nRESET voltage is lower than 0.2 VDD and the state keeps longer than 36 us (glitch filter), chip will be reset. The nRESET reset will control the chip in reset state until the nRESET voltage rises above 0.7 VDD and the state keeps longer than 36 us (glitch filter). The PINRF (SYS_RSTSTS[1]) will be set to 1 if the previous reset source is nRESET reset. Figure 4-2 shows the nRESET reset waveform.

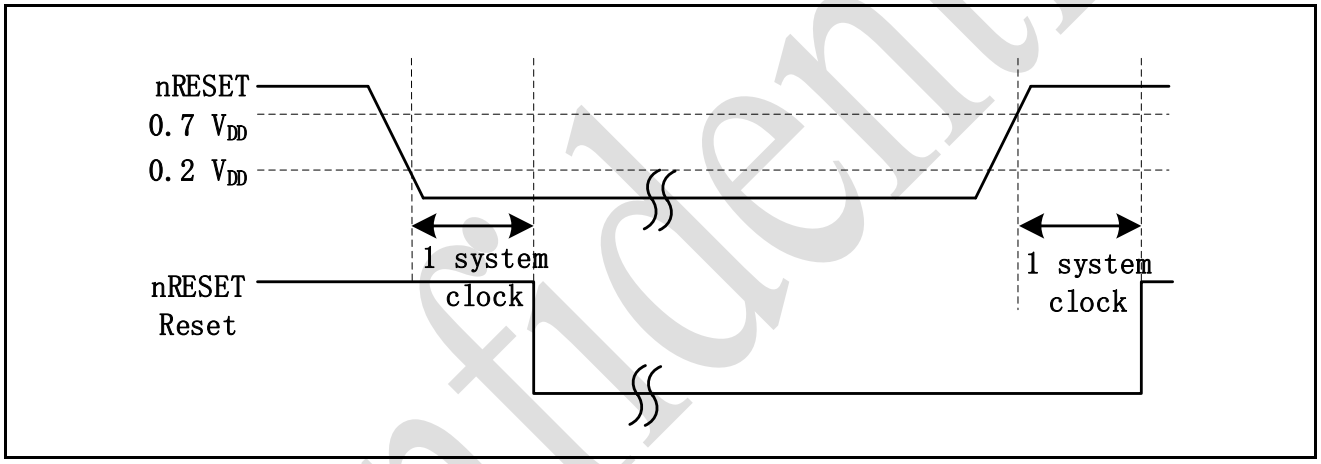


Figure 4-2 nRESET Reset Waveform

4.1.2.2 Power-On Reset (POR)

The Power-on reset (POR) is used to generate a stable system reset signal and forces the system to be reset when power-on to avoid unexpected behavior of MCU. When applying the power to MCU, the POR module will detect the rising voltage and generate reset signal to system until the voltage is ready for MCU operation. At POR reset, the PINRF (SYS_RSTSTS[1]) will be set to 1 to indicate there is a POR reset event. The PINRF (SYS_RSTSTS[1]) bit can be cleared by writing 1 to it. Figure 4-3 shows the waveform of Power-On reset.

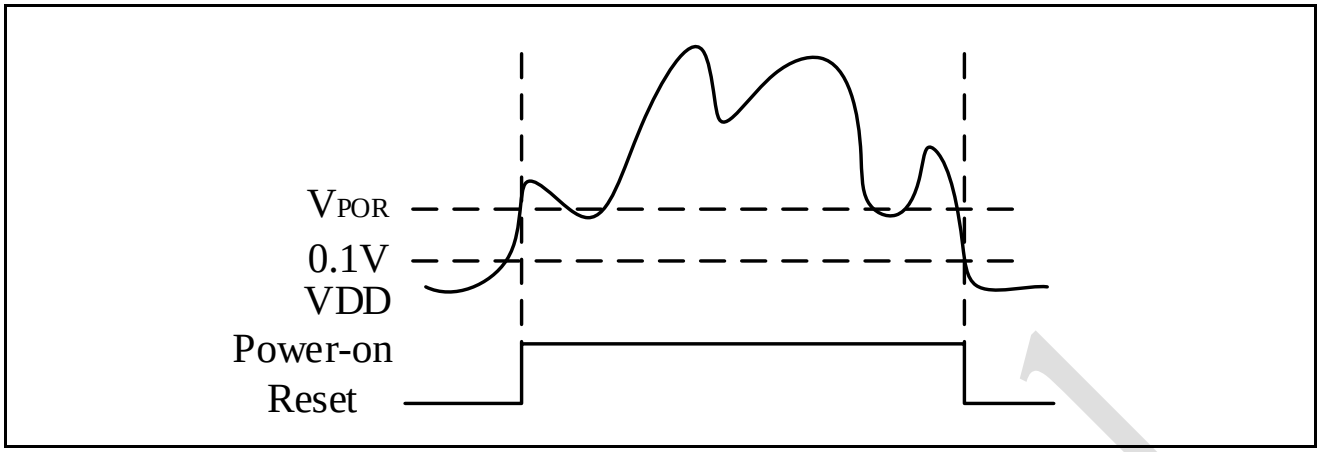


Figure 4-3 Power-on Reset (POR) Waveform

4.1.2.3 Low Voltage Reset (LVR)

Low Voltage Reset detects AVDD during system operation. When the AVDD voltage is lower than VLVR and the state keeps longer than De-glitch time (see SYS_BLDBCTL register), chip will be reset. The LVR reset will control the chip in reset state until the AVDD voltage rises above VLVR and the state keeps longer than De-glitch time. The PINRF (SYS_RSTSTS[1]) will be set to 1 if the previous reset source is nRESET reset. Figure 4-4 shows the Low Voltage Reset waveform.

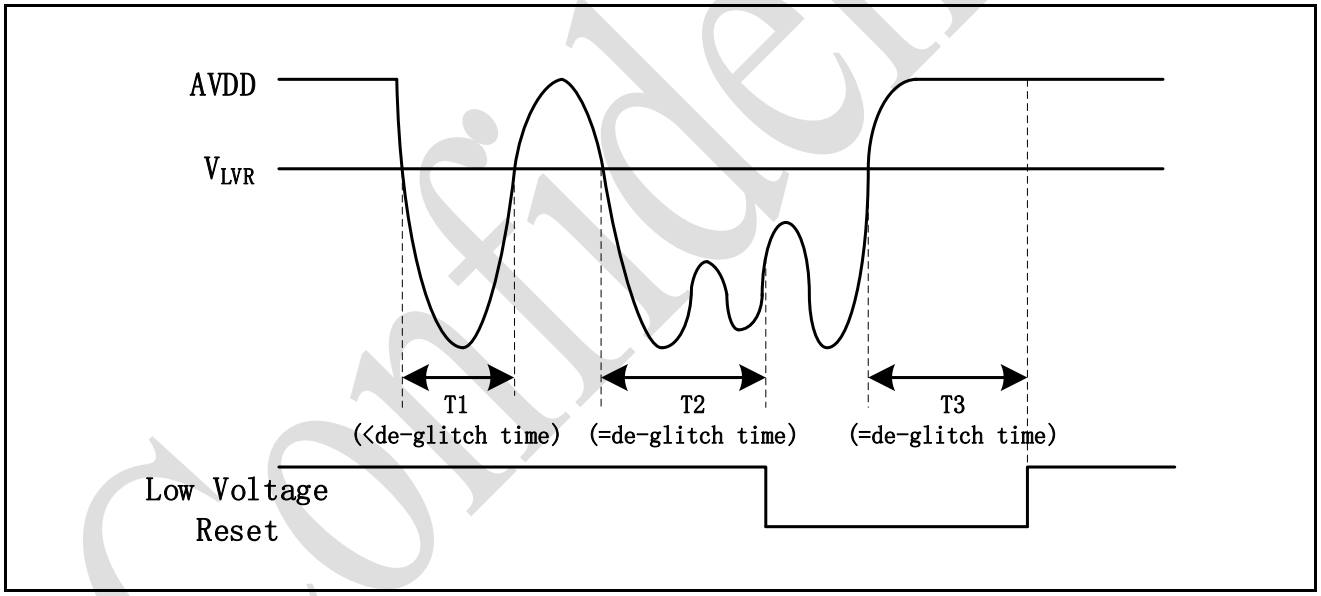


Figure 4-4 Low Voltage Reset (LVR) Waveform

4.1.2.4 Brown-out Detector Reset (BOD Reset)

If the Brown-out Detector (BOD) function is enabled by setting the Brown-out Detector Enable Bit BODEN (ANAC_RCCCTL [24]), Brown-Out Detector function will detect AVDD during system operation. When the AVDD voltage is lower than VBOD which is decided by BODEN (ANAC_RCCCTL [24]) and BODVL (ANAC_RCCCTL [26:25]) and the state keeps longer than De-glitch time (see SYS_BLDBCTL register), chip will be reset. The BOD reset will control the chip in reset state until the AVDD voltage rises above VBOD and the state keeps longer than De-glitch time. The default value of BODEN, BODVL and BODRSTEN is set by flash controller user configuration register CBOVEXT (CONFIG0[23]), CBOV (CONFIG0[22:21]) and CBORST

(CONFIG0[20]) respectively. User can determine the initial BOD setting by setting the CONFIG0 register. Figure 4-5 shows the Brown-Out Detector waveform.

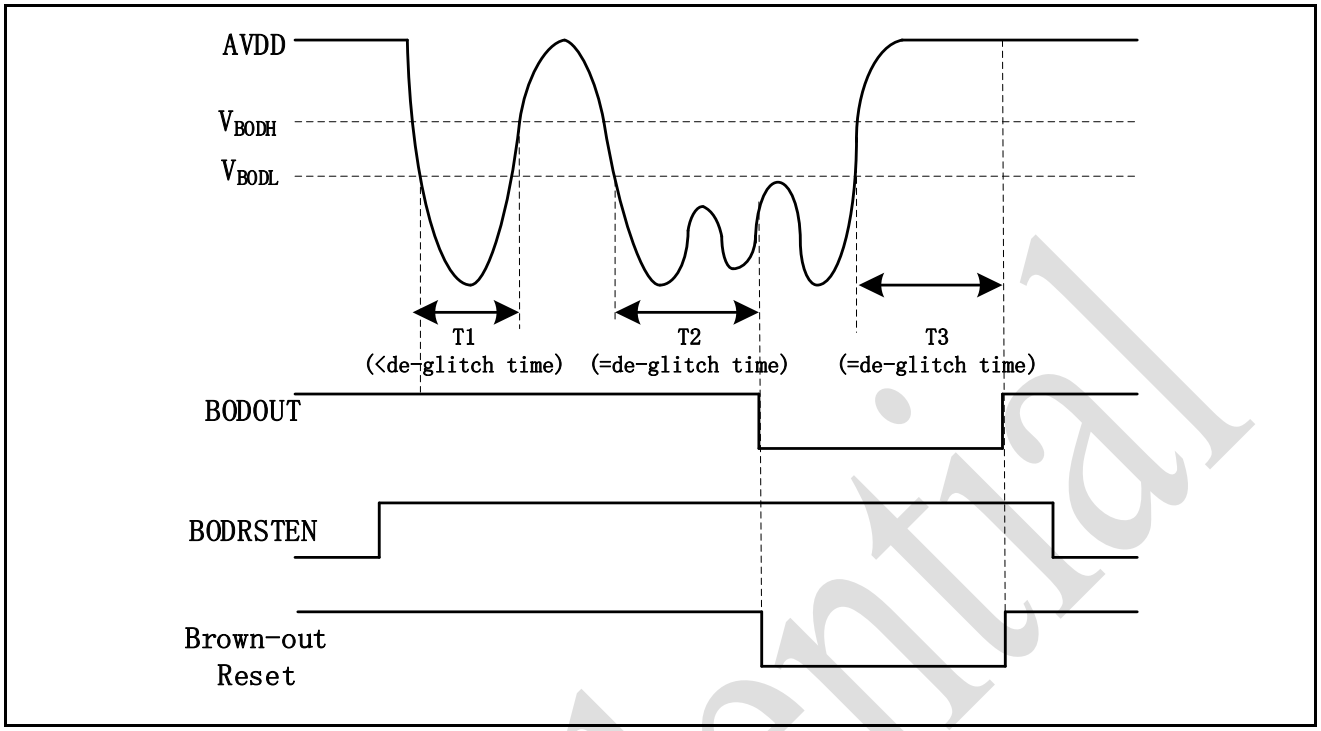


Figure 4-5 Brown-out Detector (BOD) Waveform

4.1.2.5 Watchdog Timer Reset

In most industrial applications, system reliability is very important. To automatically recover the MCU from failure status is one way to improve system reliability. The watch dog timer (WDT) is widely used to check if the system works fine. If the MCU is crashed or out of control, it may cause the watch dog time-out. User may decide to enable system reset during watch dog time-out to recover the system and take action for the system crash/out-of-control after reset.

Software can check if the reset is caused by watch dog time-out to indicate the previous reset is a watch dog reset and handle the failure of MCU after watch dog time-out reset by checking WDTRF (SYS_RSTSTS[2]).

4.1.2.6 CPU Reset, CHIP Reset and SYSTEM Reset

The CPU Reset means only PAN1020MCU is reset and all other peripherals remain the same status after CPU reset. BS (FMC_ISPCTL[1]) will not be reloaded from CONFIG setting and keep its original software setting for booting from APROM or LDROM. User can set the CPURST (SYS_IPRST0[1]) to 1 to assert the CPU Reset signal.

The CHIP Reset is same with Power-On Reset. The CPU and all peripherals are reset and BS (FMC_ISPCTL[1]) bit is automatically reloaded from CONFIG setting. User can set the CHIPRST (SYS_IPRST0[0]) to 1 to assert the CHIP Reset signal.

The System Reset is similar with CHIP Reset. User can set the SYSRESETREQ (SCS_AIRCR[2]) to 1 to assert the System Reset.

4.1.3 Power Modes and Wake-up Sources

There are three level of low power mode. In different application, use can use different low power mode to get the lowest power and best performance. Table 4-2 lists the related descriptions of each power mode.

Table 4-2 Power Mode Difference Table

Chip work mode	Descriptions	Wake sources	Wake up time	Power consumption
Deep_Sleep	Almost all region is power down excepts 3V region(a little digital control region).	GPIO (P5.2);	Just as long as chip power up.	The lowest power consumption which is near 0.
Sleep_Mode_0	3V digital region is work. 1.2V(main region) region is work in 0.7V(just SRAM can maintain datas), other logic is out of work.	GPIO (P5.2); 32K clock;	Just as long as chip power up.	The lower power consumption.
Sleep_Mode_1	All region is in right voltage. CPU is in low power.	GPIO (all); 32K clock; I2C, BOD, WDT, Timer	Just depend on wake up source active.	The low power consumption.
Normal mode	All region is in right voltage.	-	-	Work power consumption.
Power down	Power down	-	Chip power up.	0

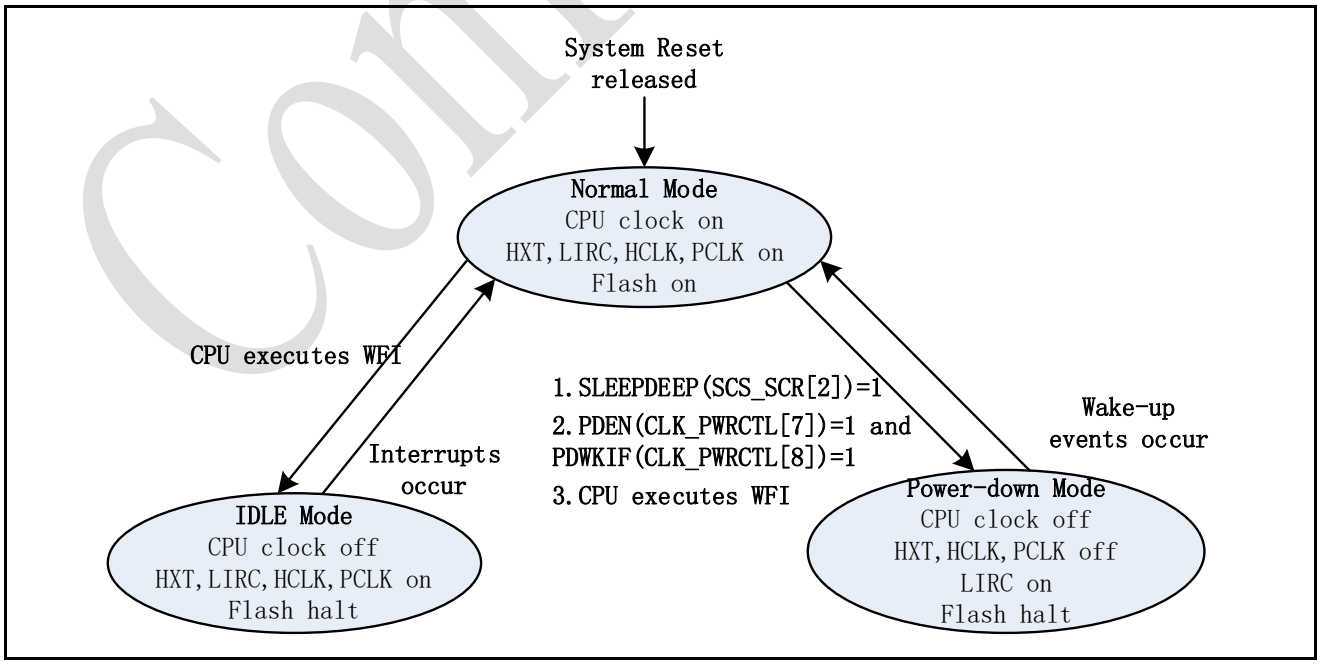


Figure 4-6 Power Mode State Machine

1. LIRC (32 kHz OSC) ON or OFF depends on S/W setting in run mode.

2. If TIMER clock source is selected as LIRC and LIRC is on.

3. If WDT clock source is selected as LIRC and LIRC is on.

Table 4-3 Clocks in Power Modes

	Normal Mode	Idle Mode	Power-down Mode
HXT (4~20 MHz XTL)	ON	ON	Halt
LXT (32768 Hz XTL)	ON	ON	ON/OFF1
LIRC (32 kHz OSC)	ON	ON	ON/OFF2
PLL	ON	ON	ON/OFF2
LDO	ON	ON	ON
CPU	ON	Halt	Halt
HCLK/HCLK	ON	ON	Halt
SRAM retention	ON	ON	ON
FLASH	ON	ON	Halt
GPIO	ON	ON	Halt
TIMER	ON	ON	ON/OFF3
PWM	ON	ON	Halt
WDT	ON	ON	ON/OFF4
WWDT	ON	ON	Halt
I2C	ON	ON	Halt
SPI	ON	ON	Halt
ADC	ON	ON	Halt

Low power process:

(1). User needs to set register ANAC_3VCTL, BLEBB->DEEPSLWKUP, BLEBB->ENBPRESET, BLEBB->DEEPSLCNTL.

(2). Whether select input pin wake up source(BLEBB->DEEPSLCNTL[31]).

(3). Set configact(ANAC_3VCTL[0]) to 1, and wait configact, configact32k(ANAC_3VCTL[1:0]) become 0.

(4). Set BLEBB->DEEPSLCNTL[2:0] to 3'b1xx.

(5). User needs to wait this condition before setting PDEN (CLK_PWRCTL[7]) and execute WFI to enter Power-down mode.

(6). Wait wake up source occur to wake up chip.

4.1.4 System Power Architecture

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

- Analog power from AVDD and AVSS provides the power for analog components operation. AVDD must be equal to VDD to avoid leakage current.

- Digital power from VDD and VSS supplies power to the I/O pins and internal regulator which provides a fixed 1.8V power for digital operation.
- Built-in a capacitor for internal voltage regulator

The output of internal voltage regulator (DVDD) requires an external capacitor which should be located close to the corresponding pin. Analog power (AVDD) should be the same voltage level as the digital power (VDD). Figure 4-7 shows the power distribution of the PAN1020.

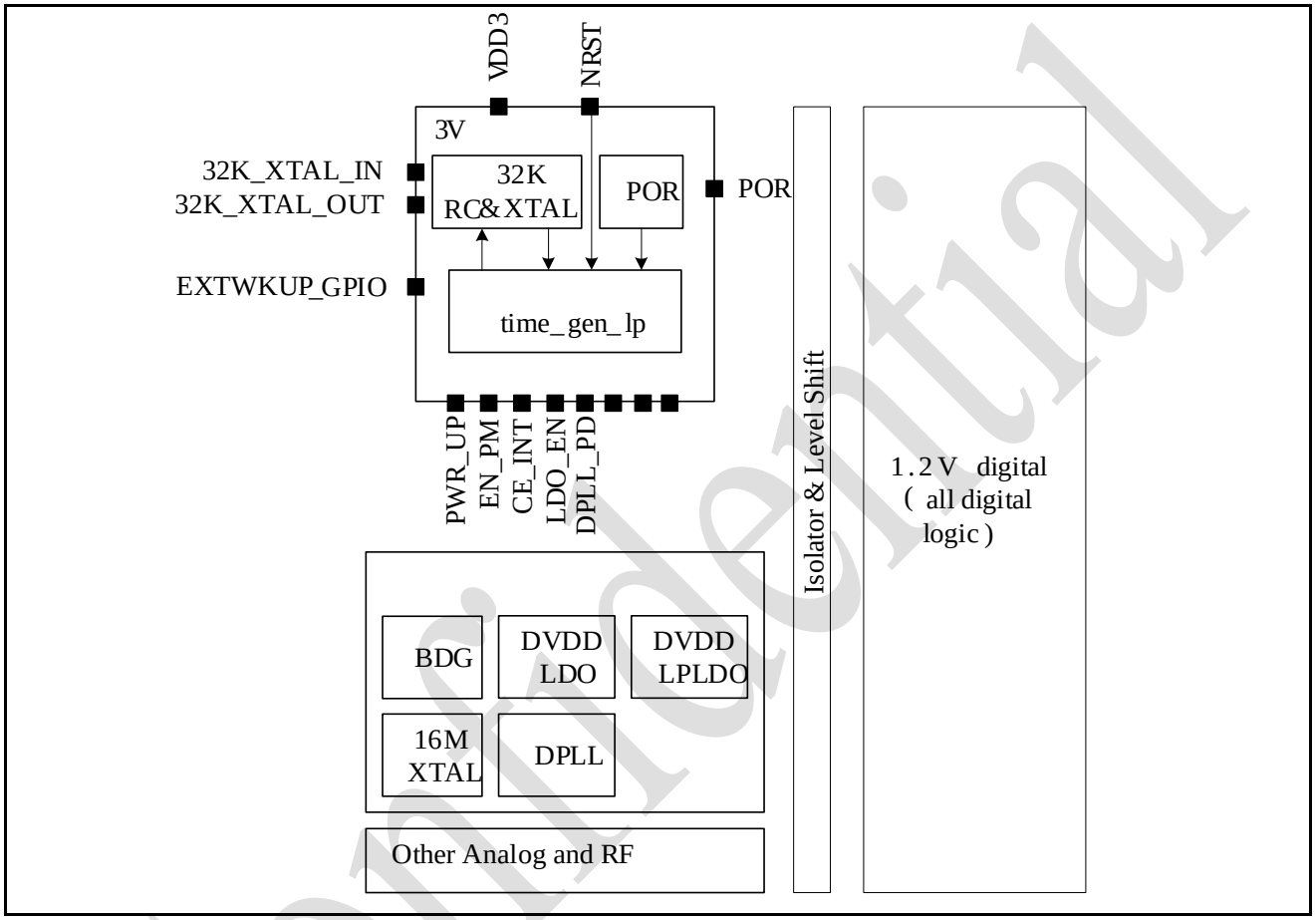
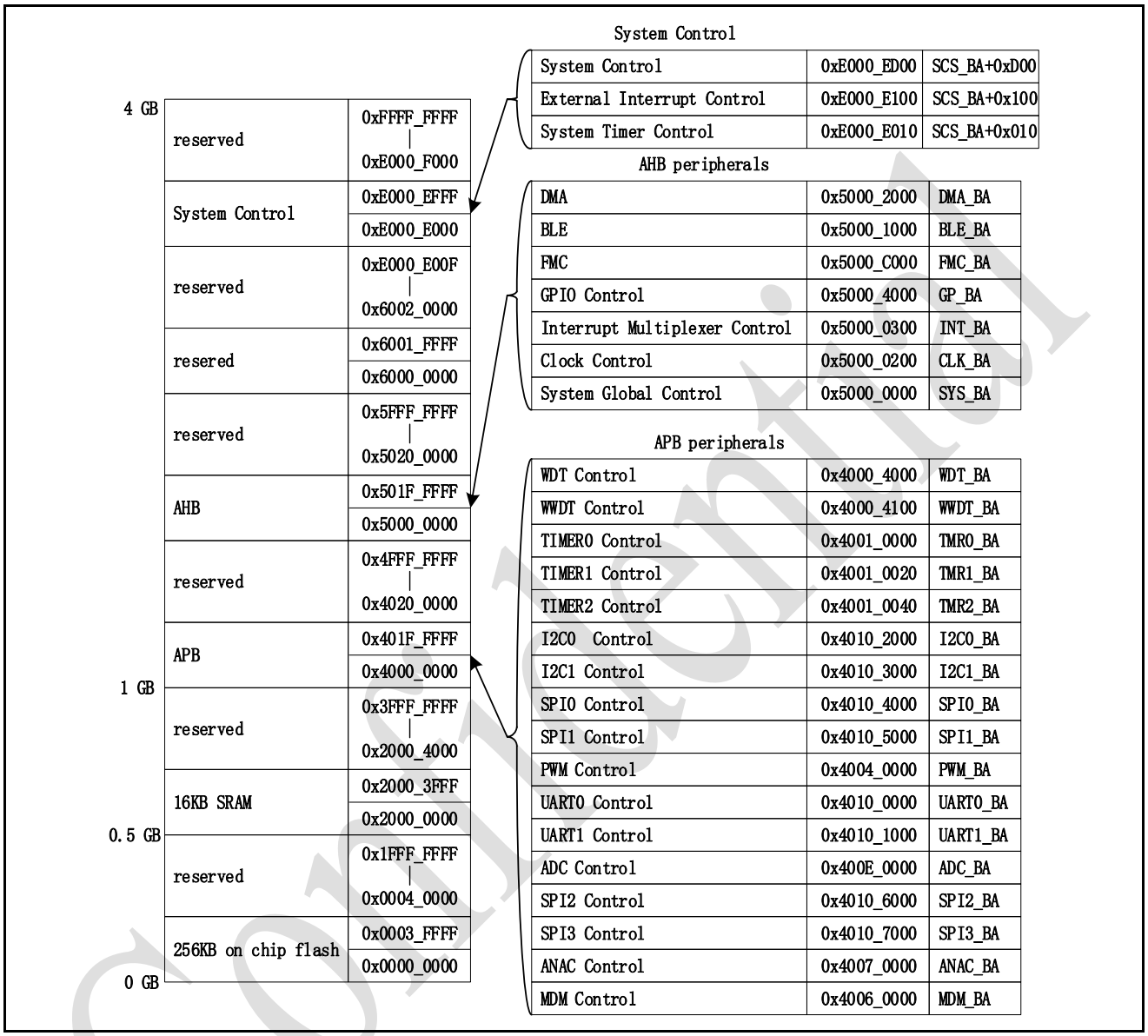


Figure 4-7 PAN1020 Power Architecture Diagram

4.1.5 System Memory Mapping

Table 4-4 Memory Mapping Table



4.1.6 Memory Organization

4.1.6.1 Overview

The PAN1020 provides 4G-byte addressing space. The addressing space assigned to each on-chip controllers is shown the Table 4-5. The detailed register definition, addressing space, and programming details will be described in the following sections for each on-chip peripheral. The PAN1020 only supports little-endian data format.

4.1.6.2 System Memory Map

The memory locations assigned to each on-chip controllers are shown in the Table 4-5.

Table 4-5 Address Space Assignments for On-Chip Modules

Addressing Space	Token	Modules
Flash and SRAM Memory Space		
0x0000_0000 – 0x0003_FFFF	FLASH_BA	Flash Memory Space (256 KB)
0x2000_0000 – 0x2000_3FFF	SRAM_BA	SRAM Memory Space (16 KB)
APB Modules Space (0x4000_0000 – 0x400F_FFFF)		
0x4000_4000 – 0x4004_00FF	WDT_BA	Watchdog Timer Control Registers
0x4000_4100 – 0x4004_41FF	WWDT_BA	Window Watchdog Timer Control Registers
0x4001_0000 – 0x4001_001F	TMR0_BA	Timer0 Control Registers
0x4001_0020 – 0x4001_003F	TMR1_BA	Timer1 Control Registers
0x4001_0040 – 0x4001_005F	TMR2_BA	Timer2 Control Registers
0x4004_0000 – 0x4004_00FF	PWM_BA	PWM Control Registers
0x4006_0000 – 0x4006_03FF	MDM_BA	BLE-MDM Control Registers
0x4007_0000 – 0x4007_00FF	ANAC_BA	ANA-CTL Control Registers
0x4010_0000 – 0x4010_0FFF	UART0_BA	UART0 Control Registers
0x4010_1000 – 0x4010_1FFF	UART1_BA	UART1 Control Registers
0x4010_2000 – 0x4010_2FFF	I2C0_BA	I2C0 Interface Control Registers
0x4010_3000 – 0x4010_3FFF	I2C1_BA	I2C1 Interface Control Registers
0x4010_4000 – 0x4010_4FFF	SPI0_BA	SPI0 with Master/slave Function Control Registers
0x4010_5000 – 0x4010_5FFF	SPI1_BA	SPI1 with Master/slave Function Control Registers
0x4010_6000 – 0x4010_6FFF	SPI2_BA	SPI2 with Master/slave Function Control Registers
0x4010_7000 – 0x4010_7FFF	SPI3_BA	SPI3 with Master/slave Function Control Registers
0x400E_0000 – 0x400E_00FF	ADC_BA	Analog-Digital-Converter (ADC) Control Registers
AHB Modules Space (0x5000_0000 – 0x501F_FFFF)		
0x5000_2000 – 0x5000_23FF	DMA_BA	DMA Control Registers
0x5000_0000 – 0x5000_01FF	SYS_BA	System Global Control Registers
0x5000_0200 – 0x5000_02FF	CLK_BA	Clock Control Registers
0x5000_0300 – 0x5000_03FF	INT_BA	Interrupt Multiplexer Control Registers
0x5000_1000 – 0x5000_17FF	BLE_BA	BLE Baseband Control Registers and EM, 2KB
0x5000_4000 – 0x5000_7FFF	GP_BA	GPIO (P0~P5) Control Registers
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash Memory Control Registers
System Control Space (0xE000_E000 – 0xE000_EFFF)		
0xE000_E010 – 0xE000_E0FF	SCS_BA+ 0×010	System Timer Control Registers
0xE000_E100 – 0xE000_ECFF	SCS_BA+ 0×100	Nested Vectored Interrupt Control Registers
0xE000_ED00 – 0xE000_ED8F	SCS_BA+ 0×D00	System Control Block Registers

4.1.7 Register Map

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

Register	Offset	R/W	Description	Reset Value
SYS Base Address: **SYS_BA = 0x5000_0000**				
SYS_RSTSTS	SYS_BA+0x04	R/W	System Reset Status Register	0x0000_00XX
SYS_IPRST0	SYS_BA+0x08	R/W	Peripheral Reset Control Register 0	0x0000_0000
SYS_IPRST1	SYS_BA+0x0C	R/W	Peripheral Reset Control Register 1	0x0000_0000
SYS_BODCTL	SYS_BA+0x18	R/W	Brown-out Detector Control Register	0x0000_000X
SYS_P0_MFP	SYS_BA+0x30	R/W	P0 Multiple Function and Input Type Control Register	0x0000_0000
SYS_P1_MFP	SYS_BA+0x34	R/W	P1 Multiple Function and Input Type Control Register	0x0000_0100
SYS_P2_MFP	SYS_BA+0x38	R/W	P2 Multiple Function and Input Type Control Register	0x0000_0000
SYS_P3_MFP	SYS_BA+0x3C	R/W	P3 Multiple Function and Input Type Control Register	0x0000_0000
SYS_P4_MFP	SYS_BA+0x40	R/W	P4 Multiple Function and Input Type Control Register	0x0000_00C0
SYS_P5_MFP	SYS_BA+0x44	R/W	P5 Multiple Function and Input Type Control Register	0x0000_0800
SYS_TESTCTL	SYS_BA+0x48	R/W	Analog Test Control	0x0000_0000
SYS_BLDBCTL	SYS_BA+0x4C	R/W	BOD LVR De-Bounce Control Register	0x0000_2020
SYS_REGLCTL	SYS_BA+0x100	R/W	Register Write-Protection Control Register	0x0000_0000
SYS_STATUS	SYS_BA+0x104	RO	Register to reflect the status of ROM Mode	0x0000_0000

4.1.8 Register Description

4.1.8.1 System Reset Status Register (SYS_RSTSTS)

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

Register	Offset	R/W	Description	Reset Value
SYS_RSTSTS	SYS_BA+0x04	R/W	System Reset Status Register	0x0000_00XX

Bits	Description	
[31:8]	Reserved	Reserved.
[7]	CPURF	CPU Reset Flag The CPU reset flag is set by hardware if software writes CPURST (SYS_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]	Reserved	Reserved.
[5]	SYSRF	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 SYSRESETREQ (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	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]	Reserved	Reserved.
[2]	WDTRF	WDT Reset Flag The WDT reset flag is set by the “Reset Signal” from the Watchdog Timer or Window Watchdog Timer to indicate the previous reset source. 0 : No reset from watchdog timer or window watchdog timer. 1 : The watchdog timer or window watchdog timer had issued the reset signal to reset the system. **Note**: Software can write 1 to clear this bit to zero.
[1]	PINRF	NRESET Pin Reset Flag The nRESET pin reset flag is set by the “Reset Signal” from the nRESET pin or Power-on Reset (POR) Controller or Low-Voltage-Reset (LVR) to indicate the previous reset source. 0 : No reset from nRESET pin, POR or LVR. 1 : Pin nRESET, POR or LVR had issued the reset signal to reset the system. **Note**: Software can write 1 to clear this bit to zero.
[0]	CHIPRF	CHIP Reset Flag The POR reset flag is set by the “Reset Signal” from the CHIPRST (SYS_IPRST0[0]) to indicate the previous reset source. 0 : CHIPRST. 1 : CHIPRST had issued the reset signal to reset the system. **Note**: Software can write 1 to clear this bit to zero.

4.1.8.2 Peripheral Reset Control Register 0 (SYS_IPRST0)

Register	Offset	R/W	Description	Reset Value
SYS_IPRST0	SYS_BA+0x08	R/W	Peripheral Reset Control Register 0	0x0000_0000

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

4.1.8.3 Peripheral Reset Control Register 1 (SYS_IPRST1)

Setting these bits 1 will generate asynchronous reset signals to the corresponding module controller. Users need to set these bits to 0 to release corresponding module controller from reset state.

Register	Offset	R/W	Description	Reset Value
SYS_IPRST1	SYS_BA+0x0C	R/W	Peripheral Reset Control Register 1	0x0000_0000

Bits	Description	
[31:30]	Reserved	Reserved.
[29]	ANACRST	ANALOG Controller Reset 0 : ANALOG Controller normaloperation 1 : ANALOG Controller reset
[28]	ADCRST	ADC Controller Reset 0 : ADC controller normal operation. 1 : ADC controller reset.
[27:21]	Reserved	Reserved.
[20]	PWM0RST	PWM0 Controller Reset 0 : PWM0 controller normal operation. 1 : PWM0 controller reset.
[19]	WWDTRST	WWDT Controller Reset 0 : WWDT controller normal operation. 1 : WWDT controller reset.
[18]	WDTRST	WDT Controller Reset 0 : WDT controller normal operation. 1 : WDT controller reset.

[17]	UART1RST	UART1 Controller Reset 0 : UART1 controller normal operation. 1 : UART1 controller reset.
[16]	UART0RST	UART0 Controller Reset 0 : UART0 controller normal operation. 1 : UART0 controller reset.
[15]	SPI3RST	SPI3 Controller Reset 0 : SPI3 controller normal operation. 1 : SPI3 controller reset.
[14]	SPI2RST	SPI2 Controller Reset 0 : SPI2 controller normal operation. 1 : SPI2 controller reset.
[13]	SPI1RST	SPI1 Controller Reset 0 : SPI1 controller normal operation. 1 : SPI1 controller reset.
[12]	SPI0RST	SPI0 Controller Reset 0 : SPI0 controller normal operation. 1 : SPI0 controller reset.
[11:10]	Reserved	Reserved.
[9]	I2C1RST	I2C1 Controller Reset 0 : I2C1 controller normal operation. 1 : I2C1 controller reset.
[8]	I2C0RST	I2C0 Controller Reset 0 : I2C0 controller normal operation. 1 : I2C0 controller reset.
[7:6]	Reserved	Reserved.
[5]	MDMRST	MDM Controller Reset 0 : MDM controller normal operation. 1 : MDM controller reset.
[4]	TMR2RST	Timer2 Controller Reset 0 : Timer2 controller normal operation. 1 : Timer2 controller reset.
[3]	TMR1RST	Timer1 Controller Reset 0 : Timer1 controller normal operation. 1 : Timer1 controller reset.
[2]	TMR0RST	Timer0 Controller Reset 0 : Timer0 controller normal operation. 1 : Timer0 controller reset.
[1]	GPIORST	GPIO (P0~P5) Controller Reset 0 : GPIO controller normal operation. 1 : GPIO controller reset.
[0]	BBRST	BASEBAND Controller Reset 0 : BASEBAND controller normal operation.

		1 : BASEBAND controller reset.

4.1.8.4 Brown-out Detector Control Register (SYS_BODCTL)

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

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

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

Bits	Description	
[31:7]	Reserved	Reserved.
[6]	BODOUT	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.
[5]	Reserved	Reserved.
[4]	BODIF	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.
[3]	BODRSTEN	Brown-out Reset Enable Bit (Write Protect) The default value is set by flash controller user configuration register CBORST(CONFIG0[20]) bit. If CBORST is set to 1, default value of BODRSTEN is 0. If CBORST is set to 0, default value of BODRSTEN is 1. 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 BODEN is enabled and the interrupt is asserted, the interrupt will be kept till the BODEN is set to 0. The interrupt for CPU can be blocked by disabling the BODEN and then re-enabling the BODEN function if the BOD function is required.
[2:0]	Reserved	Reserved.

4.1.8.5 Multiple Function Port0 Control Register (SYS_P0_MFP)

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

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]	MFP [7]	P0.7 Alternate Function Select Bit Bits EXT[7] (SYS_P0_MFP[23]), ALT[7] (SYS_P0_MFP[15]), and MFP[7] (SYS_P0_MFP[7]) determine the P0.7 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : I2C1_SCL function is selected. (0, 1, 0) : SPI0_CLK function is selected. (0, 1, 1) : PWM0_CH0 function is selected. (1, 0, 0) : SPI2_CLK function is selected. (1, 0, 1) : DBG[12] function is selected. (1, 1, 0) : DBG[4] function is selected. (1, 1, 1) : FRACN_OUT_FPGA[3] function is selected.
[6]	MFP [6]	P0.6 Alternate Function Select Bit Bits EXT[6] (SYS_P0_MFP[22]), ALT[6] (SYS_P0_MFP[14]), and MFP[6] (SYS_P0_MFP[6]) determine the P0.6 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : DBG[13] function is selected. (0, 1, 0) : SPI0_MISO function is selected. (0, 1, 1) : PWM0_CH1 function is selected. (1, 0, 0) : SPI2_MISO function is selected. (1, 0, 1) : UART1_TXD function is selected. (1, 1, 0) : UART0_RTS function is selected. (1, 1, 1) : 32K XC2 function is selected.
[5]	MFP [5]	P0.5 Alternate Function Select Bit Bits EXT[5] (SYS_P0_MFP[21]), ALT[5] (SYS_P0_MFP[13]), and MFP[5] (SYS_P0_MFP[5]) determine the P0.5 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : DBG[14] function is selected. (0, 1, 0) : SPI0_MOSI function is selected. (0, 1, 1) : PWM0_CH4 function is selected. (1, 0, 0) : SPI2_MOSI function is selected. (1, 0, 1) : UART1_RXD function is selected. (1, 1, 0) : UART0_CTS function is selected.

		(1, 1, 1) : 32K XC1 function is selected.
[4]	MFP [4]	P0.4 Alternate Function Select Bit Bits EXT[4] (SYS_P0_MFP[20]), ALT[4] (SYS_P0_MFP[12]), and MFP[4] (SYS_P0_MFP[4]) determine the P0.4 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : TADC_DATH function is selected. (0, 1, 0) : SPI0_SS function is selected. (0, 1, 1) : PWM0_CH5 function is selected. (1, 0, 0) : SPI2_SS function is selected. (1, 0, 1) : DBG[8] function is selected. (1, 1, 0) : DBG[0] function is selected. (1, 1, 1) : R_CLK function is selected.
[3]	MFP [3]	P0.3 Alternate Function Select Bit Bits EXT[3] (SYS_P0_MFP[19]), ALT[3] (SYS_P0_MFP[11]), and MFP[3] (SYS_P0_MFP[3]) determine the P0.3 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : SPI1_SS function is selected. (0, 1, 0) : DBG[23] function is selected. (0, 1, 1) : DBG[31] function is selected. (1, 0, 0) : SPI3_SS function is selected. (1, 0, 1) : I2C0_SDA function is selected. (1, 1, 0) : BPF_GC[2] function is selected. (1, 1, 1) : FRACN_OUT_FPGA[8] function is selected.
[2]	MFP [2]	P0.2 Alternate Function Select Bit Bits EXT[2] (SYS_P0_MFP[18]), ALT[5] (SYS_P0_MFP[10]), and MFP[2] (SYS_P0_MFP[2]) determine the P0.2 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : CLKO_32K function is selected. (0, 1, 0) : CLKO_16M function is selected. (0, 1, 1) : CLKO_26M function is selected. (1, 0, 0) : CLKO_13M function is selected. (1, 0, 1) : BPF_GC[3] function is selected. (1, 1, 0) : FRACN_OUT_FPGA[9] function is selected. (1, 1, 1) : GPIO function is selected.
[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) : SPI0_SS function is selected. (0, 1, 0) : PWM0_CH3 function is selected. (0, 1, 1) : UART0_RXD function is selected. (1, 0, 0) : UART1_CTS function is selected. (1, 0, 1) : SPI2_SS function is selected. (1, 1, 0) : LVR_OUT function is selected.

		(1, 1, 1) : LNA_GC[0] function is selected.
[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) : TADC_CLK function is selected. (0, 1, 0) : PWM0_CH2 function is selected. (0, 1, 1) : UART0_TXD function is selected. (1, 0, 0) : UART1_RTS function is selected. (1, 0, 1) : BOD_OUT function is selected. (1, 1, 0) : LNA_GC[1] function is selected. (1, 1, 1) : GPIO function is selected.

4.1.8.6 Multiple Function Port1 Control Register (SYS_P1_MFP)

Register	Offset	R/W	Description	Reset Value
SYS_P1_MFP	SYS_BA+0x34	R/W	P1 Multiple Function and Input Type Control Register	0x0000_0100

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) : SPI1_MISO function is selected. (0, 1, 0) : DBG[28] function is selected. (0, 1, 1) : SPI3_MISO function is selected. (1, 0, 0) : UART0_TXD function is selected. (1, 0, 1) : UART1_RTS function is selected. (1, 1, 0) : DBG[20] function is selected. (1, 1, 1) : FRACN_OUT_FPGA[6] 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) : SPI1_MOSI function is selected. (0, 1, 0) : DBG[29] function is selected.

		(0, 1, 1) : SPI3_MOSI function is selected. (1, 0, 0) : UART0_RXD function is selected. (1, 0, 1) : UART1_CTS function is selected. (1, 1, 0) : DBG[21] function is selected. (1, 1, 1) : BPF_GC[0] 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) : ADC_CH5 function is selected. (0, 1, 0) : UART1_TXD function is selected. (0, 1, 1) : UART0_RTS function is selected. (1, 0, 0) : PWM0_CH6 function is selected. (1, 0, 1) : DBG[17] function is selected. (1, 1, 0) : DBG[25] function is selected. (1, 1, 1) : GPIO 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) : ADC_CH4 function is selected. (0, 1, 0) : UART1_RXD function is selected. (0, 1, 1) : UART0_CTS function is selected. (1, 0, 0) : PWM0_CH4 function is selected. (1, 0, 1) : DBG[16] function is selected. (1, 1, 0) : DBG[24] function is selected. (1, 1, 1) : GPIO 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) : ADC_CH3 function is selected. (0, 1, 0) : UART0_TXD function is selected. (0, 1, 1) : UART1_RTS function is selected. (1, 0, 0) : PWM0_CH1 function is selected. (1, 0, 1) : DBG[3] function is selected. (1, 1, 0) : DBG[11] function is selected. (1, 1, 1) : GPIO 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) : ADC_CH2 function is selected. (0, 1, 0) : UART0_RXD function is selected.

		(0, 1, 1) : UART1_CTS function is selected. (1, 0, 0) : PWM0_CH0 function is selected. (1, 0, 1) : DBG[2] function is selected. (1, 1, 0) : DBG[10] function is selected. (1, 1, 1) : GPIO 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) : SPI1_CLK function is selected. (0, 1, 0) : DBG[30] function is selected. (0, 1, 1) : SPI3_CLK function is selected. (1, 0, 0) : I2C0_SCL function is selected. (1, 0, 1) : DBG[22] function is selected. (1, 1, 0) : BPF_GC[1] function is selected. (1, 1, 1) : FRACN_OUT_FPGA[7] 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) : ADC_CH1 function is selected. (0, 1, 0) : PWM0_CH3 function is selected. (0, 1, 1) : DBG[1] function is selected. (1, 0, 0) : DBG[9] function is selected. (1, 0, 1) : SPI1_MOSI function is selected. (1, 1, 0) : SPI2_MOSI function is selected. (1, 1, 1) : GPIO function is selected.

4.1.8.7 Multiple Function Port2 Control Register (SYS_P2_MFP)

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

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]	MFP [7]	P2.7 Alternate Function Select Bit Bits EXT[7] (SYS_P2_MFP[23]), ALT[7] (SYS_P2_MFP[15]), and MFP[7] (SYS_P2_MFP[7]) determine the P2.7 function. (0, 0, 0) : GPIO function is selected.

		(0, 0, 1) : default. (0, 1, 0) : TM2_EXT/TM2 function is selected. (0, 1, 1) : TADC_VLD function is selected. (1, 0, 0) : SPI0_CLK function is selected. (1, 0, 1) : SPI2_CLK function is selected. (1, 1, 0) : DBG[6] function is selected. (1, 1, 1) : DBG[14] function is selected.
[6]	MFP [6]	P2.6 Alternate Function Select Bit Bits EXT[6] (SYS_P2_MFP[22]), ALT[6] (SYS_P2_MFP[14]), and MFP[6] (SYS_P2_MFP[6]) determine the P2.6 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : SPI0_SS function is selected. (0, 1, 0) : PWM0_CH4 function is selected. (0, 1, 1) : SPI2_SS function is selected. (1, 0, 0) : RF_SPI_CSK function is selected. (1, 0, 1) : VGA_GC[3] function is selected. (1, 1, 0) : FRACN_OUT_FPGA[2] function is selected. (1, 1, 1) : default.
[5]	MFP [5]	P2.5 Alternate Function Select Bit Bits EXT[5] (SYS_P2_MFP[21]), ALT[5] (SYS_P2_MFP[13]), and MFP[5] (SYS_P2_MFP[5]) determine the P2.5 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : UART1_TXD function is selected. (0, 1, 0) : PWM0_CH3 function is selected. (0, 1, 1) : UART0_RTS function is selected. (1, 0, 0) : DBG[27] function is selected. (1, 0, 1) : DBG[19] function is selected. (1, 1, 0) : VGA_GC[2] function is selected. (1, 1, 1) : FRACN_OUT_FPGA[1] function is selected.
[4]	MFP [4]	P2.4 Alternate Function Select Bit Bits EXT[4] (SYS_P2_MFP[20]), ALT[4] (SYS_P2_MFP[12]), and MFP[4] (SYS_P2_MFP[4]) determine the P2.4 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : UART1_RXD function is selected. (0, 1, 0) : PWM0_CH2 function is selected. (0, 1, 1) : UART0_CTS function is selected. (1, 0, 0) : DBG[26] function is selected. (1, 0, 1) : DBG[18] function is selected. (1, 1, 0) : VGA_GC[1] function is selected. (1, 1, 1) : FRACN_OUT_FPGA[0] function is selected.
[3]	MFP [3]	P2.3 Alternate Function Select Bit Bits EXT[3] (SYS_P2_MFP[19]), ALT[3] (SYS_P2_MFP[11]), and MFP[3] (SYS_P2_MFP[3]) determine the P2.3 function. (0, 0, 0) : GPIO function is selected.

		(0, 0, 1) : DBG[25] function is selected. (0, 1, 0) : PWM0_CH1 function is selected. (0, 1, 1) : I2C1_SDA function is selected. (1, 0, 0) : SPI1_SS function is selected. (1, 0, 1) : SPI3_SS function is selected. (1, 1, 0) : DBG[17] function is selected. (1, 1, 1) : VGA_GC[0] function is selected.
[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) : DA_IN_FPGA[5] function is selected. (0, 1, 0) : PWM0_CH0 function is selected. (0, 1, 1) : I2C1_SCL function is selected. (1, 0, 0) : SPI1_CLK function is selected. (1, 0, 1) : SPI3_CLK function is selected. (1, 1, 0) : DBG[24] function is selected. (1, 1, 1) : DBG[16] function is selected.
[1]	MFP [1]	P2.1 Alternate Function Select Bit Bits EXT[1] (SYS_P2_MFP[17]), ALT[1] (SYS_P2_MFP[9]), and MFP[1] (SYS_P2_MFP[1]) determine the P2.1 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : RXMIX_GC function is selected. (0, 1, 0) : FRACN_OUT_FPGA[4] function is selected. (0, 1, 1) : default. (1, 0, 0) : default. (1, 0, 1) : default. (1, 1, 0) : default. (1, 1, 1) : default.
[0]	MFP [0]	P2.0 Alternate Function Select Bit Bits EXT[0] (SYS_P2_MFP[16]), ALT[0] (SYS_P2_MFP[8]), and MFP[0] (SYS_P2_MFP[0]) determine the P2.0 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : SPI1_SS function is selected. (0, 1, 0) : FRACN_OUT_FPGA[5] function is selected. (0, 1, 1) : DBG[15] function is selected. (1, 0, 0) : SPI3_SS function is selected. (1, 0, 1) : I2C1_SDA function is selected. (1, 1, 0) : DBG[7] function is selected. (1, 1, 1) : GPIO function is selected.

4.1.8.8 Multiple Function Port3 Control Register (SYS_P3_MFP)

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

Bits	Description	
[31:24]	Reserved	Reserved.
[23:16]	EXT[7:0]	P3[7:0] Alternate Function Selection Extension The pin function of P3 depends on EXT, MFP and ALT.
[15:8]	ALT[7:0]	P3[7:0] Alternate Function Select Bit The pin function of P3 depends on EXT, MFP and ALT.
[7]	MFP [7]	P3.7 Alternate Function Select Bit Bits EXT[7] (SYS_P3_MFP[23]), ALT[7] (SYS_P3_MFP[15]), and MFP[7] (SYS_P3_MFP[7]) determine the P3.7 function.
[6]	MFP [6]	P3.6 Alternate Function Select Bit Bits EXT[6] (SYS_P3_MFP[22]), ALT[6] (SYS_P3_MFP[14]), and MFP[6] (SYS_P3_MFP[6]) determine the P3.6 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : TM1_EXT/TM1 function is selected. (0, 1, 0) : GPIO function is selected. (0, 1, 1) : SPI1_SS function is selected. (1, 0, 0) : SPI3_SS function is selected. (1, 0, 1) : DBG[23] function is selected. (1, 1, 0) : DBG[15] function is selected. (1, 1, 1) : DA_IN_FPGA[1] function is selected.
[5]	MFP [5]	P3.5 Alternate Function Select Bit Bits EXT[5] (SYS_P3_MFP[21]), ALT[5] (SYS_P3_MFP[13]), and MFP[5] (SYS_P3_MFP[5]) determine the P3.5 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : TM1_CNT_OUT function is selected. (0, 1, 0) : I2C0_SCL function is selected. (0, 1, 1) : DBG[22] function is selected. (1, 0, 0) : DBG[14] function is selected. (1, 0, 1) : GPIO function is selected. (1, 1, 0) : DA_IN_FPGA[0] function is selected. (1, 1, 1) : default.
[4]	MFP [4]	P3.4 Alternate Function Select Bit Bits EXT[4] (SYS_P3_MFP[20]), ALT[4] (SYS_P3_MFP[12]), and MFP[4] (SYS_P3_MFP[4]) determine the P3.4 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : TM0_CNT_OUT function is selected. (0, 1, 0) : I2C0_SDA function is selected. (0, 1, 1) : DBG[21] function is selected.

		(1, 0, 0) : DBG[13] function is selected. (1, 0, 1) : GPIO function is selected. (1, 1, 0) : P_CLK function is selected. (1, 1, 1) : default.
[3]	MFP [3]	P3.3 Alternate Function Select Bit Bits EXT[3] (SYS_P3_MFP[19]), ALT[3] (SYS_P3_MFP[11]), and MFP[3] (SYS_P3_MFP[3]) determine the P3.3 function. Reserved.
[2]	MFP [2]	P3.2 Alternate Function Select Bit Bits EXT[2] (SYS_P3_MFP[18]), ALT[5] (SYS_P3_MFP[10]), and MFP[2] (SYS_P3_MFP[2]) determine the P3.2 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : INT0 function is selected. (0, 1, 0) : TM0_EXT/TM0 function is selected. (0, 1, 1) : STADC function is selected. (1, 0, 0) : SPI0_MISO function is selected. (1, 0, 1) : SPI2_MISO function is selected. (1, 1, 0) : DBG[20] function is selected. (1, 1, 1) : DBG[12] function is selected.
[1]	MFP [1]	P3.1 Alternate Function Select Bit Bits EXT[1] (SYS_P3_MFP[17]), ALT[1] (SYS_P3_MFP[9]), and MFP[1] (SYS_P3_MFP[1]) determine the P3.1 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : PWM0_CH5 function is selected. (0, 1, 0) : UART1_TXD function is selected. (0, 1, 1) : ADC_CH7 function is selected. (1, 0, 0) : UART0_RTS function is selected. (1, 0, 1) : DBG[19] function is selected. (1, 1, 0) : DBG[27] function is selected. (1, 1, 1) : default.
[0]	MFP [0]	P3.0 Alternate Function Select Bit Bits EXT[0] (SYS_P3_MFP[16]), ALT[0] (SYS_P3_MFP[8]), and MFP[0] (SYS_P3_MFP[0]) determine the P3.0 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : PWM0_CH7 function is selected. (0, 1, 0) : UART1_RXD function is selected. (0, 1, 1) : ADC_CH6 function is selected. (1, 0, 0) : UART0_CTS function is selected. (1, 0, 1) : DBG[18] function is selected. (1, 1, 0) : DBG[26] function is selected. (1, 1, 1) : default.

4.1.8.9 Multiple Function Port4 Control Register (SYS_P4_MFP)

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

Bits	Description	
[31:24]	Reserved	Reserved.
[23:16]	EXT[7:0]	P4[7:0] Alternate Function Selection Extension The pin function of P4 depends on EXT, MFP and ALT.
[15:8]	ALT[7:0]	P4[7:0] Alternate Function Select Bit The pin function of P4 depends on EXT, MFP and ALT.
[7]	MFP [7]	P4.7 Alternate Function Select Bit Bits EXT[7] (SYS_P4_MFP[23]), ALT[7] (SYS_P4_MFP[15]), and MFP[7] (SYS_P4_MFP[7]) determine the P4.7 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : ICE_DAT function is selected. (0, 1, 0) : UART1_TXD function is selected. (0, 1, 1) : I2C0_SDA function is selected. (1, 0, 0) : SPI1_SS function is selected. (1, 0, 1) : SPI2_SS function is selected. (1, 1, 0) : FLASH_DIO function is selected. (1, 1, 1) : reserved.
[6]	MFP [6]	P4.6 Alternate Function Select Bit Bits EXT[6] (SYS_P4_MFP[22]), ALT[6] (SYS_P4_MFP[14]), and MFP[6] (SYS_P4_MFP[6]) determine the P4.6 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : ICE_CLK function is selected. (0, 1, 0) : UART1_RXD function is selected. (0, 1, 1) : I2C0_SCL function is selected. (1, 0, 0) : SPI1_CLK function is selected. (1, 0, 1) : SPI2_CLK function is selected. (1, 1, 0) : FLAHS_CLK function is selected. (1, 1, 1) :default.
[5]	MFP [5]	P4.5 Alternate Function Select Bit Bits EXT[5] (SYS_P4_MFP[21]), ALT[5] (SYS_P4_MFP[13]), and MFP[5] (SYS_P4_MFP[5]) determine the P4.5 function. Reserved.
[4]	MFP [4]	P4.4 Alternate Function Select Bit Bits EXT[4] (SYS_P4_MFP[20]), ALT[4] (SYS_P4_MFP[12]), and MFP[4] (SYS_P4_MFP[4]) determine the P4.4 function. Reserved.
[3]	MFP [3]	P4.3 Alternate Function Select Bit Bits EXT[3] (SYS_P4_MFP[19]), ALT[3] (SYS_P4_MFP[11]), and MFP[3] (SYS_P4_MFP[3])

		determine the P4.3 function. Reserved.
[2]	MFP [2]	P4.2 Alternate Function Select Bit Bits EXT[2] (SYS_P4_MFP[18]), ALT[5] (SYS_P4_MFP[10]), and MFP[2] (SYS_P4_MFP[2]) determine the P4.2 function. Reserved.
[1]	MFP [1]	P4.1 Alternate Function Select Bit Bits EXT[1] (SYS_P4_MFP[17]), ALT[1] (SYS_P4_MFP[9]), and MFP[1] (SYS_P4_MFP[1]) determine the P4.1 function. Reserved.
[0]	MFP [0]	P4.0 Alternate Function Select Bit Bits EXT[0] (SYS_P4_MFP[16]), ALT[0] (SYS_P4_MFP[8]), and MFP[0] (SYS_P4_MFP[0]) determine the P4.0 function. Reserved.

4.1.8.10 Multiple Function Port5 Control Register (SYS_P5_MFP)

Register	Offset	R/W	Description	Reset Value
SYS_P5_MFP	SYS_BA+0x44	R/W	P5 Multiple Function and Input Type Control Register	0x0000_0800

Bits	Description	
[31:24]	Reserved	Reserved.
[23:16]	EXT[7:0]	P5[7:0] Alternate Function Selection Extension The pin function of P5 depends on EXT, MFP and ALT.
[15:8]	ALT[7:0]	P5[7:0] Alternate Function Select Bit The pin function of P5 depends on EXT, MFP and ALT.
[7]	MFP [7]	P5.7 Alternate Function Select Bit Bits EXT[7] (SYS_P5_MFP[23]), ALT[7] (SYS_P5_MFP[15]), and MFP[7] (SYS_P5_MFP[7]) determine the P5.7 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : DA_IN_FPGA[4] function is selected. (0, 1, 0) : I2C0_SDA function is selected. (0, 1, 1) : PWM0_CH7 function is selected. (1, 0, 0) : SPI1_MISO function is selected. (1, 0, 1) : SPI3_MISO function is selected. (1, 1, 0) : DBG[11] function is selected. (1, 1, 1) : DBG[3] function is selected.
[6]	MFP [6]	P5.6 Alternate Function Select Bit Bits EXT[6] (SYS_P5_MFP[22]), ALT[6] (SYS_P5_MFP[14]), and MFP[6] (SYS_P5_MFP[6]) determine the P5.6 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : DA_IN_FPGA[3] function is selected. (0, 1, 0) : I2C0_SCL function is selected.

		(0, 1, 1) : PWM0_CH6 function is selected. (1, 0, 0) : SPI1_MOSI function is selected. (1, 0, 1) : SPI3_MOSI function is selected. (1, 1, 0) : DBG[10] function is selected. (1, 1, 1) : DBG[2] function is selected.
[5]	MFP [5]	P5.5 Alternate Function Select Bit Bits EXT[5] (SYS_P5_MFP[21]), ALT[5] (SYS_P5_MFP[13]), and MFP[5] (SYS_P5_MFP[5]) determine the P5.5 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : RF_SPI_MISO function is selected. (0, 1, 0) : SPI0_SS function is selected. (0, 1, 1) : PWM0_CH5 function is selected. (1, 0, 0) : SPI2_SS function is selected. (1, 0, 1) : DBG[9] function is selected. (1, 1, 0) : DBG[1] function is selected. (1, 1, 1) : DA_IN_FPGA[2] function is selected.
[4]	MFP [4]	P5.4 Alternate Function Select Bit Bits EXT[4] (SYS_P5_MFP[20]), ALT[4] (SYS_P5_MFP[12]), and MFP[4] (SYS_P5_MFP[4]) determine the P5.4 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : TM2_CNT_OUT function is selected. (0, 1, 0) : TADC_DATL function is selected. (0, 1, 1) : RF_SPI_MOSI function is selected. (1, 0, 0) : SPI0_MOSI function is selected. (1, 0, 1) : SPI2_MOSI function is selected. (1, 1, 0) : DBG[7] function is selected. (1, 1, 1) : DBG[15] function is selected.
[3]	MFP [3]	P5.3 Alternate Function Select Bit Bits EXT[3] (SYS_P5_MFP[19]), ALT[3] (SYS_P5_MFP[11]), and MFP[3] (SYS_P5_MFP[3]) determine the P5.3 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : ADC_CH0 function is selected. (0, 1, 0) : PWM0_CH2 function is selected. (0, 1, 1) : RF_SPI_CSN function is selected. (1, 0, 0) : DBG[0] function is selected. (1, 0, 1) : DBG[8] function is selected. (1, 1, 0) : SPI1_MISO function is selected. (1, 1, 1) : SPI2_MISO function is selected.
[2]	MFP [2]	P5.2 Alternate Function Select Bit Bits EXT[2] (SYS_P5_MFP[18]), ALT[5] (SYS_P5_MFP[10]), and MFP[2] (SYS_P5_MFP[2]) determine the P5.2 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : INT1 function is selected. (0, 1, 0) : EXT_WAKEUP function is selected.

		(0, 1, 1) : FLASH_ENABLE function is selected. (1, 0, 0) : default. (1, 0, 1) : default. (1, 1, 0) : default. (1, 1, 1) : default.
[1]	MFP [1]	P5.1 Alternate Function Select Bit Bits EXT[1] (SYS_P5_MFP[17]), ALT[1] (SYS_P5_MFP[9]), and MFP[1] (SYS_P5_MFP[1]) determine the P5.1 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : UART1_CTS function is selected. (0, 1, 0) : I2C1_SCL function is selected. (0, 1, 1) : UART0_RXD function is selected. (1, 0, 0) : PWM0_CH6 function is selected. (1, 0, 1) : DBG[4] function is selected. (1, 1, 0) : DBG[12] function is selected. (1, 1, 1) : PLL_RSTN_FPGA function is selected.
[0]	MFP [0]	P5.0 Alternate Function Select Bit Bits EXT[0] (SYS_P5_MFP[16]), ALT[0] (SYS_P5_MFP[8]), and MFP[0] (SYS_P5_MFP[0]) determine the P5.0 function. (0, 0, 0) : GPIO function is selected. (0, 0, 1) : UART1_RTS function is selected. (0, 1, 0) : I2C1_SDA function is selected. (0, 1, 1) : UART0_TXD function is selected. (1, 0, 0) : PWM0_CH7 function is selected. (1, 0, 1) : DBG[5] function is selected. (1, 1, 0) : DBG[13] function is selected. (1, 1, 1) : CAL_DONE_FPGA function is selected.

4.1.8.11 Test Control Register (SYS_TESTCTL)

Register	Offset	R/W	Description	Reset Value
SYS_TESTCTL	SYS_BA+0x48	R/W	MDM Test Control	0x0000_0000

Bits	Description	
[31:27]	Reserved	Reserved
[26:24]	DGP3_SEL	Select which internal 8bit debug info muxed to DBG[31:24] Same as DGP0_SEL description
[23:19]	Reserved	Reserved
[18:16]	DGP2_SEL	Select which internal 8bit debug info muxed to DBG[23:16] Same as DGP0_SEL description
[15:11]	Reserved	Reserved
[10:8]	DGP1_SEL	Select which internal 8bit debug info muxed to DBG[15:8] Same as DGP0_SEL description

[7:3]	Reserved	Reserved
[2:0]	DGP0_SEL	Select which internal 8bit debug info muxed to DBG[7:0] 3'b000:MDM_DBG[7:0] 3'b001:MDM_DBG[15:8] 3'b010:MDM_DBG[23:16] 3'b011:MDM_DBG[31:24] 3'b100:BLE_DBG[7:0] 3'b101:BLE_DBG[15:8] 3'b110:BLE_DBG[23:16] 3'b111:BLE_DBG[31:24]

4.1.8.12 BOD LVR De-Bounce Control Register (SYS_BLDBCTL)

Register	Offset	R/W	Description	Reset Value
SYS_BLDBCTL	SYS_BA+0x4C	R/W	BOD LVR De-Bounce Control Register	0x0000_2020

Bits	Description	
[31:14]	Reserved	Reserved.
[13:8]	LVRDBSEL	LVR De-Bounce (glitch) time Control register [0]=1: about 2^4 ahb clk clock [1]=1: about 2^7 ahb clk clock [2]=1: about 2^9 ahb clk clock [3]=1: about 2^11 ahb clk clock [4]=1: about 2^13 ahb clk clock [5]=1: about 2^15 ahb clk 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.
[7:6]	Reserved	Reserved.
[5:0]	BODDBSEL	BOD De-Bounce (glitch) time Control register [0]=1: about 2^4 PLL clock [1]=1: about 2^7 PLL clock [2]=1: about 2^9 PLL clock [3]=1: about 2^11 PLL clock [4]=1: about 2^13 PLL clock [5]=1: about 2^15 PLL 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.

4.1.8.13 Register Write-Protection Control Register (SYS_REGLCTL)

Some of the system control registers need to be protected to avoid inadvertent write and disturb the chip operation. These system control registers are protected after the power on reset till user to

disable register protection. For user to programs these protected registers, a register protection disable sequence needs to be followed by a special programming. The register protection disable sequence is writing the data 0x59, 0x16, 0x88 to the register SYS_REGLCTL address at 0x5000_0100 continuously. Any different data value, different sequence or any other write to other address during these three data writing will abort the whole sequence.

After the protection is disabled, user can check the protection disable bit at address 0x5000_0100 bit 0, 1 is protection disable, 0 is protection enable. Then user can update the target protected register value and then write any data to the address 0x5000_0100 to enable the register protection.

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

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

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	REGLCTL	Register Write-protection Code (Write Only) Some registers have write-protection function. Writing these registers have to disable the protected function by writing the sequence value 0x59, 0x16, 0x88 to this field. After this sequence is completed, the REGLCTL bit will be set to 1 and write-protection registers can be normal write. Register Write-protection Disable Index (Read Only) 0 = Write-protection Enabled for writing protected registers. Any write to the protected register is ignored. 1 = Write-protection Disabled for writing protected registers. Protected registers are listed below:

Register	Address	Note
SYS_IPRST0	0x5000_0008	Peripheral Reset Control Register 0
SYS_BODCTL	0x5000_0018	Brown-out Detect Control Register
SYS_PWRCTL	0x5000_0200	Bit[6] is not protected for power wake-up interrupt clear
CLK_APBCLK bit[0]	0x5000_0208	Bit[0] is watchdog clock enable
CLK_CLKSEL0	0x5000_0210	HCLK and CPU STCLK BB clock source select
CLK_CLKSEL1 bit1[1:0]	0x5000_0214	Watchdog clock source select
ISPCTL	0x5000_C000	Flash ISP Control
ISPTRG	0x5000_C010	ISP Trigger Control
FATCTL	0x5000_C018	Flash Access Time Control
WDT_CTL	0x4000_4000	Watchdog Timer Control
WDT_ALTCTL	0x4000_4004	Watchdog Timer Alternative Control

Note: The bits which are write-protected will be noted as" **(Write Protect)**" beside the description.

4.1.8.14 Register to reflect the status of ROM Mode (SYS_STATUS)

Register	Offset	R/W	Description	Reset Value
SYS_ STATUS	SYS_BA+0x104	R/W	Register to reflect the status of ROM Mode	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	STATUS	Register to reflect the status of ROM Mode(Read Only) 0xA5,0xA6,0xA7,0xA8,0xA9: ROM Mode Others: Flash Mode **Note**: the value of this register is reflecting the command Host has issued, when PAD_NRST is active low, host transmits command by P46 P47.

4.1.9 System Timer (SysTick)

The PAN1020MCU 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.

4.1.9.1 System Timer Control Register Map

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

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
SYST_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

4.1.9.2 System Timer Control Register

4.1.9.2.1. 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.

4.1.9.2.2. 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.

4.1.9.2.3. 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).

4.1.10 Nested Vectored Interrupt Controller (NVIC)

4.1.10.1 Overview

The PAN1020MCU 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.

4.1.10.2 Features

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

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

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

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

4.1.10.3 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 4-6 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 4-7 System Interrupt Map Vector Table

Exception Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Source Mod-ule	Interrupt Description	Power-Down Wake-Up
1 ~ 15	-	-	-	System exceptions	-
16	0	DMA	DMA	DMA interrupt	Yes
17	1	WDT_INT	WDT/WWDT	Watchdog Timer interrupt	Yes
18	2	EINT0	GPIO	External signal interrupt from P3.2 pin	Yes
19	3	EINT1	GPIO	External signal interrupt from P5.2 pin	Yes
20	4	GP0/1_INT	GPIO	External signal interrupt from GPIO group P0~P1	Yes
21	5	GP2/3/4_INT	GPIO	External signal interrupt from GPIO group P2~P4 ex-cept P3.2	Yes
22	6	PWM_INT	PWM	PWM interrupt	No
23	7	SPI2_INT	SPI2	SPI2 interrupt	No
24	8	TMR0_INT	TMR0	Timer 0 interrupt	Yes
25	9	TMR1_INT	TMR1	Timer 1 interrupt	Yes
26	10	TMR2_INT	TMR2	Timer 2 interrupt	Yes
27	11	BOD_INT	BOD	BOD interrupt	No
28	12	UART0_INT	UART0	UART0 interrupt	Yes
29	13	UART1_INT	UART1	UART1 interrupt	Yes
30	14	SPI0_INT	SPI0	SPI0 interrupt	No
31	15	SPI1_INT	SPI1	SPI1 interrupt	No
32	16	GP5_INT	GPIO	External signal interrupt from GPIO group P5 except	Yes

				P5.2	
33	17	ANAC_INT	ANAC	ANAC interrupt	No
34	18	I2C0_INT	I2C0	I2C0 interrupt	Yes
35	19	I2C1_INT	I2C1	I2C1 interrupt	No
36	20	BLE_RADIOCNTL_INT	BLE	BLE_RADIOCNTL interrupt	No
37	21	BLE_FINETGTIM_INT	BLE	BLE_FINE_TGTIM interrupt	No
38	22	BLE_GROSSTGTIM_INT	BLE	BLE_ GROSSTGTIM interrupt	No
39	23	BLE_ERROR_INT	BLE	BLE_ERROR interrupt	No
40	24	BLE_CRYPT_INT	BLE	BLE_CRYPT interrupt	No
41	25	SPI3_INT	SPI3	SPI3 interrupt	Yes
42	26	BLE_EVENT_INT	BLE	BLE_EVENT interrupt	No
43	27	BLE_SLP_INT	BLE	BLE_SLP interrupt	No
44	28	PWRWU_INT	CLKC	Clock controller interrupt for chip wake-up from Power-down state	Yes
45	29	ADC_INT	ADC	ADC interrupt	No
46	30	BLE_RX_INT	BLE	BLE_RX interrupt	No
47	31	BLE_CSCNT_INT	BLE	BLE_CSCNT interrupt	No

4.1.10.4 Vector Table

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

Table 4-8 Vector Table Format

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

4.1.10.5 Operation Description

NVIC interrupts can be enabled and disabled by writing to their corresponding Interrupt Set-Enable or Interrupt Clear-Enable register bit-field. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current enabled state of the corresponding interrupts. When an interrupt is disabled, interrupt assertion will cause the interrupt to become Pending; however, the

interrupt will not be activated. If an interrupt is Active when it is disabled, it remains in its Active state until cleared by reset or an exception return. Clearing the enable bit prevents new activations of the associated interrupt.

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

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

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

4.1.10.6 NVIC Control Register Map

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

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

4.1.10.7 NVIC Control Register Description

4.1.10.7.1. 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.

4.1.10.7.2. 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.

4.1.10.7.3. 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.

4.1.10.7.4. 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.

4.1.10.7.5. 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.

4.1.10.7.6. 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.

4.1.10.7.7. 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.

4.1.10.7.8. 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.

4.1.10.7.9. 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.

4.1.10.7.10. 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.

4.1.10.7.11. 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.

4.1.10.7.12. 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.

4.1.11 System Control Block Registers (SCB)

The PAN1020MCU status and operating mode control are managed System Control Block Registers. Including CPUID, PAN1020MCU interrupt priority and PAN1020MCU power management can be controlled through these system control registers.

4.1.11.1 System Control Block Register Map

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

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
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

4.1.11.2 System Control Block Register Description

4.1.11.2.1. 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.

4.1.11.2.2. 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.

4.1.11.2.3. 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.

4.1.11.2.4. 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.

4.1.11.2.5. 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.

4.1.11.2.6. 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.

4.2 Clock Controller

4.2.1 Overview

The clock controller generates clocks for the whole chip, including system clocks and all peripheral clocks. The clock controller also implements the power control function with the individually clock ON/OFF control, clock source selection and clock divider. The chip enters Power-down mode when PAN1020MCU executes the WFI instruction only if the PDEN (CLK_PWRCTL[7]) bit is set to 1. After that, chip enters Power-down mode and waits for wake-up interrupt source triggered to exit Power-down mode. In Power-down mode, the clock controller turns off the 16 MHz external high speed crystal (HXT) to reduce the overall system power consumption. The following figures show the clock generator and the overview of the clock source control.

The clock generator consists of 2 external clock sources as listed below:

- 16 MHz external high speed crystal oscillator (HXT)
- 32 KHz internal low speed RC oscillator (LIRC)

There are four clock for digital region, which are generated by the 2 external clock source(HXT and LIRC).

- 26 MHz internal DPLL(DPLL_26M)
- 16 MHz external high speed crystal oscillator (HXT)
- 32 KHz internal low speed RC oscillator (LIRC)
- 32 KHz external high speed crystal oscillator (LXT)

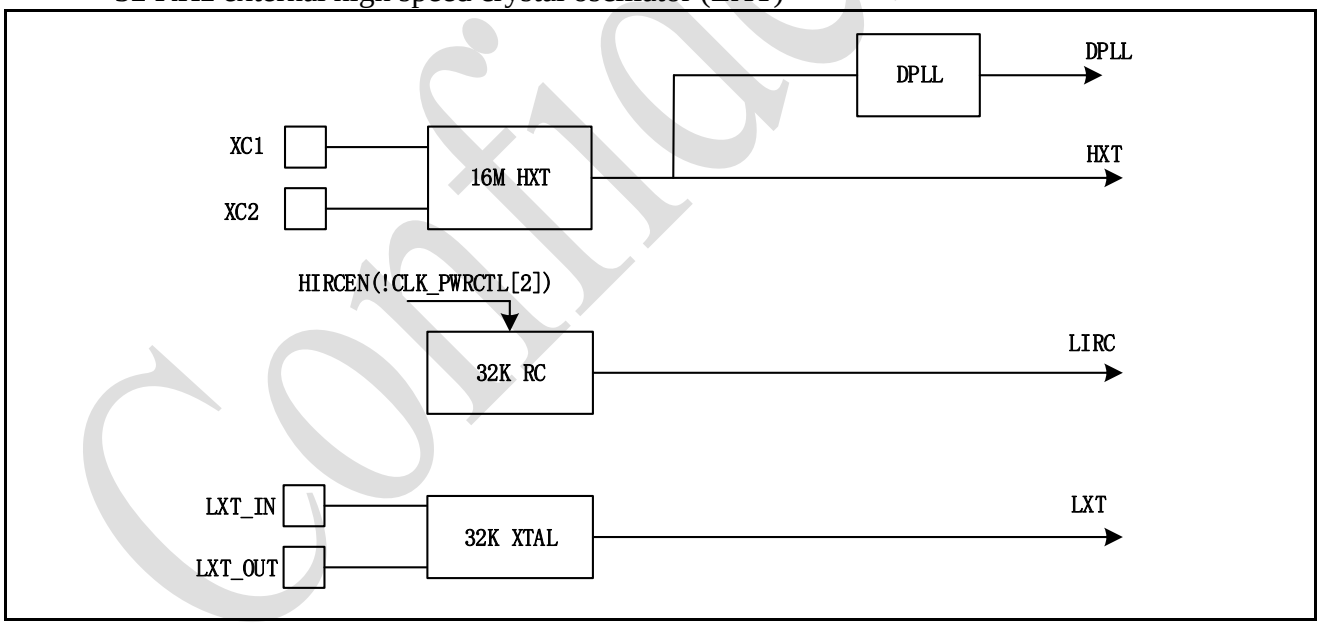


Figure 4-8 Clock Generator Block Diagram

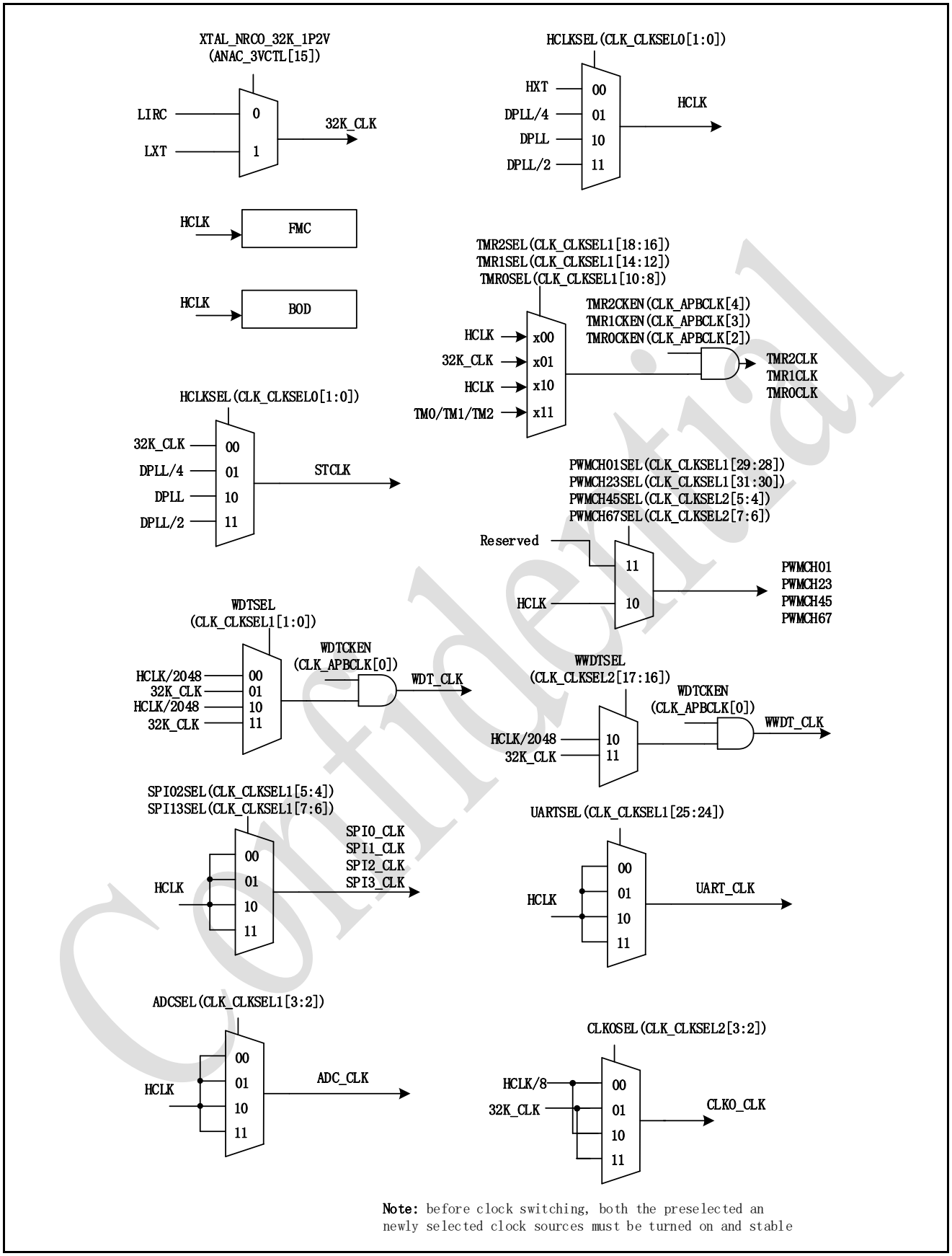


Figure 4-9 Clock Generator Global View Diagram

4.2.2 System Clock and SysTick Clock

The system clock(HCLK) has 4 clock sources which were generated from clock generator block. The clock source switch depends on the register HCLKSEL (CLK_CLKSEL0[1:0]). The block diagram is shown in Figure 4-10.

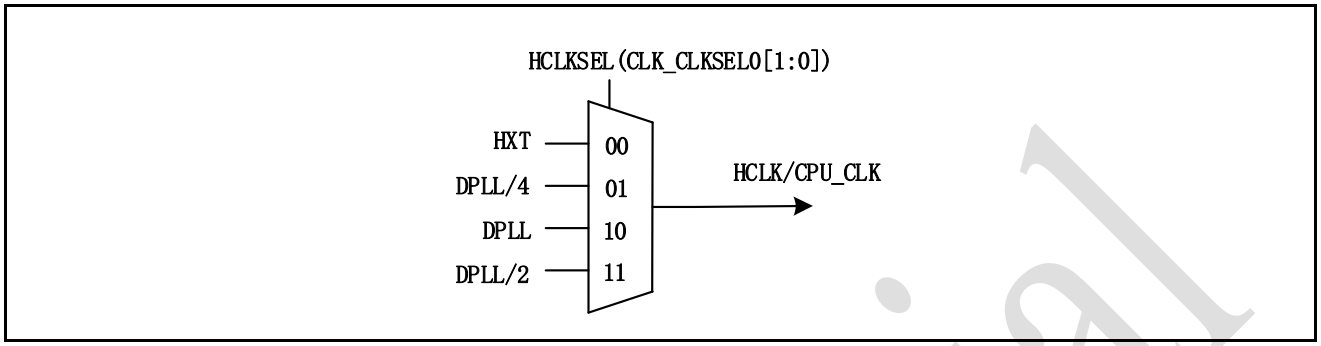


Figure 4-10 System Clock Block Diagram

The clock source of SysTick in PAN1020MCU can use CPU clock or external clock CLKSRC(SYST_CTRL[2]). If using external clock, the SysTick clock (STCLK) has 4 clock sources. The clock source switch depends on the setting of the register STCLKSEL (CLK_CLKSEL0[1:0]). The block diagram is shown in Figure 4-11.

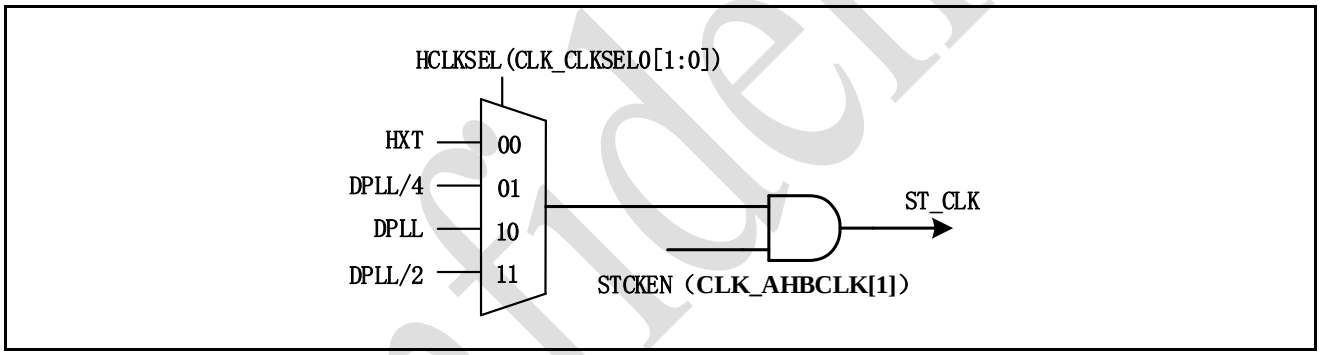


Figure 4-11 SysTick Clock Control Block Diagram

4.2.3 Peripherals Clock Source Selection

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

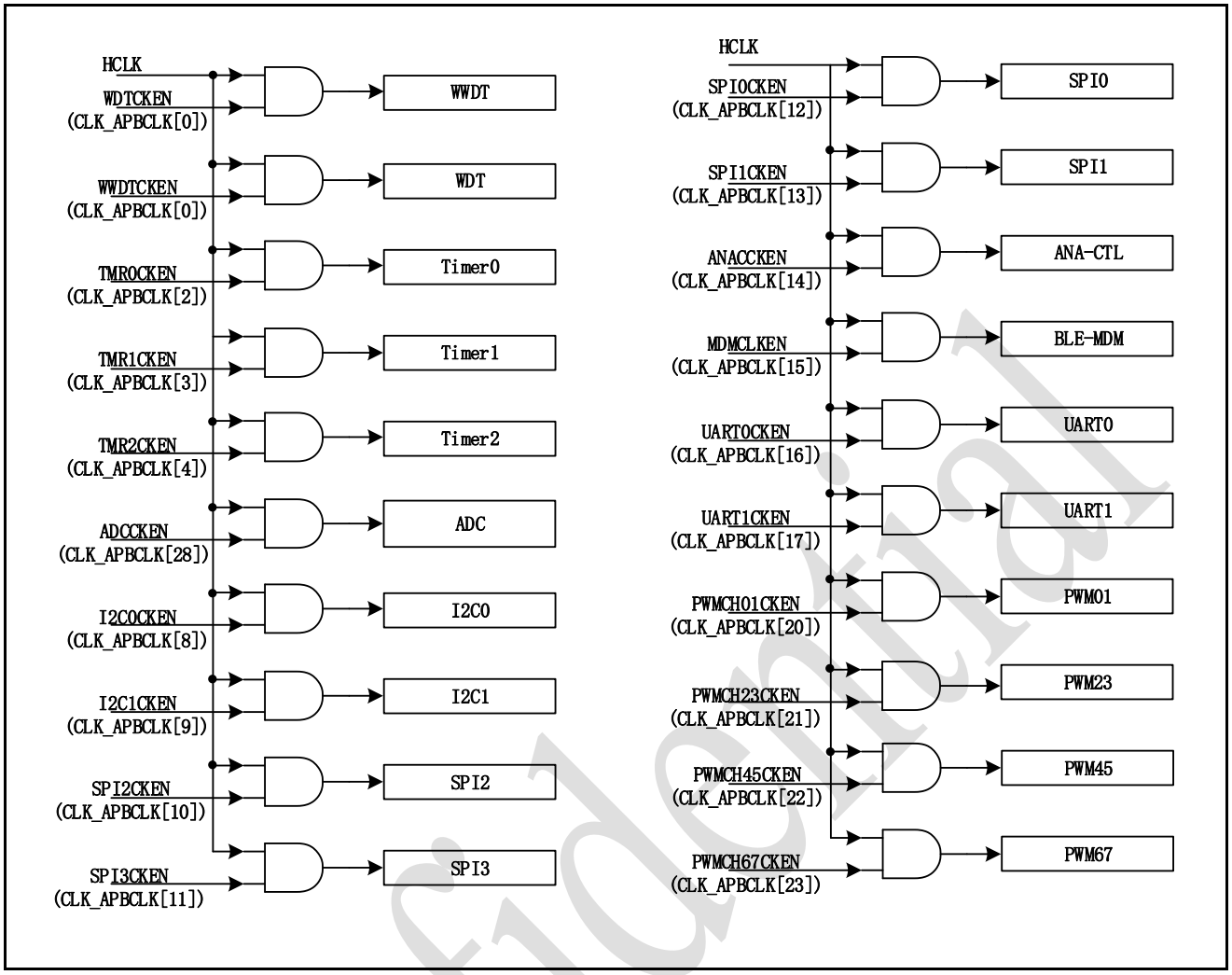


Figure 4-12 Peripherals Bus Clock Source Selection for HCLK

4.2.4 Power-down Mode Clock

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

The clocks still kept active are listed below:

- Clock Generator
- 32 KHz clock (32K_CLK)
- Peripherals Clock (When 32 kHz low speed oscillator is adopted as clock source)
  - Watchdog Clock
  - Timer 0/1/2 Clock

4.2.5 Frequency Divider Output

The clock output pin is port0.2(P0.2)

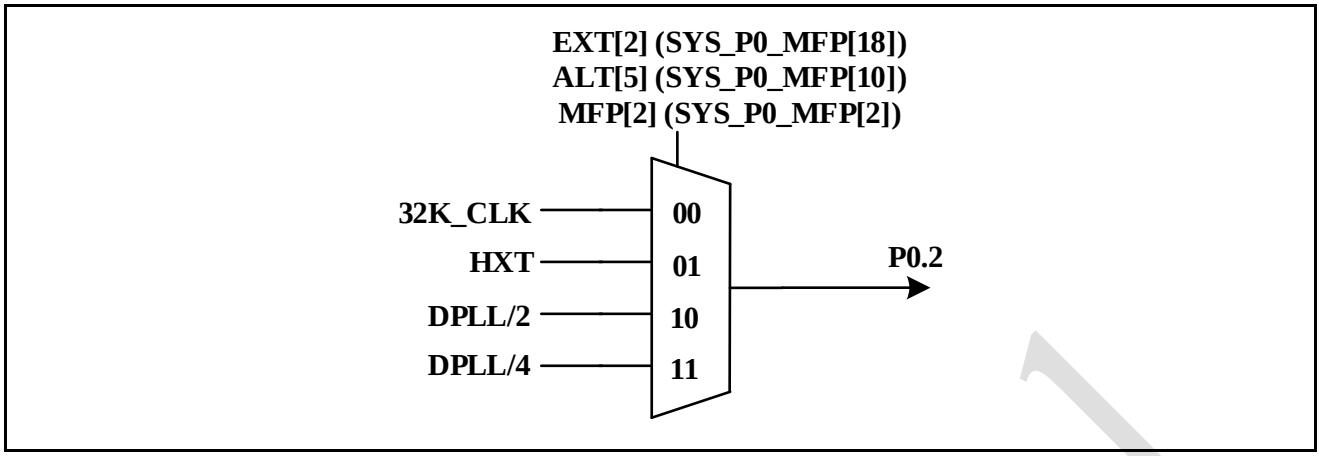


Figure 4.2-6 Block Diagram of Frequency Divider

4.2.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
CLK Base Address: **CLK_BA = 0x5000_0200**				
CLK_PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x0000_0023
CLK_AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_0006
CLK_APBCLK	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register	0x0000_0001
CLK_STATUS	CLK_BA+0x0C	R/W	Clock Status Monitor Register	0x0000_000D
CLK_CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0000_0000
CLK_CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0x0000_0000
CLK_CLKDIV	CLK_BA+0x18	R/W	Clock Divider Number Register	0x0000_0000
CLK_CLKSEL2	CLK_BA+0x1C	R/W	Clock Source Select Control Register 2	0x0000_0000
CLK_WAKEUP-CTL	CLK_BA+0x28	R/W	Wake Up stable time control register	0x0840_0040
CLK_LPCTL	CLK_BA+0x2C	R/W	Low Power control register	0x0000_0000

4.2.7 Register Description

4.2.7.1 Power-down Control Register (CLK_PWRCTL)

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

Register	Offset	R/W	Description	Reset Value
CLK_PWRCTL	CLK_BA+0x00	R/W	System Power-down Control Register	0x0000_0023

Bits	Description	
[31:11]	Reserved	Reserved.

[10]	PDVD18	Reserved (Write Protect)
[9]	Reserved	Reserved.
[8]	PDIBG	Reserved (Write Protect)
[7]	PDEN	System Power-down Enable Bit (Write Protect) When chip wakes up from Power-down mode. User needs to set this bit again for next Power-down. In Power-down mode,16MHz external high speed crystal oscillator (HXT) will be disabled in this mode, and 32 kHz internal low speed RC oscillator (LIRC) are not controlled by Power-down mode. In Power-down mode, the system clock are disabled, and ignored the clock source selection. The clocks of peripheral are not controlled by Power-down mode, if the peripheral clock source is from 32 kHz internal low speed oscillator. 0 = Chip operating normally or chip in Idle mode because of WFI command. 1 = Chip enters Power-down mode instantly or waits CPU sleep command WFI.
[6]	PDWKIF	Power-down Mode Wake-up Interrupt Status Set by “Power-down wake-up event”, which indicates that resume from Power-down mode” The flag is set if the GPIO, UART, WDT, Timer or BOD wake-up occurred. **Note**: This bit works only if PDWKIEN (CLK_PWRCTL[5]) set to 1. Write 1 to clear the bit to 0.
[5]	PDWKIEN	Power-down Mode Wake-up Interrupt Enable Bit (Write Protect) 0 = Power-down mode wake-up interrupt Disabled. 1 = Power-down mode wake-up interrupt Enabled. **Note**: The interrupt will occur when both PDWKIF and PDWKIEN are high.
[4]	Reserved	Reserved.
[3]	PDLIRC	Reserved (Write Protect)
[2]	PDHIRC	Reserved (Write Protect)
[1]	PDXTLBUF	Reserved (Write Protect)
[0]	PDXTL	Reserved (Write Protect)

Table 4-9 Power-down Mode Control

Register Or Instruction Mode	**SLEEPDEEP (SCR[2])**	**PDEN (CLK_PWRCTL[7])**	**CPU Run WFI Instruction**	**Clock Disable**
Normal operation	0	0	NO	All clocks disabled by control register
Idle mode (CPU entering Sleep mode)	0	0	YES	Only CPU clock disabled
Power-down mode (CPU entering Deep Sleep mode)	1	1	YES	Most clocks are disabled except 32 kHz and only WDT peripheral clock still enable if its peripheral clock source is selected as 32 kHz.

When chip enters Power-down mode, user can wake-up this chip using some interrupt sources. The related interrupt sources and NVIC IRQ enable bits (NVIC_ISER) should be enabled before setting PDEN bit in CLK_PWRCTL[7] to ensure chip can enter Power-down and wake-up successfully.

4.2.7.2 AHB Devices Clock Enable Control Register (CLK_AHBCLK)

The bits in this register are used to enable/disable clock for system clock.

Register	Offset	R/W	Description	Reset Value
CLK_AHBCLK	CLK_BA+0x04	R	AHB Devices Clock Enable Control Register	0x0000_0006

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

4.2.7.3 APB Devices Clock Enable Control Register (CLK_APBCLK)

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

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

Bits	Description	
[31:29]	Reserved	Reserved.
[28]	ADCCKEN	Analog-digital-converter (ADC) Clock Enable Bit 0 = ADC peripheral clock Disabled. 1 = ADC peripheral clock Enabled.
[27:24]	Reserved	Reserved.
[23]	PWMCH67CKEN	PWM_67 Clock Enable Bit 0 = PWM67 clock Disabled. 1 = PWM67 clock Enabled.
[22]	PWMCH45CKEN	PWM_45 Clock Enable Bit 0 = PWM45 clock Disabled. 1 = PWM45 clock Enabled.

[21]	PWMCH23CKEN	PWM_23 Clock Enable Bit 0 = PWM23 clock Disabled. 1 = PWM23 clock Enabled.
[20]	PWMCH01CKEN	PWM_01 Clock Enable Bit 0 = PWM01 clock Disabled. 1 = PWM01 clock Enabled.
[19:18]	Reserved	Reserved.
[17]	UART1CKEN	UART1 Clock Enable Bit 0 = UART1 clock Disabled. 1 = UART1 clock Enabled.
[16]	UART0CKEN	UART0 Clock Enable Bit 0 = UART0 clock Disabled. 1 = UART0 clock Enabled.
[15]	MDMCLKEN	BLE-MDM CLK Clock Enable Bit 0 = MDM CLK Disabled. 1 = MDM CLK Enabled.
[14]	ANACCKEN	Analog Ctrl CLK Enable Bit 0 = ANAC CLK Disabled 1 = ANAC CLK Enabled
[13]	SPI1CKEN	SPI1 Clock Enable Bit 0 = SPI1 peripheral clock Disabled. 1 = SPI1 peripheral clock Enabled.
[12]	SPI0CKEN	SPI0 Clock Enable Bit 0 = SPI0 peripheral clock Disabled. 1 = SPI0 peripheral clock Enabled.
[11]	SPI3CKEN	SPI3 Clock Enable Bit 0 = SPI3 peripheral clock Disabled. 1 = SPI3 peripheral clock Enabled.
[10]	SPI2CKEN	SPI2 Clock Enable Bit 0 = SPI2 peripheral clock Disabled. 1 = SPI2 peripheral clock Enabled.
[9]	I2C1CKEN	I2C1 Clock Enable Bit 0 = I2C1 clock Disabled. 1 = I2C1 clock Enabled.
[8]	I2C0CKEN	I2C0 Clock Enable Bit 0 = I2C0 clock Disabled. 1 = I2C0 clock Enabled.
[7:6]	Reserved	Reserved.
[5]	Reserved	Reserved.
[4]	TMR2CKEN	Timer2 Clock Enable Bit 0 = Timer2 clock Disabled. 1 = Timer2 clock Enabled.
[3]	TMR1CKEN	Timer1 Clock Enable Bit

		0 = Timer1 clock Disabled. 1 = Timer1 clock Enabled.
[2]	TMR0CKEN	Timer0 Clock Enable Bit 0 = Timer0 clock Disabled. 1 = Timer0 clock Enabled.
[1]	Reserved	Reserved.
[0]	WDTCKEN	WDT/WWDT Clock Enable Bit (Write Protect) 0 = WDT/WWDT clock Disabled. 1 = WDT/WWDT clock Enabled. **Note**: This bit is the protected bit, and programming it needs to write 0x59, 0x16, and 0x88 to address 0x5000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address SYS_BA + 0x100.

4.2.7.4 Clock Status Register (CLK_STATUS)

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

Register	Offset	R/W	Description	Reset Value
CLK_STATUS	CLK_BA+0x0C	R	Clock Status Monitor Register	0x0000_000D

Bits	Description	
[31:6]	Reserved	Reserved.
[5]	CLK_STA-TUS	Sleep mode identifier (Read Only) 0 = active mode 1 = sleep mode
[4]	32k LXT stb	32K RCO Clock Source Stable Flag (Read Only) 0 = LXT clock is not stable or disabled. 1 = LXT clock is stable and enabled.
[3]	32k LIRC stb	32K XTAL Clock Source Stable Flag (Read Only) 0 = LIRC clock is not stable or disabled. 1 = LIRC clock is stable and enabled.
[2]	DPLLSTB	Internal DPLL Clock Source Stable Flag (Read Only) 0 = Internal DPLL clock is not stable or disabled. 1 = Internal DPLL clock is stable and enabled.
[1]	Reserved	Reserved.
[0]	HXTSTB	HXT Clock Source Stable Flag(Read Only) 0 = HXT clock is not stable or disabled. 1 = HXT clock is stable and enabled.

4.2.7.5 Clock Source Select Control Register 0 (CLK_CLKSEL0)

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0000_0000

Bits	Description	
[31:2]	Reserved	Reserved.
[1:0]	HCLKSEL Or STCLKSEL	HCLK Clock Source Selection (it is STCLKSEL too) (Write Protect) 00 = Clock source is from HXT (default). 01 = Clock source is from 13M. 10 = Reserved 11 = Clock source is from 26M (normal work). **Note1**: Before clock switching, the related clock sources (both pre-select and new-select) must be turn-on and stable. **Note2**: These bits are protected bit, and programming them needs to write 0x59, 0x16, and 0x88 to address 0x5000_0100 to disable register protection. Refer to the register SYS_REGLCTL at address SYS_BA + 0x100.

4.2.7.6 Clock Source Select Control Register 1 (CLK_CLKSEL1)

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

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0x0000_0000

Bits	Description	
[31:30]	PWMCH23SEL	PWM2 and PWM3 use the same peripheral clock source; Both of them use the same pre-scaler, and Clock source is always from HCLK
[29:28]	PWMCH01SEL	PWM0 and PWM1 use the same peripheral clock source; Both of them use the same pre-scaler, and Clock source is always from HCLK
[27:26]	Reserved	Reserved.
[25:24]	UARTSEL	UART Clock Source is always from HCLK.
[23:19]	Reserved	Reserved.
[18:16]	TMR2SEL	TIMER2 Clock Source Selection x00 = Clock source is from HCLK. x01 = Clock source is from 32K_CLK. x10 = Clock source is from HCLK. x11 = Clock source is from external trigger TM2.
[15]	Reserved	Reserved.
[14:12]	TMR1SEL	TIMER1 Clock Source Selection x00 = Clock source is from HCLK. x01 = Clock source is from 32K_CLK. x10 = Clock source is from HCLK.

		x11 = Clock source is from external trigger TM1.
[11]	Reserved	Reserved.
[10:8]	TMR0SEL	TIMER0 Clock Source Selection x00 = Clock source is from HCLK. x01 = Clock source is from 32K_CLK. x10 = Clock source is from HCLK. x11 = Clock source is from external trigger TM0.
[7:6]	SPI13SEL	SPI1 and SPI3 Clock Source is always from HCLK
[5:4]	SPI02SEL	SPI0 and SPI2 Clock Source is always from HCLK
[3:2]	ADCSEL	ADC Peripheral Clock is always from HCLK
[1:0]	WDTSEL	WDT CLK Clock Source Selection (Write Protect) 00 = Clock source is from HCLK/2048 clock. 01 = Clock source is from 32K_CLK. 10 = Clock source is from HCLK/2048 clock. 11 = Clock source is from 32K_CLK.

4.2.7.7 Clock Divider Register (CLK_CLKDIV)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV	CLK_BA+0x18	R/W	Clock Divider Number Register	0x0000_0000

Bits	Description	
[31:0]	Reserved	Reserved.

4.2.7.8 Clock Source Select Control Register (CLK_CLKSEL2)

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

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL2	CLK_BA+0x1C	R/W	Clock Source Select Control Register 2	0x0000_0000

Bits	Description	
[31:18]	Reserved	Reserved.
[17:16]	WWDTSEL	Window Watchdog Timer Clock Source Selection 00 = Reserved. 01 = Reserved. 10 = Clock source from HCLK/2048 clock. 11 = Clock source from 32K_CLK.
[15:8]	Reserved	Reserved.
[7:6]	PWMCH67SEL	PWM6 and PWM7 use the same peripheral clock source; Both of them use the same pre-scaler, and Clock source is always from HCLK

[5:4]	PWMCH45SEL	PWM4 and PWM5 use the same peripheral clock source; Both of them use the same pre-scaler, and Clock source is always from HCLK
[3:0]	Reserved	Reserved.

4.2.7.9 Wakeup Control Register (CLK_WAKEUPCTL)

The Register takes effect by setting COFIG_ACT (see ANAC_3VCTL[0]), and be synchronized to 3V low power domain

Register	Offset	R/W	Description	Reset Value
CLK_WAKEUP-CTL	CLK_BA+0x28	R/W	Wakeup Control Register	0x0840_0040

Bits	Description	
[31:30]	Reserved	Reserved.
[29:24]	OSCLWU-SEL	LIRC stable time select The default value is set by flash controller user configuration register (CONFIG0[31:26]) bit. [0]=1: about 2^0 32K clock [1]=1: about 2^1 32K clock [2]=1: about 2^2 32K clock [3]=1: about 2^3 32K clock (default) [4]=1: about 2^4 32K clock [5]=1: about 2^5 32K clock **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.
[23:16]	PLLWUSEL	PLL stable time select [0]=1: about 2^1 PLL clock [1]=1: about 2^3 PLL clock [2]=1: about 2^4 PLL clock [3]=1: about 2^7 PLL clock [4]=1: about 2^9 PLL clock [5]=1: about 2^11 PLL clock [7]=1: about 2^13 PLL clock (default) [8]=1: about 2^15 PLL clock **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.
[15:8]	Reserved	Reserved.
[7:0]	HXTLWU-SEL	HXTL stable time select [0]=1: about 2^1 HXT clock [1]=1: about 2^3 HXT clock [2]=1: about 2^5 HXT clock [3]=1: about 2^7 HXT clock [4]=1: about 2^9 HXT clock

		[5]=1: about 2^11 HXT clock [6]=1: about 2^13 HXT clock (default) [7]=1: about 2^15 HXT clock **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.

4.2.7.10 Low Power Register (CLK_LPCTL)

Register	Offset	R/W	Description	Reset Value
CLK_LPCTL	CLK_BA+0x2C	R/W	Low Power control register	0x0000_0000

Bits	Description	
[31:10]	Reserved	Reserved.
[9]	PDLVR	Reserved.
[8]	PDBOD	Reserved.
[7]	Reserved.	Reserved.
[6]	PDADC	Reserved.
[5]	PDVD18	Reserved.
[4]	PDLIRC	Reserved.
[3]	PDHIRC	Reserved.
[2]	PDPLL	Reserved.
[1]	PDIBG	Reserved.
[0]	PDXTL	Reserved.

4.3 Flash Memory Controller (FMC)

4.3.1 Overview

The Panchip PAN1020 series is equipped with 256 Kbytes on-chip embedded flash for application and Data Flash to store some application dependent data. A User Configuration block provides for system initialization. A size configurable loader ROM (LDROM) shared with 256K APROM is used for In-System-Programming (ISP) function. A 512bytes security protection ROM (SPROM) can conceal user program. This chip also supports In-Application-Programming (IAP) function, user switches the code executing without the chip reset after the embedded flash updated.

4.3.2 Features

- Supports 253KB (default) application ROM (APROM).
- Supports 2KB (default) loader ROM (LDROM).
- Supports size configurable for APROM and LDROM, the total size of them is 255KB.
- Supports configurable Data Flash size to share with APROM.
- Supports 512 bytes security protection ROM (SPROM) to conceal user program.
- Supports 16 bytes User Configuration block to control system initialization.
- Supports 512 bytes page erase for all embedded flash.
- Supports CRC-32 checksum calculation function.
- Supports In-System-Programming (ISP) / In-Application-Programming (IAP) to update embedded flash memory.

4.3.3 Block Diagram

The flash memory controller (FMC) consists of AHB slave interface, flash control registers, flash initialization controller, flash operation control and embedded flash memory. Figure 4-13 shows the block diagram of flash memory controller.

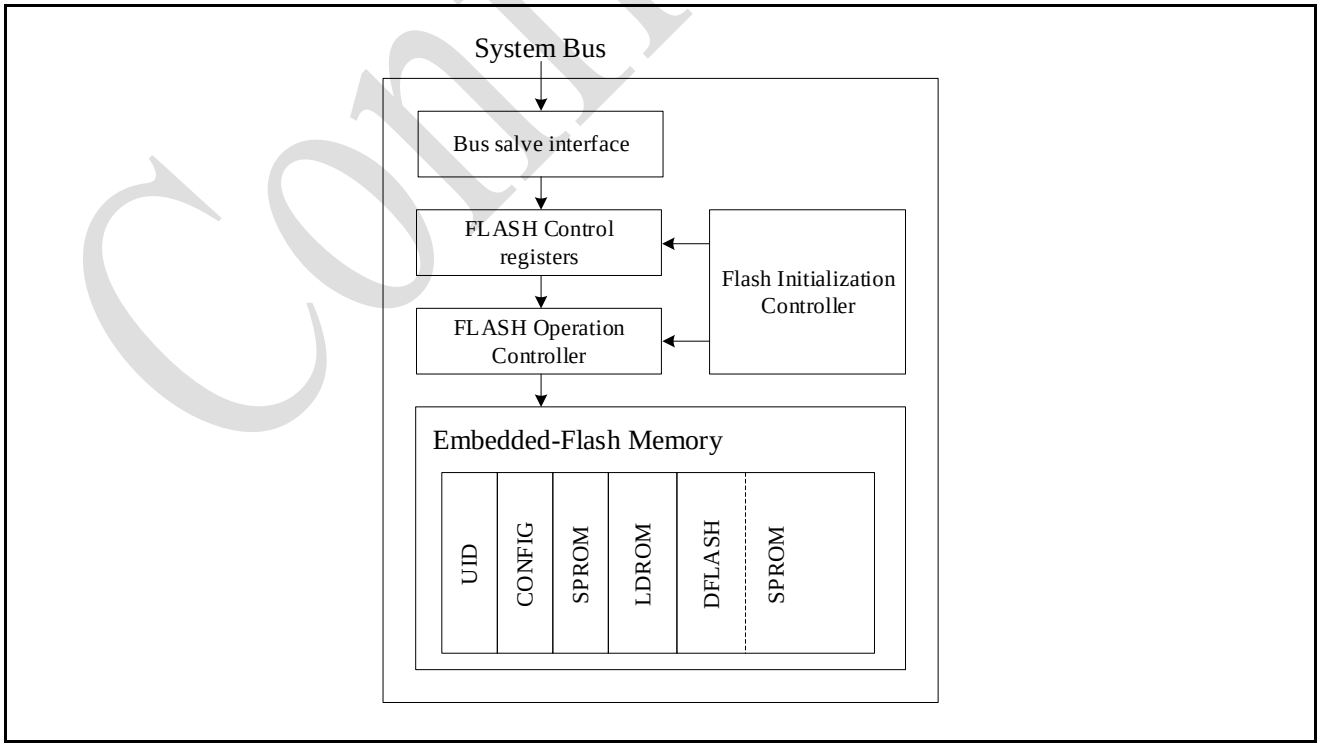


Figure 4-13 Flash Memory Control Block Diagram

- Bus Slave Interface

There is single bus slave interfaces in flash memory controller for CPU to perform the instruction data fetch and ISP control registers.

- Flash Control Registers

All of ISP control and status registers are in the flash control registers. The detail registers description is in the Register Description section

- Flash Initialization Controller

When chip is power on or active from reset, the flash initialization controller will start to access flash automatically and check the flash stability, and also reload User Configuration content to the flash control registers for system initialization.

- Flash Operation Controller

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

- Embedded Flash Memory

The embedded flash memory is the main memory for user application code and parameters. It consists of the user configuration block, 2 KB LDROM, 512B SPROM and 253 KB APROM with Data Flash. The page erase flash size is 512B, and program bit width is 32 bits.The size of APROM and LDROM can be configured by APAEND_DIS (CONFIG3[18]) and APAEND (FMC_ISPCFG3[17:0]), and the total size of APROM and LDROM is 255KB.

4.3.4 Functional Description

The FMC functions include the memory organization, boot selection, IAP, ISP, the embedded flash programming, and checksum calculation.

4.3.4.1 Memory Organization

The FMC memory consists of the embedded flash memory. The embedded flash memory is programmable, and includes APROM, LDROM, SPROM, Data Flash and the User Configuration block. The address map includes flash memory map and four system address maps: LDROM with IAP, LDROM without IAP, APROM with IAP, and APROM without IAP functions.

4.3.4.2 LDROM, APROM and Data Flash

LDROM is designed for a loader to implement In-System-Programming (ISP) function by user. The size is configured by user: the flash starting address is 0x0010_0000 and lasting to 0x0010_07FF (default). APROM is main memory for user applications the flash starting address is 0x0000_0000, and lasting to 0x0003_F3FF (default). Data Flash is used to store application parameters (not instruction). Data Flash is shared with APROM and size is configurable. The base address of Data Flash is determined by DFBA (CONFIG1[19:0]) and end address is APROM end address. All of embedded flash memory is 512 bytes page erased.

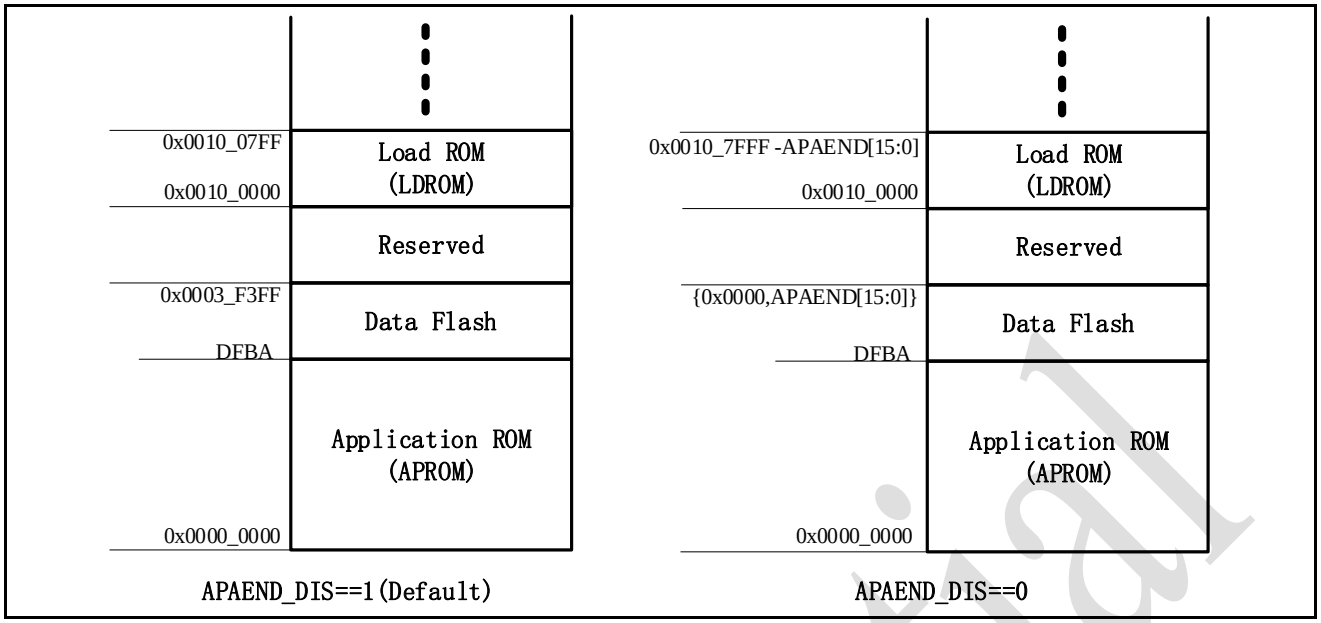


Figure 4-14 APROM, LDROM and Data Flash share with a 256K Flash

4.3.4.3 User Configuration Block

User Configuration block is internal programmable configuration area for boot options, such as flash security lock, boot select, brown-out voltage level,APROM Flash end address, and Data Flash base address. It works like a fuse for power on setting. It is loaded from flash memory to its corresponding control registers during chip power on. User can set these bits according to different application requests. User Configuration block can be updated by ISP function and located at 0x0030_0000 with four 32 bits words (CONFIG0, CONFIG1,CONFIG2 and CONFIG3). Any change on User Configuration block will take effect after system reboot.

CONFIG0 (Address = 0x0030_0000)

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

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:6]	CBS	Chip Booting Selection When CBS[0] = 0 with IAP mode, the LDROM base address is mapping to 0x100000 and APROM base address is mapping to 0x0. User could access both APROM and LDROM without boot switching. In other words, the code in LDROM and APROM can

		be called by each other. CBS value is valid. 00 = Boot from LDROM with IAP mode. 01 = Boot from LDROM without IAP mode. 10 = Boot from APROM with IAP mode. 11 = Boot from APROM without IAP mode. **Note**:BS (FMC_ISPCTL[1]) is only be used to control boot switching when CBS[0] = 1. VECMAP (FMC_ISPSTS[23:9]) is only be used to remap 0x0~0x1FF when CBS[0] = 0.
[5:2]	Reserved	Reserved
[1]	LOCK	Security Lock Control 0 = Flash memory content is locked. 1 = Flash memory content is locked except ALOCK (CONFIG2[7:0]) is 0x5A.
[0]	DFEN	Data Flash Enable Bit The Data Flash is shared with APROM, and the base address of Data Flash is decided by DFBA (CONFIG1[19:0]) when DFEN is 0. 0 = Data Flash Enabled. 1 = Data Flash Disabled.

CONFIG1 (Address = 0x0030_0004)

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

Bits	Descriptions	
[31:20]	Reserved	Reserved
[19:0]	DFBA	Data Flash Base Address This register works only when DFEN (CONFIG0[0]) is set to 0. If DFEN (CONFIG0[0]) is set to 0, the Data Flash base address is defined by user. Since on-chip flash erase unit is 512 bytes, it is mandatory to keep bit 8-0 as 0.

CONFIG2 (Address = 0x0030_0008)

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							

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

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

CONFIG3 (Address = 0x0030_000C)

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

Bits	Descriptions	
[31:19]	Reserved	Reserved
[18]	APAEND_DIS	APROM Address END Disable 1=APROM is 253K, and LDROM is 2K. APROM is starting at 0x0000_0000 and lasting to {0x0000_3F3FF}.LDROM is stating at 0x0010_0000 and lasting to {0x0100_07FF}. 0=APROM end address is configured by APAEND[17:0], APROM is starting at 0x0000_0000 and lasting to {0x0000, APAEND[17:0]}.LDROM is stating at 0x0010_0000 and lasting to {0x0100, 7BFFF-APAEND[17:0]}.The total size of APROM and LDROM is 255KB.
[17:0]	APAEND	APROM Address END; APAEND_DIS=1: APROM is 253K, and LDROM is 2K. APROM is starting at 0x0000_0000 and lasting to {0x0000_3F3FF }.LDROM is stating at 0x0010_0000 and lasting to {0x0100_07FF}. APAEND_DIS=0:APROM end address is configured by APAEND[17:0], APROM is starting at 0x0000_0000 and lasting to {0x0000, APAEND[17:0]}. LDROM is stating at 0x0010_0000 and lasting to {0x0100, 7BFFF-APAEND[17:0]}. The total size of APROM and LDROM is 255KB.

		If Data Flash used, DFBA[17:0]should be litter than APAEND[17:0], Since on-chip flash erase unit is 512 bytes, APAEND is mandatory to keep bit 8-0 as 1.

4.3.4.4 Security Protection Memory (SPROM)

The security protection memory (SPROM) is used to store instructions for security application. The SPROM includes 512 bytes at location address 0x0020_0000 ~ 0x20_01FF and doesn't support "whole chip erase command".Figure 4-15 shows that the last byte of SPROM is used to identify the SPROM code is non-secured, debug secured or secured mode.

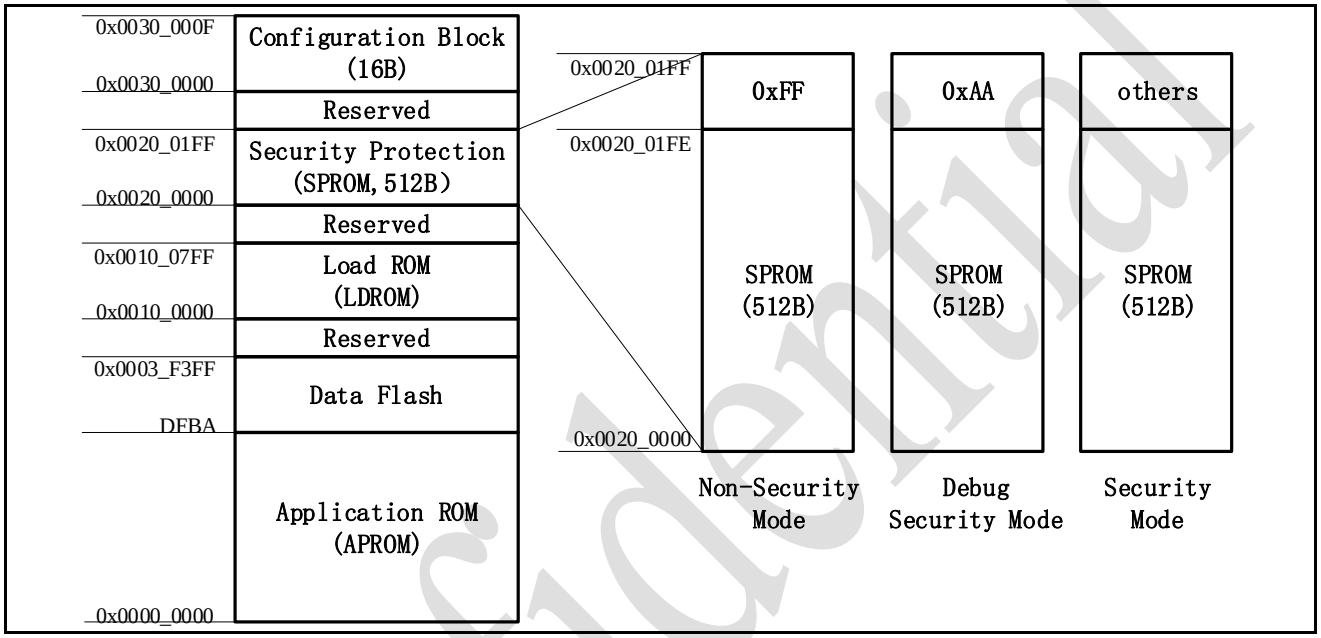


Figure 4-15 SPROM Security Mode

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

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

(3) SPROM secured mode (the last byte is neither 0xFF nor 0xAA). In order to conceal SPROM code in secured mode, CPU only can perform instruction fetch and get data from SPROM when CPU is run at SPROM area. Otherwise, CPU will get all zero (0x0000_0000) for data access. In order to protect SPROM, the CPU instruction fetch will also get zero value when CPU ICE (In-Circuit-Emulator) port is connected in secured code. At this mode, SPROM doesn't support ISP program and read flash command and only supports page erase command.

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

The SEC_MODE (FMC_CONFIG3[31:24]) indicate the last by of SPROM.

4.3.4.5 Flash Memory Map

In the PAN1020 series, the flash memory map is different to system memory map. The system memory map is used by CPU fetch code or data from FMC memory. The flash memory map is used for ISP function to read, program or erase FMC memory. Figure 4-16 shows the flash memory map.

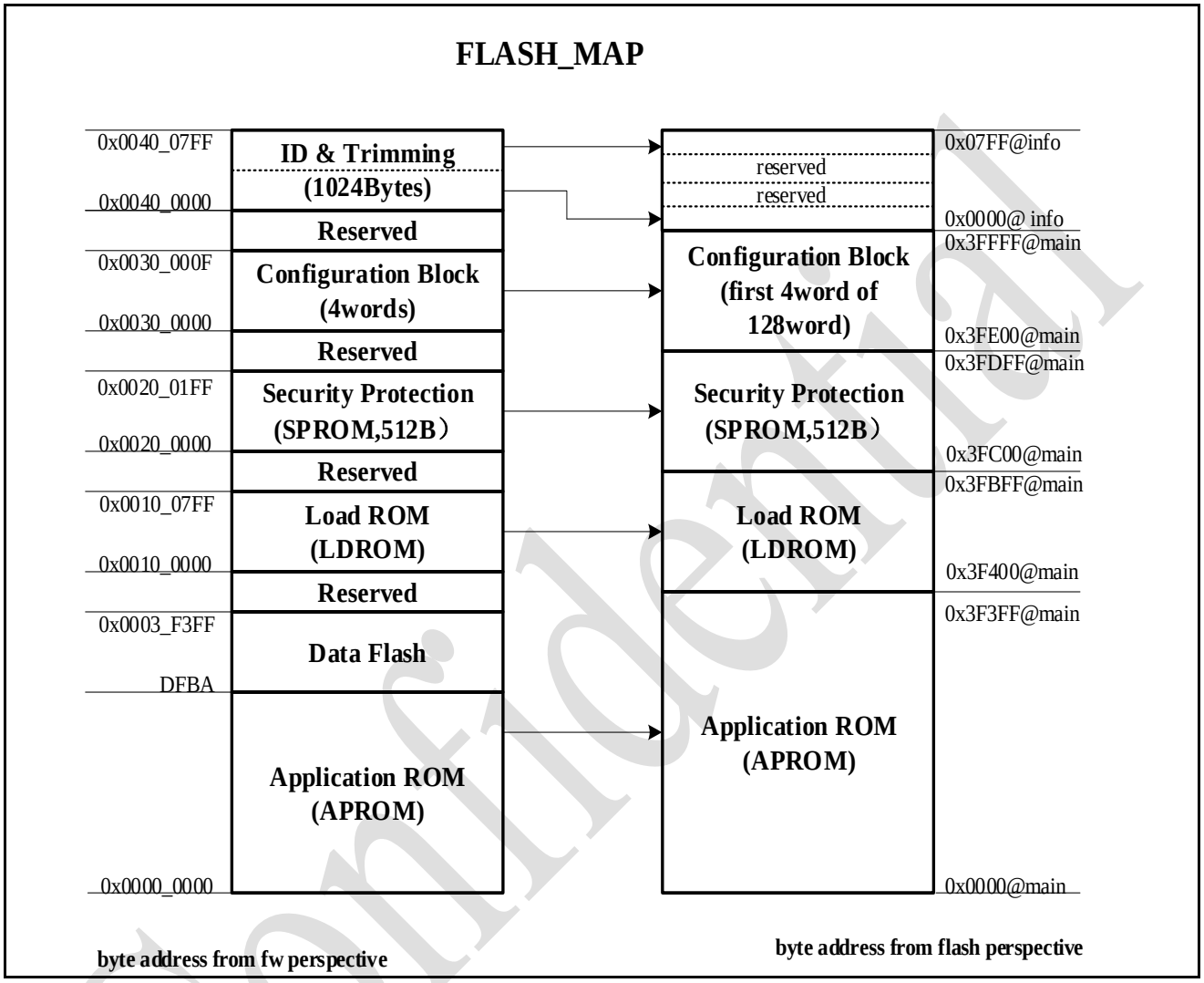


Figure 4-16 Flash Memory Map

4.3.4.6 System Memory Map with IAP Mode

The system memory map is used by CPU to fetch code or data from FMC memory. SPROM(0x0020_0000~0x0020_01FF) and LDROM(0x0010_0000~0x0010_07FF) address map are the same as in the flash memory map. The Data Flash is shared with APROM and the Data Flash base address is defined by CONFIG1. The content of CONFIG1 is loaded into DFBA (Data Flash Base Address Register) at the flash initialization. The DFBA~APAEND is the Data Flash region for CPU data access, and 0x0000_0200~(DFBA-1) is APROM region for CPU instruction access.

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

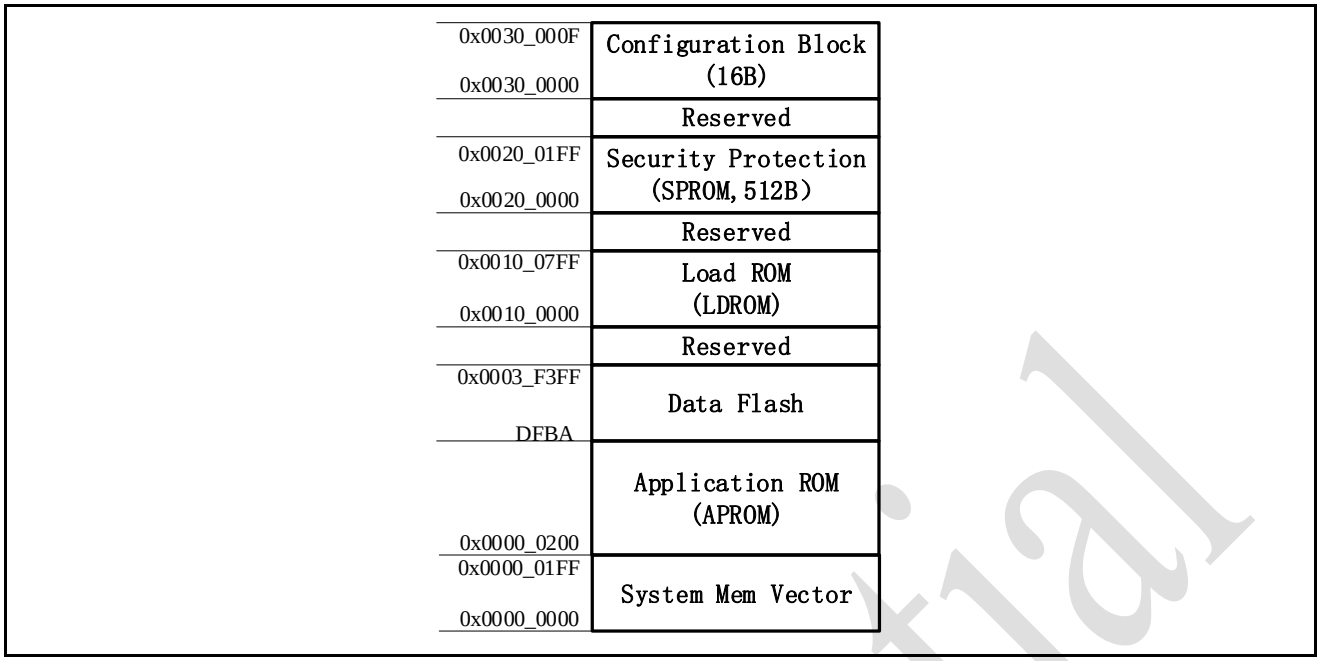


Figure 4-17 System Memory Map with IAP Mode

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

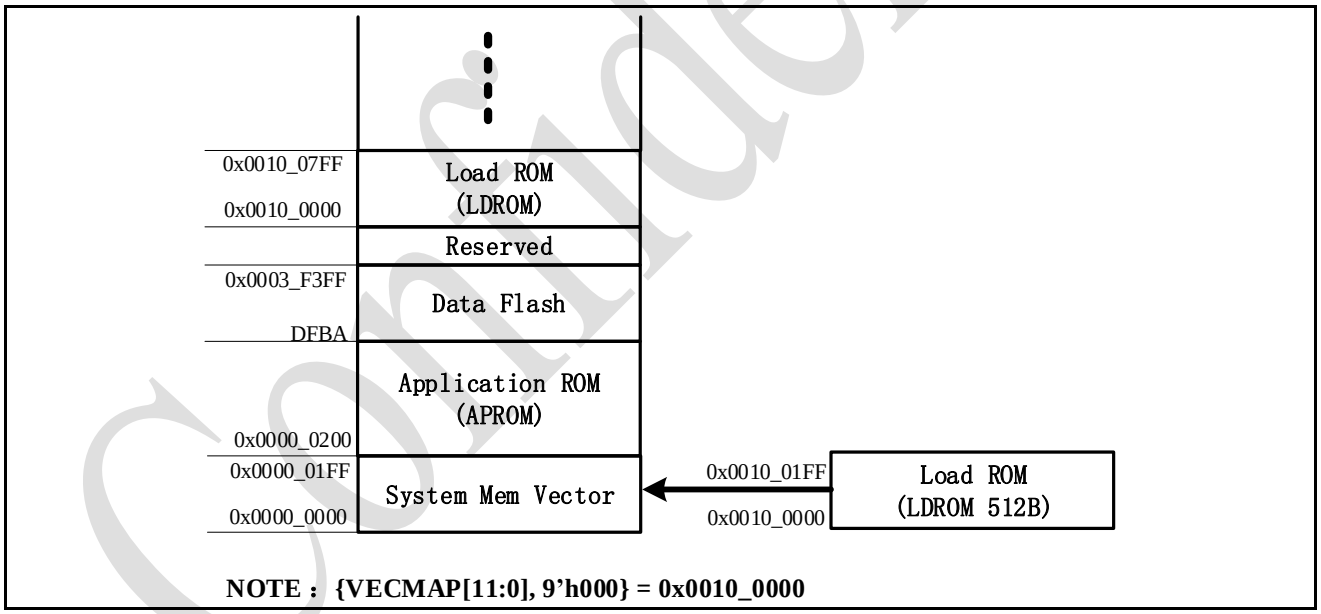


Figure 4-18 LDROM with IAP Mode

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

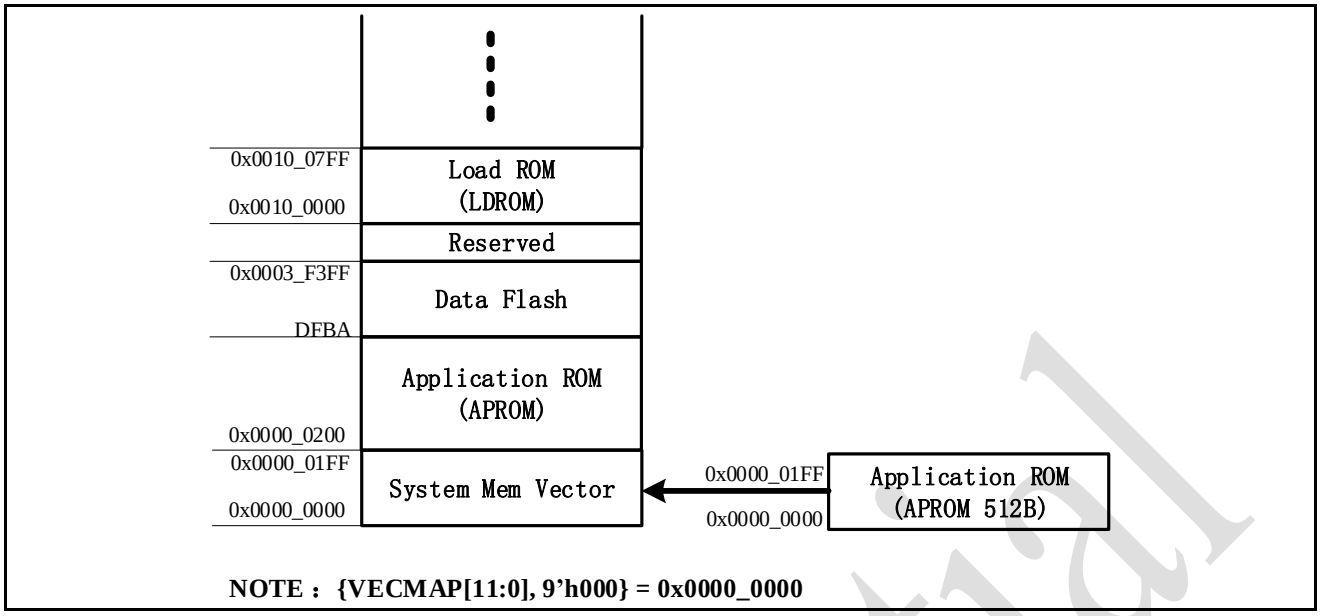


Figure 4-19 APROM with IAP Mode

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

4.3.4.7 System Memory Map without IAP mode

SPROM(0x0020_0000~0x0020_01EF), but the system memory vector mapping is not supported. There are two kinds of system memory map without IAP mode when chip booting: (1) LDROM without IAP, (2) APROM without IAP. In LDROM without IAP mode, LDROM base is mapping to 0x0000_0000. CPU program cannot run to access APROM. In APROM without IAP mode, APROM base is mapping to 0x0000_0000. CPU program cannot run to access LDROM. The Data Flash is shared with APROM and the Data Flash base address is defined by CONFIG1. The content of CONFIG1 is loaded into DFBA (Data Flash Base Address Register) at the flash initialization. The DFBA~0x0003_F3FF is the Data Flash region for CPU data access, and 0x0000_0000~(DFBA-1) is APROM region for CPU instruction access.

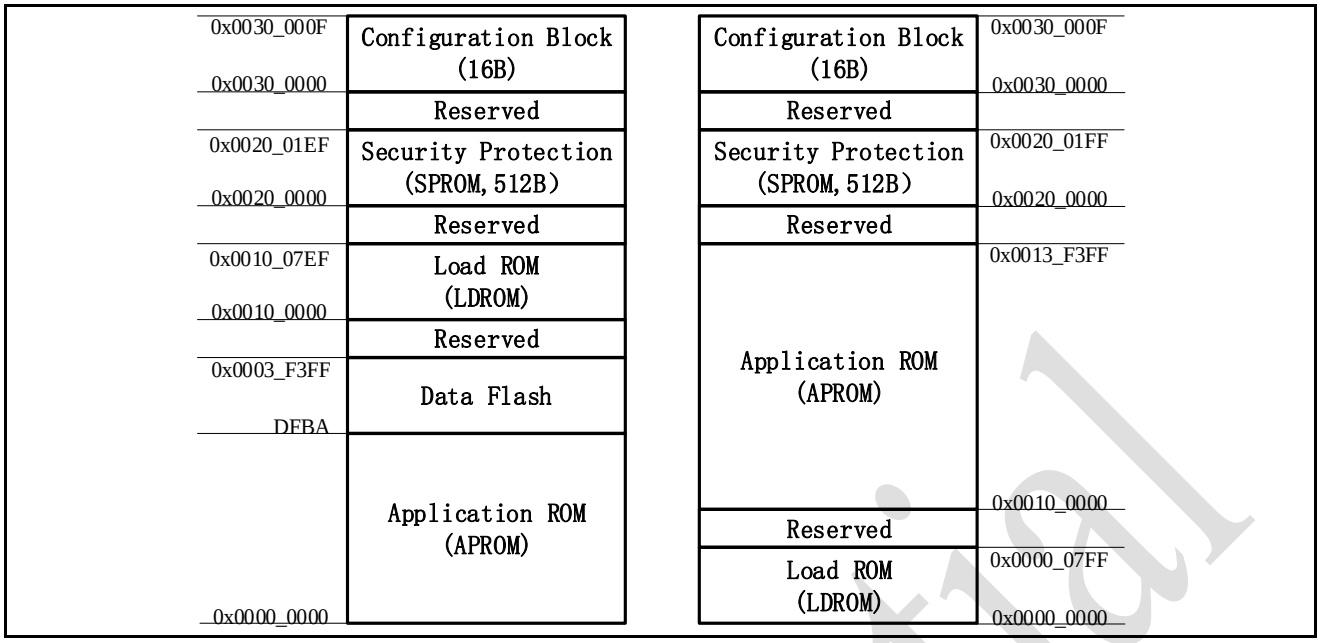


Figure 4-20 System Memory Map without IAP Mode

4.3.4.8 Boot Selection

The PAN1020 provides four booting sources for user select. They are LDROM with IAP, LDROM without IAP, APROM with IAP, and APROM without IAP. The booting source and system memory map are setting by CBS (CONFIG0[7:6]).

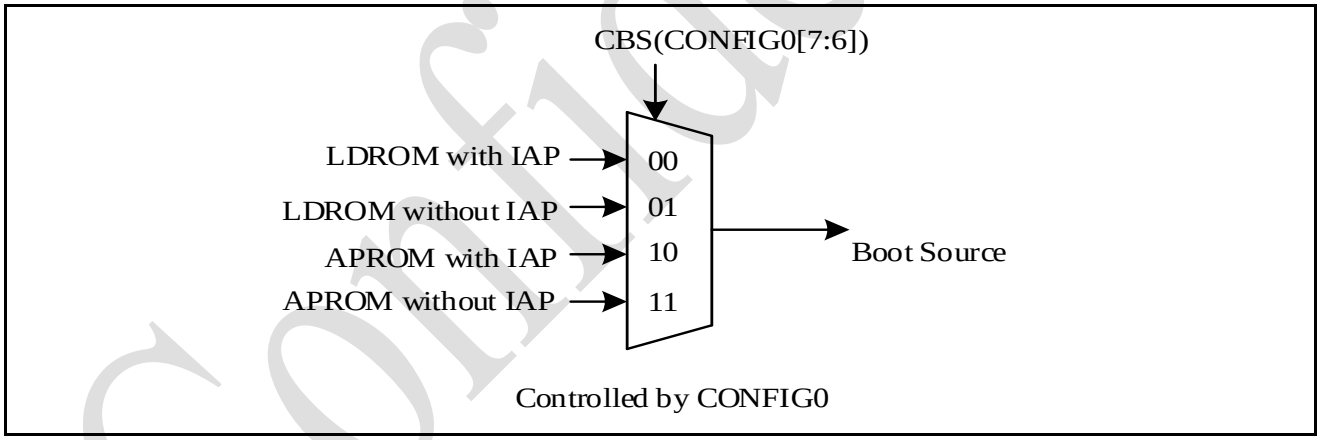


Figure 4-21 Boot Source Selection

Table 4-10 Vector Mapping Support Table

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

4.3.4.9 In-Application-Programming (IAP)

The PAN1020 Series provides In-Application-Programming (IAP) function for user to switch the code executing between APROM, LDROM and SPROM. User can enable the IAP function by booting chip and setting the chip boot selection bits in CBS (CONFIG0[7:6]) as 10 or 00.

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

4.3.4.10 In-System-Programming (ISP)

The PAN1020 series supports In-System-Programming (ISP) function allowing the embedded flash memory to be reprogrammed under software control. ISP is performed without removing the microcontroller from the system through the firmware and on-chip connectivity interface, such as UART, I2C, and SPI.

The PAN1020 ISP provides the following functions for embedded flash memory.

- Supports flash page erase function
- Supports flash data program function
- Supports flash data read function
- Supports company ID read function
- Supports device ID read function
- Supports unique ID read function
- Supports memory checksum calculation function
- Supports system memory vector remap function

4.3.4.10.1. ISP Commands

Table 4-11 ISP Command List

ISP Command	FMC_ISPCMD	FMC_ISPADDR	FMC_ISPDAT
FLASH Page Erase	0x22	Valid address of flash memory origination. It must be 512 bytes page alignment.	N/A
SPROM Page Erase	0x22	0x0020_0000	0x0055_AA03
FLASH 32-bit Program	0x21	Valid address of flash memory origination	FMC_ISPDAT: Programming Data
FLASH Read	0x00	Valid address of flash memory origination	FMC_ISPDAT: Return Data
Read Company ID	0x0B	0x0000_0000(user mode) 0x0040_0040(ROM mode)	FMC_ISPDAT: 0xFFFF_FFFF
Read Product ID	0x0C	0x0000_0004(user mode) 0x0040_0084(ROM mode)	FMC_ISPDAT: 0xFFFF_FFFF
Read CRC32 Checksum	0x0D	Keep address of "Run Checksum Calculation"	FMC_ISPDAT: Return Checksum
Run CRC32 Checksum Calculation	0x2D	Valid start address of memory origination It must be 512 bytes page alignment	FMC_ISPDAT: Size It must be 512 bytes alignment

Read Unique ID	0x04	0x0000_0000 (user mode) 0x0040_0020(ROM mode) **Note**: only care FMC_ISPADDR[4:2]	FMC_ISPDAT: Unique ID Word 0
		0x0000_0004 (user mode) 0x0040_0024(ROM mode) **Note**: only care FMC_ISPADDR[4:2]	FMC_ISPDAT: Unique ID Word 1
		0x0000_0008 (user mode) 0x0040_0028(ROM mode) **Note**: only care FMC_ISPADDR[4:2]	FMC_ISPDAT: Unique ID Word2
		0x0000_000C (user mode) 0x0040_002C(ROM mode) **Note**: only care FMC_ISPADDR[4:2]	FMC_ISPDAT: Unique ID Word3
Read Unique Company ID	0x04	0x0000_0010 (user mode) 0x0040_0030(ROM mode) **Note**: only care FMC_ISPADDR[4:2]	FMC_ISPDAT: Unique Company ID Word 0
		0x0000_0014 (user mode) 0x0040_0034(ROM mode) **Note**: only care FMC_ISPADDR[4:2]	FMC_ISPDAT: Unique Company ID Word 1
		0x0000_0018 (user mode) 0x0040_0038(ROM mode) **Note**: only care FMC_ISPADDR[4:2]	FMC_ISPDAT: Unique Company ID Word 2
		0x0000_001C (user mode) 0x0040_003C(ROM mode) **Note**: only care FMC_ISPADDR[4:2]	FMC_ISPDAT: Unique Company ID Word 3
Vector Remap	0x2E	Valid address in APROM or LDROM. It must be 512 bytes alignment	N/A
Wafer ID		0x0040_00C0	2 Words
ROM Version		0x0040_01C0	4 Words

4.3.4.10.2. ISP Procedure

The FMC provides embedded flash memory read, erase and program operation. Several control bits of FMC control register are write-protected, thus it is necessary to unlock before setting.

After unlocking the protected register bits, user needs to set the FMC_ISPCTL control register to decide to update LDROM, APROM, SPROM or user configuration block, and then set ISPEN (FMC_ISPCTL[0]) to enable ISP function.

Once the FMC_ISPCTL register is set properly, user can set FMC_ISPCMD (refer above ISP command list) for specify operation. Set FMC_ISPADDR for target flash memory based on flash memory origination. FMC_ISPDAT can be used to set the data to program or used to return the read data according to FMC_ISPCMD.

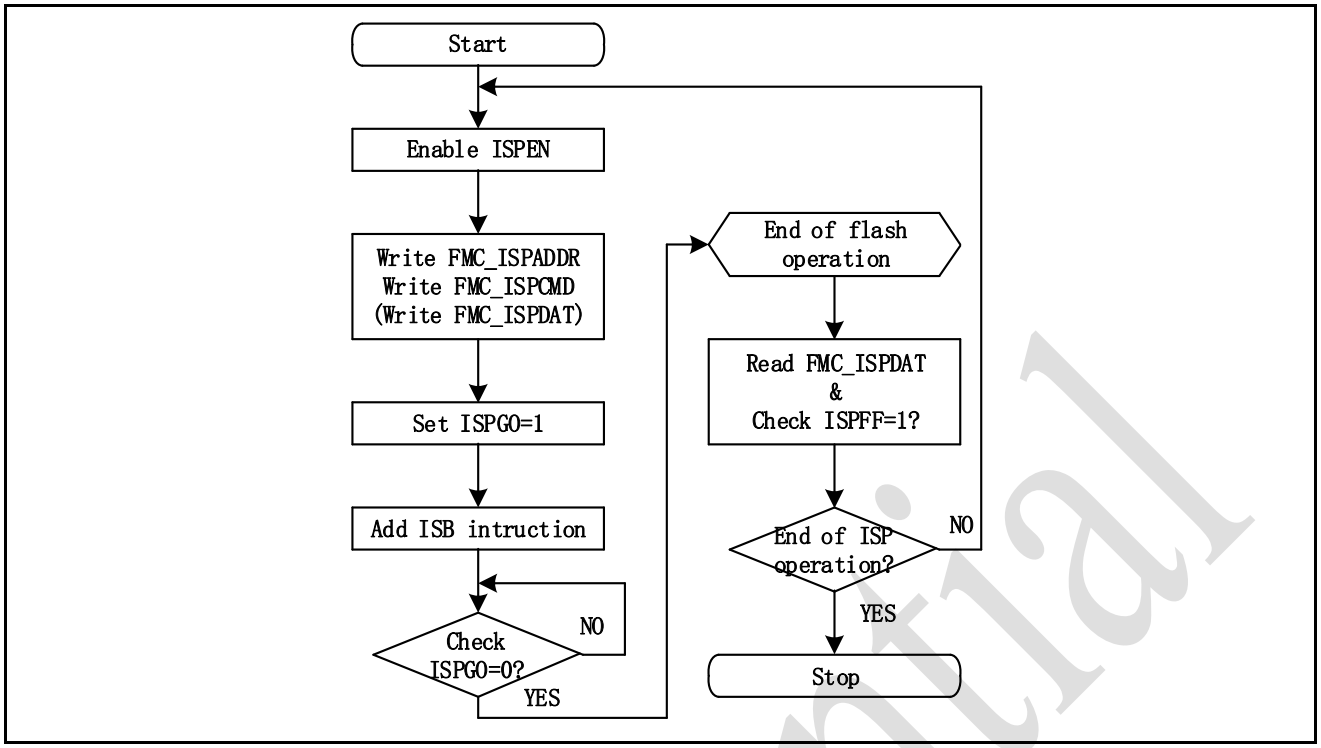


Figure 4-22 ISP Procedure Example

Finally, set ISPGO (FMC_ISPTRG[0]) register to perform the relative ISP function. The ISPGO (FMC_ISPTRG[0]) bit is self-cleared when ISP function has been done. To make sure ISP function has been finished before CPU goes ahead, ISB (Instruction Synchronization Barrier) instruction is used right after ISPGO (FMC_ISPTRG[0]) setting.

Several error conditions will be checked after ISP is completed. If an error condition occurs, ISP operation is not started and the ISP fail flag will be set instead. ISPFF (FMC_ISPSTS[6]) flag can only be cleared by software. The next ISP procedure can be started even ISPFF (FMC_ISPSTS[6]) bit is kept as 1. Therefore, it is recommended to check the ISPFF (FMC_ISPSTS[6]) bit and clear it after each ISP operation if it is set to 1.

When the ISPGO (FMC_ISPTRG[0]) bit is set, CPU will wait for ISP operation to finish during this period; the peripheral still keeps working as usual. If any interrupt request occurs, CPU will not service it till ISP operation is finished. When ISP operation is finished, the ISPGO bit will be cleared by hardware automatically. User can check whether ISP operation is finished or not by the ISPGO (FMC_ISPTRG[0]) bit. User should add ISB (Instruction Synchronization Barrier) instruction next to the instruction in which ISPGO (FMC_ISPTRG[0]) bit is set 1 to ensure correct execution of the instructions following ISP operation.

4.3.4.11 CRC32 Checksum Calculation

The PAN1020 series supports the Cyclic Redundancy Check (CRC-32) checksum calculation function to help user quickly check the memory content includes APROM, LDROM and SPROM. The CRC-32 polynomial is below:

CRC32:X32+X26+X23+X22+X16+X12+X11+X10+X8+X7+X5+X4+X2+X+1

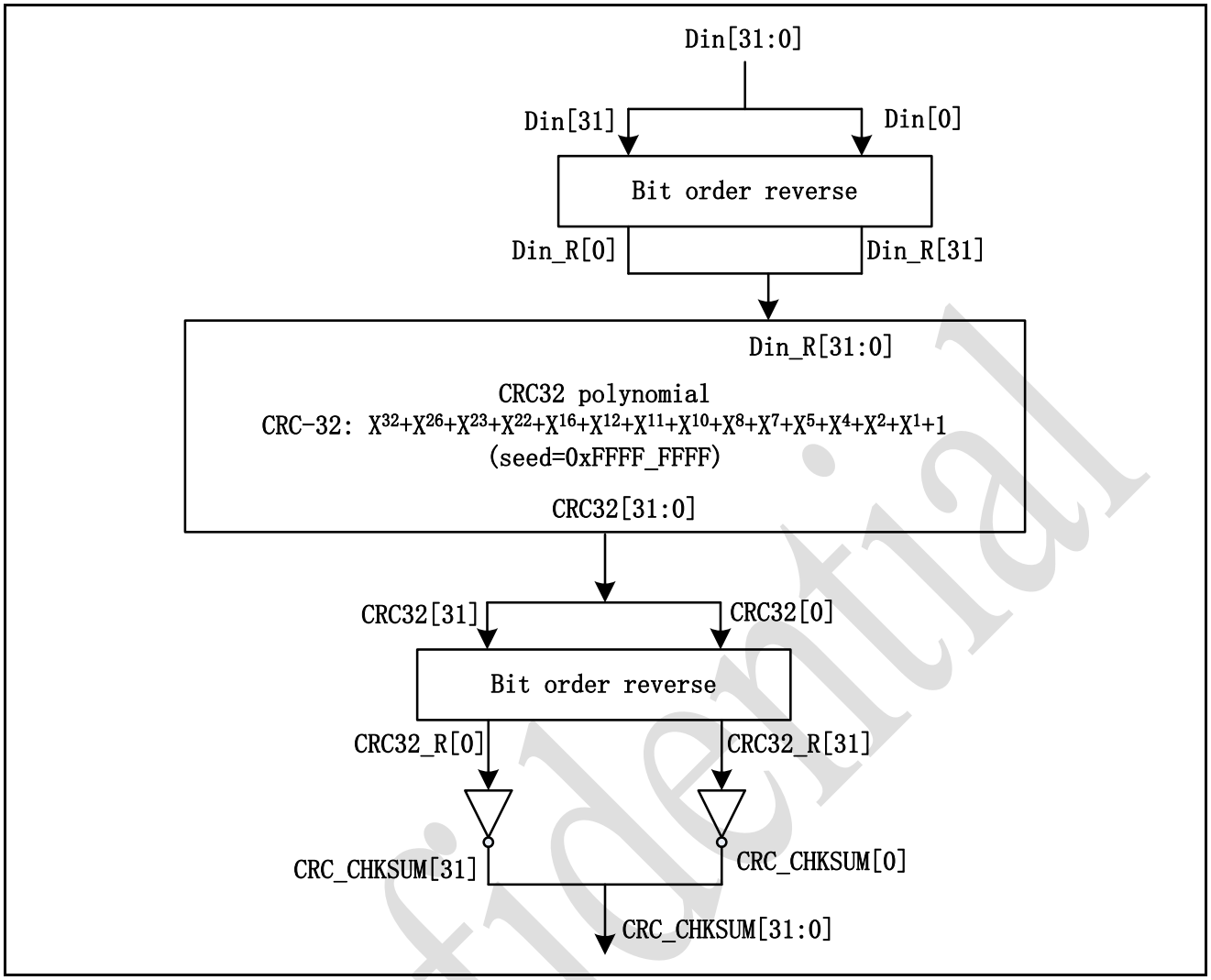


Figure 4-23 CRC-32 Checksum Calculation

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

1. Perform ISP “Run Memory Checksum” operation: user has to set the memory starting address (FMC_ISPADDR) and size (FMC_ISPDAT) to calculate. Both address and size have to be 4 bytes alignment, the size should be must be multiples of 4 bytes and the starting address is absolute address, in Figure 4-16 Flash Memory Map, it refers to byte address from flash perspect.

2. Perform ISP “Read Memory Checksum” operation: the FMC_ISPADDR should be kept as the same as step 1.

3. Read FMC_ISPDAT to get checksum: The checksum is read from FMC_ISPDAT. If the checksum is 0x0000_0000, It must be one of two conditions (1) Checksum calculation is in-progress, (2) Address and size is over device limitation.

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

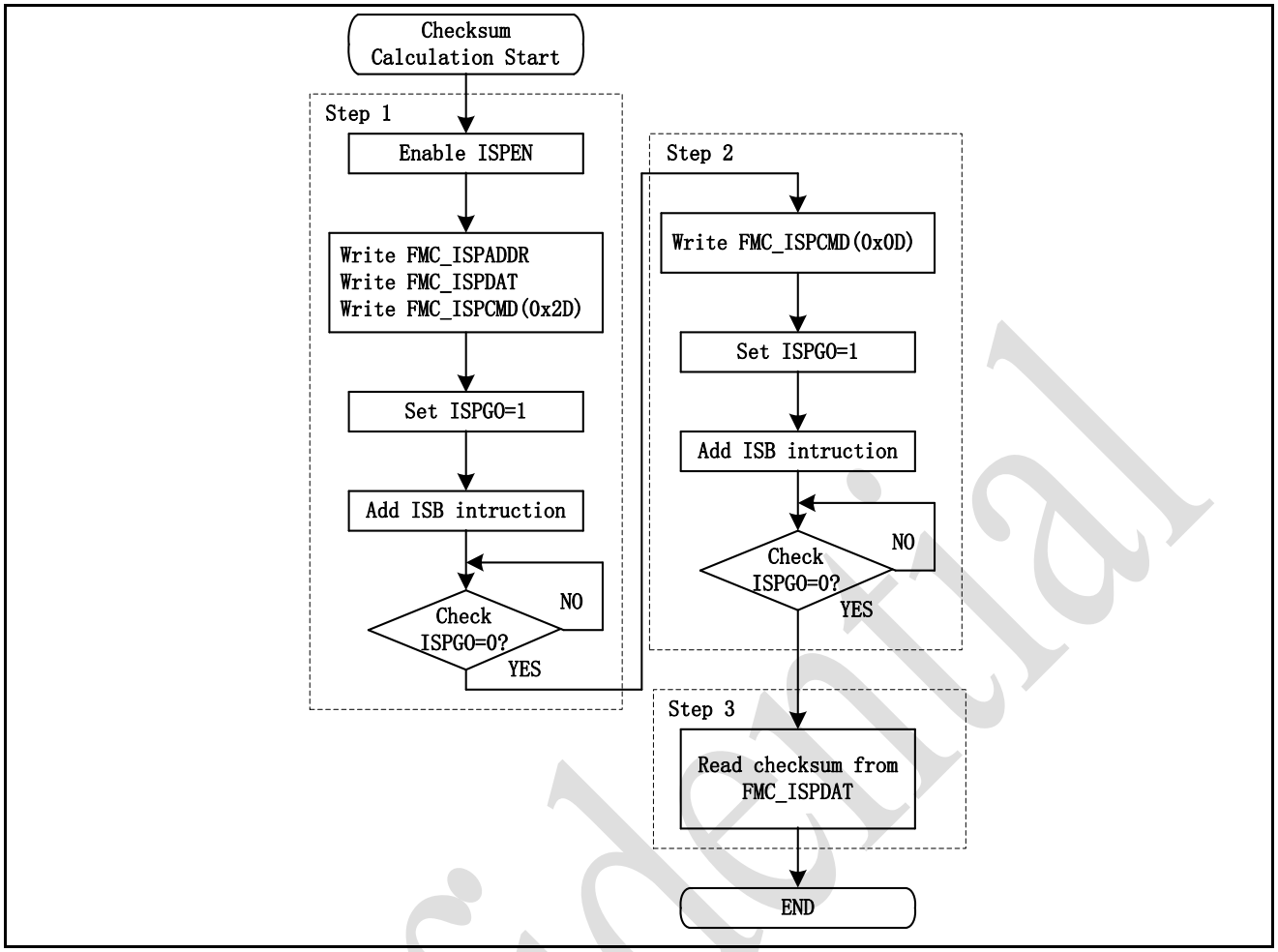


Figure 4-24 CRC-32 Checksum Calculation Flow

4.3.5 Flash Control Register Map

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

Register	Offset	R/W	Description	Reset Value
FMC Base Address: **FMC_BA = 0x5000_C000**				
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP Control Register	0x0000_000X
FMC_ISPADDR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000
FMC_ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000
FMC_ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Control Register	0x0000_0000
FMC_DFBA	FMC_BA+0x14	R	Data Flash Base Address	0xXXXX_XXXX
FMC_FATCTL	FMC_BA+0x18	R/W	Flash Access Time Control Register	0x0000_0000
FMC_ISPSTS	FMC_BA+0x40	R/W	ISP Status Register	0xX0X0_000X
FMC_ISPCFG0	FMC_BA+0x50	R	ISP Configure0 Register	0xFFFF_FFFF
FMC_ISPCFG1	FMC_BA+0x54	R	ISP Configure1 Register	0xFFFF_FFFF
FMC_ISPCFG2	FMC_BA+0x58	R	ISP Configure2 Register	0xFFFF_FFFF
FMC_ISPCFG3	FMC_BA+0x5C	R	ISP Configure3 Register	0xFFFF_FFFF

4.3.6 Flash Control Register Description

4.3.6.1 ISP Control Register (FMC_ISPCTL)

Register	Offset	R/W	Description	Reset Value
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP Control Register	0x0000_000X

Bits	Descriptions	
[31:7]	Reserved	Reserved.
[6]	ISPFF	ISP Fail Flag (Write Protect) This bit is set by hardware when a triggered ISP meets any of the following conditions: This bit needs to be cleared by writing 1 to it. (1) APROM writes to itself if APUEN is set to 0. (2) LDROM writes to itself if LDUEN is set to 0. (3) CONFIG is erased/programmed if CFGUEN is set to 0. (4) SPROM is erased/programmed if SPUEN is set to 0. (5) SPROM is programmed at SPROM secured mode. (6) Destination address is illegal, such as over an available range. (7) Invalid ISP commands.
[5]	LDUEN	LDROM Update Enable Bit (Write Protect) 0 = LDROM cannot be updated. 1 = LDROM can be updated.
[4]	CFGUEN	CONFIG Update Enable Bit (Write Protect) 0 = CONFIG cannot be updated. 1 = CONFIG can be updated.
[3]	APUEN	APROM Update Enable Bit (Write Protect) 0 = APROM cannot be updated when the chip runs in APROM. 1 = APROM can be updated when the chip runs in APROM.
[2]	SPUEN	SPROM Update Enable Bit (Write Protect) 0 = SPROM cannot be updated. 1 = SPROM can be updated.
[1]	BS	Boot Select (ReadOnly) This bit also functions as chip booting status flag, which can be used to check where chip booted from. This bit is initiated with the inversed value of CBS[1] (CONFIG0[7]) after any reset is happened except CPU reset (RSTS_CPU is 1) or system reset (RSTS_SYS) is happened 0 = Booting from APROM. 1 = Booting from LDROM.
[0]	ISPEN	ISP Enable Bit (Write Protect) Set this bit to enable the ISP function. 0 = ISP function Disabled. 1 = ISP function Enabled.

4.3.6.2 ISP Address (FMC_ISPADDR)

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

Bits	Descriptions	
[31:0]	ISPADDR	ISP Address The PAN1020 is equipped with embedded flash. ISPADDR[1:0] must be kept 00 for ISP 32-bit operation. and ISPADDR[8:0] must be kept all 0 for Vector Page Re-map Command For CRC32 Checksum Calculation command, this field is the flash starting address for checksum calculation, 512 bytes alignment is necessary for checksum calculation. The specific relationship between ISP command and FMC_ISPADDR can be related to Table 4-11.

4.3.6.3 ISP Data Register (FMC_ISPDAT)

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

Bits	Descriptions	
[31:0]	ISPDAT	ISP Data Write data to this register before ISP program operation. Read data from this register after ISP read operation. For Run CRC32 Checksum Calculation command, ISPDAT is the memory size (byte) and 512 bytes alignment. For ISP Read Checksum command, ISPDAT is the checksum result. If ISPDAT = 0x0000_0000, it means that (1) the checksum calculation is in progress, or (2) the memory range for checksum calculation is incorrect. The specific relationship between ISP command and FMC_ISPDAT can be related to Table 4-11.

4.3.6.4 ISP Command Register (FMC_ISPCMD)

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

Bits	Descriptions	
[31:7]	Reserved	Reserved
[6:0]	CMD	ISP CMD ISP command table is shown below: 0x00= FLASH Read.

		0x04= Read Unique ID. 0x0B= Read Company ID. 0x0C= Read Product ID. 0x0D= Read CRC32 Checksum. 0x21= FLASH 32-bit Program. 0x22= FLASH Page Erase. 0x23= FLASH whole chip Erase (ROM mode). 0x2D= Run CRC32 Checksum Calculation. 0x2E= Vector Remap. The other commands are invalid.

4.3.6.5 ISP Trigger Control Register (FMC_ISPTRG)

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

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

4.3.6.6 Data Flash Base Address Register (FMC_DFBA)

Register	Offset	R/W	Description	Reset Value
FMC_DFBA	FMC_BA+0x14	R	Data Flash Base Address	0xXXXX_XXXX

Bits	Descriptions	
[31:0]	DFBA	Data Flash Base Address This register indicates Data Flash start address. It is a read only register. The Data Flash is shared with APROM. The content of this register is loaded from CONFIG1 This register is valid when DFEN (CONFIG0[0]) =0 .

4.3.6.7 Flash Access Time Control Register (FMC_FATCTL)

Register	Offset	R/W	Description	Reset Value
FMC_FATCTL	FMC_BA+0x18	R/W	Flash Access Time Control Register	0x0000_0000

Bits	Descriptions	
[31:7]	Reserved	Reserved
[6:4]	FOM	Frequency Optimization Mode (Write Protect) The PAN1020 supports adjustable flash access timing to optimize the flash access cycles in different working frequency. 0x1 = Frequency ≤ 24MHz. Others = Frequency ≤ 50MHz
[3:0]	Reserved	Reserved

4.3.6.8 ISP Status Register (FMC_ISPSTS)

Register	Offset	R/W	Description	Reset Value
FMC_ISPSTS	FMC_BA+0x40	R/W	ISP Status Register	0xX0X0_000X

Bits	Descriptions	
[31]	SCODE	Security Code Active Flag(read only) 0 = SPROM secured code is inactive. 1 = SPROM secured code is active.
[30:21]	Reserved	Reserved
[20:9]	VECMAP	Vector Page Mapping Address (Read Only) All access to 0x0000_0000~0x0000_01FF is remapped to the flash memory address {VECMAP[11:0], 9'h000} ~ {VECMAP[11:0], 9'h1FF}
[8:7]	Reserved	Reserved
[6]	ISPFF	ISP Fail Flag (Write Protect) This bit is the mirror of ISPFF (FMC_ISPCTL[6]), it needs to be cleared by writing 1 to FMC_ISPCTL[6] or FMC_ISPSTS[6]. This bit is set by hardware when a triggered ISP meets any of the following conditions: (1) APROM writes to itself if APUEN is set to 0. (2) LDROM writes to itself if LDUEN is set to 0. (3) CONFIG is erased/programmed if CFGUEN is set to 0. (4) SPROM is erased/programmed if SPUEN is set to 0. (5) SPROM is programmed at SPROM secured mode. (6) Page Erase command at LOCK mode with ICE connection. (7) Erase or Program command at brown-out detected. (8) Destination address is illegal, such as over an available range. (9) Invalid ISP commands.
[5:3]	Reserved	Reserved
[2:1]	CBS	Boot Selection Of CONFIG (Read Only) This bit is initiated with the CBS (CONFIG0[7:6]) after any reset is happened except CPU reset (RSTS_CPU is 1) or system reset (RSTS_SYS) is happened. 00 = LDROM with IAP mode. 01 = LDROM without IAP mode. 10 = APROM with IAP mode.

		11 = APROM without IAP mode.
[0]	ISPBUSY	ISP BUSY (Read Only) 0 = ISP operation is finished. 1 = ISP operation is busy.

4.3.6.9 ISP Configure0 Register (FMC_ISPCFG0)

Register	Offset	R/W	Description	Reset Value
FMC_ISPCFG0	FMC_BA+0x50	R	ISP configure Register	0xFFFF_FFFF

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7:6]	CBS	Chip Booting Selection (Read Only) When CBS[0] = 0 with IAP mode, the LDROM base address is mapping to 0x100000 and APROM base address is mapping to 0x0. User could access both APROM and LDROM without boot switching. In other words, the code in LDROM and APROM can be called by each other. CBS value is valid. 00 = Boot from LDROM with IAP mode. 01 = Boot from LDROM without IAP mode. 10 = Boot from APROM with IAP mode. 11 = Boot from APROM without IAP mode. **Note**: BS (FMC_ISPCTL[1]) is only be used to control boot switching when CBS[0] = 1. VECMAP (FMC_ISPSTS[23:9]) is only be used to remap 0x0~0x1FF when CBS[0] = 0.
[5:2]	Reserved	Reserved
[1]	LOCK	Security Lock Control (Read Only) 0 = Flash memory content is locked. 1 = Flash memory content is locked except ALOCK (CONFIG2[7:0]) is 0x5A.
[0]	DFEN	Data Flash Enable Bit (Read Only) The Data Flash is shared with APROM, and the base address of Data Flash is decided by DFBA (CONFIG1[19:0]) when DFEN is 0. 0 = Data Flash Enabled. 1 = Data Flash Disabled.

4.3.6.10 ISP Configure1 Register (FMC_ISPCFG1)

Register	Offset	R/W	Description	Reset Value
FMC_ISPCFG1	FMC_BA+0x54	R	ISP configure Register	0xFFFF_FFFF

Bits	Descriptions

[31:20]	Reserved	Reserved
[19:0]	DFBA	Data Flash Base Address (Read Only) This register works only when DFEN (CONFIG0[0]) is set to 0. If DFEN (CONFIG0[0]) is set to 0, the Data Flash base address is defined by user. Since on-chip flash erase unit is 512 bytes, it is mandatory to keep bit 8-0 as 0.

4.3.6.11 ISP Configure2 Register (FMC_ISPCFG2)

Register	Offset	R/W	Description	Reset Value
FMC_ISPCFG2	FMC_BA+0x58	R	ISP Configure2 Register	0xFFFF_FFFF

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

4.3.6.12 ISP Configure3 Register (FMC_ISPCFG3)

Register	Offset	R/W	Description	Reset Value
FMC_ISPCFG3	FMC_BA+0x5C	R	ISP Configure3 Register	0xFFFF_FFFF

Bits	Descriptions	
[31:24]	SEC_MODE	Security Mode, the last byte of SPROM (Read Only) 0xFF= non-secured mode 0xAA= debug secured mode Others= secured mode
[23:19]	Reserved	Reserved
[18]	APAEND_DIS	APROM Address END Disable (Read Only) 1=APROM is 253K, and LDROM is 2K. APROM is starting at 0x0000_0000 and lasting to {0x0003_F3FF}.LDROM is stating at 0x0010_0000 and lasting to {0x0100_07FF}. 0=APROM end address is configured by APAEND[17:0], APROM is starting at 0x0000_0000 and lasting to {14'h0,APAEND[17:0]}.LDROM is stating at 0x0010_0000 and lasting to 0x0010_00000+(18'h3FC00-APAEND[17:0]}. The total size of APROM and LDROM is 255KB.
[17:0]	APAEND	APROM Address END (Read Only) APAEND_DIS=1: APROM is 253K, and LDROM is 2K. APROM is starting at 0x0000_0000 and lasting to {0x0003_F3FF}.LDROM is stating at 0x0010_0000 and lasting to {0x0010_07FF}.

		APAEND_DIS=0: APROM end address is configured by APAEND[17:0], APROM is starting at 0x0000_0000 and lasting to {14'h0, APAEND[17:0]}.LDROM is stating at 0x0010_0000 and lasting to 0x0010_0000+(18'h3FC00-APAEND[17:0]}.The total size of APROM and LDROM is 255KB. **Note**: the separation of APROM and LDROM is page aligned, in other words, the size of APROM or LDROM is a integer times of 512 bytes. The page address of APROM is {9'd0 --- APAEND[17:9]}(we have 512 pages totally in main flash)

4.4 General Purpose I/O (GPIO)

4.4.1 Overview

The PAN1020 has up to 41 General Purpose I/O pins to be shared with other function pins depending on the chip configuration. There are 25 GPIOs for QFN32 package, 40 GPIOs for PAN1020DY, 41 GPIOs for PAN1020BY and 15 GPIOs for QSOP24 package. These 30 pins are arranged in 6 ports named as P0, P1, P2, P3, P4 and P5. Each of the 30 pins is independent and has the corresponding register bits to control the pin mode function and data.

The I/O type of each pin can be configured by software individually as Input, Push-pull output, Open-drain output, or Quasi-bidirectional mode. After the chip is reset, the I/O mode of all pins is stay in input mode and each port data register Px_DOUT[n] resets to 1. For Quasi-bidirectional mode, each I/O pin is equipped with a very weak individual pull-up resistor about 110k~ 300kΩ for VDD is from 2.2 V to 3.6 V.

4.4.2 Features

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

4.4.3 Block Diagram

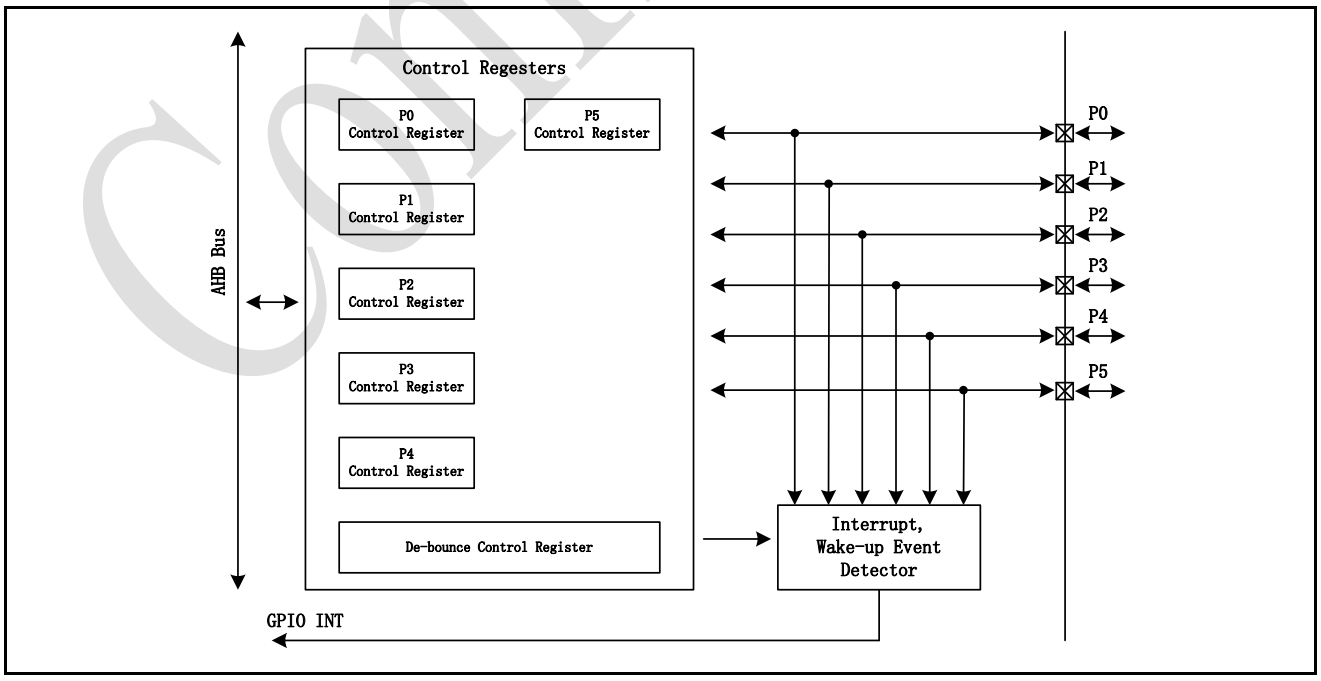


Figure 4-25 GPIO Controller Block Diagram

4.4.4 Basic Configuration

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

4.4.5 Functional Description

4.4.5.1 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.

4.4.5.2 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.

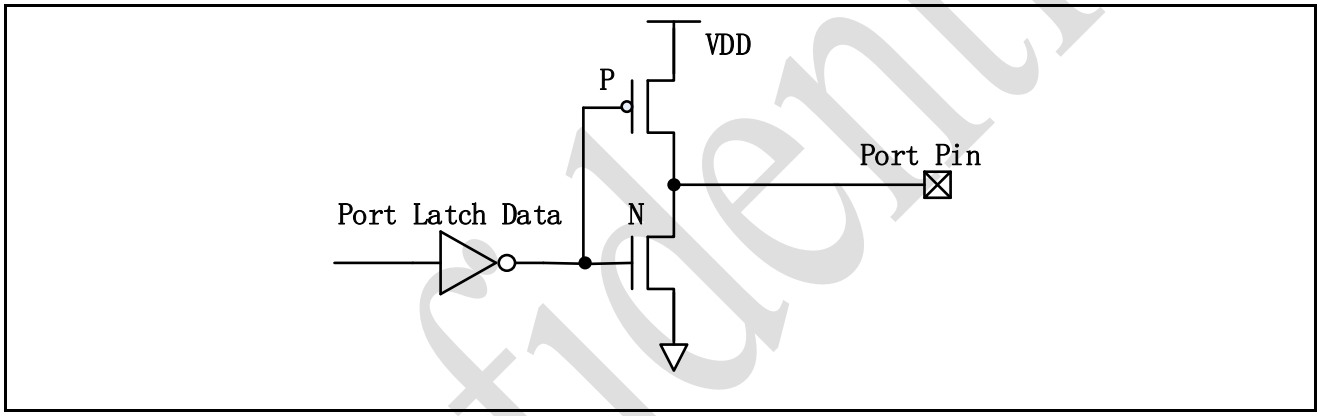


Figure 4-26 Push-Pull Output

4.4.5.3 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.

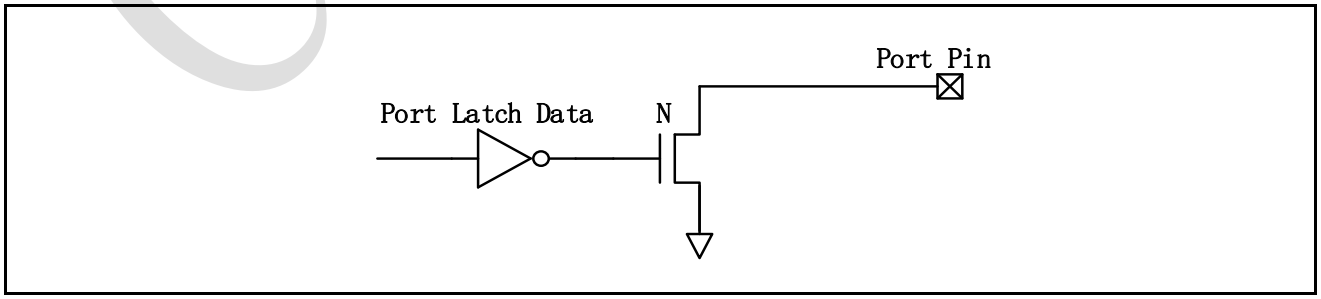


Figure 4-27 Open-Drain Output

4.4.5.4 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.

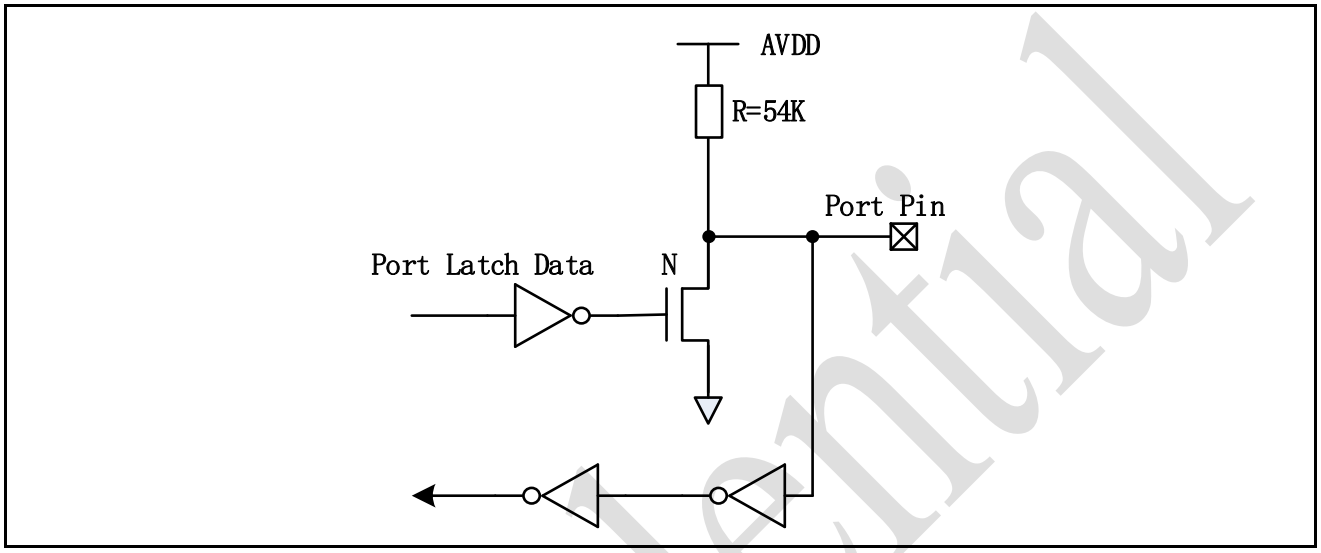


Figure 4-28 Quasi-Bidirectional I/O Mode

4.4.6 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.

4.4.7 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 = 0x5000_4000**				
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	0x00FF_0000
P0_DOUT	GP_BA+0x008	R/W	P0 Data Output Value	0x0000_0000
P0_DATMSK	GP_BA+0x00C	R/W	P0 Data Output Write Mask	0x0000_0000

P0_PIN	GP_BA+0x010	R	P0 Pin Value	0x0000_00XX
P0_DBEN	GP_BA+0x014	R/W	P0 De-bounce Enable Control	0x0000_0000
P0_INTTYPE	GP_BA+0x018	R/W	P0 Interrupt Mode Control	0x0000_0000
P0_INTEN	GP_BA+0x01C	R/W	P0 Interrupt Enable Control	0x0000_0000
P0_INTSRC	GP_BA+0x020	R/W	P0 Interrupt Source Flag	0x0000_0000
P1_MODE	GP_BA+0x040	R/W	P1 I/O Mode Control	0x0000_0000
P1_DINOFF	GP_BA+0x044	R/W	P1 Digital Input Path Disable Control	0x00FE_0000
P1_DOUT	GP_BA+0x048	R/W	P1 Data Output Value	0x0000_0000
P1_DATMSK	GP_BA+0x04C	R/W	P1 Data Output Write Mask	0x0000_0000
P1_PIN	GP_BA+0x050	R	P1 Pin Value	0x0000_00XX
P1_DBEN	GP_BA+0x054	R/W	P1 De-bounce Enable Control	0x0000_0000
P1_INTTYPE	GP_BA+0x058	R/W	P1 Interrupt Mode Control	0x0000_0000
P1_INTEN	GP_BA+0x05C	R/W	P1 Interrupt Enable Control	0x0000_0000
P1_INTSRC	GP_BA+0x060	R/W	P1 Interrupt Source Flag	0x0000_0000
P2_MODE	GP_BA+0x080	R/W	P2 I/O Mode Control	0x0000_0000
P2_DINOFF	GP_BA+0x084	R/W	P2 Digital Input Path Disable Control	0x00FF_0000
P2_DOUT	GP_BA+0x088	R/W	P2 Data Output Value	0x0000_0000
P2_DATMSK	GP_BA+0x08C	R/W	P2 Data Output Write Mask	0x0000_0000
P2_PIN	GP_BA+0x090	R	P2 Pin Value	0x0000_00XX
P2_DBEN	GP_BA+0x094	R/W	P2 De-bounce Enable Control	0x0000_0000
P2_INTTYPE	GP_BA+0x098	R/W	P2 Interrupt Mode Control	0x0000_0000
P2_INTEN	GP_BA+0x09C	R/W	P2 Interrupt Enable Control	0x0000_0000
P2_INTSRC	GP_BA+0x0A0	R/W	P2 Interrupt Source Flag	0x0000_0000
P3_MODE	GP_BA+0x0C0	R/W	P3 I/O Mode Control	0x0000_0000
P3_DINOFF	GP_BA+0x0C4	R/W	P3 Digital Input Path Disable Control	0x0077_0000
P3_DOUT	GP_BA+0x0C8	R/W	P3 Data Output Value	0x0000_0000
P3_DATMSK	GP_BA+0x0CC	R/W	P3 Data Output Write Mask	0x0000_0000
P3_PIN	GP_BA+0x0D0	R	P3 Pin Value	0x0000_00XX
P3_DBEN	GP_BA+0x0D4	R/W	P3 De-bounce Enable Control	0x0000_0000
P3_INTTYPE	GP_BA+0x0D8	R/W	P3 Interrupt Mode Control	0x0000_0000
P3_INTEN	GP_BA+0x0DC	R/W	P3 Interrupt Enable Control	0x0000_0000
P3_INTSRC	GP_BA+0x0E0	R/W	P3 Interrupt Source Flag	0x0000_0000
P4_MODE	GP_BA+0x100	R/W	P4 I/O Mode Control	0x0000_0000
P4_DINOFF	GP_BA+0x104	R/W	P4 Digital Input Path Disable Control	0x0000_0000
P4_DOUT	GP_BA+0x108	R/W	P4 Data Output Value	0x0000_0000
P4_DATMSK	GP_BA+0x10C	R/W	P4 Data Output Write Mask	0x0000_0000
P4_PIN	GP_BA+0x110	R	P4 Pin Value	0x0000_00XX
P4_DBEN	GP_BA+0x114	R/W	P4 De-bounce Enable Control	0x0000_0000
P4_INTTYPE	GP_BA+0x118	R/W	P4 Interrupt Mode Control	0x0000_0000
P4_INTEN	GP_BA+0x11C	R/W	P4 Interrupt Enable Control	0x0000_0000
P4_INTSRC	GP_BA+0x120	R/W	P4 Interrupt Source Flag	0x0000_0000
P5_MODE	GP_BA+0x140	R/W	P5 I/O Mode Control	0x0000_0000

P5_DINOFF	GP_BA+0x144	R/W	P5 Digital Input Path Disable Control	0x00F7_0000
P5_DOUT	GP_BA+0x148	R/W	P5 Data Output Value	0x0000_0000
P5_DATMSK	GP_BA+0x14C	R/W	P5 Data Output Write Mask	0x0000_0000
P5_PIN	GP_BA+0x150	R	P5 Pin Value	0x0000_00XX
P5_DBEN	GP_BA+0x154	R/W	P5 De-bounce Enable Control	0x0000_0000
P5_INTTYPE	GP_BA+0x158	R/W	P5 Interrupt Mode Control	0x0000_0000
P5_INTEN	GP_BA+0x15C	R/W	P5 Interrupt Enable Control	0x0000_0000
P5_INTSRC	GP_BA+0x160	R/W	P5 Interrupt Source Flag	0x0000_0000
GPIO_DBCTL	GP_BA+0x180	R/W	De-bounce Cycle Control	0x0000_0000
P00_PDIO	GP_BA+0x200	R/W	GPIO P0.0 Pin Data Input/Output	0x0000_0001
P01_PDIO	GP_BA+0x204	R/W	GPIO P0.1 Pin Data Input/Output	0x0000_0001
P02_PDIO	GP_BA+0x208	R/W	GPIO P0.2 Pin Data Input/Output	0x0000_0001
P03_PDIO	GP_BA+0x20C	R/W	GPIO P0.3 Pin Data Input/Output	0x0000_0001
P04_PDIO	GP_BA+0x210	R/W	GPIO P0.4 Pin Data Input/Output	0x0000_0001
P05_PDIO	GP_BA+0x214	R/W	GPIO P0.5 Pin Data Input/Output	0x0000_0001
P06_PDIO	GP_BA+0x218	R/W	GPIO P0.6 Pin Data Input/Output	0x0000_0001
P07_PDIO	GP_BA+0x21C	R/W	GPIO P0.7 Pin Data Input/Output	0x0000_0001
P10_PDIO	GP_BA+0x220	R/W	GPIO P1.0 Pin Data Input/Output	0x0000_0001
P11_PDIO	GP_BA+0x224	R/W	GPIO P1.1 Pin Data Input/Output	0x0000_0001
P12_PDIO	GP_BA+0x228	R/W	GPIO P1.2 Pin Data Input/Output	0x0000_0001
P13_PDIO	GP_BA+0x22C	R/W	GPIO P1.3 Pin Data Input/Output	0x0000_0001
P14_PDIO	GP_BA+0x230	R/W	GPIO P1.4 Pin Data Input/Output	0x0000_0001
P15_PDIO	GP_BA+0x234	R/W	GPIO P1.5 Pin Data Input/Output	0x0000_0001
P16_PDIO	GP_BA+0x238	R/W	GPIO P1.6 Pin Data Input/Output	0x0000_0001
P17_PDIO	GP_BA+0x23C	R/W	GPIO P1.7 Pin Data Input/Output	0x0000_0001
P20_PDIO	GP_BA+0x240	R/W	GPIO P2.0 Pin Data Input/Output	0x0000_0001
P21_PDIO	GP_BA+0x244	R/W	GPIO P2.1 Pin Data Input/Output	0x0000_0001
P22_PDIO	GP_BA+0x248	R/W	GPIO P2.2 Pin Data Input/Output	0x0000_0001
P23_PDIO	GP_BA+0x24C	R/W	GPIO P2.3 Pin Data Input/Output	0x0000_0001
P24_PDIO	GP_BA+0x250	R/W	GPIO P2.4 Pin Data Input/Output	0x0000_0001
P25_PDIO	GP_BA+0x254	R/W	GPIO P2.5 Pin Data Input/Output	0x0000_0001
P26_PDIO	GP_BA+0x258	R/W	GPIO P2.6 Pin Data Input/Output	0x0000_0001
P27_PDIO	GP_BA+0x24C	R/W	GPIO P2.7 Pin Data Input/Output	0x0000_0001
P30_PDIO	GP_BA+0x260	R/W	GPIO P3.0 Pin Data Input/Output	0x0000_0001
P31_PDIO	GP_BA+0x264	R/W	GPIO P3.1 Pin Data Input/Output	0x0000_0001
P32_PDIO	GP_BA+0x268	R/W	GPIO P3.2 Pin Data Input/Output	0x0000_0001
P34_PDIO	GP_BA+0x270	R/W	GPIO P3.4 Pin Data Input/Output	0x0000_0001
P35_PDIO	GP_BA+0x274	R/W	GPIO P3.5 Pin Data Input/Output	0x0000_0001
P36_PDIO	GP_BA+0x278	R/W	GPIO P3.6 Pin Data Input/Output	0x0000_0001
P46_PDIO	GP_BA+0x298	R/W	GPIO P4.6 Pin Data Input/Output	0x0000_0001
P47_PDIO	GP_BA+0x29C	R/W	GPIO P4.7 Pin Data Input/Output	0x0000_0001
P50_PDIO	GP_BA+0x2A0	R/W	GPIO P5.0 Pin Data Input/Output	0x0000_0001

P51_PDIO	GP_BA+0x2A4	R/W	GPIO P5.1 Pin Data Input/Output	0x0000_0001
P52_PDIO	GP_BA+0x2A8	R/W	GPIO P5.2 Pin Data Input/Output	0x0000_0001
P53_PDIO	GP_BA+0x2AC	R/W	GPIO P5.3 Pin Data Input/Output	0x0000_0001
P54_PDIO	GP_BA+0x2B0	R/W	GPIO P5.4 Pin Data Input/Output	0x0000_0001
P55_PDIO	GP_BA+0x2B4	R/W	GPIO P5.5 Pin Data Input/Output	0x0000_0001
P56_PDIO	GP_BA+0x2B8	R/W	GPIO P5.6 Pin Data Input/Output	0x0000_0001
P57_PDIO	GP_BA+0x2BC	R/W	GPIO P5.7 Pin Data Input/Output	0x0000_0001

4.4.8 GPIO Register Description

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

Register	Offset	R/W	Description	Reset Value
P0_MODE	GP_BA+0x000	R/W	P0 I/O Mode Control	0x0000_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
P3_MODE	GP_BA+0x0C0	R/W	P3 I/O Mode Control	0x0000_0000
P4_MODE	GP_BA+0x100	R/W	P4 I/O Mode Control	0x0000_0000
P5_MODE	GP_BA+0x140	R/W	P5 I/O Mode Control	0x0000_0000

Bits	Descriptions	
[31:16]	Reserved	Reserved.
[2n+1:2n] n=0,1..7	MODEn	Port 0-5 I/O Pin[N] Mode Control Determine each I/O mode of Px.n pins. 00 = Px.n is in Input mode. 01 = Px.n is in Push-pull Output mode. 10 = Px.n is in Open-drain Output mode. 11 = Px.n is in Quasi-bidirectional mode. **Note1**: Max. n=7 for port 0. Max. n=7 for port 1. Max. n=7 for port 2. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,.5 are reserved. Max. n=7 for port 5

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

Register	Offset	R/W	Description	Reset Value
P0_DINOFF	GP_BA+0x004	R/W	P0 Digital Input Path Disable Control	0x00FF_0000
P1_DINOFF	GP_BA+0x044	R/W	P1 Digital Input Path Disable Control	0x00FE_0000
P2_DINOFF	GP_BA+0x084	R/W	P2 Digital Input Path Disable Control	0x00FF_0000

P3_DINOFF	GP_BA+0x0C4	R/W	P3 Digital Input Path Disable Control	0x0077_0000
P4_DINOFF	GP_BA+0x104	R/W	P4 Digital Input Path Disable Control	0x0000_00C0
P5_DINOFF	GP_BA+0x144	R/W	P5 Digital Input Path Disable Control	0x00F7_0000

Bits	Descriptions	
[31:24]	Reserved	Reserved.
[n+16] n=0,1..7	DINOFF[n]	Port 0-5 Pin[N] Digital Input Path Disable Control Each of these bits is used to control if the digital input path of corresponding Px.n pin is disabled. If input is analog signal, users can disable Px.n digital input path to avoid input current leakage. 0 = Px.n digital input path Enabled. 1 = Px.n digital input path Disabled (digital input tied to low). Note1: Max. n=7 for port 0. Max. n=7 for port 1. Max. n=7 for port 2. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,.5 are reserved. Max. n=7 for port 5
[15:8]	Reserved	Reserved.
[n] n=0,1..7	PUEN[n]	Port 0-5 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.

4.4.8.3 Port 0-5 Data Output Value (Px_DOUT)

Register	Offset	R/W	Description	Reset Value
P0_DOUT	GP_BA+0x008	R/W	P0 Data Output Value	0x0000_0000
P1_DOUT	GP_BA+0x048	R/W	P1 Data Output Value	0x0000_0000
P2_DOUT	GP_BA+0x088	R/W	P2 Data Output Value	0x0000_0000
P3_DOUT	GP_BA+0x0C8	R/W	P3 Data Output Value	0x0000_0000
P4_DOUT	GP_BA+0x108	R/W	P4 Data Output Value	0x0000_0000
P5_DOUT	GP_BA+0x148	R/W	P5 Data Output Value	0x0000_0000

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[n] n=0,1..7	DOUT[n]	Port 0-5 Pin[N] Output Value Each of these bits controls the status of a Px.n pin when the Px.n is configured as Push-pull output, Open-drain output or Quasi-bidirectional mode.

		0 = Px.n will drive Low if the Px.n pin is configured as Push-pull output, Open-drain output or Quasi-bidirectional mode. 1 = Px.n will drive High if the Px.n pin is configured as Push-pull output or Quasi-bidirectional mode. **Note1**: Max. n=7 for port 0. Max. n=7 for port 1. Max. n=7 for port 2. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,.5 are reserved. Max. n=7 for port 5

4.4.8.4 Port 0-5 Data Output Write Mask (Px_DATMSK)

Register	Offset	R/W	Description	Reset Value
P0_DATMSK	GP_BA+0x00C	R/W	P0 Data Output Write Mask	0x0000_0000
P1_DATMSK	GP_BA+0x04C	R/W	P1 Data Output Write Mask	0x0000_0000
P2_DATMSK	GP_BA+0x08C	R/W	P2 Data Output Write Mask	0x0000_0000
P3_DATMSK	GP_BA+0x0CC	R/W	P3 Data Output Write Mask	0x0000_0000
P4_DATMSK	GP_BA+0x10C	R/W	P4 Data Output Write Mask	0x0000_0000
P5_DATMSK	GP_BA+0x14C	R/W	P5 Data Output Write Mask	0x0000_0000

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

4.4.8.5 Port 0-5 Pin Value (Px_PIN)

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

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[n] n=0,1..7	PIN[n]	Port 0-5 Pin[N] Pin Value Each bit of the register reflects the actual status of the respective Px.n pin. If the bit is 1, it indicates the corresponding pin status is high; else the pin status is low. **Note1**: Max. n=7 for port 0. Max. n=7 for port 1. Max. n=7 for port 2. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,.5 are reserved. Max. n=7 for port 5

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

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

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[n] n=0,1..7	DBEN[n]	Port 0-5 Pin[N] Input Signal De-bounce Enable Bit The DBEN[n] bit is used to enable the de-bounce function for each corresponding bit. If the input signal pulse width cannot be sampled by continuous two de-bounce sample cycle, the input signal transition is seen as the signal bounce and will not trigger the interrupt. The de-bounce clock source is controlled by DBCLKSRC (GPIO_DBCTL [4]), one

		de-bounce sample cycle period is controlled by DBCLKSEL (GPIO_DBCTL [3:0]). 0 = Px.n de-bounce function Disabled. 1 = Px.n de-bounce function Enabled. The de-bounce function is valid only for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignore. **Note**: 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. **Note1**: Max. n=7 for port 0. Max. n=7 for port 1. Max. n=7 for port 2. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,.5 are reserved. Max. n=7 for port 5

4.4.8.7 Port 0-5 Interrupt Mode Control (Px_ INTTYPE)

Register	Offset	R/W	Description	Reset Value
P0_INTTYPE	GP_BA+0x018	R/W	P0 Interrupt Mode Control	0x0000_0000
P1_INTTYPE	GP_BA+0x058	R/W	P1 Interrupt Mode Control	0x0000_0000
P2_INTTYPE	GP_BA+0x098	R/W	P2 Interrupt Mode Control	0x0000_0000
P3_INTTYPE	GP_BA+0x0D8	R/W	P3 Interrupt Mode Control	0x0000_0000
P4_INTTYPE	GP_BA+0x118	R/W	P4 Interrupt Mode Control	0x0000_0000
P5_INTTYPE	GP_BA+0x158	R/W	P5 Interrupt Mode Control	0x0000_0000

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

		Note1: Max. n=7 for port 0. Max. n=7 for port 1. Max. n=7 for port 2. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,.5 are reserved. Max. n=7 for port 5

4.4.8.8 Port 0-5 Interrupt Enable Control (Px_INTEN)

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

Bits	Descriptions	
[31:24]	Reserved	Reserved.
[n+16] n=0,1..7	RHIEN[n]	Port 0-5 Pin[N] Rising Edge Or High Level Interrupt Trigger Type Enable Bit The RHIEN (Px_INTEN[n+16]) bit is used to enable the interrupt for each of the corresponding input Px.n pin. Set bit to 1 also enable the pin wake-up function. When setting the RHIEN (Px_INTEN[n+16]) bit to 1: If the interrupt is level trigger (TYPE (Px_INTTYPE[n]) bit is set to 1), the input Px.n pin will generate the interrupt while this pin state is at high level. If the interrupt is edge trigger (TYPE (Px_INTTYPE[n]) bit is set to 0), the input Px.n pin will generate the interrupt while this pin state changed from low to high. 0 = Px.n level high or low to high interrupt Disabled. 1 = Px.n level high or low to high interrupt Enabled. **Note1**: Max. n=7 for port 0. Max. n=7 for port 1. Max. n=7 for port 2. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,.5 are reserved. Max. n=7 for port 5
[15:8]	Reserved	Reserved.
[n] n=0,1..7	FLIEN[n]	Port 0-5 Pin[N] Falling Edge Or Low Level Interrupt Trigger Type Enable Bit The FLIEN (Px_INTEN[n]) bit is used to enable the interrupt for each of the corresponding input Px.n pin. Set bit to 1 also enable the pin wake-up function. When setting the FLIEN (Px_INTEN[n]) bit to 1: If the interrupt is level trigger (TYPE (Px_INTTYPE[n]) bit is set to 1), the input Px.n

		pin will generate the interrupt while this pin state is at low level. If the interrupt is edge trigger (TYPE (Px_INTTYPE[n]) bit is set to 0), the input Px.n pin will generate the interrupt while this pin state changed from high to low. 0 = Px.n level low or high to low interrupt Disabled. 1 = Px.n level low or high to low interrupt Enabled. **Note1**: Max. n=7 for port 0. Max. n=7 for port 1. Max. n=7 for port 2. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,.5 are reserved. Max. n=7 for port 5

4.4.8.9 Port 0-5 Interrupt Source Flag (Px_INTSRC)

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

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[n] n=0,1..7	INTSRC[n]	Port 0-5 Pin[N] Interrupt Source Flag Write Operation: 0 = No action. 1 = Clear the corresponding pending interrupt. Read Operation: 0 = No interrupt at Px.n. 1 = Px.n generates an interrupt. **Note1**: Max. n=7 for port 0. Max. n=7 for port 1. Max. n=7 for port 2. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,.5 are reserved. Max. n=7 for port 5

4.4.8.10 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_0000

Bits	Descriptions	
[31:5]	Reserved	Reserved.
[4]	DBCLKSRC	De-bounce Counter Clock Source Selection 0 = De-bounce counter clock source is HCLK. 1 = De-bounce counter clock source is 32 kHz internal low speed RC oscillator (LIRC).
[3:0]	DBCLKSEL	De-bounce Sampling Cycle Selection 0000 = Sample interrupt input once per 1 clock. 0001 = Sample interrupt input once per 2 clocks. 0010 = Sample interrupt input once per 4 clocks. 0011 = Sample interrupt input once per 8 clocks. 0100 = Sample interrupt input once per 16 clocks. 0101 = Sample interrupt input once per 32 clocks. 0110 = Sample interrupt input once per 64 clocks. 0111 = Sample interrupt input once per 128 clocks. 1000 = Sample interrupt input once per 256 clocks. 1001 = Sample interrupt input once per 2*256 clocks. 1010 = Sample interrupt input once per 4*256 clocks. 1011 = Sample interrupt input once per 8*256 clocks. 1100 = Sample interrupt input once per 16*256 clocks. 1101 = Sample interrupt input once per 32*256 clocks. 1110 = Sample interrupt input once per 64*256 clocks. 1111 = Sample interrupt input once per 128*256 clocks.

4.4.8.11 GPIO Px.n Data Input/Output (Pxn_PDIO)

Register	Offset	R/W	Description	Reset Value
P00_PDIO	GP_BA+0x200	R/W	GPIO P0.0 Pin Data Input/Output	0x0000_0001
P01_PDIO	GP_BA+0x204	R/W	GPIO P0.1 Pin Data Input/Output	0x0000_0001
P02_PDIO	GP_BA+0x208	R/W	GPIO P0.2 Pin Data Input/Output	0x0000_0001
P03_PDIO	GP_BA+0x20C	R/W	GPIO P0.3 Pin Data Input/Output	0x0000_0001
P04_PDIO	GP_BA+0x210	R/W	GPIO P0.4 Pin Data Input/Output	0x0000_0001
P05_PDIO	GP_BA+0x214	R/W	GPIO P0.5 Pin Data Input/Output	0x0000_0001
P06_PDIO	GP_BA+0x218	R/W	GPIO P0.6 Pin Data Input/Output	0x0000_0001
P07_PDIO	GP_BA+0x21C	R/W	GPIO P0.7 Pin Data Input/Output	0x0000_0001
P10_PDIO	GP_BA+0x220	R/W	GPIO P1.0 Pin Data Input/Output	0x0000_0001
P11_PDIO	GP_BA+0x224	R/W	GPIO P1.1 Pin Data Input/Output	0x0000_0001
P12_PDIO	GP_BA+0x228	R/W	GPIO P1.2 Pin Data Input/Output	0x0000_0001
P13_PDIO	GP_BA+0x22C	R/W	GPIO P1.3 Pin Data Input/Output	0x0000_0001
P14_PDIO	GP_BA+0x230	R/W	GPIO P1.4 Pin Data Input/Output	0x0000_0001

P15_PDIO	GP_BA+0x234	R/W	GPIO P1.5 Pin Data Input/Output	0x0000_0001
P16_PDIO	GP_BA+0x238	R/W	GPIO P1.6 Pin Data Input/Output	0x0000_0001
P17_PDIO	GP_BA+0x23C	R/W	GPIO P1.7 Pin Data Input/Output	0x0000_0001
P20_PDIO	GP_BA+0x240	R/W	GPIO P2.0 Pin Data Input/Output	0x0000_0001
P21_PDIO	GP_BA+0x244	R/W	GPIO P2.1 Pin Data Input/Output	0x0000_0001
P22_PDIO	GP_BA+0x248	R/W	GPIO P2.2 Pin Data Input/Output	0x0000_0001
P23_PDIO	GP_BA+0x24C	R/W	GPIO P2.3 Pin Data Input/Output	0x0000_0001
P24_PDIO	GP_BA+0x250	R/W	GPIO P2.4 Pin Data Input/Output	0x0000_0001
P25_PDIO	GP_BA+0x254	R/W	GPIO P2.5 Pin Data Input/Output	0x0000_0001
P26_PDIO	GP_BA+0x258	R/W	GPIO P2.6 Pin Data Input/Output	0x0000_0001
P27_PDIO	GP_BA+0x24C	R/W	GPIO P2.7 Pin Data Input/Output	0x0000_0001
P30_PDIO	GP_BA+0x260	R/W	GPIO P3.0 Pin Data Input/Output	0x0000_0001
P31_PDIO	GP_BA+0x264	R/W	GPIO P3.1 Pin Data Input/Output	0x0000_0001
P32_PDIO	GP_BA+0x268	R/W	GPIO P3.2 Pin Data Input/Output	0x0000_0001
P34_PDIO	GP_BA+0x270	R/W	GPIO P3.4 Pin Data Input/Output	0x0000_0001
P35_PDIO	GP_BA+0x274	R/W	GPIO P3.5 Pin Data Input/Output	0x0000_0001
P36_PDIO	GP_BA+0x278	R/W	GPIO P3.6 Pin Data Input/Output	0x0000_0001
P46_PDIO	GP_BA+0x298	R/W	GPIO P4.6 Pin Data Input/Output	0x0000_0001
P47_PDIO	GP_BA+0x29C	R/W	GPIO P4.7 Pin Data Input/Output	0x0000_0001
P50_PDIO	GP_BA+0x2A0	R/W	GPIO P5.0 Pin Data Input/Output	0x0000_0001
P51_PDIO	GP_BA+0x2A4	R/W	GPIO P5.1 Pin Data Input/Output	0x0000_0001
P52_PDIO	GP_BA+0x2A8	R/W	GPIO P5.2 Pin Data Input/Output	0x0000_0001
P53_PDIO	GP_BA+0x2AC	R/W	GPIO P5.3 Pin Data Input/Output	0x0000_0001
P54_PDIO	GP_BA+0x2B0	R/W	GPIO P5.4 Pin Data Input/Output	0x0000_0001
P55_PDIO	GP_BA+0x2B4	R/W	GPIO P5.5 Pin Data Input/Output	0x0000_0001
P56_PDIO	GP_BA+0x2B8	R/W	GPIO P5.6 Pin Data Input/Output	0x0000_0001
P57_PDIO	GP_BA+0x2BC	R/W	GPIO P5.7 Pin Data Input/Output	0x0000_0001

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

		Max. n=7 for port 2. Max. n=7 for port 3, n=3, n=7 are reserved. Max. n=7 for port 4, n=0,.5 are reserved. Max. n=7 for port 5

4.5 Timer Controller (TMR)

4.5.1 Overview

The Timer Controller includes three 32-bit timers, TMR0, TMR1 and TMR2, allowing user to easily implement a timer control for applications. The timer can perform functions, such as frequency measurement, delay timing, clock generation, event counting by external input pins, and interval measurement by external capture pins.

4.5.2 Features

- Three sets of 32-bit timer with 24-bit up counter and one 8-bit prescale counter
- Independent clock source for each timer
- Provides one-shot, periodic, toggle-output and continuous counting operation modes
- 24-bit up counter value is readable through CNT (TIMRTx_CNT[23:0])
- Supports event counting function
- 24-bit capture value is readable through CAPDAT (TIMERx_CAP[23:0])
- Supports external capture pin event for interval measurement
- Supports external capture pin event to reset 24-bit up counter
- Supports chip wake-up from Idle/Power-down mode if a timer interrupt signal is generated
- Supports 32KHz clock capture for interval measurement

4.5.3 Block Diagram

The timer controller block diagram and clock control are shown in Figure 4-29 and Figure 4-30.

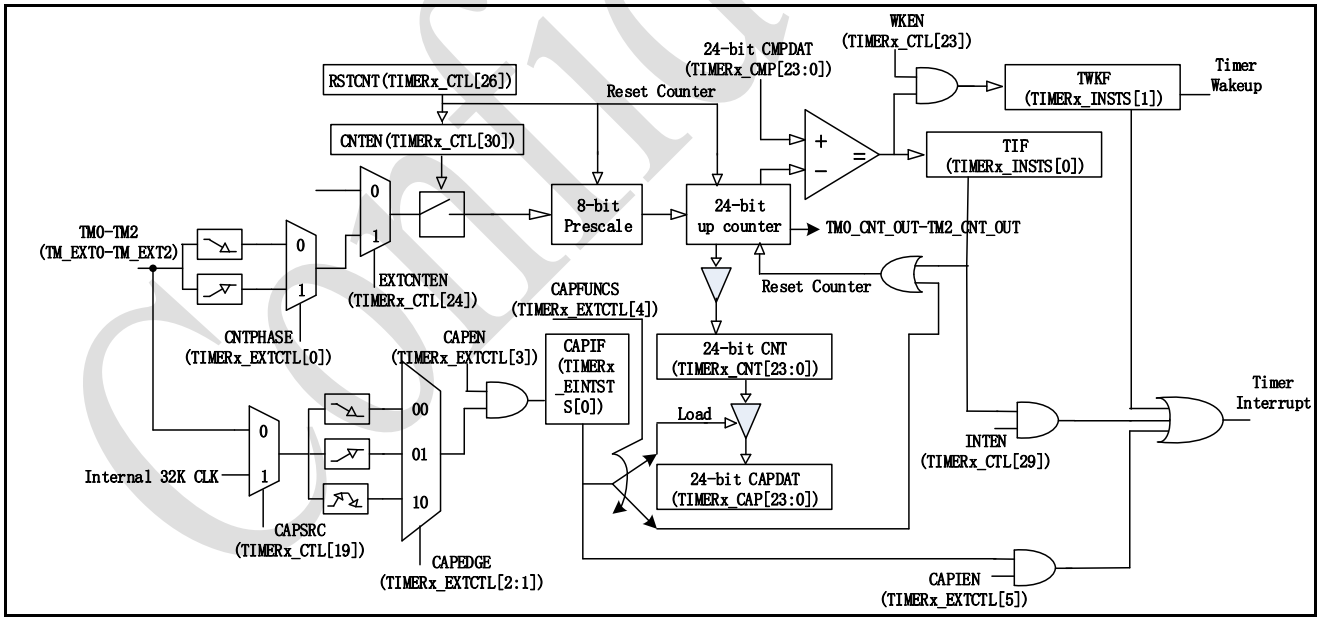


Figure 4-29 Timer Controller Block Diagram (Three TRM)

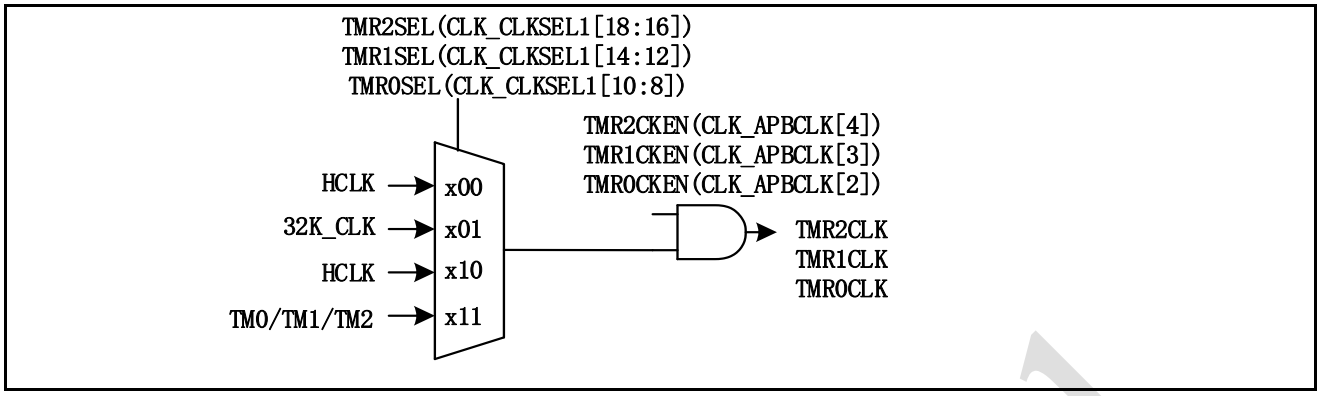


Figure 4-30 Clock Source of Timer Controller (Three TRM)

4.5.4 Basic Configuration

The peripheral clock source of TMR0, TMR1, and TMR2 can be enabled in TMRxCKEN (CLK_APBCLK[4:2]) and selected as clock source in TMR0SEL (CLK_CLKSEL1[10:8]) for TMR0, TMR1SEL (CLK_CLKSEL1[14:12]) for TMR1, and TMR1SEL (CLK_CLKSEL1[18:16]) for TMR2.

4.5.5 Functional Description

4.5.5.1 Timer Flag

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

4.5.5.2 Timer Interrupt Flag

The timer controller supports two interrupt flags; one is TIF (TIMERx_INTSTS[0]) which is set while timer counter value CNT (TIMERx_CNT[23:0]) matches the timer compared value CMPDAT (TIMERx_CMP[23:0])and INTEN (TIMERx_CTL[29]) is set to 1.And the other is CAPIF (TIMERx_EINTSTS[0]) which is set when the transition on the TMx_EXT pin associated CAPEDGE (TIMERx_EXTCTL[2:1]) setting, and CAPIEN (TIMERx_EXTCTL[5]) is set to 1.

4.5.5.3 Timer Counting Operation Mode

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

4.5.5.3.1. One-shot Mode

If the timer controller is configured at one-shot mode OPMODE (TIMERx_CTL[28:27] is 2’b00) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. Once the CNT (TIMERx_CNT[23:0]) value reaches CMPDAT (TIMERx_CMP[23:0]) value, the TIF (TIMERx_INTSTS[0]) will be set to 1, CNT value and CNTEN bit is cleared automatically by timer controller then timer counting operation stops. In the meantime, if the INTEN (TIMERx_CTL[29]) is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU also.

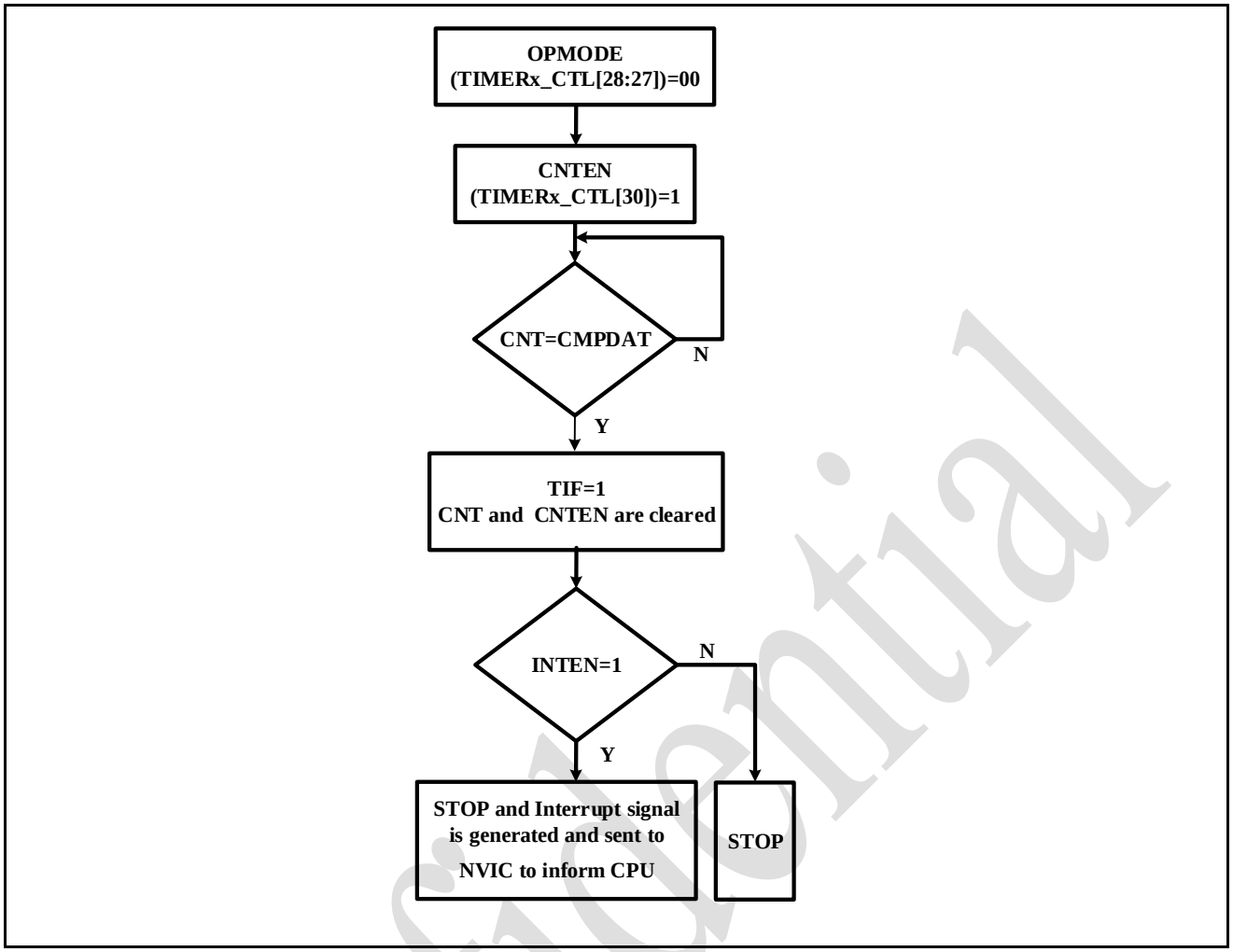


Figure 4-31 One-shot Mode

4.5.5.3.2. Periodic Mode

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

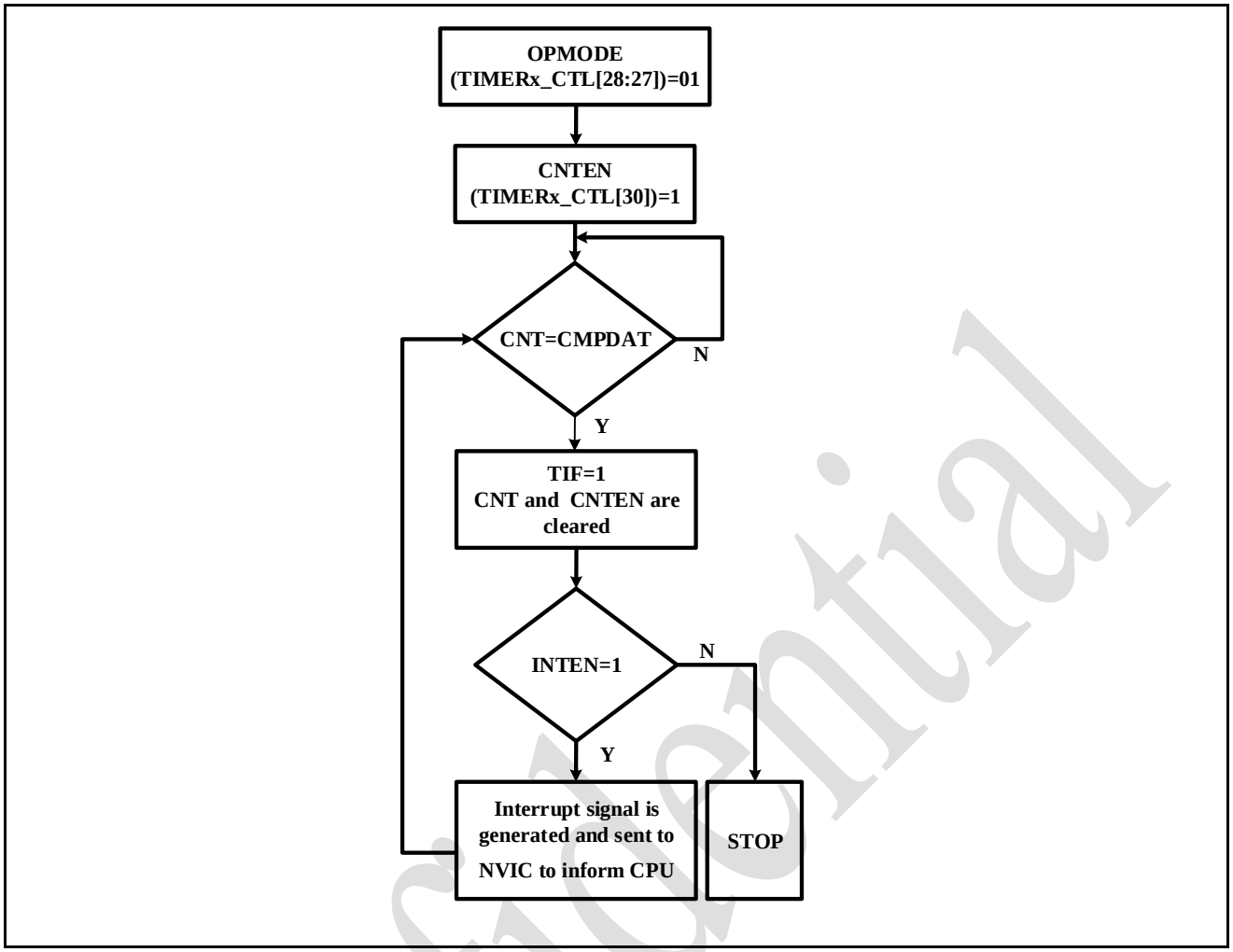


Figure 4-32 Periodic Mode

4.5.5.3.3. Toggle-output Mode

If the timer controller is configured at toggle-output mode (TIMERx_CTL[28:27] is 2'b10) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. The counting operation of toggle-output mode is almost the same as periodic mode, except toggle-output mode has associated TM0 ~ TM2 pin to output signal while specify TIF (TIMERx_INTSTS[0]) is set. Thus, the toggle-output signal on TMx_CNT_OUT（x=0,1,2）pin is high and changing back and forth with 50% duty cycle.

4.5.5.3.4. Continuous Counting Mode

If the timer controller is configured at continuous counting mode (TIMERx_CTL[28:27] is 2'b11) and CNTEN (TIMERx_CTL[30]) is set, the timer counter starts up counting. Once the CNT (TIMERx_CNT[23:0]) value reaches the CMPDAT (TIMERx_CMP[23:0]) value, the TIF (TIMERx_INTSTS[0]) will be set to 1 and the CNT value keeps up counting. In the meantime, if the INTEN (TIMERx_CTL[29]) is enabled, the timer interrupt signal is generated and sent to NVIC to inform CPU. User can change different CMPDAT value immediately without disabling timer counting and restarting timer counting in this mode.

For example, CMPDAT value is set as 80, first. The TIF will set to 1 when CNT value is equal to 80, the timer counter is kept counting and CNT value will not go back to 0, it continues to count 81,

82, 83,…to 2^{24}-1, 0, 1, 2, 3, …to 2^{24}-1 again and again. Next, if user programs the CMPDAT value as 200 and clears TIF, the TIF will be set to 1 again when CNT value reaches 200. At last, user programs CMPDAT as 500 and clears TIF, the TIF will set to 1 again when CNT value reaches to 500. In this mode, the timer counting is continuous.

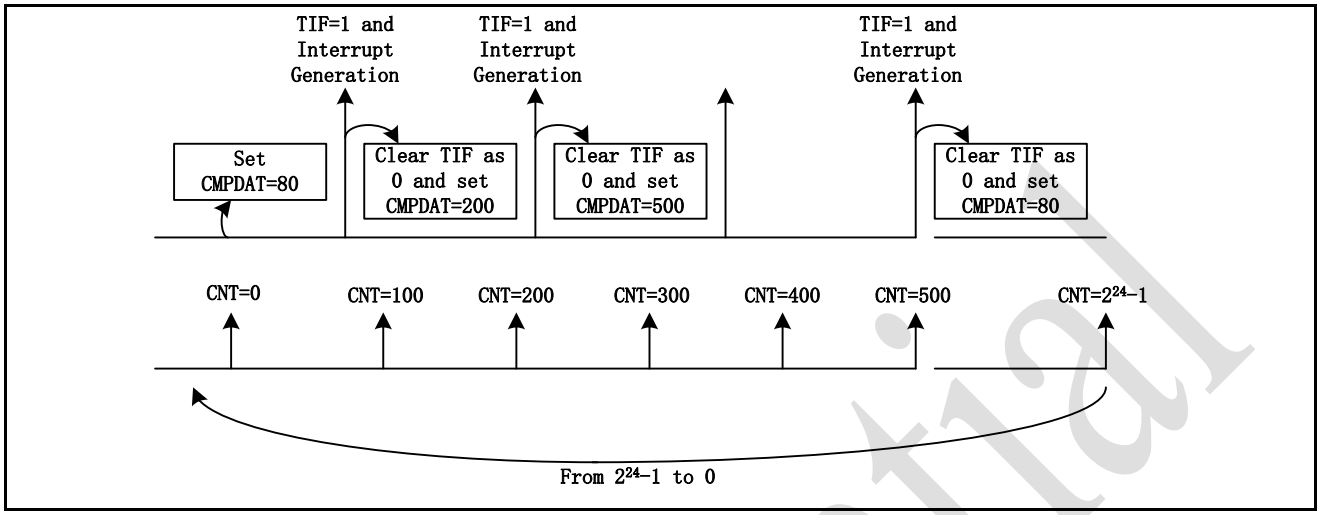


Figure 4-33 Continuous Counting Mode

4.5.5.4 Event Counting Mode

The timer controller also provides an application which can count the input event from TMx (x= 0~2) pin and the number of event will reflect to CNT (TIMERx_CNT[23:0]) value. It is also called as event counting function. In this function, EXTCNTEN (TIMERx_CTL[24]) should be set and the timer peripheral clock source should be set as HCLK.

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

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

4.5.5.5 Input Capture Function

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

4.5.5.5.1. Free-Counting Capture Mode

The event capture function is used to load CNT (TIMERx_CNT[23:0]) value to CAPDAT (TIMERx_CAP[23:0]) value while edge transition detected on TMx_EXT (x= 0~2) pin. In this mode, CAPSEL (TIMERx_EXTCTL[8]) and CAPFUNCS (TIMERx_EXTCTL[4]) should be as 0 for select TMx_EXT transition is using to trigger event capture function and the timer peripheral clock source should be set as HCLK.

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

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

Users must consider the Timer will keep register TIMERx_CAP unchanged and drop the new capture value, if the CPU does not clear the CAPIF (TIMERx_EINTSTS[0]) status. The operation method is described in Figure 4-34.

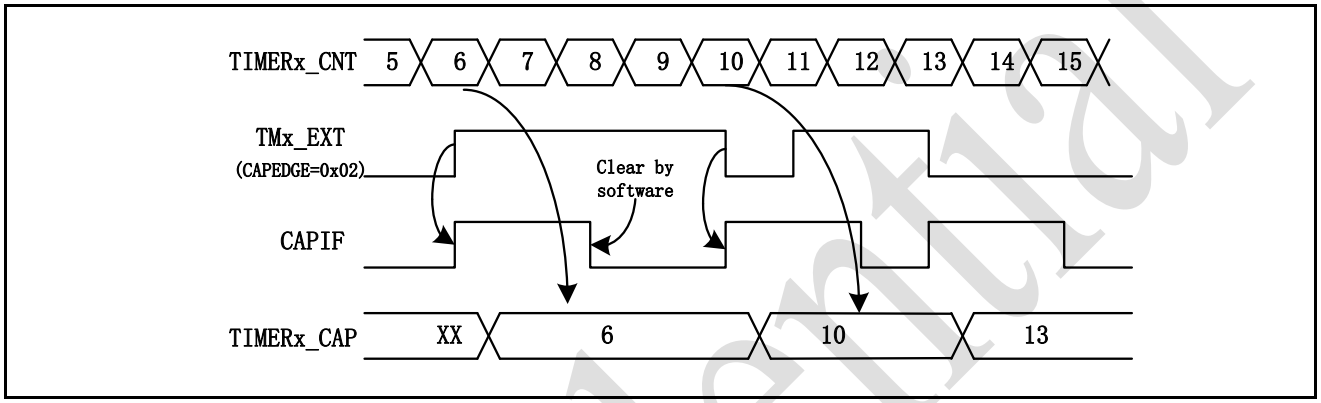


Figure 4-34 Free-Counting Capture Mode

4.5.5.5.2. External Reset Counter Mode

The timer controller also provides reset counter function to reset CNT (TIMERx_CNT[23:0]) value while edge transition detected on TMx_EXT (x= 0~2). In this mode, most the settings are the same as event capture mode except CAPFUNCS (TIMERx_EXTCTL[4]) should be as 1 for select TMx_EXT transition is using to trigger reset counter value. The operation method is also described in Figure 4-35.

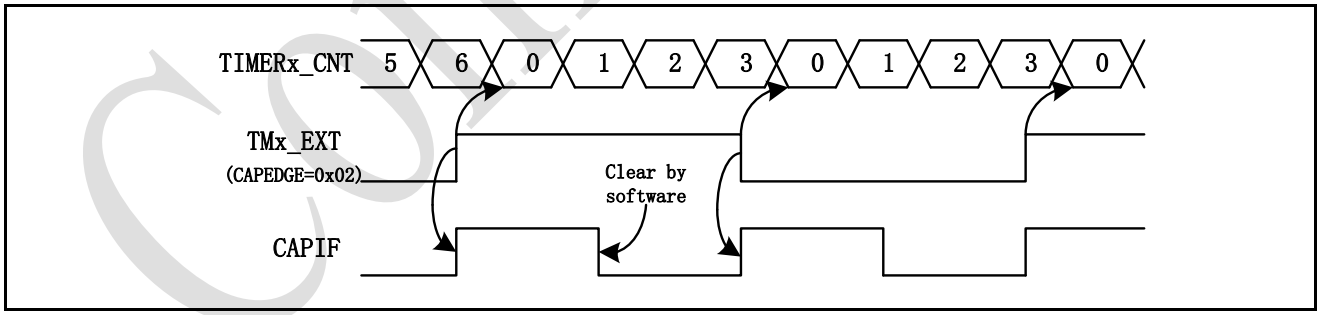


Figure 4-35 External Reset Counter Mode

4.5.5.5.3. Trigger-Counting Capture Mode

If CAPSEL (TIMERx_EXTCTL[8]) is set to 1, CAPEN (TIMERx_EXTCTL[3]) is set to 1 and CAPFUNCS (TIMERx_EXTCTL[4]) is set to 0, the CNT will be reset to 0 then captured into CAPDAT register when TMx_EXT (x= 0~2) pin trigger condition occurred. The TMx_EXT triggeredge can be chosen by CAPEDGE (TIMERx_EXTCTL[2:1]). The detailed operation method is described in

Table 4-12.

When TMx_EXT trigger occurred, CAPIF (TIMERx_EINTSTS[0]) is set to 1, and the interrupt signal is generated, then sent to NVIC to inform CPU if CAPIEN (TIMERx_EXTCTL[5]) is 1. And, the TMx_EXT source operating frequency should be less than 1/3 HCLK frequency if disable TMx_EXT de-bounce or less than 1/8 HCLK frequency if enabling TMx_EXT de-bounce. It also provides TMx_EXT enabled or disabled capture de-bounce function by CAPDBEN (TIMERx_EXTCTL[6]).

Table 4-12 Input Capture Mode Operation

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

				counting, while 1 to 0 transition stops counting.

4.5.6 TMR Register Map

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

Register	Offset	R/W	Description	Reset Value
TRM Base Address: **TMR0_BA = 0x4001_0000** **TMR1_BA = 0x4001_0020** **TMR2_BA = 0x4001_0040**				
TIMER0_CTL	TMR0_BA+0x00	R/W	Timer0 Control and Status Register	0x0000_0005
TIMER0_CMP	TMR0_BA+0x04	R/W	Timer0 Compare Register	0x0000_0000
TIMER0_INTSTS	TMR0_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER0_CNT	TMR0_BA+0x0C	R	Timer0 Data Register	0x0000_0000
TIMER0_CAP	TMR0_BA+0x10	R	Timer0 Capture Data Register	0x0000_0000
TIMER0_EXTCTL	TMR0_BA+0x14	R/W	Timer0 External Control Register	0x0000_0000
TIMER0_EINTSTS	TMR0_BA+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TIMER1_CTL	TMR1_BA+0x00	R/W	Timer1 Control and Status Register	0x0000_0005
TIMER1_CMP	TMR1_BA+0x04	R/W	Timer1 Compare Register	0x0000_0000
TIMER1_INTSTS	TMR1_BA+0x08	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER1_CNT	TMR1_BA+0x0C	R	Timer1 Data Register	0x0000_0000
TIMER1_CAP	TMR1_BA+0x10	R	Timer1 Capture Data Register	0x0000_0000
TIMER1_EXTCTL	TMR1_BA+0x14	R/W	Timer1 External Control Register	0x0000_0000
TIMER1_EINTSTS	TMR1_BA+0x18	R/W	Timer1 External Interrupt Status Register	0x0000_0000
TIMER2_CTL	TMR2_BA+0x00	R/W	Timer1 Control and Status Register	0x0000_0005
TIMER2_CMP	TMR2_BA+0x04	R/W	Timer1 Compare Register	0x0000_0000
TIMER2_INTSTS	TMR2_BA+0x08	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER2_CNT	TMR2_BA+0x0C	R	Timer1 Data Register	0x0000_0000
TIMER2_CAP	TMR2_BA+0x10	R	Timer1 Capture Data Register	0x0000_0000
TIMER2_EXTCTL	TMR2_BA+0x14	R/W	Timer1 External Control Register	0x0000_0000
TIMER2_EINTSTS	TMR2_BA+0x18	R/W	Timer1 External Interrupt Status Register	0x0000_0000

4.5.7 TMR Register Description

4.5.7.1 Timer Control Register (TIMERx_CTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_CTL	TMR0_BA+0x00	R/W	Timer0 Control and Status Register	0x0000_0005
TIMER1_CTL	TMR1_BA+0x00	R/W	Timer1 Control and Status Register	0x0000_0005
TIMER2_CTL	TMR2_BA+0x00	R/W	Timer2 Control and Status Register	0x0000_0005

Bits	Descriptions	
[31]	ICEDEBUG	ICE Debug Mode Acknowledge Disable Bit (Write Protect)

		0 = ICE debug mode acknowledgement effects TIMER counting. TIMER counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. TIMER counter will keep going no matter CPU is held by ICE or not. **Note**: This bit is write protected. Refer to the SYS_REGLCTL register.
[30]	CNTEN	Timer Counting Enable Bit 0 = Stops/Suspends counting. 1 = Starts counting. **Note1**: In stop status, and then setting CNTEN to 1 will enable the 24-bit up counter to keep counting from the last stop counting value. **Note2**: This bit is auto-cleared by hardware in one-shot mode (TIMERx_CTL[28:27] = 2'b00) when the timer interrupt flag TIF (TIMERx_INTSTS[0]) is generated.
[29]	INTEN	Timer Interrupt Enable Bit 0 = Timer Interrupt Disabled. 1 = Timer Interrupt Enabled. **Note**: If this bit is enabled, when the timer interrupt flag TIF is set to 1, the timer interrupt signal will be generated and inform CPU.
[28:27]	OPMODE	Timer Counting Mode Selection 00 = The Timer controller is operated in one-shot mode. 01 = The Timer controller is operated in periodic mode. 10 = The Timer controller is operated in toggle-output mode. 11 = The Timer controller is operated in continuous counting mode.
[26]	RSTCNT	Timer Counter Reset Setting this bit will reset the 24-bit up counter value CNT (TIMERx_CNT[23:0]) and also force CNTEN (TIMERx_CTL[30]) to 0 if ACTSTS (TIMERx_CTL[25]) is 1. 0 = No effect. 1 = Reset internal 8-bit prescale counter, 24-bit up counter value and CNTEN bit.
[25]	ACTSTS	Timer Active Status (Read Only) This bit indicates the 24-bit up counter status. 0 = 24-bit up counter is not active. 1 = 24-bit up counter is active.
[24]	EXTCNTEN	Event Counter Mode Enable Bit This bit is for external counting pin function enabled. 0 = Event counter mode Disabled. 1 = Event counter mode Enabled. **Note**: When timer is used as an event counter, this bit should be set to 1 and select HCLK as timer clock source
[23]	WKEN	Wake-up Function Enable Bit If this bit is set to 1, while the timer interrupt flag TIF (TIMERx_INTSTS[0]) is 1 and INTEN (TIMERx_CTL[29]) is enabled, the timer interrupt signal will generate a wake-up trigger event to CPU. 0 = Wake-up function Disabled if timer interrupt signal generated. 1 = Wake-up function Enabled if timer interrupt signal generated.
[22:20]	Reserved	Reserved.

[19]	CAPSRC	Capture Pin Source Select Bit 0 = Capture Function source is from TMx_EXT (x= 0~2) pin. 1 = Capture Function source is from internal 32K CLK.
[18:8]	Reserved	Reserved.
[7:0]	PSC	Prescale Counter Timer input clock or event source is divided by (PSC+1) before it is fed to the timer up counter. If this field is 0 (PSC = 0), then there is no scaling.

4.5.7.2 Timer Compare Register (TIMERx_CMP)

Register	Offset	R/W	Description	Reset Value
TIMER0_CMP	TMR0_BA+0x04	R/W	Timer0 Compare Register	0x0000_0000
TIMER1_CMP	TMR1_BA+0x04	R/W	Timer1 Compare Register	0x0000_0000
TIMER2_CMP	TMR2_BA+0x04	R/W	Timer2 Compare Register	0x0000_0000

Bits	Descriptions	
[31:24]	Reserved	Reserved.
[23:0]	CMPDAT	Timer Compared Value CMPDAT is a 24-bit compared value register. When the internal 24-bit up counter value is equal to CMPDAT value, the TIF (TIMERx_INTSTS[0] Timer Interrupt Flag) will set to . Time-out period = (Period of timer clock input) * (8-bit PSC + 1) * (24-bit CMPDAT). **Note1**: Never write 0x0 or 0x1 in CMPDAT field, or the core will run into unknown state. **Note2**: When timer is operating at continuous counting mode, the 24-bit up counter will keep counting continuously even if user writes a new value into CMPDAT field. But if timer is operating at other modes, the 24-bit up counter will restart counting from 0 and using newest CMPDAT value to be the timer compared value while user writes a new value into the CMPDAT field.

4.5.7.3 Timer Interrupt Status Register (TIMERx_INTSTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_INTSTS	TMR0_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER1_INTSTS	TMR1_BA+0x08	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER2_INTSTS	TMR2_BA+0x08	R/W	Timer2 Interrupt Status Register	0x0000_0000

Bits	Descriptions	
[31:3]	Reserved	Reserved.
[2]	TF	Timer Flag This bit indicates the interrupt flag status of Timer while 24-bit timer up counter CNT

		(TIMERx_CNT[23:0]) value reaches to CMPDAT (TIMERx_CMP[23:0]) value. 0 = No effect. 1 = CNT value matches the CMPDAT value. **Note**: This bit is cleared by writing 1 to it.
[1]	TWKF	Timer Wake-up Flag This bit indicates the interrupt wake-up flag status of timer. 0 = Timer does not cause CPU wake-up. 1 = CPU wake-up from Idle or Power-down mode if timer time-out interrupt signal generated. **Note**: This bit is cleared by writing 1 to it.
[0]	TIF	Timer Interrupt Flag This bit indicates the interrupt flag status of Timer while 24-bit timer up counter CNT (TIMERx_CNT[23:0]) value reaches to CMPDAT (TIMERx_CMP[23:0]) value. 0 = No effect. 1 = CNT value matches the CMPDAT value. **Note**: This bit is cleared by writing 1 to it.

4.5.7.4 Timer Data Register (TIMERx_CNT)

Register	Offset	R/W	Description	Reset Value
TIMER0_CNT	TMR0_BA+0x0C	R	Timer0 Data Register	0x0000_0000
TIMER1_CNT	TMR1_BA+0x0C	R	Timer1 Data Register	0x0000_0000
TIMER2_CNT	TMR2_BA+0x0C	R	Timer2 Data Register	0x0000_0000

Bits	Descriptions	
[31:24]	Reserved	Reserved.
[23:0]	CNT	Timer Data Register This field can be reflected the internal 24-bit timer counter value or external event input counter value from TMx (x=0~2) pin. If EXTCNTEN (TIMERx_CTL[24]) is 0, user can read CNT value for getting current 24- bit counter value. If EXTCNTEN (TIMERx_CTL[24]) is 1, user can read CNT value for getting current 24- bit event input counter value.

4.5.7.5 Timer Capture Data Register (TIMERx_CAP)

Register	Offset	R/W	Description	Reset Value
TIMER0_CAP	TMR0_BA+0x10	R	Timer0 Capture Data Register	0x0000_0000
TIMER1_CAP	TMR1_BA+0x10	R	Timer1 Capture Data Register	0x0000_0000
TIMER2_CAP	TMR2_BA+0x10	R	Timer2 Capture Data Register	0x0000_0000

Bits	Descriptions

[31:24]	Reserved	Reserved.
[23:0]	CAPDAT	Timer Capture Data Register When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on TMx_EXT pin matched the CAPEDGE (TIMERx_EXTCTL[2:1]) setting, CAPIF (TIMERx_EINTSTS[0]) will set to 1 and the current timer counter value CNT (TIMERx_CNT[23:0]) will be auto-loaded into this CAPDAT field.

4.5.7.6 Timer External Control Register (TIMERx_EXTCTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_EXTCTL	TMR0_BA+0x14	R/W	Timer0 External Control Register	0x0000_0000
TIMER1_EXTCTL	TMR1_BA+0x14	R/W	Timer1 External Control Register	0x0000_0000
TIMER2_EXTCTL	TMR2_BA+0x14	R/W	Timer2 External Control Register	0x0000_0000

Bits	Descriptions	
[31:9]	Reserved	Reserved.
[8]	CAPSEL	Capture Mode Select Bit 0 = Timer counter reset function or free-counting mode of timer capture function. 1 = Trigger-counting mode of timer capture function.
[7]	CNTDBEN	Timer Counter Pin De-bounce Enable Bit 0 = TMx (x= 0~2) pin de-bounce Disabled. 1 = TMx (x= 0~2) pin de-bounce Enabled. **Note**: If this bit is enabled, the edge detection of TMx pin is detected with de-bounce circuit.
[6]	CAPDBEN	Timer External Capture Pin De-bounce Enable Bit 0 = TMx_EXT (x= 0~2) pin de-bounce Disabled. 1 = TMx_EXT (x= 0~2) pin de-bounce Enabled. **Note1**: If this bit is enabled, the edge detection of TMx_EXT pin is detected with de-bounce circuit.
[5]	CAPIEN	Timer External Capture Interrupt Enable Bit 0 = TMx_EXT (x= 0~2) pin detection Interrupt Disabled. 1 = TMx_EXT (x= 0~2) pin detection Interrupt Enabled. **Note**: CAPIEN is used to enable timer external interrupt. If CAPIEN enabled, timer will generate an interrupt when CAPIF (TIMERx_EINTSTS[0]) is . For example, while CAPIEN = 1, CAPEN = 1, and CAPEDGE = 00, an 1 to 0 transition on the TMx_EXT pin will cause the CAPIF to be set then the interrupt signal is generated and sent to NVIC to inform CPU.
[4]	CAPFUNCS	Capture Function Select Bit 0 = External Capture Mode Enabled. 1 = External Reset Mode Enabled. **Note1**: When CAPFUNCS is 0, transition on TMx_EXT (x= 0~2) pin is used to

		save the 24-bit timer counter value to CAPDAT register. **Note2**: When CAPFUNCS is 1, transition on TMx_EXT (x= 0~2) pin is used to reset the 24-bit timer counter value.
[3]	CAPEN	Timer External Capture Pin Enable Bit This bit enables the TMx_EXT pin. 0 = TMx_EXT (x= 0~2) pin Disabled. 1 = TMx_EXT (x= 0~2) pin Enabled.
[2:1]	CAPEDGE	Timer External Capture Pin Edge Detection 00 = A falling edge on TMx_EXT (x= 0~2) pin will be detected. 01 = A rising edge on TMx_EXT (x= 0~2) pin will be detected. 10 = Either rising or falling edge on TMx_EXT (x= 0~2) pin will be detected. 11 = Reserved.
[0]	CNTPHASE	Timer External Count Phase This bit indicates the detection phase of external counting pin TMx (x= 0~2). 0 = A falling edge of external counting pin will be counted. 1 = A rising edge of external counting pin will be counted.

4.5.7.7 Timer External Interrupt Status Register (TIMERx_EINTSTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_EINTSTS	TMR0_BA+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000
TIMER1_EINTSTS	TMR1_BA+0x18	R/W	Timer1 External Interrupt Status Register	0x0000_0000
TIMER2_EINTSTS	TMR2_BA+0x18	R/W	Timer2 External Interrupt Status Register	0x0000_0000

Bits	Descriptions	
[31:2]	Reserved	Reserved.
[1]	CAPF	Timer External Capture Flag This bit indicates the timer external capture interrupt flag status. 0 = TMx_EXT (x= 0~2) pin interrupt did not occur. 1 = TMx_EXT (x= 0~2) pin interrupt occurred. **Note1**: This bit is cleared by writing 1 to it. **Note2**: When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on TMx_EXT (x= 0~2) pin matched the CAPEDGE (TIMERx_EXTCTL[2:1]) setting, this bit will set to 1 by hardware. Note3: There is a new incoming capture event detected before CPU clearing the CAPIF status. If the above condition occurred, the Timer will keep register TIMERx_CAP unchanged and drop the new capture value.
[0]	CAPIF	Timer External Capture Interrupt Flag This bit indicates the timer external capture interrupt flag status.

		0 = TMx_EXT (x= 0~2) pin interrupt did not occur. 1 = TMx_EXT (x= 0~2) pin interrupt occurred. **Note1**: This bit is cleared by writing 1 to it. **Note2**: When CAPEN (TIMERx_EXTCTL[3]) bit is set, CAPFUNCS (TIMERx_EXTCTL[4]) bit is 0, and a transition on TMx_EXT (x= 0~2) pin matched the CAPEDGE (TIMERx_EXTCTL[2:1]) setting, this bit will set to 1 by hardware. **Note3**: There is a new incoming capture event detected before CPU clearing the CAPIF status. If the above condition occurred, the Timer will keep register TIMERx_CAP unchanged and drop the new capture value.

4.6 Enhanced PWM Generator (PWM)

4.6.1 Overview

The PAN1020 series has built in one PWM unit (PWM0) which is specially designed for motor driving control applications. The PWM0 supports eight PWM generators which can be configured as eight independent PWM outputs, PWM0_CH0~PWM0_CH7, or as four complementary PWM pairs, (PWM0_CH0, PWM0_CH1), (PWM0_CH2, PWM0_CH3), (PWM0_CH4, PWM0_CH5) and (PWM0_CH6, PWM0_CH7) with four programmable dead-time generators, or as four synchronous PWM pairs, with each pin in a pair in-phase – (PWM0_CH0, PWM0_CH1), (PWM0_CH2, PWM0_CH3), (PWM0_CH4, PWM0_CH5)and (PWM0_CH6, PWM0_CH7).

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 eight PWM generators provide sixteen 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.

4.6.2 Features

- Eight independent 16-bit PWM duty control units with maximum eight port pins:
  - Eight independent PWM outputs – PWM0_CH0, PWM0_CH1, PWM0_CH2, PWM0_CH3, PWM0_CH4, PWM0_CH5,PWM0_CH6, and PWM0_CH7
  - Four complementary PWM pairs, with each pin in a pair mutually complement to each other and capable of programmable dead-time insertion – (PWM0_CH0, PWM0_CH1), (PWM0_CH2, PWM0_CH3), (PWM0_CH4, PWM0_CH5) and (PWM0_CH6, PWM0_CH7)
  - Four synchronous PWM pairs, with each pin in a pair in-phase – (PWM0_CH0, PWM0_CH1), (PWM0_CH2, PWM0_CH3), (PWM0_CH4, PWM0_CH5)and (PWM0_CH6, PWM0_CH7)
- Group control bit – PWM0_CH2, PWM0_CH4 and PWM0_CH6 are synchronized with PWM0_CH0, PWM0_CH3, PWM0_CH5 and PWM0_CH7 are synchronized with PWM0_CH1
- Auto-reload mode PWM
- Up to 16-bit resolution
- Supports edge-aligned, center-aligned and precise center-aligned mode
- Supports asymmetric PWM generating in center-aligned and precise center-aligned mode
- Supports center loading in center-aligned and precise center-aligned mode
- Programmable dead-time insertion between complementary paired PWMs
- Each pin of PWM0_CH0 to PWM0_CH7 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 mask aligned function
- Supports independently rising CMP matching, PERIOD matching, falling CMP matching (in Center-aligned type), PERIOD matching to trigger ADC conversion
- Timer comparing matching event trigger PWM to do phase change in BLDC application
- Provides interrupt accumulation function

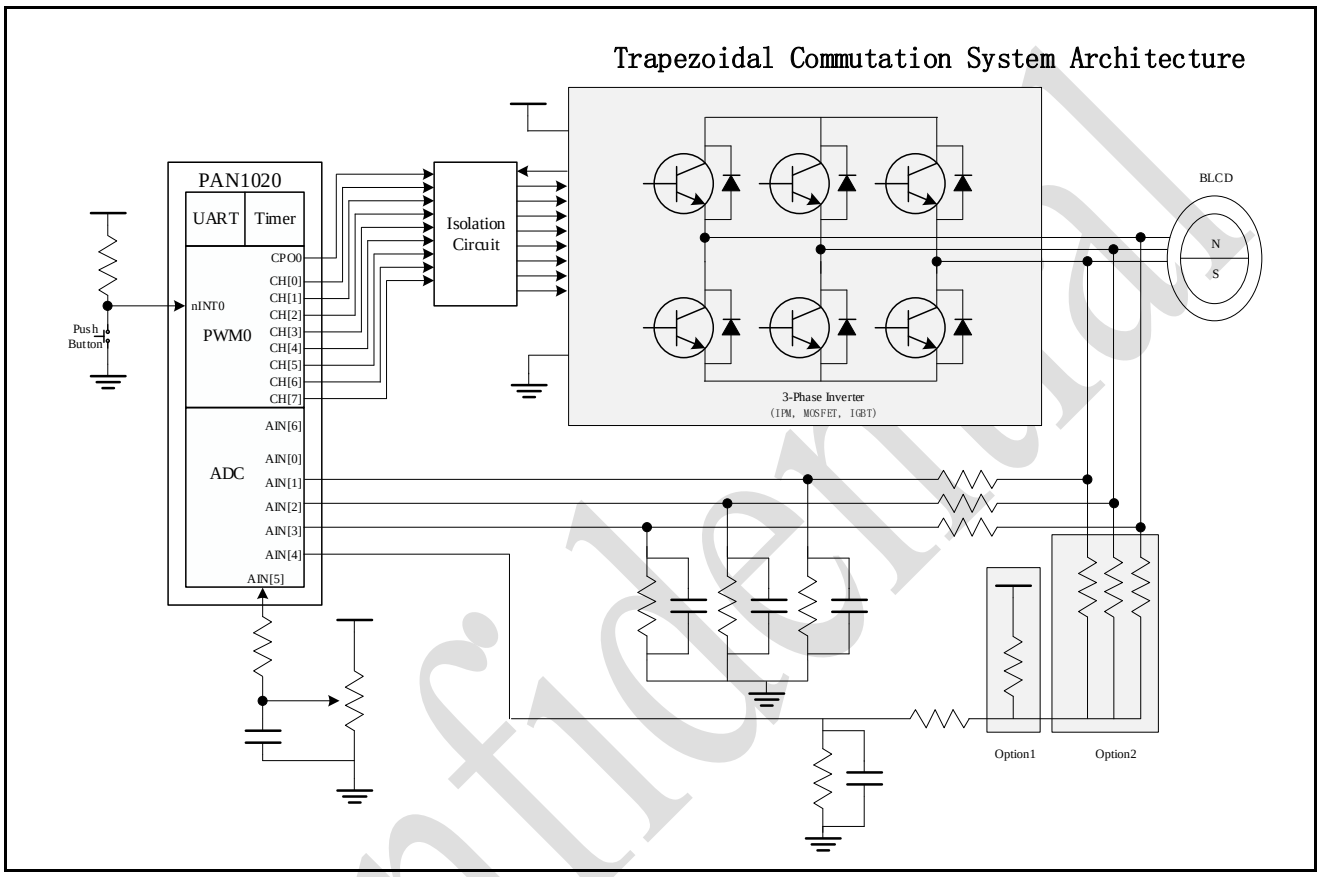


Figure 4-36 Application Circuit Diagram

4.6.3 Block Diagram

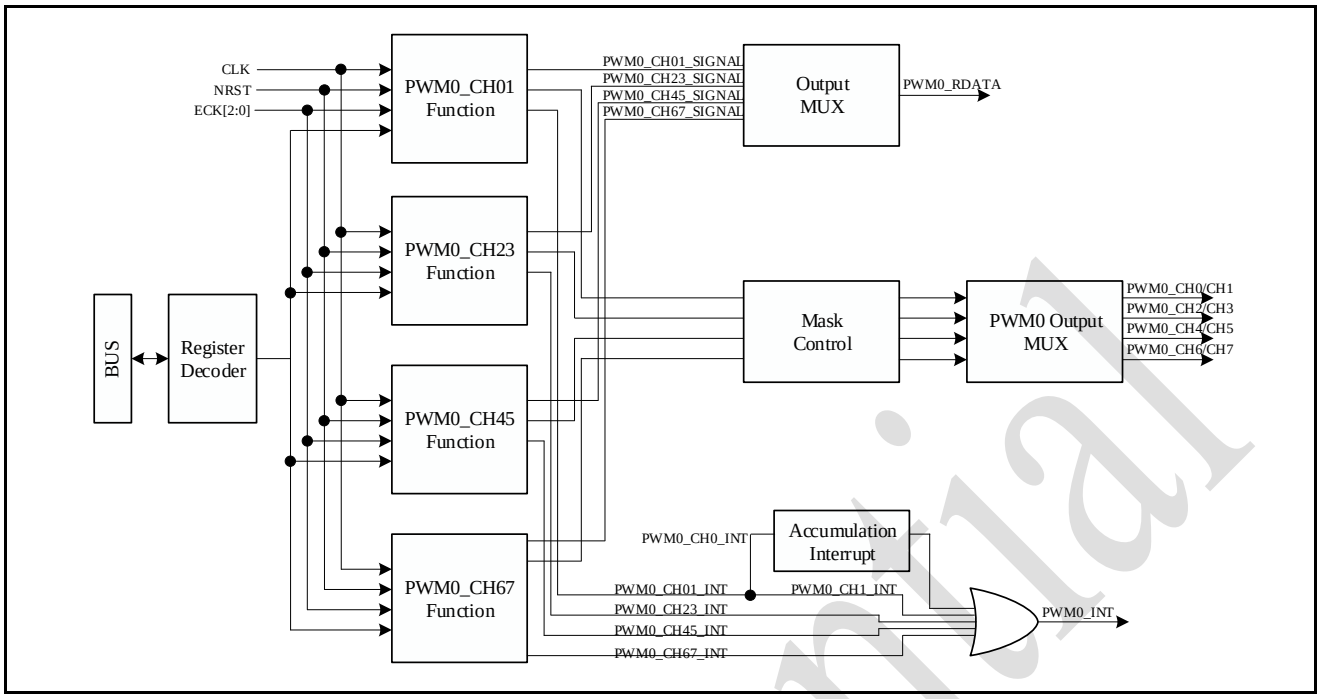


Figure 4-37 PWM Block Diagram

Figure 4-38 shows the architecture of PWM0 in pair (e.g. PWM Counter 0/1 are in one pair and PWM Counter 2/3 are in another one, and so on).

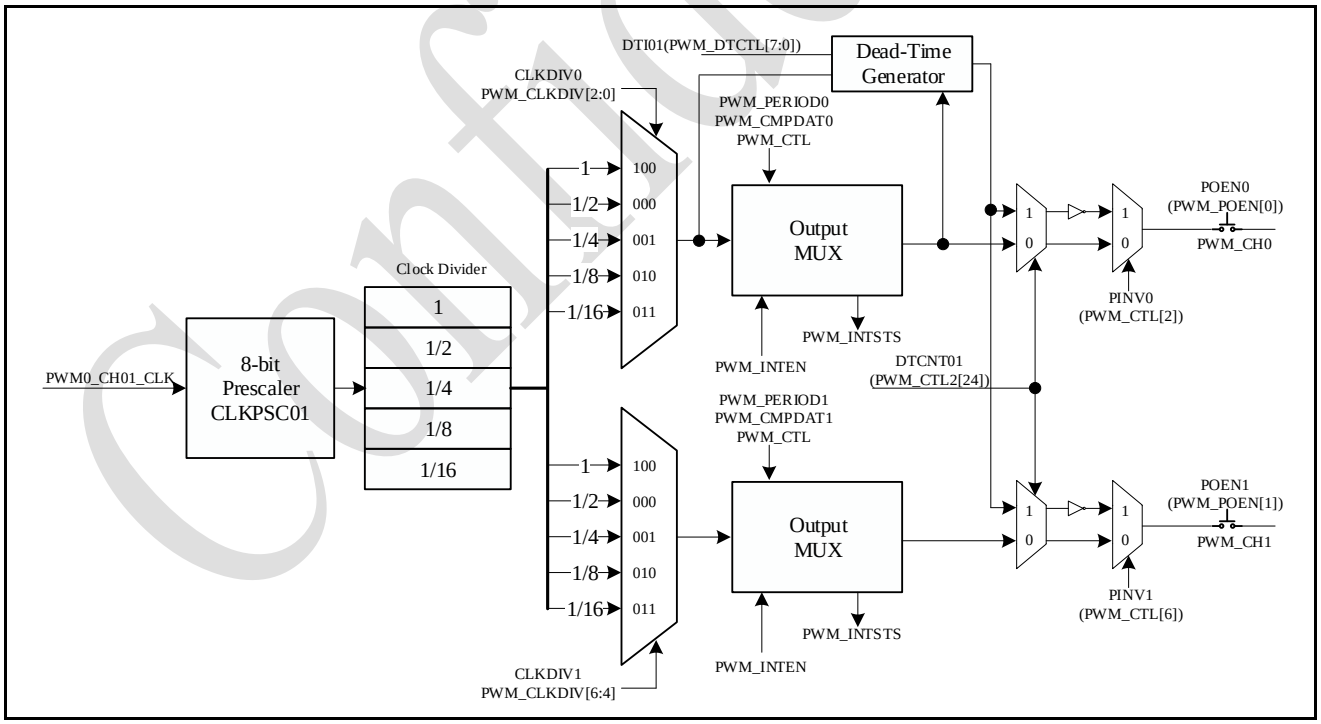


Figure 4-38 PWM Generator 0 Architecture Diagram

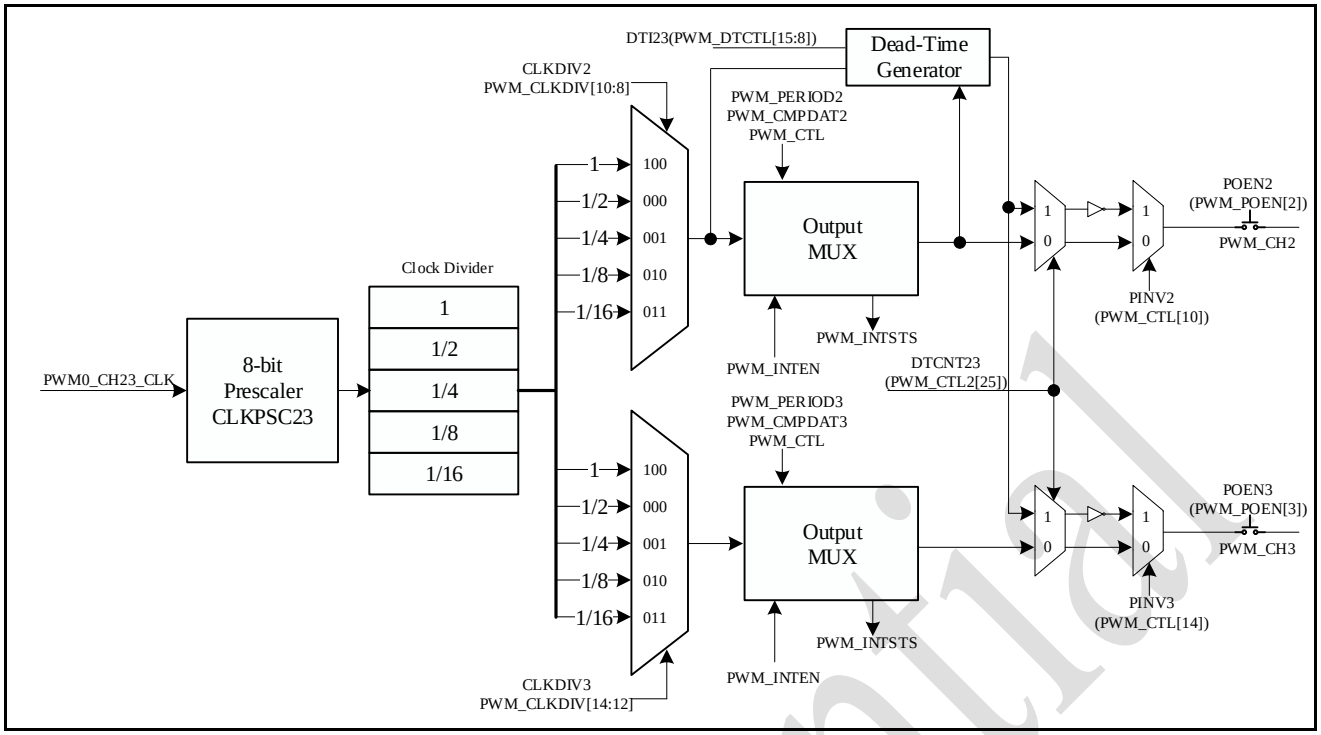


Figure 4-39 PWM Generator 2 Architecture Diagram

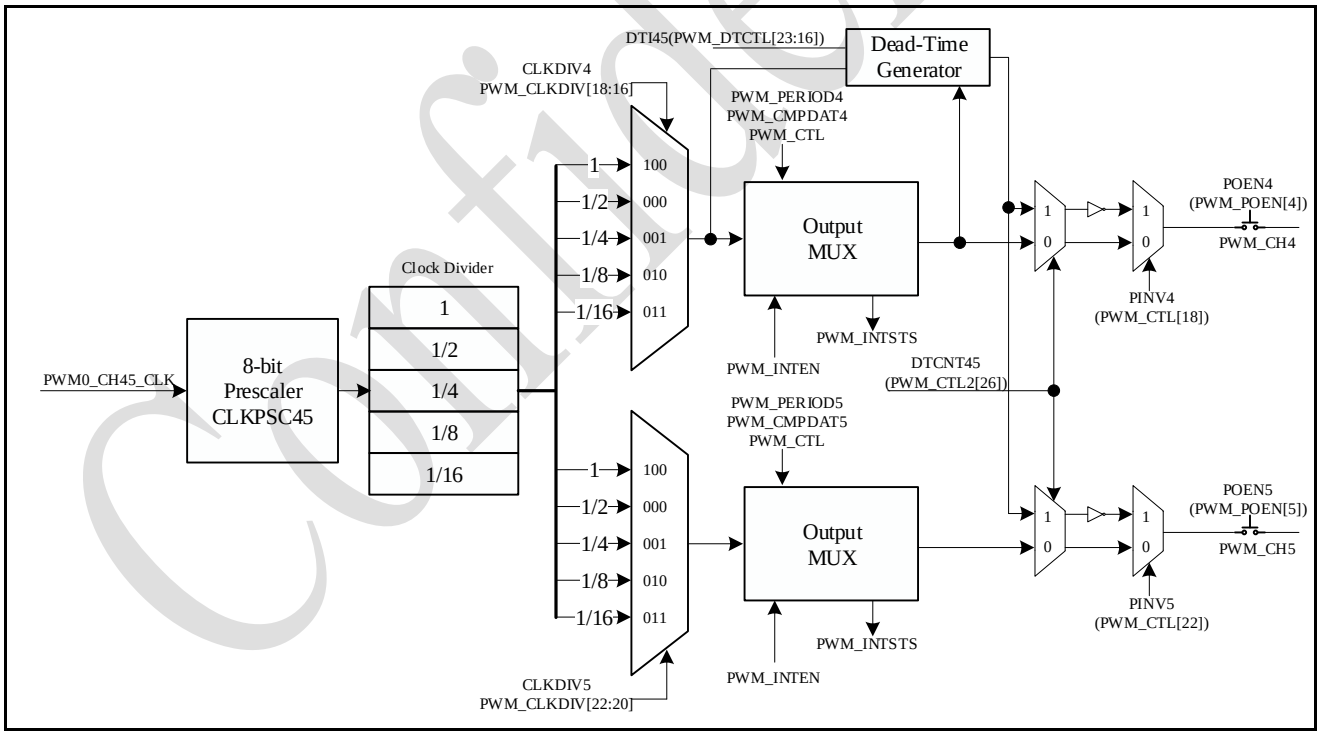


Figure 4-40 PWM Generator 4 Architecture Diagram

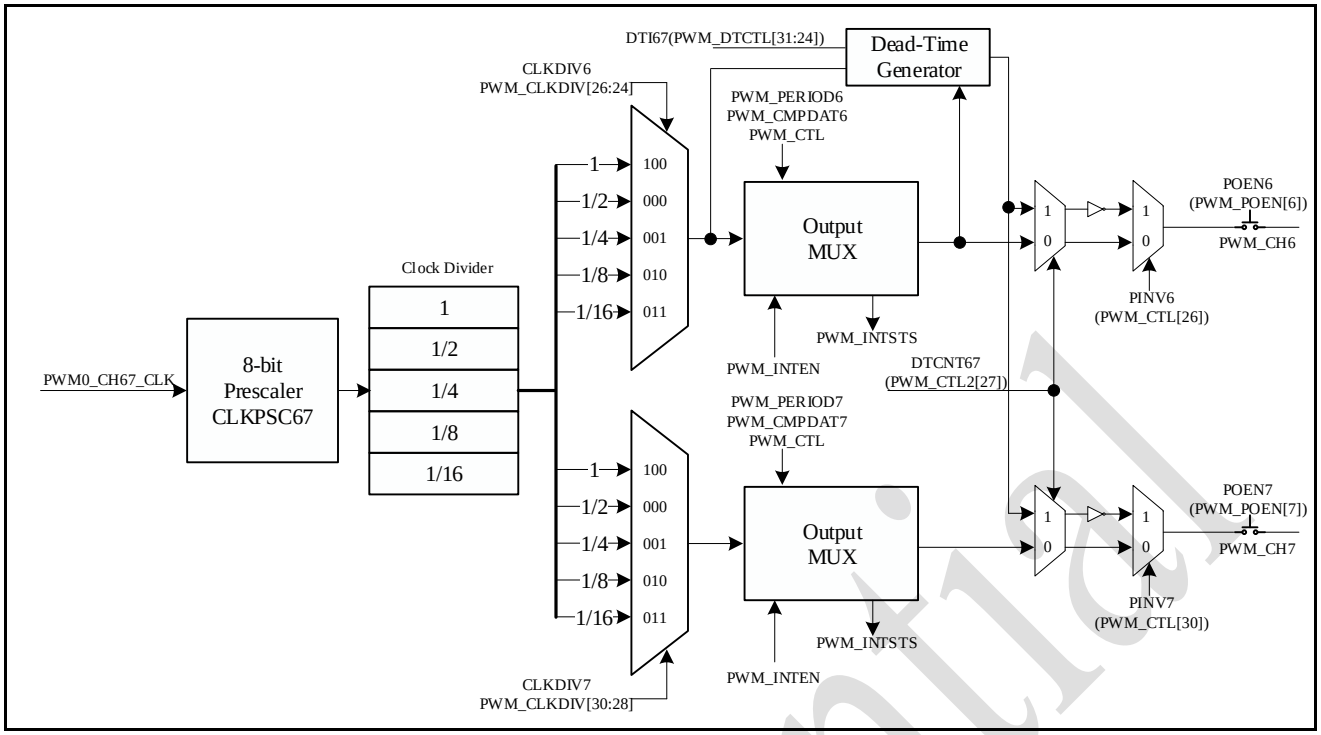


Figure 4-41 PWM Generator 6 Architecture Diagram

4.6.4 Basic Configuration

The PWM0 pin functions are configured in SYS_P0_MFP, SYS_P1_MFP, SYS_P2_MFP and SYS_P5_MFP registers.

The PWM0 clock can be enabled in CLK_APBCLK[23:20]. The PWM0 clock source must be HCLK.

4.6.5 Functional Description

4.6.5.1 PWM Counter Operation

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

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

Edge-aligned (Down counter):

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

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

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

Center-aligned (Up and Down Counter):

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

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

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

Precise Center-aligned (Up and Down Counter):

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

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

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

4.6.5.1.1. Edge-aligned PWM (Down-counter)

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

Figure 4-42, Figure 4-43 and Figure 4-44 show the Edge-aligned PWM timing and operation flow.

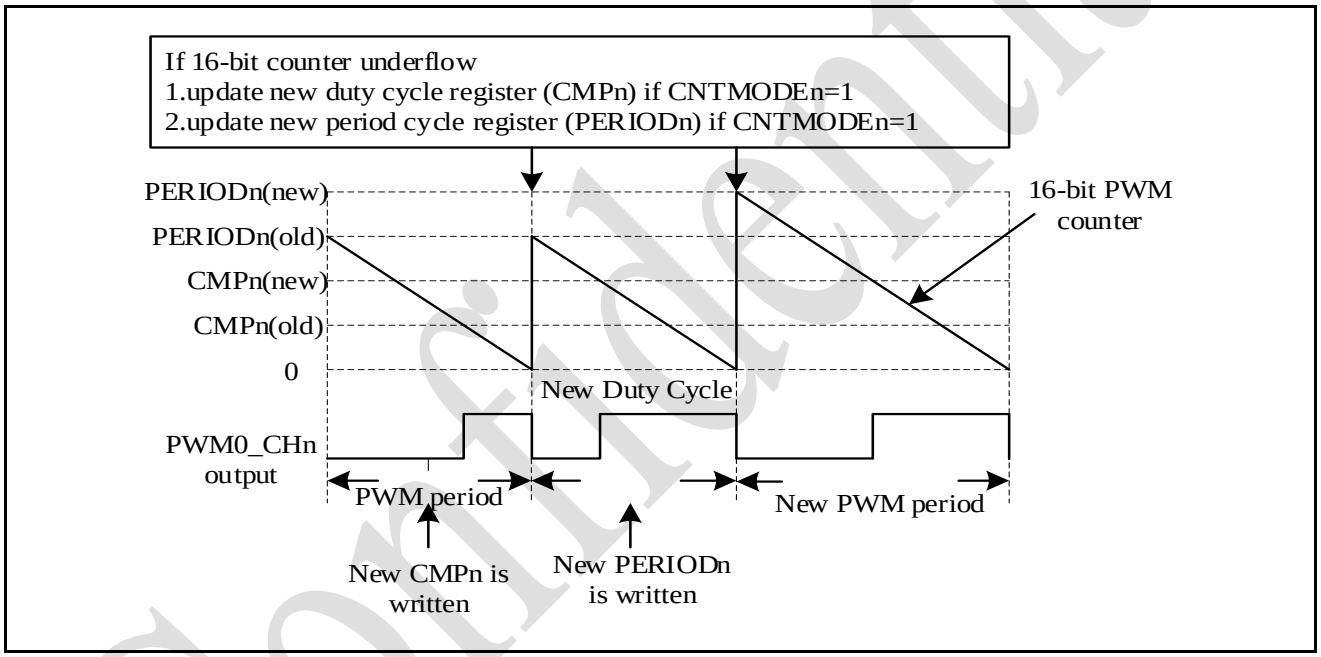


Figure 4-42 Edge-aligned Type PWM

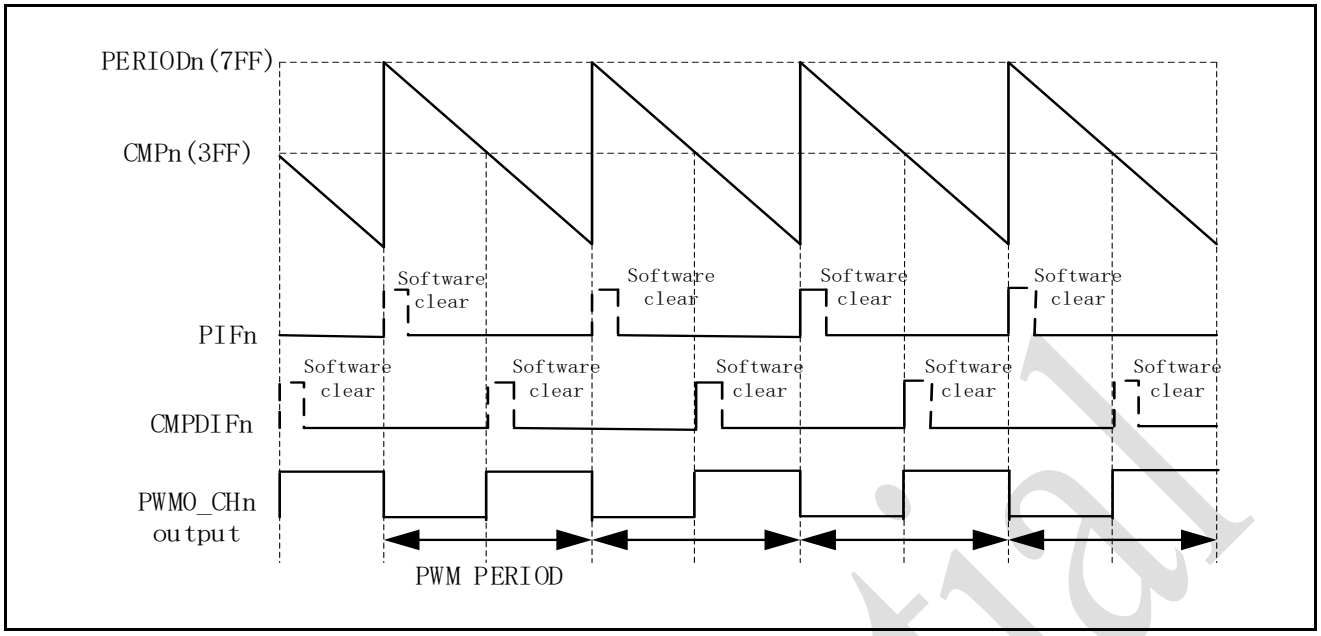


Figure 4-43 PWM Edge-aligned Waveform Timing Diagram

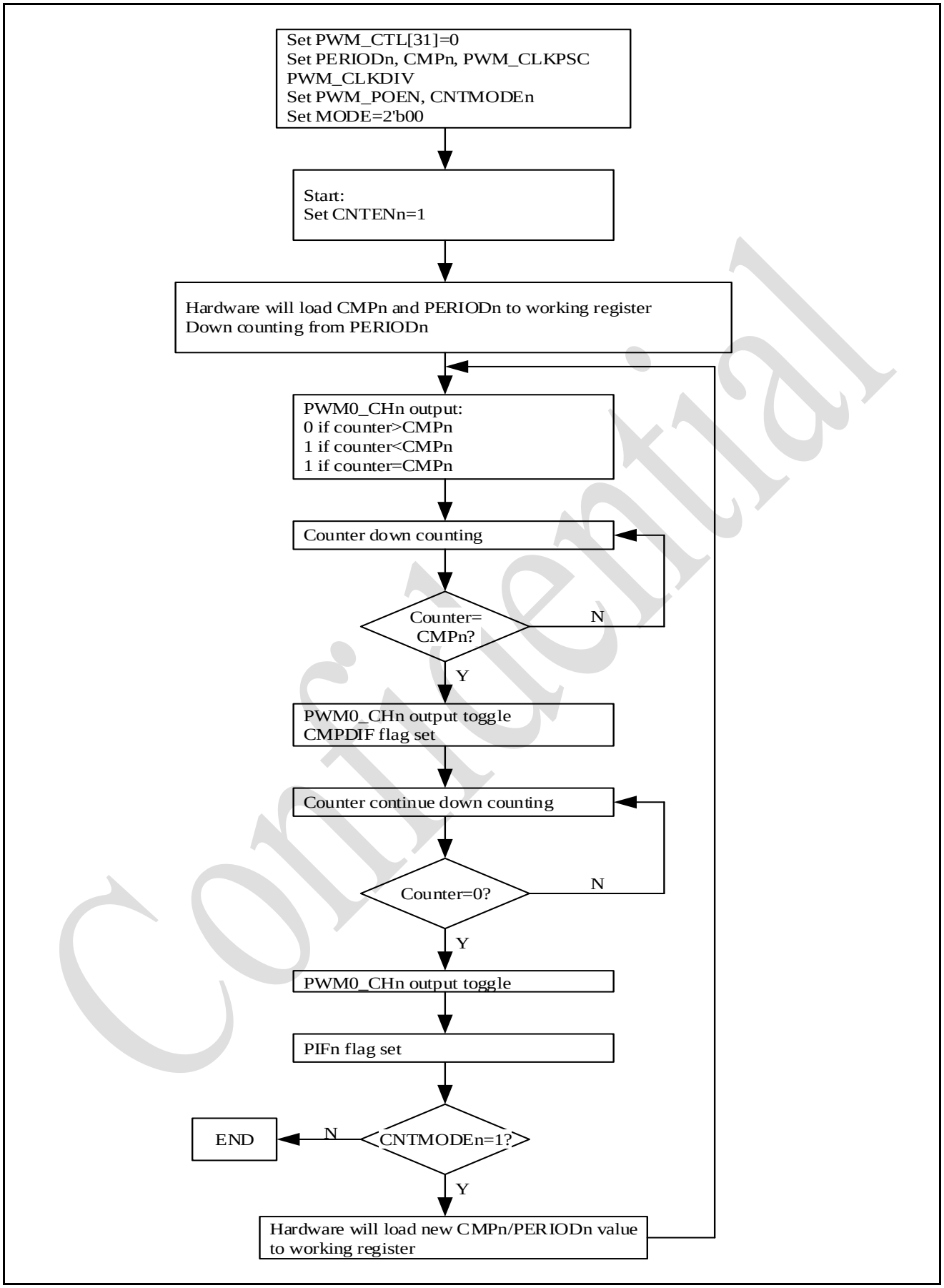


Figure 4-44 Edge-aligned Flow Diagram

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 4-46. The pulse width modulation follows the formula below and the legend of PWM counter Comparator is shown in Figure 4-45. Note that the corresponding GPIO pins must be configured as PWM function (enable PWM_POEN) for the corresponding PWM channel.

*PWM frequency = HCLK / ((CLKPSCnm + 1) * (clock divider)) / (PERIOD + 1)*; where nm, could be 01, 23, 45 or 67 depending on the selected PWM channel

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

PERIOD =0: PWM output is always low

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

CMP ≥ PERIOD: PWM output is always high

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

CMP = 0: PWM is always low

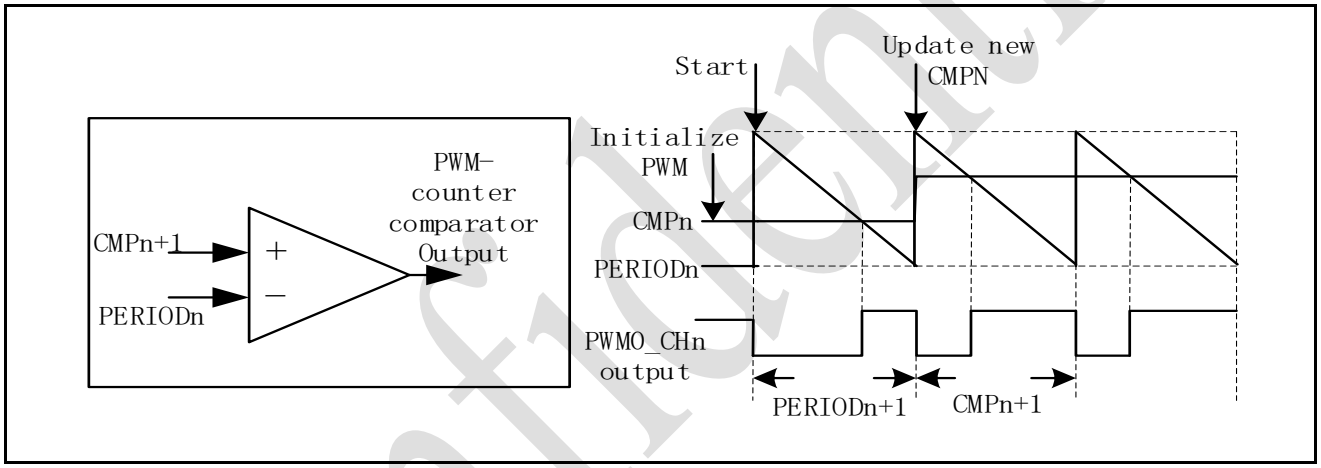


Figure 4-45 Legend of Internal Comparator Output of PWM Counter

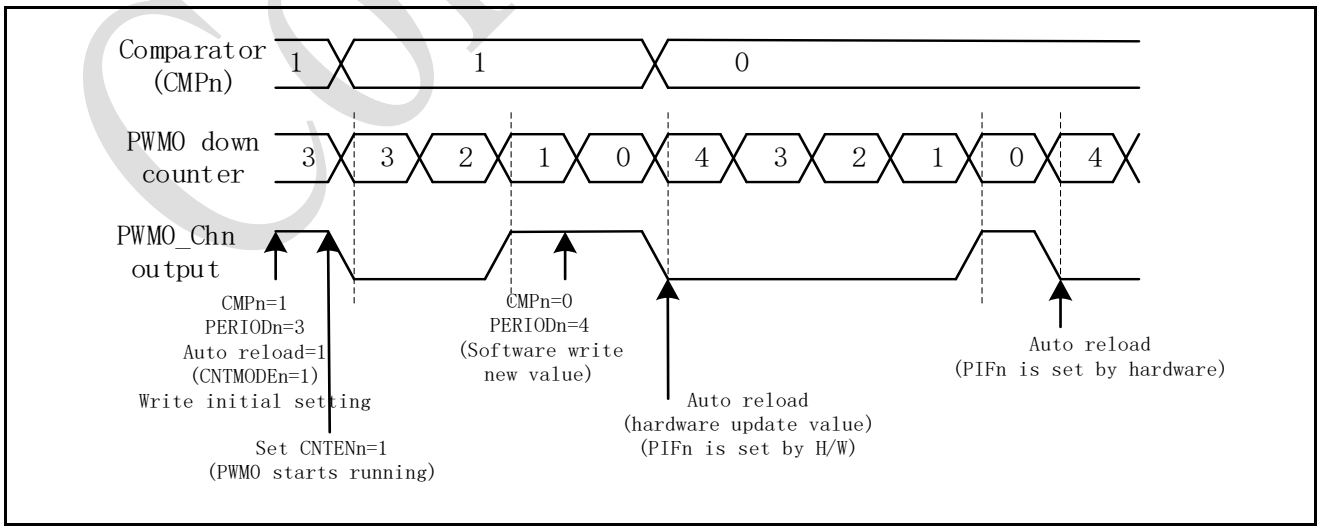


Figure 4-46 PWM Counter Operation Timing

4.6.5.1.2. Center-Aligned PWM (up/down counter)

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

In Center-aligned type, the PWM period interrupt can also be requested at down-counter underflow if PINTTYPE (PWM_CTL2[17]) =0, i.e. at start (end) of each PWM cycle or at up-counter matching with PERIODn if PINTTYPE (PWM_CTL2[17]) =1, i.e. at center point of PWM cycle.

*PWM frequency = HCLK / ((CLKPSCnm + 1) * (clock divider)) / (PERIOD + 1)*; where nm, could be 01, 23, 45 or 67 depending on the selected PWM channel

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

PERIOD =0: PWM output is always low

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

CMP ≥ PERIOD: PWM output is always low

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

CMP = 0: PWM is always high

Note: 1. Unit = one PWM clock cycle.

Figure 4-47, Figure 4-48, Figure 4-49 and Figure 4-50 show the Center-aligned PWM timing and operation flow.

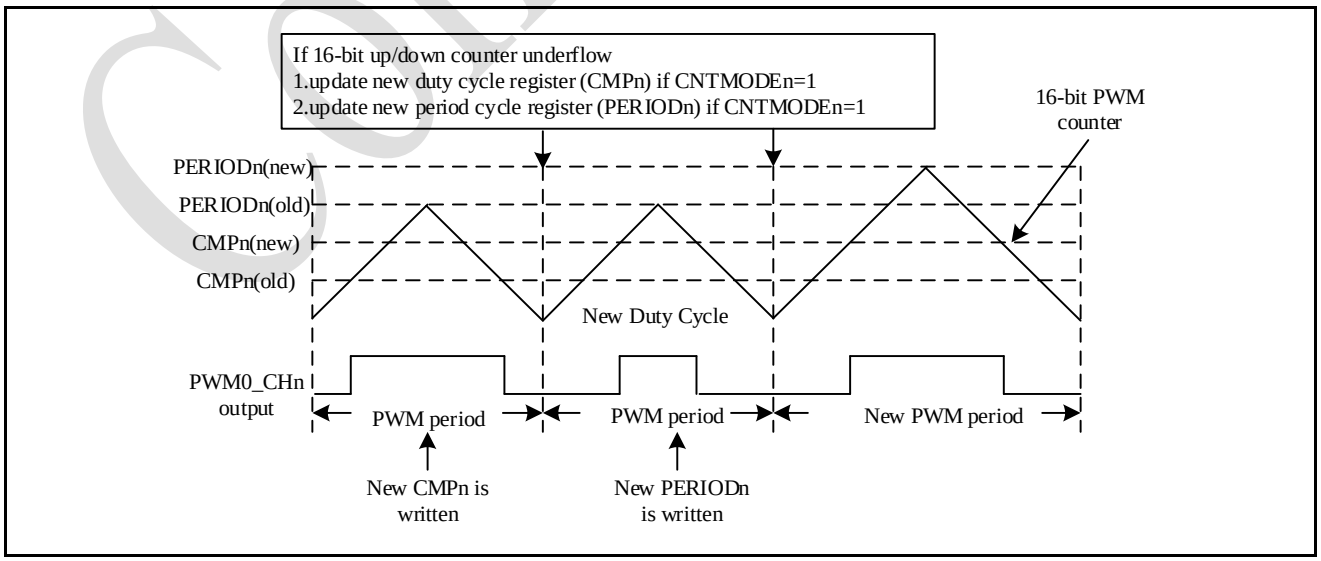


Figure 4-47 Center-aligned Type PWM

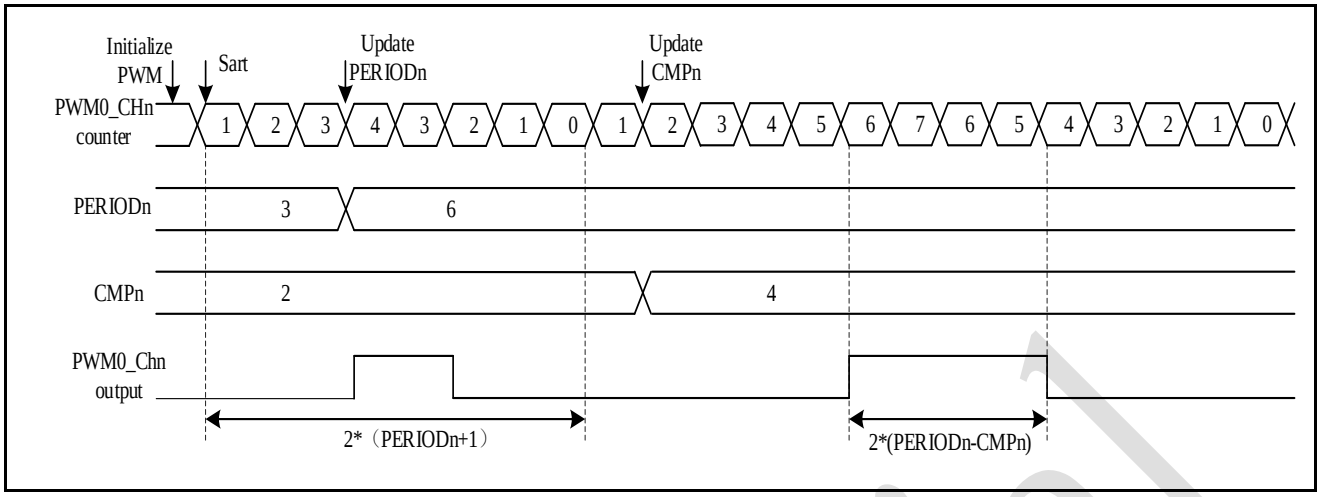


Figure 4-48 Center-aligned Type Operation Timing

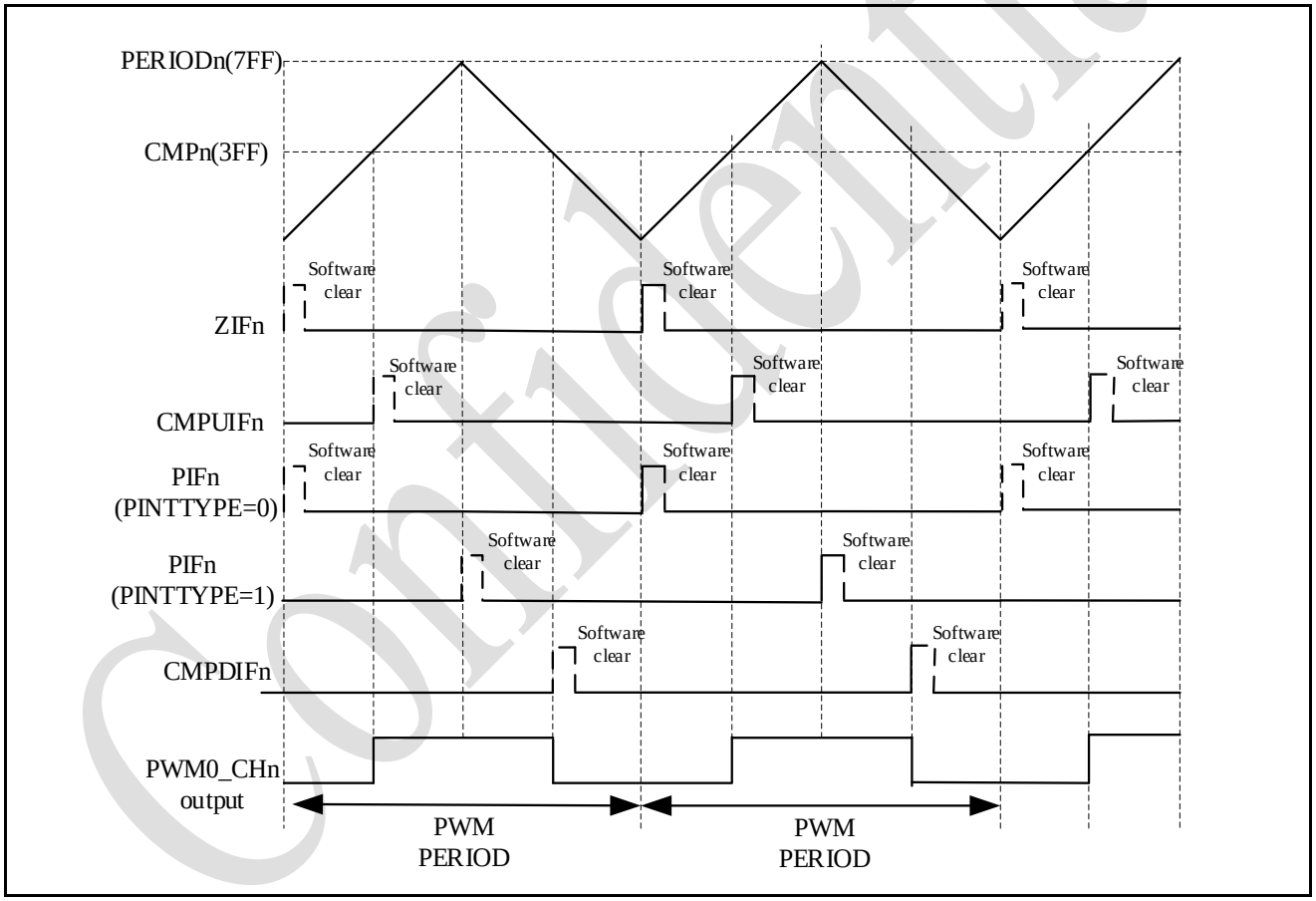


Figure 4-49 PWM Center-aligned Waveform Timing Diagram

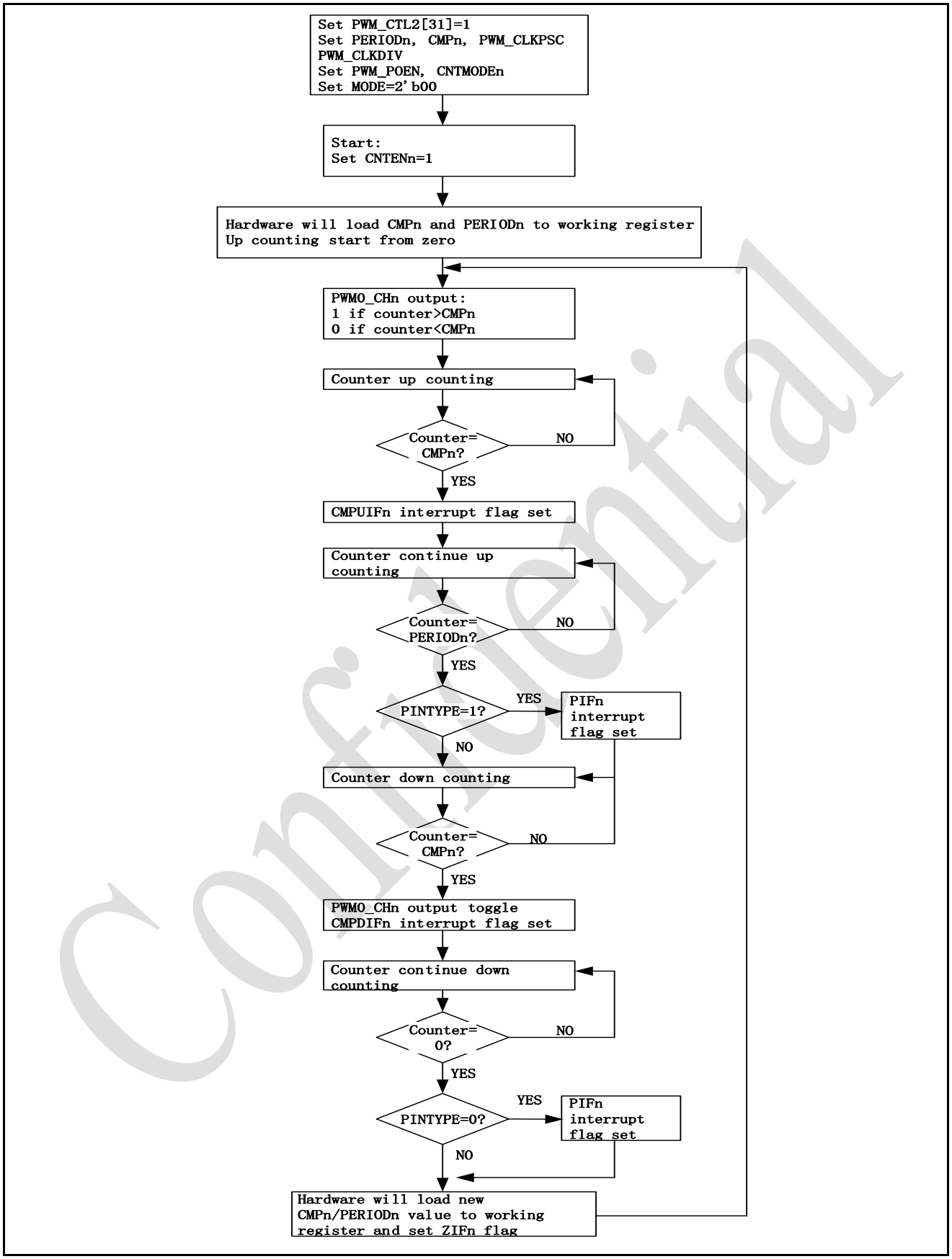


Figure 4-50 Center-aligned Flow Diagram

4.6.5.1.3. Precise Center-Aligned PWM (Up/Down Counter)

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

In Precise Center-aligned type, the PWM period interrupt can also be requested at down-counter underflow if PINTTYPE (PWM_CTL2[17]) =0, i.e. at start (end) of each PWM cycle or at up-counter matching with PERIODn if PINTTYPE (PWM_CTL2[17]) =1, i.e. at center point of PWM cycle.

*PWM frequency = HCLK / ((CLKPSCnm + 1) * (clock divider)) / (PERIOD + 1)*; where nm, could be 01, 23, 45 or 67 depending on the selected PWM channel

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

PERIOD =0: PWM output is always low

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

CMP ≥ PERIOD: PWM output is always low

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

CMP = 0: PWM is always high

Note: 1. Unit = one PWM clock cycle.

Figure 4-51, Figure 4-52, Figure 4-53 show the Precise Center-aligned PWM timing and operation flow.

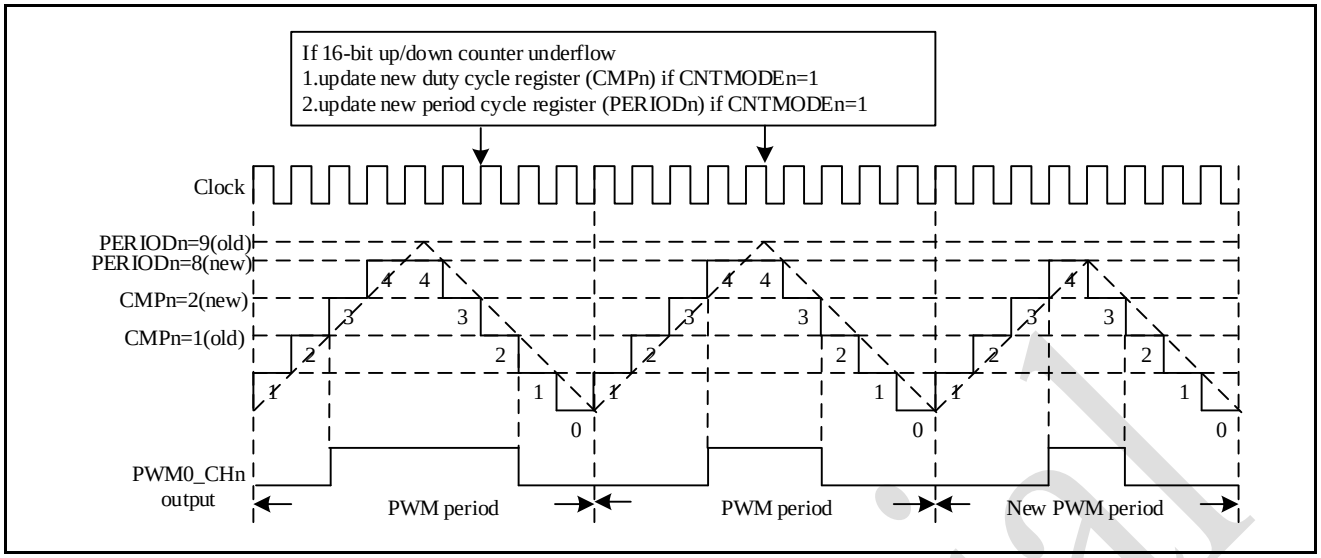


Figure 4-51 Precise Center-aligned Type PWM

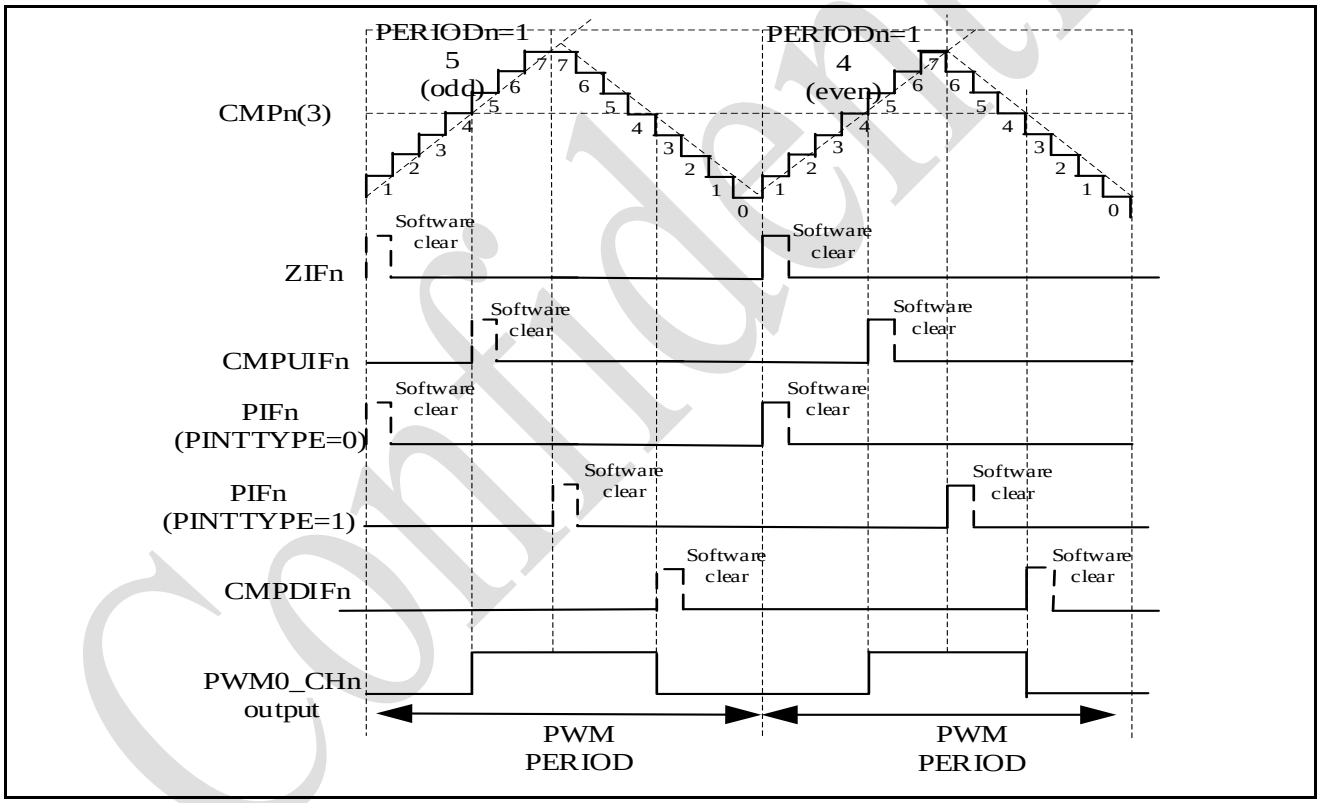


Figure 4-52 PWM Precise Center-aligned Waveform Timing Diagram

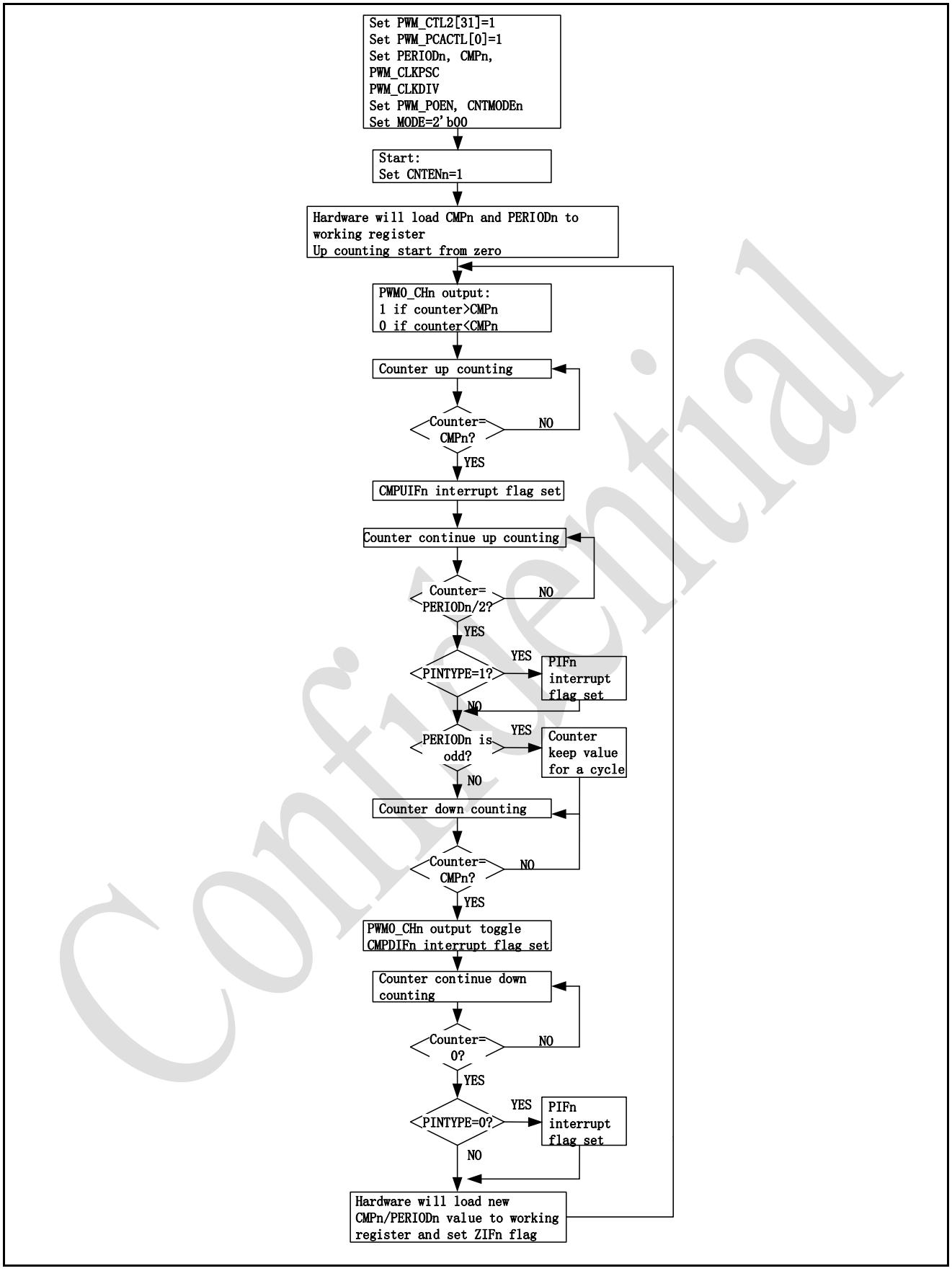


Figure 4-53 Precise Center-aligned Flow Diagram

4.6.5.2 PWM Center Loading Operation

In center-aligned or precise center-aligned type, PWM also supports loading new PERIODn, CMPn. If operating in asymmetric mode, CMPDn will also supports center loading operation.When counter counting to center of PWM period. By setting HCUPDT (PWM_CTL[5]) to enable this function. Figure 4-54 shows an example of center loading operation, when counter counts to original center 4; it updates its value to PERIODn 6 then continues counting down.

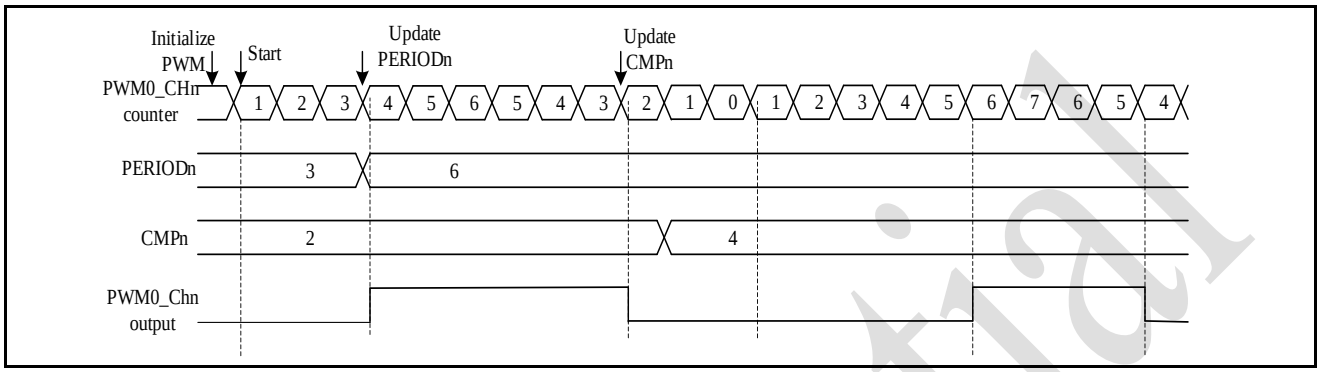


Figure 4-54 PWM Center Loading Timing Diagram

4.6.5.3 PWM Double Buffering and Auto-reload Operation

The PAN1020 series PWM have double buffering function, the reload value is updated at the start of next period without affecting current counter operation. The PWM counter value can be written into PERIODn.

PWM0_CH0 will operate in Auto-reload mode if CNTMODE0 bit is set to 1. It is recommended that switch PWM0_CH0 operating mode before set CNTEN0 bit to 1 to enable PWM0_CH0 counter to start running because the content of PERIOD0 and CMP0 will be cleared to 0 to reset the PWM0_CH0 period and duty setting when PWM0_CH0 operating mode is changed. As PWM0_CH0 operates at auto-reload mode, CMP0 and PERIOD0 should be written first and then set CNTEN0 bit to 1 to enable PWM0_CH0 counter to start running. The PERIOD0 value will be reloaded to PWM0_CH0 counter when the down counting reaches 0. If the PERIOD0 is set to 0, PWM0_CH0 counter will be held. PWM0_CH1~PWM0_CH7 performs the same function as PWM0_CH0

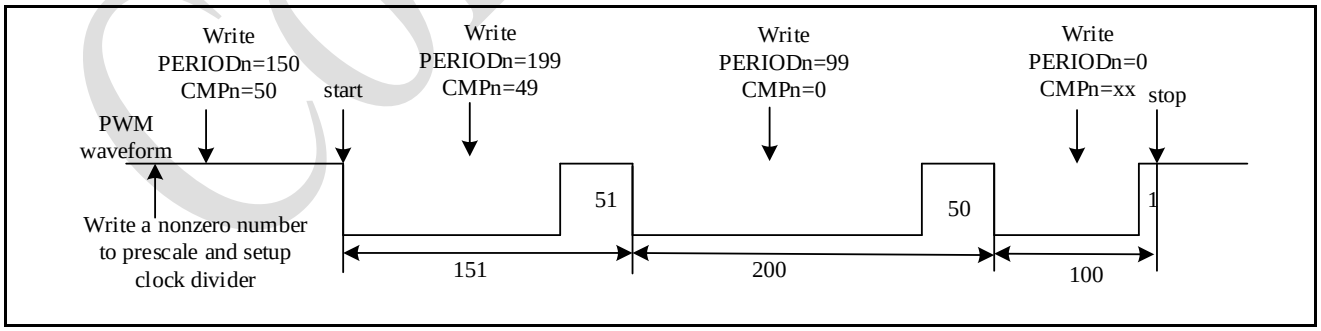


Figure 4-55 PWM Double Buffering Illustration

4.6.5.4 Modulate Duty Ratio

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

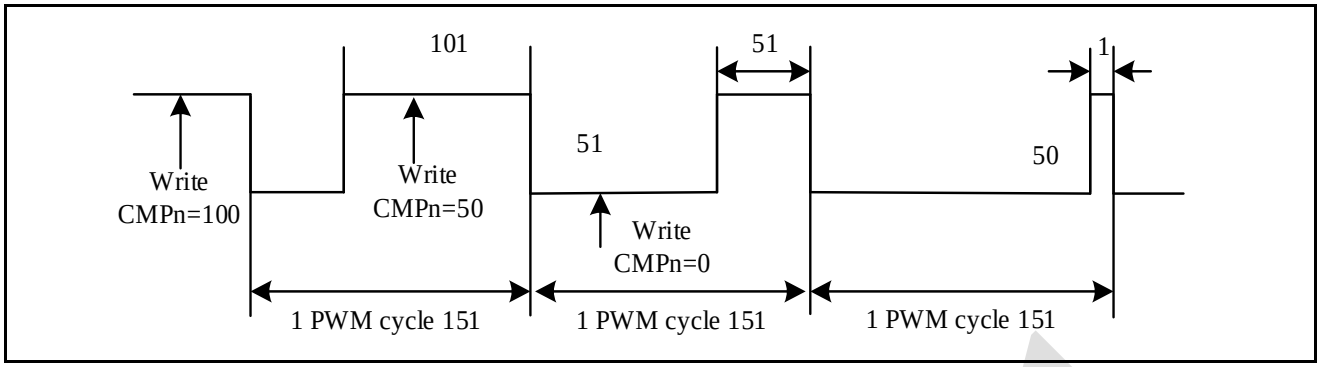


Figure 4-56 PWM Controller Output Duty Ratio

4.6.5.5 PWM Operation Modes

This powerful PWM unit supports independent mode which may be applied to DC or BLDC motor system, Complementary mode with dead-time insertion which may be used in the application of AC induction motor and synchronous motor, and Synchronous mode that makes both pins of each pair are in phase. Besides, the Group mode, which forces the PWM0_CH2, PWM0_CH4 and PWM0_CH6 synchronous with PWM0_CH0 generator, may simplify updating duty control in DC and BLDC motor applications and Asymmetric mode, to generate asymmetric PWM waveform and interrupt timing.

4.6.5.6 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.

4.6.5.7 Complementary Mode

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

In this module there are four duty-cycle generators utilized for complementary mode, with total of four PWM output pair pins in this module. The total eight PWM outputs are grouped into output pairs of even and odd numbered outputs. In complimentary modes, the internal odd PWM signal PWM0_CHn, always be the complement of the corresponding even PWM signal. For example, PWM0_CH1 will be the complement of PWM0_CH0. PWM0_CH3 will be the complement of PWM0_CH2, PWM0_CH5 will be the complement of PWM0_CH4, and PWM0_CH7 will be the complement of PWM0_CH6. The time base for the PWM module is provided by its own 16-bit counter, which also incorporates selectable pre-scalar options.

4.6.5.8 Dead-time Insertion

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

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

*dead-time = PWM_CLK * (DTInm+1)*, where nm, could be 01, 23, 45,67

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

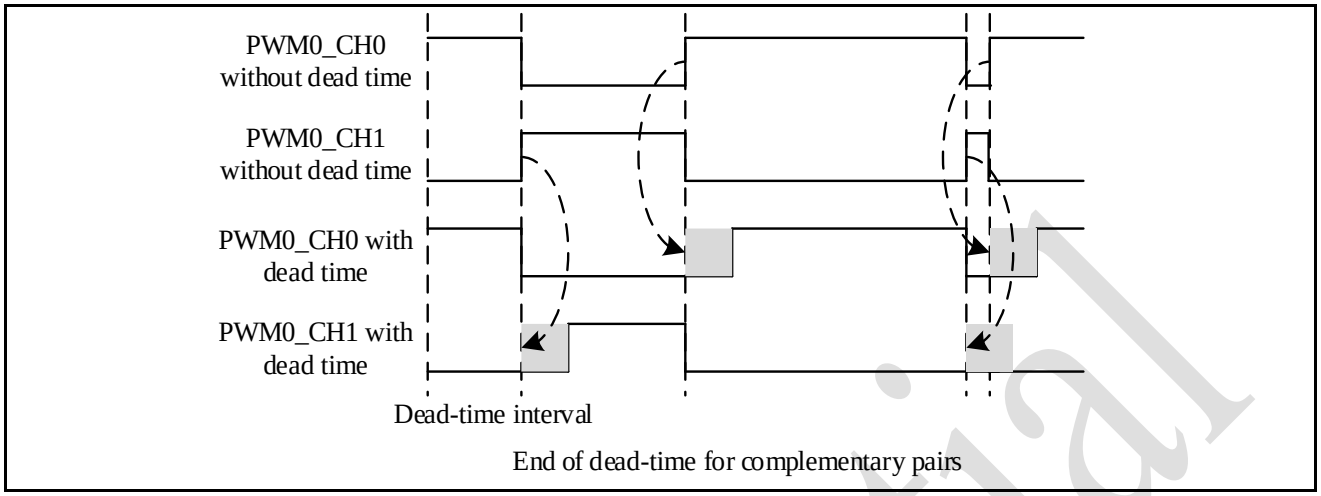


Figure 4-57 Dead-time Insertion

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.

4.6.5.9 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.

PWM0_CH1=PWM0_CH0, PWM0_CH3=PWM0_CH2, PWM0_CH5=PWM0_CH4, and PWM0_CH7=PWM0_CH6.

4.6.5.10 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 (PWM0_CH2, PWM0_CH3), (PWM0_CH4, PWM0_CH5) and (PWM0_CH6, PWM0_CH7) pairs will follow (PWM0_CH0, PWM0_CH1), which imply;

PWM0_CH6 = PWM0_CH4 = PWM0_CH2 = PWM0_CH0;

PWM0_CH7= PWM0_CH5 = PWM0_CH3 = PWM0_CH1 = invert (PWM0_CH0) if Complementary mode is enabled when MODE (PWM_CTL2[29:28]) = 01.

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

4.6.5.11 Asymmetric Mode

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

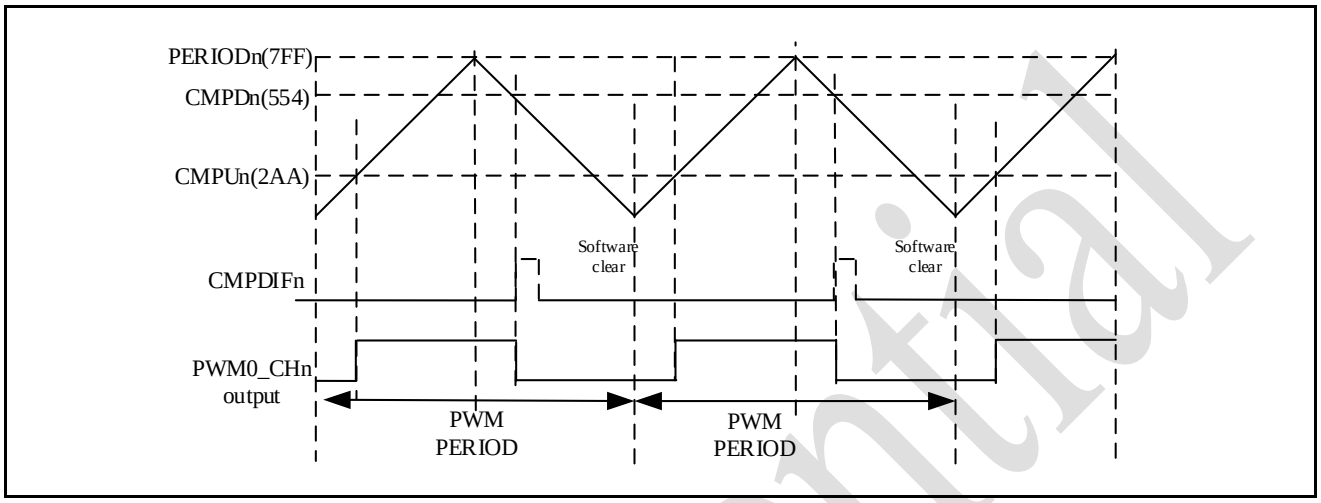


Figure 4-58 Asymmetric Mode Timing Diagram

4.6.5.12 Polarity Control

Each PWM port from PWM0_CH0 to PWM0_CH7 has independent polarity control (PINV0~7) 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 4-59 shows the initial state before PWM starts with different polarity settings.

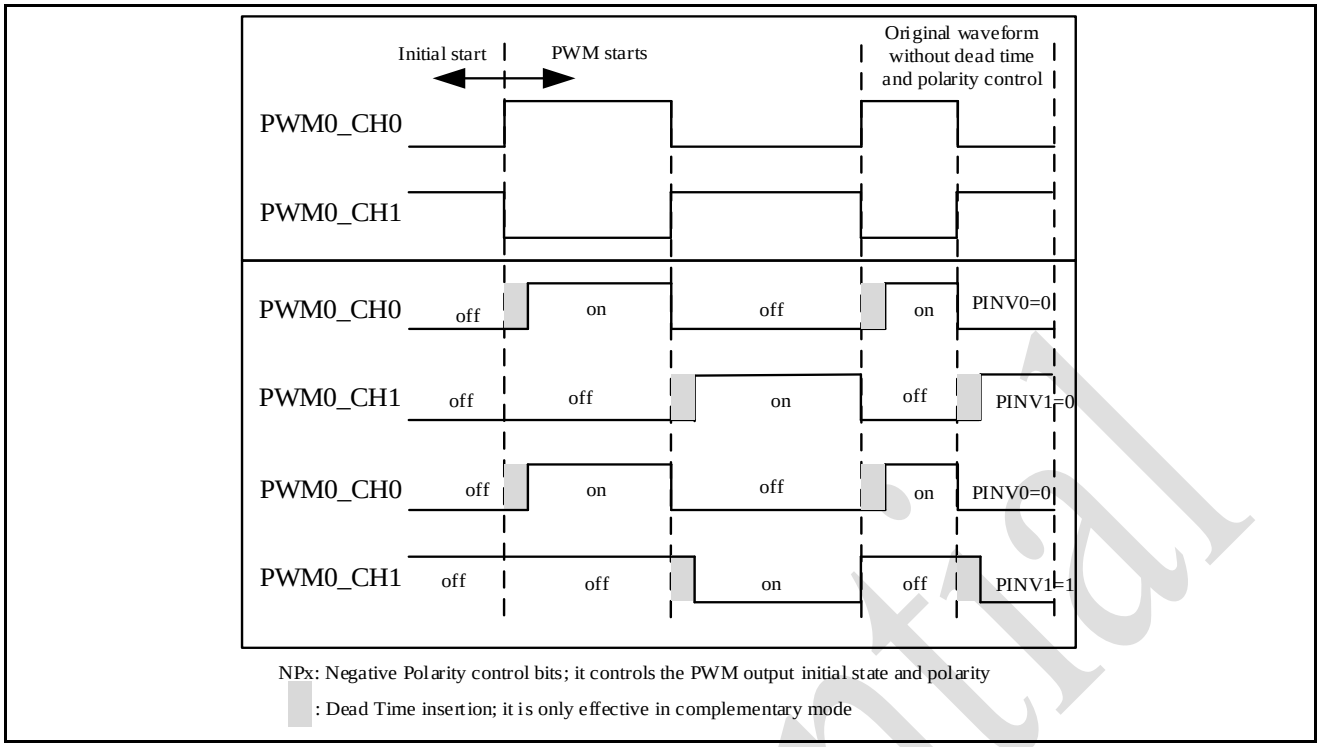


Figure 4-59 Initial State and Polarity Control with Rising Edge Dead-time Insertion

4.6.5.13 PWM for Motor Control Interrupt Architecture

There are four interrupt sources for PWM unit, which are

- ZIFn (PWM_INTSTS[7:0]) PWM counter count to zero interrupt flag;
- CMPUIFn (PWM_INTSTS[31:24]) PWM counter up-counts to CMPn (PWM_CMPDATn[15:0]) interrupt flag;
- PIFn (PWM_INTSTS[23:16]) PWM counter counts to period of edge-aligned type or counts to center of center-aligned type interrupt flag;
- CMPDIFn (PWM_INTSTS[15:8]) PWM counter down-counts to CMPn (PWM_CMPDATn[15:0]) interrupt flag, if operating in asymmetric type it down count to CMPDn (PWM_CMPDATn[31:16]).

The bits ZIENn (PWM_INTEN[7:0]) control the ZIFn interrupt enable; the bits CMPUIENn (PWM_INTEN[31:24]) control the CMPUIFn interrupt enable; the bits PIENn (PWM_INTEN[23:16]) control the PIFn interrupt enable; and the bits CMPDIENn (PWM_INTEN[15:8]) control the CMPDIFn interrupt enable. Note that all the interrupt flags are set by hardware and must be cleared by software.

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

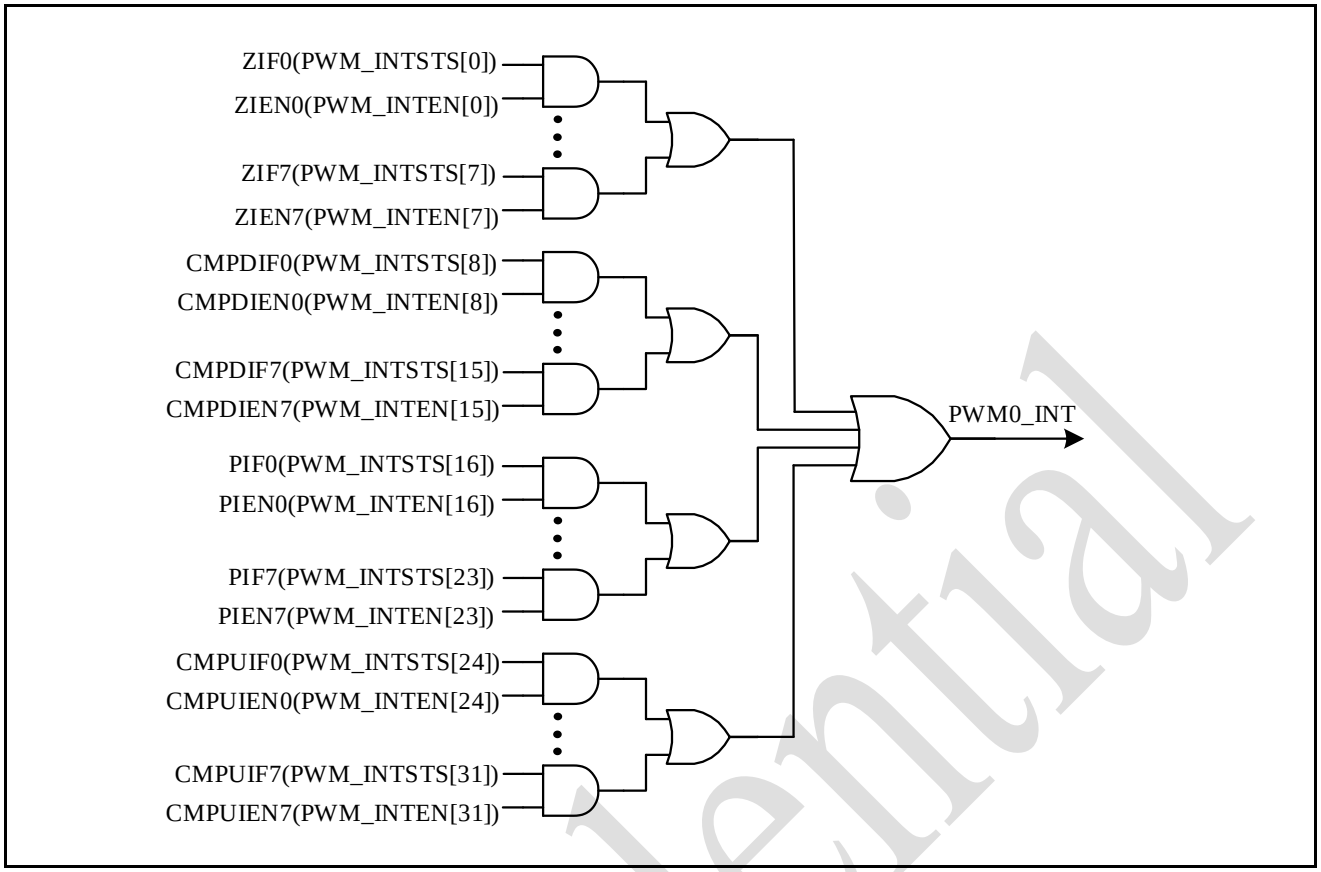


Figure 4-60 Motor Control PWM Interrupt Architecture

4.6.5.14 PWM Counter Start Procedure

The following procedure is recommended for PWM counter start

1. Configure prescaler register (PWM_CLKPSC) for setting clock prescaler (CLKPSCnm).

2. Configure clock select register (PWM_CLKDIV) for setting clock source select (CLKDIVn).

3. Configure PWM control register (PWM_CTL) for setting auto-reload mode (CNTMODEn = 1), PWM counter aligned type (CNTTYPE) and DISABLE PWM counter (CNTENn = 0).

4. Configure PWM control register (PWM_CTL) for setting inverter on/off (PINVn), and Dead-time generator on/off (DTCNTnm). (Optional)

5. Configure PWM_DTCTL register to set dead-time interval. (Optional)

6. Configure comparator register (PWM_CMPDATn) for setting PWM duty (CMPn).

7. Configure PWM counter register (PWM_PERIODn) for setting PWM counter loaded value (PERIODn).

8. Configure PWM interrupt enable register (PWM_INTEN) for setting PWM period interrupt type (INTTYPE), PWM zero interrupt enable bit (ZIENn), PWM compare up match interrupt enable bit (CMPUIENn), PWM period interrupt enable bit (PIENn), PWM compare down match interrupt enable bit (CMPDIENn). (Optional)

9. Configure PWM output enable register (PWM_POEN) to enable PWM output channel

10. Configure PWM control register (PWM_CTL) to enable PWM counter (CNTENn = 1)

4.6.5.15 PWM Counter Stop Procedure

Method 1:

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

Method 2:

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

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

4.6.6 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 = 0x4004_0000**				
PWM_CLKPSC	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_PERIOD6	PWM_BA+0x24	R/W	PWM Counter Period Register 6	0x0000_0000
PWM_PERIOD7	PWM_BA+0x28	R/W	PWM Counter Period Register 7	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_CMPDAT6	PWM_BA+0x44	R/W	PWM Comparator Register 6	0x0000_0000
PWM_CMPDAT7	PWM_BA+0x48	R/W	PWM Comparator Register 7	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_ADCTCTL0	PWM_BA+0x68	R/W	PWM Trigger Control Register 0	0x0000_0000
PWM_ADCTCTL1	PWM_BA+0x6C	R/W	PWM Trigger Control Register 1	0x0000_0000
PWM_ADCTSTS0	PWM_BA+0x70	R/W	PWM Trigger Status Register 0	0x0000_0000
PWM_ADCTSTS1	PWM_BA+0x74	R/W	PWM Trigger Status Register 1	0x0000_0000
PWM_PCACTL	PWM_BA+0x88	R/W	PWM Precise Center-Aligned Type Control Register	0x0000_0000

4.6.7 PWM Control Register Description

4.6.7.1 PWM Pre-Scale Register (PWM_CLKPSC)

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

Bits	Descriptions	
[31:24]	CLKPSC67	Clock Prescaler 6 for PWM Counter 6 and 7 Clock input is divided by (CLKPSC67 + 1) before it is fed to the corresponding PWM counter. If CLKPSC67 = 0, the clock prescaler 6 output clock will be stopped. So the corresponding PWM counter will also be stopped.
[23:16]	CLKPSC45	Clock Prescaler 4 for PWM Counter 4 and 5 Clock input is divided by (CLKPSC45 + 1) before it is fed to the corresponding PWM counter. If CLKPSC45 = 0, the clock prescaler 4 output clock will be stopped. So the corresponding PWM counter will also be stopped.
[15:8]	CLKPSC23	Clock Prescaler 2 for PWM Counter 2 and 3 Clock input is divided by (CLKPSC23 + 1) before it is fed to the corresponding PWM counter. If CLKPSC23 = 0, the clock prescaler 2 output clock will be stopped. So the corresponding PWM counter will also be stopped.
[7:0]	CLKPSC01	Clock Prescaler 0 for PWM Counter 0 and 1 Clock input is divided by (CLKPSC01 + 1) before it is fed to the corresponding PWM counter. If CLKPSC01 = 0, the clock prescaler 0 output clock will be stopped. So the corresponding PWM counter will also be stopped.

4.6.7.2 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]	Reserved	Reserved
[30:28]	CLKDIV7	Counter 7 Clock Divider Selection Select clock input for PWM counter. 000 = Clock input / (CLKPSC67*2). 001 = Clock input / (CLKPSC67*4). 010 = Clock input / (CLKPSC67*8). 011 = Clock input / (CLKPSC67*16). 100 = Clock input / CLKPSC67. Others = Clock input.
[27]	Reserved	Reserved
[26:24]	CLKDIV6	Counter 6 Clock Divider Selection Select clock input for PWM counter. 000 = Clock input / (CLKPSC67*2). 001 = Clock input / (CLKPSC67*4). 010 = Clock input / (CLKPSC67*8). 011 = Clock input / (CLKPSC67*16). 100 = Clock input / CLKPSC67. Others = Clock input.
[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.

4.6.7.3 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]	Reserved	Reserved
[30]	PINV7	PWM0_CH7 Output Inverter Enable Bit 0 = PWM0_CH7 output inverter Disabled. 1 = PWM0_CH7 output inverter Enabled.
[29]	Reserved	Reserved

[28]	CNTEN7	PWM Counter 7 Enable Start Run 0 = Corresponding PWM counter running Stopped. 1 = Corresponding PWM counter start run Enabled.
[27]	Reserved	Reserved
[26]	PINV6	PWM0_CH6 Output Inverter Enable Bit 0 = PWM0_CH6 output inverter Disabled. 1 = PWM0_CH6 output inverter Enabled.
[25]	Reserved	Reserved
[24]	CNTEN6	PWM Counter 6 Enable Start Run 0 = Corresponding PWM counter running Stopped. 1 = Corresponding PWM counter start run Enabled.
[23]	Reserved	Reserved
[22]	PINV5	PWM0_CH5 Output Inverter Enable Bit 0 = PWM0_CH5 output inverter Disabled. 1 = PWM0_CH5 output inverter Enabled.
[21]	ASYMEN	Asymmetric Mode In Center-aligned Type 0 = Symmetric mode in center-aligned type. 1 = Asymmetric mode in center-aligned type.
[20]	CNTEN5	PWM Counter 5 Enable Start Run 0 = Corresponding PWM counter running Stopped. 1 = Corresponding PWM counter start run Enabled.
[19]	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]	HCUPDT	Half Cycle Update Enable for Center-aligned Type 0 = Disable half cycle update PERIOD & CMP. 1 = Enable half cycle update PERIOD & CMP.
[4]	CNTEN1	PWM Counter 1 Enable/Disable Start Run 0 = Corresponding PWM counter running Stopped. 1 = Corresponding PWM counter start run Enabled.
[3]	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.

4.6.7.4 PWM Counter Register 0-7 (PWM_PERIOD0-7)

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
PWM_PERIOD6	PWM_BA+0x24	R/W	PWM Counter Period Register 6	0x0000_0000
PWM_PERIOD7	PWM_BA+0x28	R/W	PWM Counter Period Register 7	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. **Center-aligned type**: PWM frequency = HCLK/((prescale+1)*(clock divider))/ (2*PERIODn+1); where xy, could be 01, 23, 45, 67 depending on the selected PWM channel. *Duty ratio = (PERIODn - CMPn)/(PERIODn+1).* *PERIOD = 0: PWM is always low.* When PERIOD!=0, PWM output is as follow: CMPn ≥ PERIODn: PWM output is always low. CMPn < PERIODn: PWM low width = (CMPn + 1) * 2 unit; PWM high width = (PERIODn - CMPn) * 2 unit. CMPn = 0: PWM is always high. (Unit = One PWM clock cycle). **Note**: Any write to PERIODn will take effect in the next PWM cycle.

4.6.7.5 PWM Comparator Register 0-7 (PWM_CMPDAT0-7)

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
PWM_CMPDAT6	PWM_BA+0x44	R/W	PWM Comparator Register 6	0x0000_0000
PWM_CMPDAT7	PWM_BA+0x48	R/W	PWM Comparator Register 7	0x0000_0000

Bits	Descriptions	
[31:16] n=0,1..7	CMPDn	PWM Comparator Register for Down Counter In Asymmetric Mode CMPn ≥ PERIODn: up counter PWM output is always low. CMPDn ≥ PERIODn: down counter PWM output is always low. Others: PWM output is always high.
[15:0] n=0,1..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. Center-aligned type: PWM frequency = HCLK/((prescale+1)*(clock divider))/ (2*PERIODn+1); where xy, could be 01, 23, 45, 67 depending on the selected PWM channel. *Duty ratio = (PERIODn - CMPn)/(PERIODn+1).* *PERIOD = 0: PWM is always low.* When PERIOD!=0, PWM output is as follow: CMPn ≥ PERIODn: PWM output is always low. CMPn < PERIODn: PWM low width = (CMPn + 1) * 2 unit; PWM high width = (PERIODn - CMPn) * 2 unit. CMPn = 0: PWM is always high. (Unit = One PWM clock cycle). **Note**: Any write to CMPn will take effect in the next PWM cycle.

4.6.7.6 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]	CNTTYPE	PWM Counter-aligned Type Select Bit 0 = Edge-aligned type. 1 = Center-aligned type.
[30]	GROUPEN	Group Function Enable Bit 0 = The signals timing of all PWM channels are independent. 1 = Unify the signals timing of PWM0_CH0, PWM0_CH2, PWM0_CH4 and PWM0_CH6 in the same phase which is controlled by PWM0_CH0 and also unify the signals timing of PWM0_CH1, PWM0_CH3, PWM0_CH5 and PWM0_CH7 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]	DTCNT67	Dead-time 6 Counter Enable Bit (PWM0_CH6 and PWM0_CH7 Pair for PWMC Group) 0 = Dead-time 6 generator Disabled. 1 = Dead-time 6 generator Enabled. **Note**: When the dead-time generator is enabled, the pair of PWM0_CH6 and PWM0_CH7 becomes a complementary pair for PWMC group.

[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.
[23:18]	Reserved	Reserved
[17]	PINTTYPE	PWM Interrupt Type Selection 0 = ZIFn will be set if PWM counter underflows. 1 = ZIFn will be set if PWM counter matches PERIODn register. **Note**: This bit is effective when PWM is in center-aligned type only.
[16:0]	Reserved	Reserved

4.6.7.7 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	
[31:24] n=0,1..7	CMPUFn	PWM Compare Up Flag Flag is set by hardware when PWM0_CHn counter up count reaches CMPn. **Note**: This bit can be cleared by software writing 1.
[23:16] n=0,1..7	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.
[15:8] n=0,1..7	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] n=0,1..7	ZFn	PWM Zero Point Flag Flag is set by hardware when PWMn counter down count reaches zero point. **Note**: This bit can be cleared by software writing 1.

4.6.7.8 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	
[31:24] n=0,1..7	CMPUIENn	PWM Compare Up Interrupt Enable Bit 0 = PWM0_CHn compare up interrupt Disabled. 1 = PWM0_CHn compare up interrupt Enabled.
[23:16] n=0,1..7	PIENn	PWM Period Interrupt Enable Bit 0 = PWM0_CHn period interrupt Disabled. 1 = PWM0_CHn period interrupt Enabled.
[15:8] n=0,1..7	CMPDIENn	PWM Compare Down Interrupt Enable Bit 0 = PWM0_CHn compare down interrupt Disabled. 1 = PWM0_CHn compare down interrupt Enabled.
[7:0] n=0,1..7	ZIENn	PWM Zero Point Interrupt Enable Bit 0 = PWM0_CHn zero point interrupt Disabled. 1 = PWM0_CHn zero point interrupt Enabled.

4.6.7.9 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	
[31:24] n=0,1..7	CMPUIFn	PWM Compare Up Interrupt Flag Flag is set by hardware when PWM0_CHn counter up count reaches CMPn. **Note**: This bit can be cleared by software writing 1.
[23:16] n=0,1..7	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.
[15:8] n=0,1..7	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] n=0,1..7	ZIFn	PWM Zero Point Interrupt Flag Flag is set by hardware when PWMn counter down count reaches zero point. **Note**: This bit can be cleared by software writing 1.

4.6.7.10 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	
[31:8]	Reserved	Reserved
[7:0] n=0,1..7	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.

4.6.7.11 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]	DTI67	Dead-time Interval Register for Pair Of Channel6 and Channel7 (PWM0_CH6 and PWM0_CH7 Pair) These 8 bits determine dead-time length. The unit time of dead-time length is received from corresponding PWM_CLKDIV bits.
[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.

4.6.7.12 PWM Trigger ADC Control Register (PWM_ADCTCTL0)

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

Bits	Descriptions	
[31:28]	Reserved	Reserved
[27]	ZPTRGEN3	Channel 3 Zero Point Trigger ADC Enable Bit Enable PWM trigger ADC function while channel3's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[26]	CDTRGEN3	Channel 3 Compare Down Trigger ADC Enable Bit Enable PWM trigger ADC function while channel3's counter matching CMP3 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[25]	CPTRGEN3	Channel 3 Center Point Trigger ADC Enable Bit Enable PWM Trigger ADC Function While channel3's Counter Matching PERIOD3 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[24]	CUTRGEN3	Channel 3 Compare Up Trigger ADC Enable Bit Enable PWM trigger ADC function while channel3's counter matching CMP3 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[23:20]	Reserved	Reserved
[19]	ZPTRGEN2	Channel 2 Zero Point Trigger ADC Enable Bit Enable PWM trigger ADC function while channel2's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[18]	CDTRGEN2	Channel 2 Compare Down Trigger ADC Enable Bit Enable PWM trigger ADC function while channel2's counter matching CMP2 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[17]	CPTRGEN2	Channel 2 Center Point Trigger ADC Enable Bit

		Enable PWM Trigger ADC Function While channel2's Counter Matching PERIOD2 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[16]	CUTRGEN2	Channel 2 Compare Up Trigger ADC Enable Bit Enable PWM trigger ADC function while channel2's counter matching CMP2 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[15:12]	Reserved	Reserved
[11]	ZPTRGEN1	Channel 1 Zero Point Trigger ADC Enable Bit Enable PWM trigger ADC function while channel1's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[10]	CDTRGEN1	Channel 1 Compare Down Trigger ADC Enable Bit Enable PWM trigger ADC function while channel1's counter matching CMP1 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[9]	CPTRGEN1	Channel 1 Center Point Trigger ADC Enable Bit Enable PWM Trigger ADC Function While channel0's Counter Matching PERIOD1 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[8]	CUTRGEN1	Channel 1 Compare Up Trigger ADC Enable Bit Enable PWM trigger ADC function while channel1's counter matching CMP1 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[7:4]	Reserved	Reserved

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

4.6.7.13 PWM Trigger ADC Control Register (PWM_ADCTCTL1)

Register	Offset	R/W	Description	Reset Value
PWM_ADCTCTL1	PWM_BA+0x6C	R/W	PWM Trigger Control Register 0	0x0000_0000

Bits	Descriptions	
[31:28]	Reserved	Reserved
[27]	ZPTRGEN7	Channel 7 Zero Point Trigger ADC Enable Bit Enable PWM trigger ADC function while channel7's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[26]	CDTRGEN7	Channel 7 Compare Down Trigger ADC Enable Bit Enable PWM trigger ADC function while channel7's counter matching CMP7 in down-count direction

		0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[25]	CPTRGEN7	Channel 7 Center Point Trigger ADC Enable Bit Enable PWM Trigger ADC Function While channel7's Counter Matching PERIOD7 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[24]	CUTRGEN7	Channel 7 Compare Up Trigger ADC Enable Bit Enable PWM trigger ADC function while channel7's counter matching CMP7 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[23:20]	Reserved	Reserved
[19]	ZPTRGEN6	Channel 6 Zero Point Trigger ADC Enable Bit Enable PWM trigger ADC function while channel6's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[18]	CDTRGEN6	Channel 6 Compare Down Trigger ADC Enable Bit Enable PWM trigger ADC function while channel6's counter matching CMP6 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[17]	CPTRGEN6	Channel 6 Center Point Trigger ADC Enable Bit Enable PWM Trigger ADC Function While channel6's Counter Matching PERIOD6 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[16]	CUTRGEN6	Channel 6 Compare Up Trigger ADC Enable Bit Enable PWM trigger ADC function while channel6's counter matching CMP6 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled.

		Note: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[15:12]	Reserved	Reserved
[11]	ZPTRGEN5	Channel 5 Zero Point Trigger ADC Enable Bit Enable PWM trigger ADC function while channel5's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[10]	CDTRGEN5	Channel 5 Compare Down Trigger ADC Enable Bit Enable PWM trigger ADC function while channel5's counter matching CMP5 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[9]	CPTRGEN5	Channel 5 Center Point Trigger ADC Enable Bit Enable PWM Trigger ADC Function While channel5's Counter Matching PE-RIOD5 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[8]	CUTRGEN5	Channel 5 Compare Up Trigger ADC Enable Bit Enable PWM trigger ADC function while channel5's counter matching CMP5 in up-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is only valid for PWM in center-aligned type. When PWM is in edged-aligned type, setting this bit is meaningless and will not take any effect.
[7:4]	Reserved	Reserved
[3]	ZPTRGEN4	Channel 4 Zero Point Trigger ADC Enable Bit Enable PWM trigger ADC function while channel4's counter matching 0 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.
[2]	CDTRGEN4	Channel 4 Compare Down Trigger ADC Enable Bit Enable PWM trigger ADC function while channel4's counter matching CMP4 in down-count direction 0 = PWM condition trigger ADC function Disabled. 1 = PWM condition trigger ADC function Enabled. **Note**: This bit is valid for both center-aligned type and edged-aligned type.

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

4.6.7.14 PWM Trigger Status Register (PWM_ADCTSTS0)

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

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

		ADC. **Note**: This bit can be cleared by software writing 1.
[18]	CDTRGF2	Channel 2 Compare Down Trigger ADC Flag When the channel2's counter is counting down to CMP2, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[17]	CPTRGF2	Channel 2 Center Point Trigger ADC Flag When the channel2's counter is counting to PERIOD2, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[16]	CUTRGF2	Channel 2 Compare Up Trigger ADC Flag When the channel2's counter is counting up to CMP2, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[15:12]	Reserved	Reserved
[11]	ZPTRGF1	Channel 1 Zero Point Trigger ADC Flag When the channel1's counter is counting to zero point, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[10]	CDTRGF1	Channel 1 Compare Down Trigger ADC Flag When the channel1's counter is counting down to CMP1, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[9]	CPTRGF1	Channel 1 Center Point Trigger ADC Flag When the channel1's counter is counting to PERIOD1, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[8]	CUTRGF1	Channel 1 Compare Up Trigger ADC Flag When the channel1's counter is counting up to CMP1, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[7:4]	Reserved	Reserved
[3]	ZPTRGF0	Channel 0 Zero Point Trigger ADC Flag When the channel0's counter is counting to zero point, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[2]	CDTRGF0	Channel 0 Compare Down Trigger ADC Flag When the channel0's counter is counting down to CMP0, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[1]	CPTRGF0	Channel 0 Center Point Trigger ADC Flag When the channel0's counter is counting to PERIOD0, this bit will be set for trigger ADC.

		Note: This bit can be cleared by software writing 1.
[0]	CUTRGF0	Channel 0 Compare Up Trigger ADC Flag When the channel0's counter is counting up to CMP0, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.

4.6.7.15 PWM Trigger Status Register (PWM_ADCTSTS1)

Register	Offset	R/W	Description	Reset Value
PWM_ADCTSTS1	PWM_BA+0x74	R/W	PWM Trigger Status Register 1	0x0000_0000

Bits	Descriptions	
[31:28]	Reserved	Reserved
[27]	ZPTRGF7	Channel 7 Zero Point Trigger ADC Flag When the channel7's counter is counting to zero point, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[26]	CDTRGF7	Channel 7 Compare Down Trigger ADC Flag When the channel7's counter is counting down to CMP7, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[25]	CPTRGF7	Channel 7 Center Point Trigger ADC Flag When the channel7's counter is counting to PERIOD7, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[24]	CUTRGF7	Channel 7 Compare Up Trigger ADC Flag When the channel7's counter is counting up to CMP7, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[23:20]	Reserved	Reserved
[19]	ZPTRGF6	Channel 6 Zero Point Trigger ADC Flag When the channel6's counter is counting to zero point, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[18]	CDTRGF6	Channel 6 Compare Down Trigger ADC Flag When the channel6's counter is counting down to CMP6, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[17]	CPTRGF6	Channel 6 Center Point Trigger ADC Flag When the channel6's counter is counting to PERIOD6, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.

[16]	CUTRGF6	Channel 6 Compare Up Trigger ADC Flag When the channel6's counter is counting up to CMP6, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[15:12]	Reserved	Reserved
[11]	ZPTRGF5	Channel 5 Zero Point Trigger ADC Flag When the channel5's counter is counting to zero point, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[10]	CDTRGF5	Channel 5 Compare Down Trigger ADC Flag When the channel5's counter is counting down to CMP5, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[9]	CPTRGF5	Channel 5 Center Point Trigger ADC Flag When the channel5's counter is counting to PERIOD5, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[8]	CUTRGF5	Channel 5 Compare Up Trigger ADC Flag When the channel5's counter is counting up to CMP5, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[7:4]	Reserved	Reserved
[3]	ZPTRGF4	Channel 4 Zero Point Trigger ADC Flag When the channel4's counter is counting to zero point, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[2]	CDTRGF4	Channel 4 Compare Down Trigger ADC Flag When the channel4's counter is counting down to CMP4, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[1]	CPTRGF4	Channel 4 Center Point Trigger ADC Flag When the channel4's counter is counting to PERIOD4, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.
[0]	CUTRGF4	Channel 4 Compare Up Trigger ADC Flag When the channel4's counter is counting up to CMP4, this bit will be set for trigger ADC. **Note**: This bit can be cleared by software writing 1.

4.6.7.16 Precise PWM Center-Aligned Type Control Register (PWM_PCACTL)

Register	Offset	R/W	Description	Reset Value

PWM_PCACTL	PWM_BA+0x88	R/W	PWM Precise Center-Aligned Type Control Register	0x0000_0000

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

4.7 Watchdog Timer (WDT)

4.7.1 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.

4.7.2 Features

- 18-bit free running up counter for WDT time-out interval
- Selectable time-out interval (2^4 ~ 2^{18}) WDT_CLK cycles and the time-out interval is 0.5 ms ~ 8.192s if WDT_CLK = 32 kHz
- System kept in reset state for a period of (1 / WDT_CLK) * 63
- Supports selectable WDT reset delay period, including 1026, 130, 18 or 3 WDT_CLK reset delay period
- Supports to force WDT enabled after chip powered on or reset by setting CWDTEN[2:0] in Config0 register
- Supports WDT time-out wake-up function only if WDT clock source is selected as LIRC or LXT

4.7.3 Block Diagram

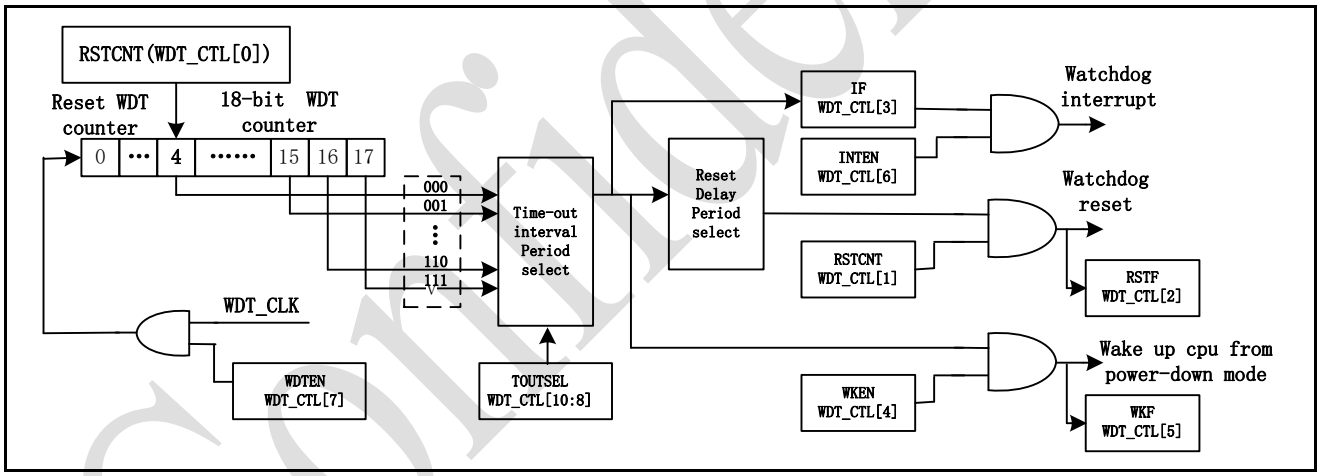


Figure 4-61 Watchdog Timer Block Diagram

4.7.4 Clock Control

The WDT clock control is shown in Figure 4-62.

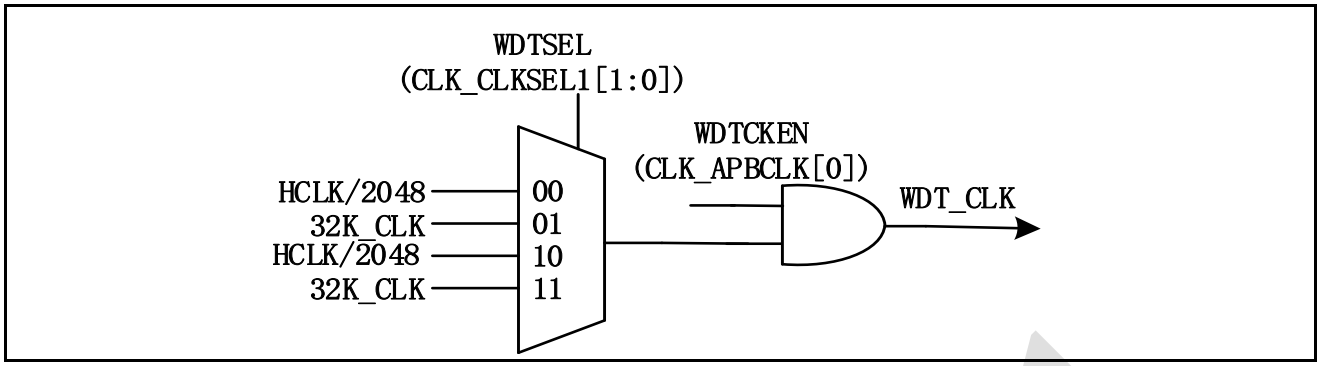


Figure 4-62 Watchdog Timer Clock Control

4.7.5 Basic Configuration

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

4.7.6 Functional Description

The WDT includes an 18-bit free running up counter with programmable time-out intervals.Table 4-13 shows the WDT time-out interval period selection and Figure 4-63 shows the WDT time-out interval and reset period timing.

4.7.6.1 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[10:8]). When the WDT up counter reaches the TOUTSEL (WDT_CTL[10:8]) setting, the WDT time-out interrupt will occur and then WDT time-out flag TOF (WDT_CTL[16]) will be set to 1 immediately.

4.7.6.2 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[10:8]). When the WDT up counter reaches the TOUTSEL (WDT_CTL[10:8]) setting, the WDT time-out interrupt will occur and then WDT time-out interrupt flag IF (WDT_CTL[3]) will be set to 1 immediatelywhen INTEN (WDT_CTL[6]) is set to 1.

4.7.6.3 WDT Reset Delay Period and Reset System

A specified T_{RSTD} reset delay period occurs when the IF (WDT_CTL[3]) is set to 1. User should set RSTCNT (WDT_CTL[0]) to reset the 18-bit WDT up counter value to avoid generating the WDT time-out reset signal before the T_{RSTD} reset delay period expires. Moreover, user should set RSTDSEL (WDT_ALTCTL [1:0]) to select reset delay period to clear WDT counter. If the WDT up counter value has not been cleared after the specified T_{RSTD} delay period expires, the WDT control will set RSTF (WDT_CTL[2]) to 1 if RSTEN (WDT_CTL[1]) bit is enabled, and then chip enters reset state immediately. Refer to Figure 4-63Watchdog Timer Time-out Interval and Reset Period Timing. The T_{RST} reset period will keep the last 63 WDT clocks and then chip restart executing program from reset vector (0x0000_0000). The RSTF (WDT_CTL[2]) will keep 1 after WDT time-out resets the chip. User can check RSTF (WDT_CTL[2]) via software to recognize if the system has been reset by WDT time-out reset or not.

Table 4-13 Watchdog Timer Time-out Interval Period Selection

TOUTSEL	Time-Out Interval Period TTIS	Reset Delay Period TRSTD
000	2^4 * TWDT	(3/18/130/1026) * TWDT
001	2^6 * TWDT	(3/18/130/1026) * TWDT
010	2^8 * TWDT	(3/18/130/1026) * TWDT
011	2^{10} * TWDT	(3/18/130/1026) * TWDT
100	2^{12} * TWDT	(3/18/130/1026) * TWDT
101	2^{14} * TWDT	(3/18/130/1026) * TWDT
110	2^{16} * TWDT	(3/18/130/1026) * TWDT
111	2^{18} * T_{WDT}	(3/18/130/1026) * T_{WDT}

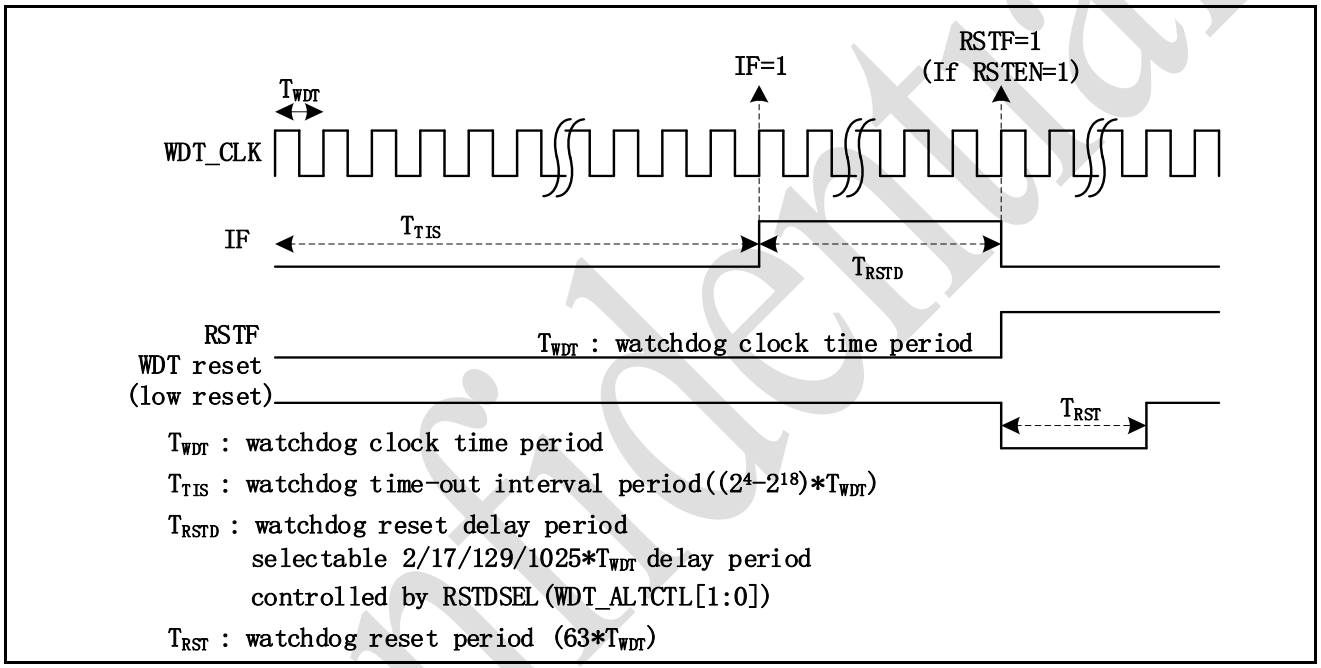


Figure 4-63 Watchdog Timer Time-out Interval and Reset Period Timing

4.7.6.4 WDT Wake-up

If WDT clock source is selected to LIRC or HCLK, 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.

4.7.7 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_0700
WDT_ALTCTL	WDT_BA+0x04	R/W	WDT Alternative Control Register	0x0000_0000

4.7.8 WDT Register Description

4.7.8.1 WDT Control Register (WDT_CTL)

Register	Offset	R/W	Description	Reset Value
WDT_CTL	WDT_BA+0x00	R/W	WDT Control Register	0x0000_0700

Bits	Descriptions	
[31]	ICEDEBUG	ICE Debug Mode Acknowledge Disable Bit (Write Protect) 0 = ICE debug mode acknowledgement affects WDT counting. WDT up counter will be held while CPU is held by ICE. 1 = ICE debug mode acknowledgement Disabled. WDT up counter will keep going no matter CPU is held by ICE or not. **Note**: This bit is write protected. Refer to the SYS_REGLCTL register.
[30:17]	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.
[15:11]	Reserved	Reserved.
[10:8]	TOUTSEL	WDT Time-out Interval Selection (Write Protect) These three bits select the time-out interval period for the WDT. 000 = 2^4 * WDT_CLK. 001 = 2^6 * WDT_CLK. 010 = 2^8 * WDT_CLK. 011 = 2^10 * WDT_CLK. 100 = 2^12 * WDT_CLK. 101 = 2^14 * WDT_CLK. 110 = 2^16 * WDT_CLK. 111 = 2^18 * WDT_CLK. **Note**: This bit is write protected. Refer to the SYS_REGLCTL register.

[7]	WDTEN	WDT Enable Bit (Write Protect) 0 = WDT Disabled (This action will reset the internal up counter value). 1 = WDT Enabled. **Note1**: This bit is write protected. Refer to the SYS_REGLCTL register. **Note2**: If CWDTEN[2:0] (combined by Config0[31] and Config0[4:3]) bits is not configure to 111, this bit is forced as 1 and user cannot change this bit to 0.
[6]	INTEN	WDT Time-out Interrupt Enable Bit (Write Protect) If this bit is enabled, the WDT time-out interrupt signal is generated and inform to CPU. 0 = WDT time-out interrupt Disabled. 1 = WDT time-out interrupt Enabled. **Note**: This bit is write protected. Refer to the SYS_REGLCTL register.
[5]	WKF	WDT Time-out Wake-up Flag (Write Protect) This bit indicates the interrupt wake-up flag status of WDT 0 = WDT does not cause chip wake-up. 1 = Chip wake-up from Idle or Power-down mode if WDT time-out interrupt signal generated. **Note1**: This bit is write protected. Refer to the SYS_REGLCTL register. **Note2**: This bit is cleared by writing 1 to it.
[4]	WKEN	WDT Time-out Wake-up Function Control (Write Protect) If this bit is set to 1, while WDT time-out interrupt flag IF (WDT_CTL[3]) is generated to 1 and interrupt enable bit INTEN (WDT_CTL[6]) is enabled, the WDT time-out interrupt signal will generate a wake-up trigger event to chip. 0 = Wake-up trigger event Disabled if WDT time-out interrupt signal generated. 1 = Wake-up trigger event Enabled if WDT time-out interrupt signal generated. **Note1**: This bit is write protected. Refer to the SYS_REGLCTL register. **Note2**: Chip can be woken-up by WDT time-out interrupt signal generated only if WDT clock source is selected to LIRC or LXT.
[3]	IF	WDT Time-out Interrupt Flag This bit will be set to 1 while WDT up counter value reaches the selected WDT time-out interval 0 = WDT time-out interrupt did not occur. 1 = WDT time-out interrupt occurred. **Note**: This bit is cleared by writing 1 to it.
[2]	RSTF	WDT Time-out Reset Flag This bit indicates the system has been reset by WDT time-out reset or not. 0 = WDT time-out reset did not occur. 1 = WDT time-out reset occurred. **Note**: This bit is cleared by writing 1 to it.
[1]	RSTEN	WDT Time-out Reset Enable Bit (Write Protect) Setting this bit will enable the WDT time-out reset function If the WDT up counter value has not been cleared after the specific WDT reset delay period expires. 0 = WDT time-out reset function Disabled. 1 = WDT time-out reset function Enabled.

		Note: This bit is write-protected. Refer to the SYS_REGLCTL register.
[0]	RSTCNT	Reset WDT Up Counter (Write Protect) 0 = No effect. 1 = Reset the internal 18-bit WDT up counter value. **Note1**: This bit is write protected. Refer to the SYS_REGLCTL register. **Note2**: This bit will be automatically cleared by hardware.

4.7.8.2 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.

4.8 Window Watchdog Timer (WWDT)

4.8.1 Overview

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

4.8.2 Feature

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

4.8.3 Block Diagram

The WWDT block diagram is shown in Figure 4-64.

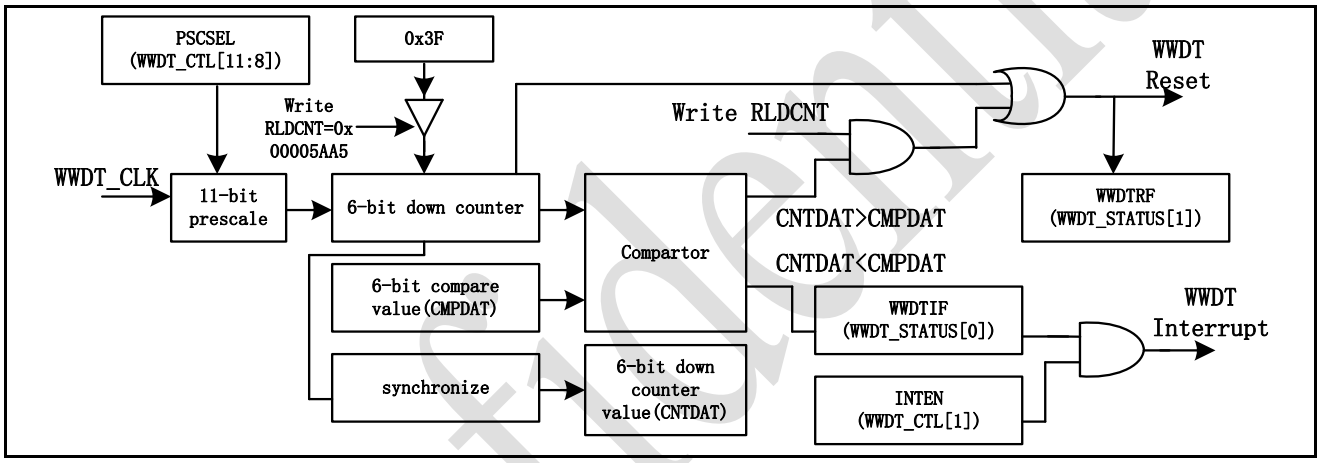


Figure 4-64WWDT Block Diagram

4.8.4 Clock Control

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

The WWDT clock control is shown in Figure 4-65.

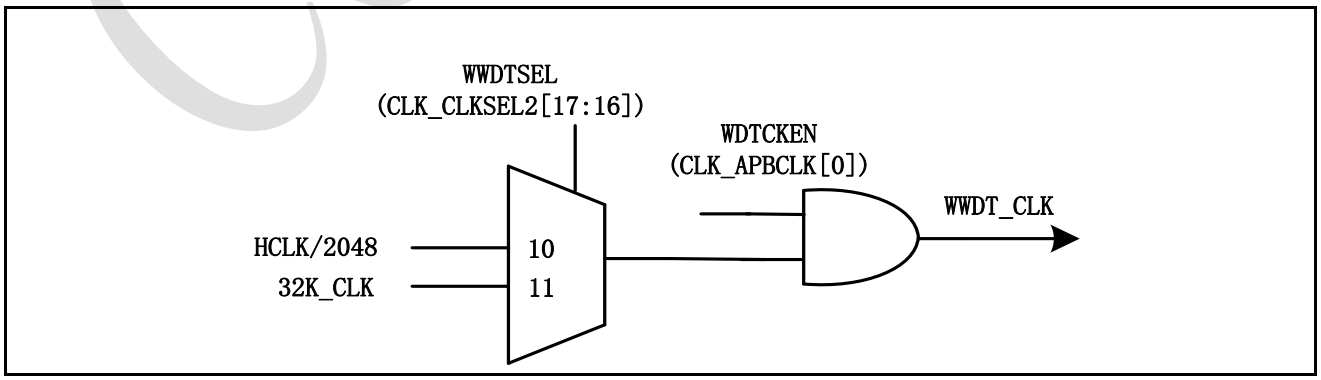


Figure 4-65 WWDT Clock Control

4.8.5 Functional Description

The WWDT includes a 6-bit down counter with programmable prescale value to define different WWDT time-out intervals. The clock source of 6-bit WWDT is based on system clock divide 2048 (HCLK/2048) or 32 kHz internal low speed RC oscillator (LIRC) with a programmable 11-bit pre-scale counter value which controlled by PSCSEL (WWDT_CTL[11:8]). Also, the correlate of PSCSEL (WWDT_CTL[11:8]) and prescale value are listed in Table 4-14.

Table 4-14 WWDT Prescaler Value Selection

PSCSEL	Prescaler Value	Max. Time-Out Period	Max. Time-Out Interval (WWDT_CLK=32 KHz)
0000	1	1 * 64 * T_{WWDT}	2ms
0001	2	2 * 64 * T_{WWDT}	4ms
0010	4	4 * 64 * T_{WWDT}	8ms
0011	8	8 * 64 * T_{WWDT}	16ms
0100	16	16 * 64 * T_{WWDT}	32ms
0101	32	32 * 64 * T_{WWDT}	64ms
0110	64	64 * 64 * T_{WWDT}	128ms
0111	128	128 * 64 * T_{WWDT}	256ms
1000	192	192 * 64 * T_{WWDT}	384ms
1001	256	256 * 64 * T_{WWDT}	512ms
1010	384	384 * 64 * T_{WWDT}	768ms
1011	512	512 * 64 * T_{WWDT}	1.024s
1100	768	768 * 64 * T_{WWDT}	1.536s
1101	1024	1024 * 64 * T_{WWDT}	2.048s
1110	1536	1536 * 64 * T_{WWDT}	3.072s
1111	2048	2048 * 64 * T_{WWDT}	4.096s

4.8.5.1 WWDT Counting

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

4.8.5.2 WWDT Compare Match Flag

During down counting by the WWDT counter, the WWDTF (WWDT_STATUS[2]) is set to 1 while the WWDT counter value (CNTDAT) is equal to window compare value (CMPDAT) and WWDTF can be cleared by user.

4.8.5.3 WWDT Compare Match Interrupt Flag

During down counting by the WWDT counter, the WWDTIF (WWDT_STATUS[0]) is set to 1 while the WWDT counter value (CNTDAT) is equal to window compare value (CMPDAT) and

INTEN(WWDT_CTL[1]) is set to 1. WWDTIF can be cleared by user;

4.8.5.4 WWDT Reset System

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

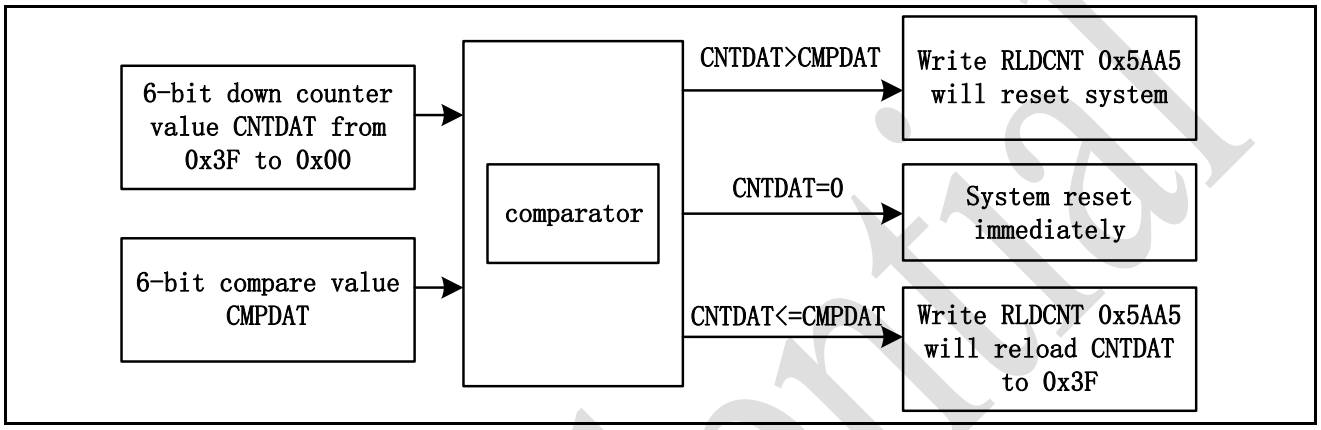


Figure 4-66 WWDT Reset and Reload Behavior

4.8.5.5 WWDT Window Setting Limitation

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

Table 4-15 CMPDAT Setting Limitation

PSCSEL	Prescale	Value
0000	1	0x03 ~ 0x3F
0001	2	0x02 ~ 0x3F
others	others	0x00 ~ 0x3F

4.8.6 WWDT Control Register Map

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

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

WWDT_STATUS	WWDT_BA+0x08	R/W	WWDT Status Register	0x0000_0000
WWDT_CNT	WWDT_BA+0x0C	R	WWDT Counter Value Register	0x0000_003F

4.8.7 WWDT Register Description

4.8.7.1 WWDT Reload Counter Register (WWDT_RLDCNT)

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

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

4.8.7.2 WWDT Control Register (WWDT_CTL)

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

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

		0101 = Pre-scale is 32; Max time-out period is 32 * 64 * WWDT_CLK. 0110 = Pre-scale is 64; Max time-out period is 64 * 64 * WWDT_CLK. 0111 = Pre-scale is 128; Max time-out period is 128 * 64 * WWDT_CLK. 1000 = Pre-scale is 192; Max time-out period is 192 * 64 * WWDT_CLK. 1001 = Pre-scale is 256; Max time-out period is 256 * 64 * WWDT_CLK. 1010 = Pre-scale is 384; Max time-out period is 384 * 64 * WWDT_CLK. 1011 = Pre-scale is 512; Max time-out period is 512 * 64 * WWDT_CLK. 1100 = Pre-scale is 768; Max time-out period is 768 * 64 * WWDT_CLK. 1101 = Pre-scale is 1024; Max time-out period is 1024 * 64 * WWDT_CLK. 1110 = Pre-scale is 1536; Max time-out period is 1536 * 64 * WWDT_CLK. 1111 = Pre-scale is 2048; Max time-out period is 2048 * 64 * WWDT_CLK.
[7:2]	Reserved	Reserved
[1]	INTEN	WWDT Interrupt Enable Bit If this bit is enabled, the WWDT counter compare match interrupt signal is generated and inform to CPU. 0 = WWDT counter compare match interrupt Disabled. 1 = WWDT counter compare match interrupt Enabled.
[0]	WWDTEN	WWDT Enable Bit Set this bit to enable WWDT counter counting. 0 = WWDT counter is stopped. 1 = WWDT counter is starting counting.

4.8.7.3 WWDT Status Register (WWDT_STATUS)

Register	Offset	R/W	Description	Reset Value
WWDT_STATUS	WWDT_BA+0x08	R/W	WWDT Status Register	0x0000_0000

Bits	Descriptions	
[31:3]	Reserved	Reserved
[2]	WWDTF	WWDT Compare Match Flag This bit indicates the flag status of WWDT while WWDT counter value matches CMPDAT (WWDT_CTL[21:16]). 0 = No effect. 1 = WWDT counter value matches CMPDAT. **Note**: This bit is cleared by writing 1 to it.
[1]	WWDTRF	WWDT Timer-out Reset Flag This bit indicates the system has been reset by WWDT time-out reset or not. 0 = WWDT time-out reset did not occur. 1 = WWDT time-out reset occurred. **Note**: This bit is cleared by writing 1 to it.
[0]	WWDTIF	WWDT Compare Match Interrupt Flag This bit indicates the interrupt flag status of WWDT while WWDT counter value matches CMPDAT (WWDT_CTL[21:16]). 0 = No effect.

		1 = WWDT counter value matches CMPDAT. **Note**: This bit is cleared by writing 1 to it.

4.8.7.4 WWDT Counter Value Register (WWDT_CNT)

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

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

4.9 UART Controller (UART)

4.9.1 Overview

The UART is a programmable Universal Asynchronous Receiver/Transmitter (UART). It converts parallel input signals into serial output signals.There is no CLK line. It transfers data by TX and RX line. UART achieves data identification by start bit, stop bit and baud rate

4.9.2 Features

- 9-bit serial data support
- Configurable parameters for the following:
  - APB data bus widths of 32
  - Additional DMA interface signals for compatibility with DMA interface
  - DMA interface signal polarity
  - Transmit and receive FIFO depths of 16
  - Internal FIFO (RAM) selection
  - Use of one clocks—just HCLK
  - Clock gate enable output(s) used to indicate that the TX and RX pipeline is clear (no data) and no activity has occurred for more than one character time, so that clocks can be gated
  - Additional FIFO status registers
  - Auto Flow Control mode, as specified in the 16750 standard
  - Transmitter Holding Register Empty (THRE) interrupt mode
- Ability to set some configuration parameters during instantiation
- Configuration identification registers present
- Functionality based on the 16550 industry standard
- FIFO support
- Programmable serial data baud rate as calculated by the following:
  - *baud rate = (serial clock frequency)/(16*divisor)*

4.9.3 Block Diagram

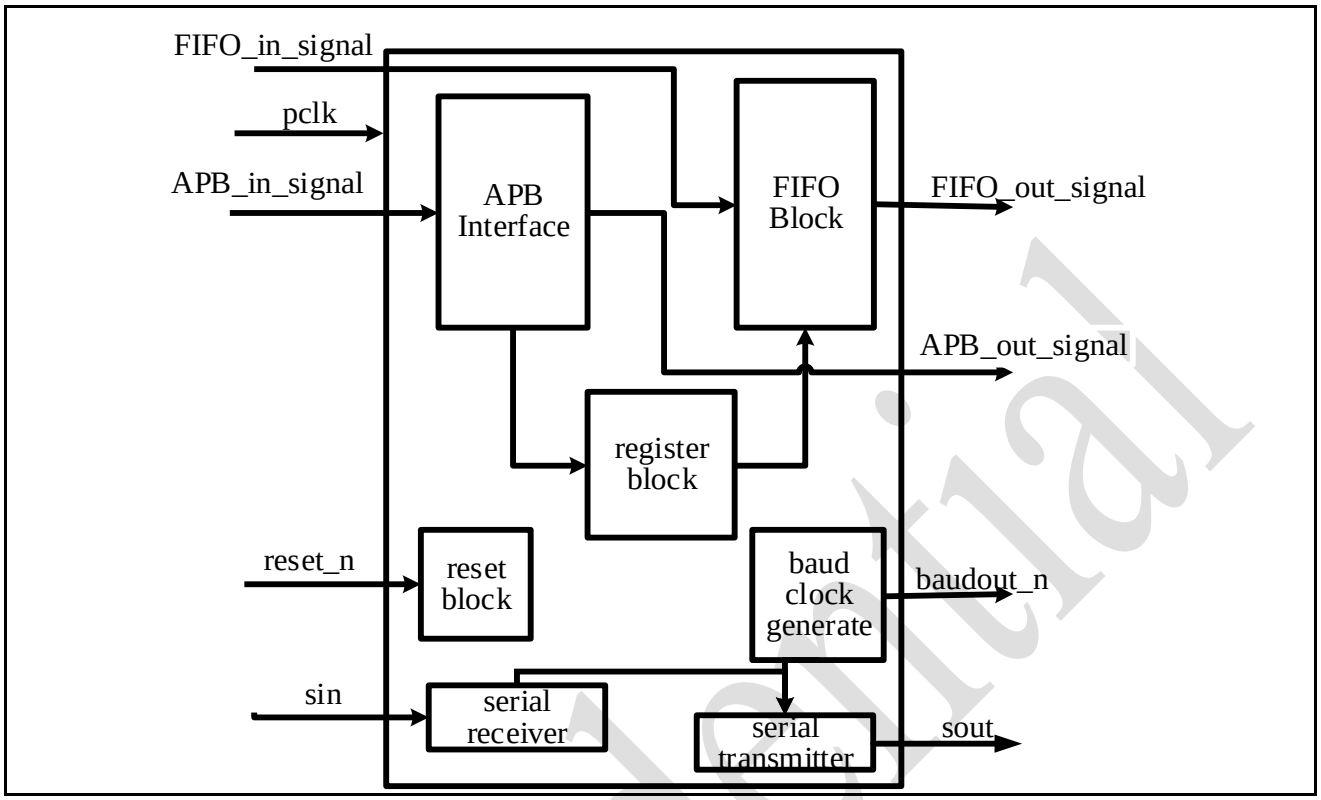


Figure 4-67 UART Controller Block Diagram

4.9.4 Functional Description

4.9.4.1 UART (RS232) Serial Protocol

Because the serial communication between the UART and a selected device is asynchronous, additional bits (start and stop) are added to the serial data to indicate the beginning and end. Utilizing these bits allows two devices to be synchronized.This structure of serial data—accompanied by start and stop bits—is referred to as a character, as shown in Figure 4-68.

An additional parity bit can be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure in order to provide the UART with the ability to perform simple error checking on the received data.

All the bits in the transmission are transmitted for exactly the same time duration; the exception to this is the half-stop bit when 1.5 stop bits are used. This duration is referred to as a Bit Period or Bit Time; one Bit Time equals sixteen baud clocks.

To ensure stability on the line, the receiver samples the serial input data at approximately the midpoint of the Bit Time once the start bit has been detected. Because the exact number of baud clocks is known for which each bit was transmitted, calculating the midpoint for sampling is not difficult; that is, every sixteen baud clocks after the midpoint sample of the start bit. Together with serial input debouncing, this sampling helps to avoid the detection of false start bits. Short glitches are filtered out by debouncing, and no transition is detected on the line. If a glitch is wide enough to avoid filtering by debouncing, a falling edge is detected. However, a start bit is detected only if the line is again sampled low after half a bit time has elapsed. Figure 4-68 shows the sampling points of the

first two bits in a serial character.

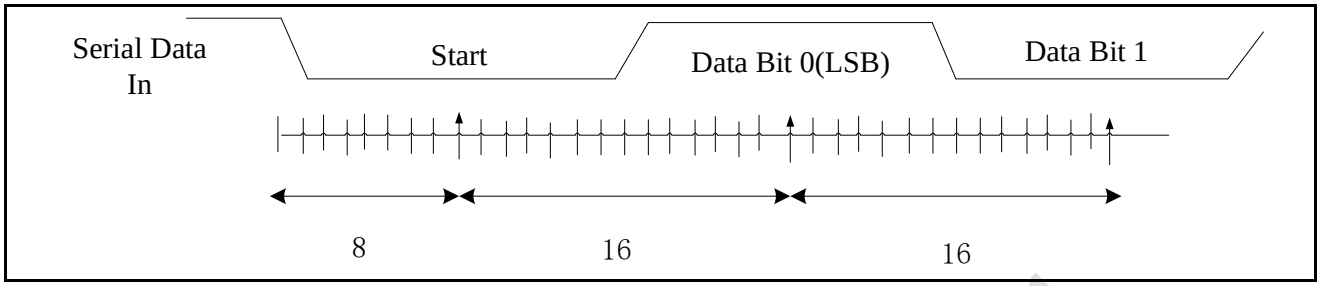


Figure 4-68 Receiver Serial Data Sample Points

As part of the 16550 standard, an optional baud clock reference output signal (baudout_n) provides timing information to receiving devices that require it. The baud rate of the UART is controlled by the serial clock—sclk or HCLK in a single clock implementation—and the Divisor Latch Register (DLH and DLL).

4.9.4.2 UART 9-bit Data Transfer

The UART can be configured to have 9-bit data transfer in both transmit and receive mode. The 9th bit in the character appears after the 8th bit and before the parity bit in the character.

By enabling 9-bit data transfer mode, UART can be used in multi-drop systems where one master isconnected to multiple slaves in a system. The master communicates with one of the slaves. When the master wants to transfer a block of data to a slave, it first sends an address byte to identify the target slave. The address byte is differentiated from the data byte by setting the 9th bit of the data byte to 1. If the 9th bit is setto 0, then the character represents an address byte. All the slave systems compare the address byte withtheir own address and only the target slave (in which the address has matched) is enabled to receive datafrom the master. The master then starts transmitting data bytes to the target slave. The non-addressed slave systems ignore the incoming data until a new address byte is received.

4.9.4.3 UART Fractional Baud Rate Support

UART supports fractional baud rate that enables a user to program the fractional part of the divisor value to generate fractional baud rate that results in reduced frequency error. The UART interface usage has been evolving to include ever increasing baud rate speeds. The UART needs to be software configurable to handle the baud rates within 2% frequency error.

The Baud rate of UART is controlled by HCLK in single clock implementation (CLOCK_MODE=1) and the Divisor Latch Register (DLH and DLL).

The baud rate is determined by the following factors:

- Serial clock operating frequency (HCLK in single clock implementation)
- The desired baud rate.
- The baud rate generator divisor value, DIVISOR (composed of DLH & DLL registers).
- The acceptable Baud-rate error, %ERROR

The programmable fractional baud rate divisor enables a finer resolution of baud clock than the conventional integer divider. The programmable fractional baud clock divider allows for the programmability of both an integer divisor as well as fractional component. The average frequency of

the baud clock from the fractional baud rate divisor is dependent upon both the integer divisor and the fractional component, thereby providing a finer resolution to the average frequency of the baud clock.

Baud Rate Divisor = Serial Clock Frequency / (16∗Required Baud Rate) = $BRD_I + BRD_F$ (1)

Where,

BRD_I - Integer part of the divisor.

BRD_F - Fractional part of the divisor.

Fractional division of clock is used by the N/N+1 divider, where N is the integer part of the divisor. N/N+1division works on the basis of achieving the required average timing over a long period by alternating the division between two numbers. If N=1 and ratio of N/N+1 is same, which means equal number of divide by 1 and divide by 2 over a period of time, average time period would come out to be divided by 1.5. Varying the ratio of N/N+1 any value can be achieved above 1 and below 2.

4.9.4.4 FIFO and Clock Support

- FIFO surpport

In PAN1020, the FIFO_MODE is set to 16 which means the UART works in the FIFO mode. Moreover, the FIFOs are selected to the Internal D-flip-flop-based RAMs (ram_r_w_s_dff). Selecting internal memory restricts the Memory Read Port Type to D-flip-flop-based Synchronous read port RAMs.

- Clock surpport

The UART can be configured to have one system clock (HCLK). When using a single-system clock, available system clock settings for accurate baud rates are greatly restricted.

4.9.4.5 UART Interrupts

Assertion of the UART interrupt output signal (intr)—a positive-level interrupt—occurs whenever one of the several prioritized interrupt types are enabled and active. When an interrupt occurs, the master accesses the IIR register.

The following interrupt types can be enabled with the IER register:

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE interrupt mode)
- Busy Detect Indication

These interrupt types are covered in more detail in Table 4-16.

Table 4-16 Interrupt Control Functions

Interrupt ID	Priority Level	Interrupt Type
0001	-	None
0110	Highest	Receiver line status

0100	Second	Received data available
1100	Second	Character timeout indication
0010	Third	Transmit holding register empty
0000	Fourth	Modem status
0111	Fifth	Busy detect indication

4.9.4.6 Auto Flow Control

The UART can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available. When Auto Flow Control is not selected, none of the corresponding logic is implemented and the mode cannot be enabled, reducing overall gate counts. Figure 4-69 shows a block diagram of the Auto Flow Control functionality. When Auto RTS and Auto CTS are enabled, the rts_n output is forced inactive when the receiver FIFO level reaches the threshold set by FCR[7:6]. When rts_n is connected to the cts_n input of another UART device, the other UART stops sending serial data until the receiver FIFO has available space.

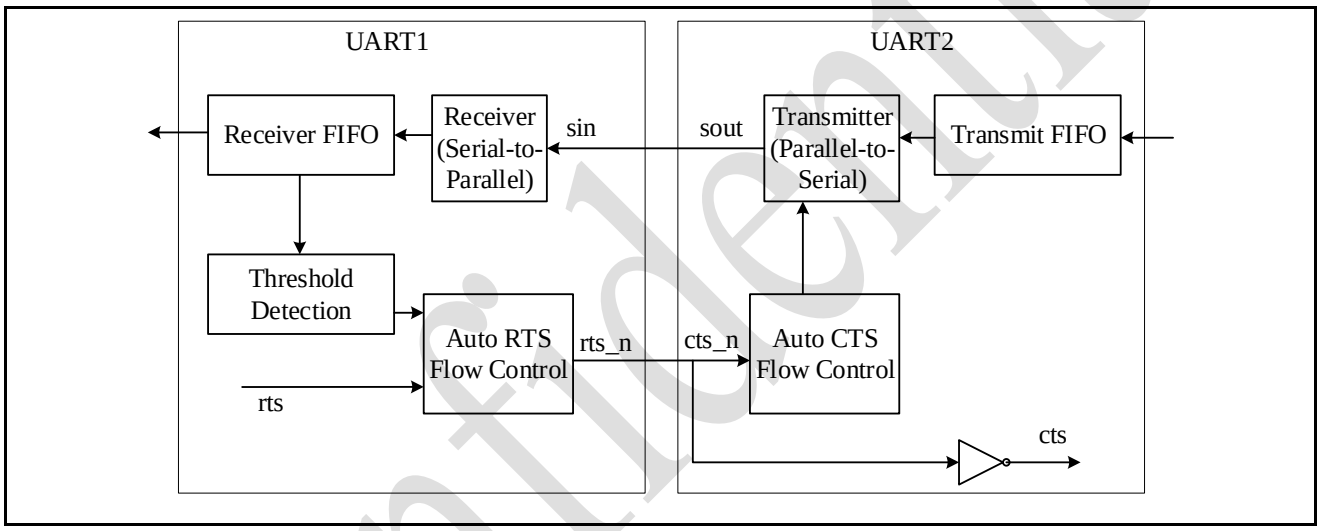


Figure 4-69 Auto Flow Control Block Diagram

4.9.4.7 Programmable THRE Interrupt

The UART can be configured for a Programmable THRE Interrupt mode in order to increase system performance; if FIFOs are not implemented, then this mode cannot be selected.

- When Programmable THRE Interrupt mode is not selected, none of the logic is implemented and the mode cannot be enabled, reducing the overall gate counts.
- When Programmable THRE Interrupt mode is selected, it can be enabled using the Interrupt Enable Register (IER[7]).

When FIFOs and THRE mode are implemented and enabled, the THRE Interrupts and dma_tx_req_n are active at, and below, a programmed transmitter FIFO empty threshold level, as opposed to empty, as shown in the flowchart in Figure 4-70.

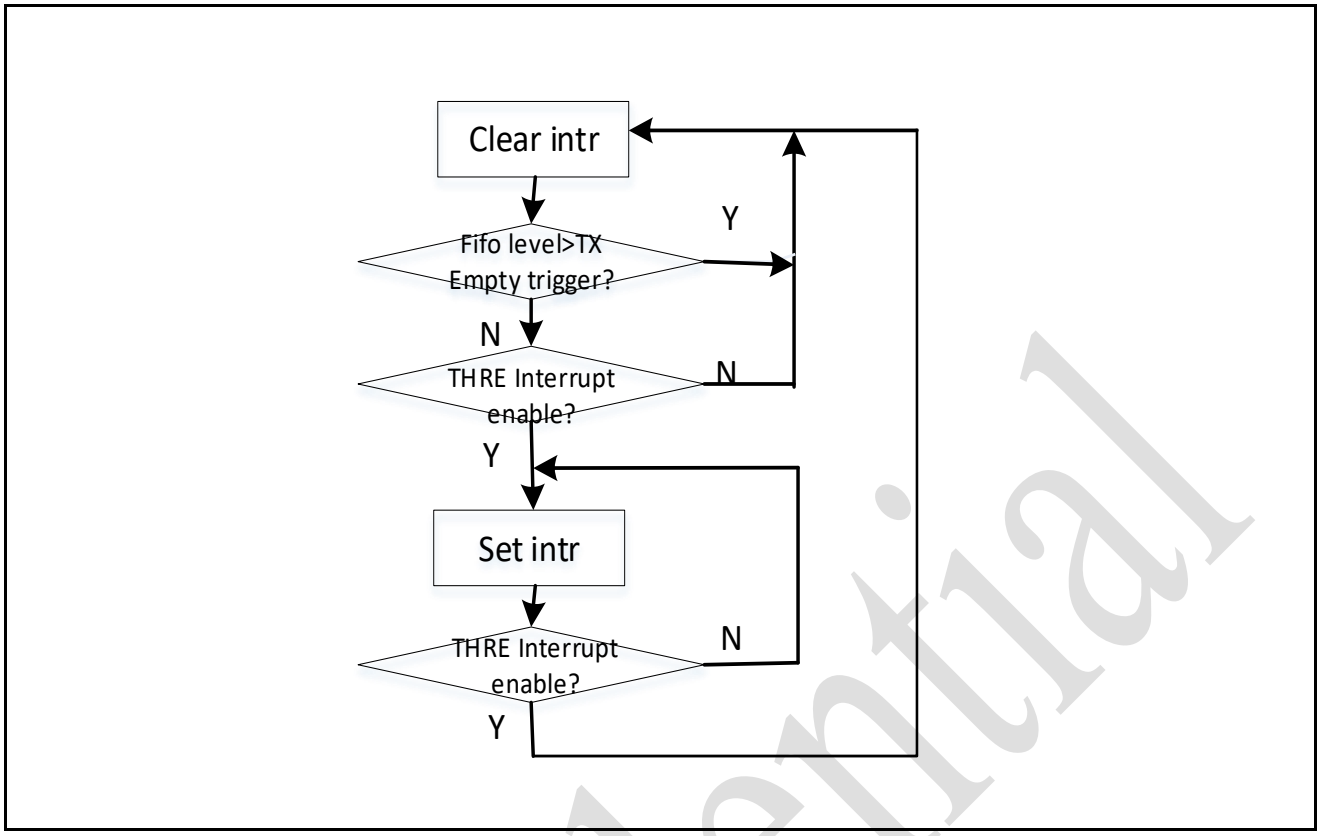


Figure 4-70 Flowchart of Interrupt Generation for Programmable THRE Interrupt Mode

The threshold level is programmed into FCR[5:4]. Available empty thresholds are:

- Empty
- 2
- 1/4
- 1/2

Selection of the best threshold value depends on the system's ability to begin a new transmission sequence in a timely manner. However, one of these thresholds should be optimal for increasing system performance by preventing the transmitter FIFO from running empty. For threshold setting details, refer to FCR.

In addition to the interrupt change, the Line Status Register (LSR[5]) also switches from indicating that the transmitter FIFO is empty to the FIFO being full. This allows software to fill the FIFO for each transmit sequence by polling LSR[5] before writing another character. The flow then allows the transmitter FIFO to be filled whenever an interrupt occurs and there is data to transmit, rather than waiting until the FIFO is completely empty. Waiting until the FIFO is empty causes a reduction in performance whenever the system is too busy to respond immediately. Further system efficiency is achieved when this mode is enabled in combination with Auto Flow Control.

4.9.4.8 Clock Gate Enable

In PAN1020, the UART can be configured to have a clock gate enable output.

The assertion of clock gate enable signals is an indication that the UART is inactive, so clocks may

be gated in order to put the device in a low-power (lp) mode. Therefore, the following must be true for at least one character time for the assertion of the clock gate enable signal(s) to occur:

- the RX FIFO is empty (in FIFO mode)
- the TX FIFO is empty (in FIFO mode)
- sin/sir_in and sout/sir_out_n are inactive (sin/sir_in are kept high and sout is high or sir_out_n is low) indicating no activity
- No change on the modem control input signals

When the assertion criteria are no longer met, the clock gate enable signal(s) are de-asserted and the clock(s) is resumed under any of these conditions:

- sin signal goes low
- Write to any of registers is performed

The time taken for the clock(s) to resume is important in preventing receive data synchronization problems, due to the UART RX block sampling:

1. At mid-point of each bit period—after approximately 8 baud clocks—in UART (RS323) mode.
2. After that, for a single clock implementation, this is 16 PCLKs.

4.9.4.9 DMA Support

The UART supports DMA signaling with use of the signal described in Table 4-17.

Table 4-17 DMA Interface Signal

Port Name	I/O	Description
dma_tx_ack_n	I	DMA Transmit Acknowledge (Active High)
dma_rx_ack_n	I	DMA Receive Acknowledge (Active High)
dma_tx_req_n	O	Transmit Buffer Ready (Active High)
dma_tx_single_n	O	DMA Transmit FIFO Single (Active High)
dma_rx_req_n	O	Receive Buffer Ready (Active High)
dma_rx_single_n	O	DMA Receive FIFO Single (Active High)

4.9.4.10 Interface Signal

The overall modem interface signals are illustrated in Table 4-18. All the ports always exists regardless of the chip operating conditions.

Table 4-18 Modem Interface Signals

Port Name	I/O	Description
cts_n	I	Clear To Send Modem Status Active State: Low Synchronous to: N/A
dsr_n	I	Data Set Ready Modem Status input Active State: Low Synchronous to: N/A
dcd_n	I	Data Carrier Detect Modedm Status input Active State: Low Synchronous to: N/A

ri_n	I	Ring Indicator Status input Active State: Low Synchronous to: N/A
dtr_n	O	Modem Control Data Terminal Ready Output Active State: Low Synchronous to: pclk
rts_n	O	Modem Control Request To Send output Active State: Low Synchronous to: pclk
out2_n	O	Modem Control Programmable output 2 Active State: Low Synchronous to: pclk
out1_n	O	Modem Control Programmable output 1 Active State: Low Synchronous to: pclk

4.9.5 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 = 0x4010_0000** **UART1_BA = 0x4010_1000**				
RBR	UARTx_BA +0x00	R	Receive Buffer Register	0x0000_0000
THR	UARTx_BA +0x00	W	Transmit Holding Register	0x0000_0000
DLL	UARTx_BA +0x00	R/W	Divisor Latch (Low)	0x0000_0000
DLH	UARTx_BA +0x04	R/W	Divisor Latch (High)	0x0000_0000
IER	UARTx_BA +0x04	R/W	Interrupt Enable Register	0x0000_0000
IIR	UARTx_BA +0x08	R	Interrupt Identification Register	0x0000_0001
FCR	UARTx_BA +0x08	W	FIFO Control Register	0x0000_0000
LCR	UARTx_BA +0x0C	R/W	Line Control Register	0x0000_0000
MCR	UARTx_BA +0x10	R/W	Modem Control Register	0x0000_0000
LSR	UARTx_BA +0x14	R	Line Status Register	0x0000_0060
MSR	UARTx_BA +0x18	R	Modem Status Register	0x0000_0000
SCR	UARTx_BA +0x1C	R/W	Scratchpad Register	0x0000_0000
USR	UARTx_BA +0x7C	R	UART Status Register	0x0000_0006
TFL	UARTx_BA +0x80	R	Transmit FIFO Level	0x0000_0000
RFL	UARTx_BA +0x84	R	Receive FIFO Level	0x0000_0000
HTX	UARTx_BA +0xA4	R/W	Halt TX	0x0000_0000
DMASA	UARTx_BA +0xA8	W	DMA Software Acknowledge	0x0000_0000
DLF	UARTx_BA +0xC0	R/W	Divisor Latch Fractional Value.	0x0000_0000

RAR	UARTx_BA +0xC4	R/W	Receive Address Register	0x0000_0000
TAR	UARTx_BA +0xC8	R/W	Transmit Address Register	0x0000_0000
LCR_EXT	UARTx_BA +0xCC	R/W	Line Extended Control Register	0x0000_0000

4.9.6 Register Description

4.9.6.1 Receive Buffe Register (RBR)

This register can be accessed only when the DLAB bit (LCR[7]) is cleared.

Register	Offset	R/W	Description	Reset Value
RBR x = 0, 1	UARTx_BA+0x00	R	Receive Buffer Register	0x0000_0000

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	RBRM	Receive Buffer register (MSB 9th bit) Data byte received on the serial input port (sin) in UART mode for the MSB 9th bit.
[7:0]	RBRL	Receive Buffer Register (LSB 8 bits) Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if DR bit (LSR[0]) is set. If FIFOs are enabled (FCR[0] set to 1), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs.

4.9.6.2 Transmit Holding Register (THR)

This register can be accessed only when the DLAB bit (LCR[7]) is cleared.

Register	Offset	R/W	Description	Reset Value
THR x = 0, 1	UARTx_BA+0x00	W	Transmit Holding Register	0x0000_0000

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	THRM	Transmit Holding Register (MSB 9th bit) Data to be transmitted on the serial output port (sout) in UART mode for the MSB 9th bit.
[7:0]	THRL	Transmit Holding Register (LSB 8 bits) Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THRE bit (LSR[5]) is set.

		If FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x(default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

4.9.6.3 Divisor Latch Low (DLL)

This register can be accessed only when the DLAB bit (LCR[7]) is cleared.

Register	Offset	R/W	Description	Reset Value
DLL x = 0, 1	UARTx_BA+0x00	R/W	Divisor Latch Low	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	DLL	Divisor Latch (Low) Lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. **Note** that with the Divisor Latch Registers (DLL and DLH) set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

4.9.6.4 Divisor Latch High (DLH)

This register can be accessed only when the DLAB bit (LCR[7]) is cleared.

Register	Offset	R/W	Description	Reset Value
DLH x = 0, 1	UARTx_BA+0x04	R/W	Divisor Latch High Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved.
[7:0]	DLH	Divisor Latch (High) Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. **Note** that with the Divisor Latch Registers (DLL and DLH) set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving.

4.9.6.5 Interrupt Enable Register (IER)

This register can be accessed only when the DLAB bit (LCR[7]) is cleared.

Register	Offset	R/W	Description	Reset Value
IER x = 0, 1	UARTx_BA+0x04	R/W	Interrupt Enable Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved.
[7]	PTIME	Programmable THRE Interrupt Mode Enable. This is used to enable/disable the generation of THRE Interrupt. 0–disabled 1–enabled
[6:4]	Reserved	Reserved
[3]	EDSSI	Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0–disabled 1–enabled
[2]	ELSI	Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0–disabled 1–enabled
[1]	ETBEI	Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 – disabled 1 – enabled
[0]	ERBFI	Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt(if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 – disabled 1 – enabled

4.9.6.6 Interrupt Identity Register (IIR)

Register	Offset	R/W	Description	Reset Value
IIR x = 0, 1	UARTx_BA+0x08	R	Interrupt Identity Register	0x0000_0001

Bits	Description	
[31:8]	Reserved	Reserved.
[7:6]	FIFOSE	FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 00 – disabled 11 – enabled
[5:4]	Reserved and read as 0	Reserved
[3:0]	IID	Interrupt ID. This indicates the highest priority pending interrupt which can be one of the following types: 0000 – modem status 0001 – no interrupt pending 0010 – THR empty 0100 – received data available 0110 – receiver line status 1100 – character timeout The interrupt priorities are split into several levels that are detailed in Table 4-19.

Table 4-19 Interrupt Control Functions

IID	Priorities Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0001	-	None	None	-
0110	Highest	Receive line status	Overrun/parity/ framing errors, break interrupt, or address received interrupt	Reading the line status register. In addition to LSR read, the Receiver line status is also cleared when RX_FIFO is read
0100	Second	Received data available	Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled)	Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO drops below the trigger level (FIFO mode and FIFOs enabled)
1100	Second	Character timeout indication	No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1 character in it during this time	Reading the receiver buffer register
0010	Third	Transmit holding register empty	Transmitter holding register empty (Prog. THRE Mode disabled) or XMIT FIFO at or below threshold (Prog. THRE Mode enabled)	Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE Mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE Mode selected and enabled)

0000	Fourth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note that if auto flow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt.	Reading the Modem status register
0111	Fifth	Busy detect indication	UART_16550_COMPATI-BLE= NO and master has tried to write to the Line Control Register while the UART is busy(USR[0] is set to 1).	Reading the UART status register

4.9.6.7 FIFO Control Register (FCR)

Register	Offset	R/W	Description	Reset Value
FCR x = 0, 1	UARTx_BA+0x08	W	FIFO Control Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved
[7:6]	RT	RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode, this trigger is used to de-termine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. 00 – 1 character in the FIFO 01 – FIFO ¼ full 10 – FIFO ½ full 11 – FIFO 2 less than full
[5:4]	TET	TX Empty Trigger. This is used to select the empty threshold level at which the THRE Interrupts are gen-erated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. 00 – FIFO empty 01 – 2 characters in the FIFO 10 – FIFO ¼ full 11 – FIFO ½ full
[3]	Reserved	Reserved
[2]	XFIFOR	XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals. **Note** that this bit is 'self-clearing'. It is not necessary to clear this bit.
[1]	RFIFOR	RCVR FIFO Reset.

		This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals. **Note** that this bit is 'self-clearing'. It is not necessary to clear this bit.
[0]	FIFOE	FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

4.9.6.8 Line Control Register (LCR)

Register	Offset	R/W	Description	Reset Value
LCR x = 0, 1	UARTx_BA+0x0C	R/W	Line Control Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved
[7]	DLAB	Divisor Latch Access Bit. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.
[6]	BC	Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to 1, the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.
[5]	SP	Stick Parity. This bit is used to force parity value. When PEN, EPS, and Stick Parity are set to 1,the parity bit is transmitted and checked as logic 0. If PEN and Stick Parity are set to 1 and EPS is a logic 0, then parity bit is transmitted and checked as a logic 1. If this bit is set to 0, Stick Parity is disabled.
[4]	EPS	Even Parity Select. This is used to select between even and odd parity to be transmitted or checked, when parity is enabled (PEN set to 1). 0 – odd parity 1 – even parity
[3]	PEN	Parity Enable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 – parity disabled 1 – parity enabled
[2]	STOP	Number of stop bits. This is used to select the number of stop bits per character that the peripheral transmits

		and receives. 0 – 1 stop bit 1 – 1.5 stop bits when DLS (LCR[1:0]) is 0, else 2 stop bit **Note** that regardless of the number of stop bits selected, the receiver checks only the first stop bit.
[1:0]	DLS	Data Length Select. When DLS_E in LCR_EXT is set to 0, this register is used to select the number of data bits per character that the peripheral transmits and receives. The number of bits that may be selected are as follows: 00 – 5 bits 01 – 6 bits 10 – 7 bits 11 – 8 bits

4.9.6.9 Modem Control Register (MCR)

Register	Offset	R/W	Description	Reset Value
MCR x = 0, 1	UARTx_BA+0x10	R/W	Modem Control Register	0x0000_0000

Bits	Description	
[31:7]	Reserved	Reserved
[6]	SIRE	SIR Mode Enable. (Write Protected) 0 – IrDA SIR Mode disabled 1 – IrDA SIR Mode enabled
[5]	AFCE	Auto Flow Control Enable. 0 – Auto Flow Control Mode disabled 1 – Auto Flow Control Mode enabled
[4]	LB	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] set to 0), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally.
[3]	OUT2	OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 0 – out2_n de-asserted (logic 1) 1 – out2_n asserted (logic 0) **Note** that in Loopback mode (MCR[4] set to 1), the out2_n output is held inactive high while the value of this location is internally looped back to an input.
[2]	OUT1	OUT1.

		This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is: 0 – out1_n de-asserted (logic 1) 1 – out1_n asserted (logic 0) **Note** that in Loopback mode (MCR[4] set to 1), the out1_n output is held inactive high while the value of this location is internally looped back to an input.
[1]	RTS	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, (MCR[5] set to 1) and FIFOs enable (FCR[0] set to 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold) only when the RTC Flow Trigger is disabled; otherwise it is gated by the receiver FIFO almost-full trigger, where “almost full” refers to two available slots in the FIFO (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to 1), the rts_n output is held inactive high while the value of this location is internally looped back to an input.
[0]	DTR	Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 – dtr_n de-asserted (logic 1) 1 – dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to 1), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.

4.9.6.10 Line Status Register (LSR)

Register	Offset	R/W	Description	Reset Value
MCR x = 0, 1	UARTx_BA+0x14	R	Line Status Register	0x0000_0060

Bits	Description	
[31:9]	Reserved	Reserved
[8]	ADDR_RCVD	Address Received bit If 9-bit data mode (LCR_EXT[0]=1) is enabled, this bit is used to indicate that the 9th bit of the receive data is set to 1. This bit can also be used to indicate whether the incoming character is an address or data. 1 - Indicates that the character is an address. 0 - Indicates that the character is data.

		In the FIFO mode, since the 9th bit is associated with the received character, it is revealed when the character with the 9th bit set to 1 at the top of the FIFO list. Reading the LSR clears the 9th bit. **Note**: You must ensure that an interrupt gets cleared (reading LSR register) before the next address byte arrives. If there is a delay in clearing the interrupt, then software will not be able to distinguish between multiple address related interrupt.
[7]	RFE	Receiver FIFO Error bit. This bit is only relevant when FIFOs are enabled (FCR[0] set to 1). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 – no error in RX FIFO 1 – error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO.
[6]	TEMT	Transmitter Empty bit. If in FIFO mode and FIFOs enabled (FCR[0] set to 1), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.
[5]	THRE	Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to 0) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If THRE_MODE_USER = Enabled and both modes are active (IER[7] set to 1 and FCR[0] set to 1 respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.
[4]	BI	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data.If in UART mode (SIR_MODE = Disabled), it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all 0s, to be received by the UART. In FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. **Note**: If a FIFO is full when a break condition is received, a FIFO overrun occurs. The break condition and all the information associated with it—parity and framing errors—is discarded; any information that a break character was received is lost.
[3]	FE	Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver.A framing error occurs when the receiver does not detect a valid STOP bit in the received data.In

		the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit; that is, data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). This happens because the break character implicitly generates a framing error by holding the sin input to logic 0 for longer than the duration of a character. 0 – no framing error 1 – framing error Reading the LSR clears the FE bit.
[2]	PE	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO.It should be noted that the Parity Error (PE) bit (LSR[2]) can be set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). In this situation, the Parity Error bit is set if parity generation and detection is enabled (LCR[3]=1) and the parity is set to odd (LCR[4]=0). 0 – no parity error 1 – parity error Reading the LSR clears the PE bit.
[1]	OE	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0 – no overrun error 1 – overrun error Reading the LSR clears the OE bit.
[0]	DR	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 – no data ready 1 – data ready This bit is cleared when the receiver FIFO is empty, in FIFO mode.

4.9.6.11 Modem Status Register (MSR)

Register	Offset	R/W	Description	Reset Value
MSR x = 0, 1	UARTx_BA+0x18	R	ModemStatus Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved
[7]	DCD	Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. 0 – dcd_n input is de-asserted (logic 1) 1 – dcd_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), DCD is the same as MCR[3] (Out2).
[6]	RI	Ring Indicator. This is used to indicate the current state of the modem control line ri_n.This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set. 0 – ri_n input is de-asserted (logic 1) 1 – ri_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), RI is the same as MCR[2] (Out1).
[5]	DSR	Data Set Ready. This is used to indicate the current state of the modem control line dsr_n.This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with the UART. 0 – dsr_n input is de-asserted (logic 1) 1 – dsr_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), DSR is the same as MCR[0] (DTR).
[4]	CTS	Clear to Send. This is used to indicate the current state of the modem control line cts_n. This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with the UART. 0 – cts_n input is de-asserted (logic 1) 1 – cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).
[3]	DDCD	Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read. 0 – no change on dcd_n since last read of MSR 1 – change on dcd_n since last read of MSR Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] = 1), DDCD reflects changes on MCR[3] (Out2). **Note**, if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset

		occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.
[2]	TERI	Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read. 0 – no change on ri_n since last read of MSR 1 – change on ri_n since last read of MSR Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] = 1), TERI reflects when MCR[2] (Out1) has changed state from a high to a low.
[1]	DDSR	Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read. 0 – no change on dsr_n since last read of MSR 1 – change on dsr_n since last read of MSR Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] = 1), DDSR reflects changes on MCR[0] (DTR). **Note**, if the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted.
[0]	DCTS	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. 0 – no change on cts_n since last read of MSR 1 – change on cts_n since last read of MSR Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). **Note**, if the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted.

4.9.6.12 Scratchpad Register (SCR)

Register	Offset	R/W	Description	Reset Value
SCR x = 0, 1	UARTx_BA+0x1C	R/W	Scratchpad Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved
[7:0]	Scratchpad Register	This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART.

4.9.6.13 UART Status Register (USR)

Register	Offset	R/W	Description	Reset Value
USR x = 0, 1	UARTx_BA+0x7C	R	UART Status Register	0x0000_0006

Bits	Description	
[31:5]	Reserved	Reserved
[4]	RFF	Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. 0 – Receive FIFO not full 1 – Receive FIFO Full This bit is cleared when the RX FIFO is no longer full.
[3]	RFNE	Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty This bit is cleared when the RX FIFO is empty.
[2]	TFE	Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.
[1]	TFNF	Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
[0]	Reserved	Reserved

4.9.6.14 Transmit FIFO Level (TFL)

Register	Offset	R/W	Description	Reset Value
TFL x = 0, 1	UARTx_BA+0x80	R	Transmit FIFO Level	0x0000_0000

Bits	Description	
[31:5]	Reserved	Reserved
[4:0]	TFL	Transmit FIFO Level. This indicates the number of data entries in the transmit FIFO.

4.9.6.15 Receive FIFO Level (RFL)

Register	Offset	R/W	Description	Reset Value
RFL x = 0, 1	UARTx_BA+0x84	R	Receive FIFO Level	0x0000_0000

Bits	Description	
[31:5]	Reserved	Reserved
[4:0]	RFL	Receive FIFO Level. This indicates the number of data entries in the receive FIFO.

4.9.6.16 Halt TX (HTX)

Register	Offset	R/W	Description	Reset Value
HTX x = 0, 1	UARTx_BA+0XA4	R/W	Halt TX	0x0000_0000

Bits	Description	
[31:1]	Reserved	Reserved
[0]	HTX	This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 = Halt TX disabled 1 = Halt TX enabled **Note**, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation.

4.9.6.17 DMA Software Acknowledge (DMASA)

Register	Offset	R/W	Description	Reset Value
DMASA x = 0, 1	UARTx_BA+0xA8	W	DMA Software Acknowledge	0x0000_0000

Bits	Description	
[31:1]	Reserved	Reserved
[0]	DMASA	DMA Software Acknowledge. This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the UART should clear its request. This causes the TX request, TX single, RX request and RX single signals to de-assert. **Note** that this bit is 'self-clearing'. It is not necessary to clear this bit.

4.9.6.18 Divisor Latch Fraction Register (DLF)

Register	Offset	R/W	Description	Reset Value
DLF x=0,1	UARTx_BA+0xC0	R/W	Divisor Latch Fraction Register	0x0000_0000

Bits	Description	
[31:4]	Reserved	Reserved
[3:0]	DLF	Fractional part of divisor. The fractional value is added to integer value set by DLH, DLL. Fractional value is determined by (Divisor Fraction value)/(2^DLF_SIZE). This table describes the DLF Values to be programmed for DLF_SIZE=4.

DLF Value	Fraction	Fractional Value
0000	0/16	0.0000
0001	1/16	0.0625
0010	2/16	0.125
0011	3/16	0.1875
0100	4/16	0.25
0101	5/16	0.3125
0110	6/16	0.375
0111	7/16	0.4375
1000	8/16	0.5
1001	9/16	0.5625
1010	10/16	0.625
1011	11/16	0.6875
1100	12/16	0.75
1101	13/16	0.8125
1110	14/16	0.875
1111	15/16	0.9375

4.9.6.19 Receive Address Register (RAR)

Register	Offset	R/W	Description	Reset Value
RAR x=0,1	UARTx_BA+0xC4	R/W	Receive Address Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved
[7:0]	RAR	This is an address matching register during receive mode. If the 9-th bit is set in the incoming character then the remaining 8-bits will be checked against this register

		value. If the match happens then sub-sequent characters with 9-th bit set to 0 will be treated as data byte until the next address byte is received. **Note**:This register is applicable only when 'ADDR_MATCH' (LCR_EXT[1]) and 'DLS_E' (LCR_EXT[0]) bits are set to 1.

4.9.6.20 Transmit Address Register (TAR)

Register	Offset	R/W	Description	Reset Value
TAT x=0,1	UARTx_BA+0xC8	R/W	Transmit Address Register	0x0000_0000

Bits	Description	
[31:8]	Reserved	Reserved
[7:0]	TAR	This is an address matching register during transmit mode. If DLS_E (LCR_EXT[0]) bit is enabled, then UART sends the 9-bit character with 9-th bit set to 1 and remaining 8-bit address will be sent from this register provided 'SEND_ADDR' (LCR_EXT[2]) bit is set to 1. **Note**: ■ This register is used only to send the address. The normal data should be sent by programming THR register. ■ Once the address is started to send on the UART serial lane, then 'SEND_ADDR' bit will be auto-cleared by the hardware.

4.9.6.21 Line Extended Control Register (LCR_EXT)

Register	Offset	R/W	Description	Reset Value
LCR_EXT x=0,1	UARTx_BA+0xCC	R/W	Line Extended Control Register	0x0000_0000

Bits	Description	
[31:4]	Reserved	Reserved
[3]	TRANSMIT_MODE	Transmit mode control bit. This bit is used to control the type of transmit mode during 9-bit data transfers. 1 = In this mode of operation, Transmit Holding Register (THR) is 9-bit wide. You must ensure that the THR register is written correctly for address/data. Address: 9th bit is set to 1, Data: 9th bit is set to 0. **Note**: Transmit address register (TAR) is not applicable in this mode of operation. 0 = In this mode of operation, Transmit Holding Register (THR) are 8-bit wide. The user needs to program the address into Transmit Address Register (TAR) and data into the THR register. SEND_ADDR bit is used as a control knob to indicate the UART on when to send the

		address.
[2]	SEND_ADDR	Send address control bit. This bit is used as a control knob for the user to determine when to send the address during transmit mode. 1 = 9-bit character will be transmitted with 9-th bit set to 1 and the remaining 8-bits will match to what is being programmed in "Transmit Address Register". 0 = 9-bit character will be transmitted with 9-th bit set to 0 and the remaining 8-bits will be taken from the TxFIFO which is programmed through 8-bit wide THR/STHR register. **Note**: 1. This bit is auto-cleared by the hardware, after sending out the address character. User is not expected to program this bit to 0. 2. This field is applicable only when DLS_E bit is set to 1 and TRANSMIT_MODE is set to 0.
[1]	ADDR_MATCH	Address Match Mode. This bit is used to enable the address match feature during receive. 1 = Address match mode; UART will wait until the incoming character with 9-th bit set to 1. and, further checks to see if the address matches with what is programmed in "Receive Address Match Register". If match is found, then sub-sequent characters will be treated as valid data and UART starts receiving data. 0 = Normal mode; UART will start to receive the data and 9-bit character will be formed and written into the receive RxFIFO. User is responsible to read the data and differentiate b/n address and data. **Note**: This field is applicable only when DLS_E is set to 1.
[0]	DLS_E	Extension for DLS. This bit is used to enable 9-bit data for transmit and receive transfers. 1 = 9 bits per character 0 = Number of data bits selected by DLS

4.10 I2C Serial Interface Controller (I2C)

4.10.1 Overview

The I2C bus is a two-wire serial interface, consisting of a serial data line (SDA) and a serial clock (SCL).These wires carry information between the devices connected to the bus. Each device is recognized by a unique address and can operate as either a “transmitter” or “receiver,” depending on the function of the device. Devices can also be considered as masters or slaves when performing data transfers. A master is a device that initiates a data transfer on the bus and generates the clock signals to permit that transfer. At that time, any device addressed is considered a slave.

The I2C module can operate in standard mode (with data rates 0 to 100 Kb/s), fast mode (with data rates less than or equal to 400 Kb/s), fast mode plus (with data rates less than or equal to 1000 Kb/s), high-speed mode (with data rates less than or equal to 3.4 Mb/s).

Depending on specific device implementation DMA capability can be available for reduced CPU overload.

4.10.2 Features

- Three speeds:
  - Standard mode (0 to 100 Kb/s)
  - Fast mode (≤ 400 Kb/s) or fast mode plus (≤ 1000 Kb/s)
  - High-speed mode (≤ 3.4 Mb/s)
- Two operation mode:
  - Slave mode
  - Master mode
- Clock synchronization
- 7- or 10-bit addressing
- 7- or 10-bit combined format transfers
- Bulk transmit mode
- Ignore CBUS addresses (an older ancestor of I2C that used to share the I2C bus)
- Transmit and receive buffers
- Interrupt or polled-mode operation
- Support DMA
- Programmable SDA hold time (tHD;DAT)
- Multiple masters arbitration

4.10.3 Block Diagram

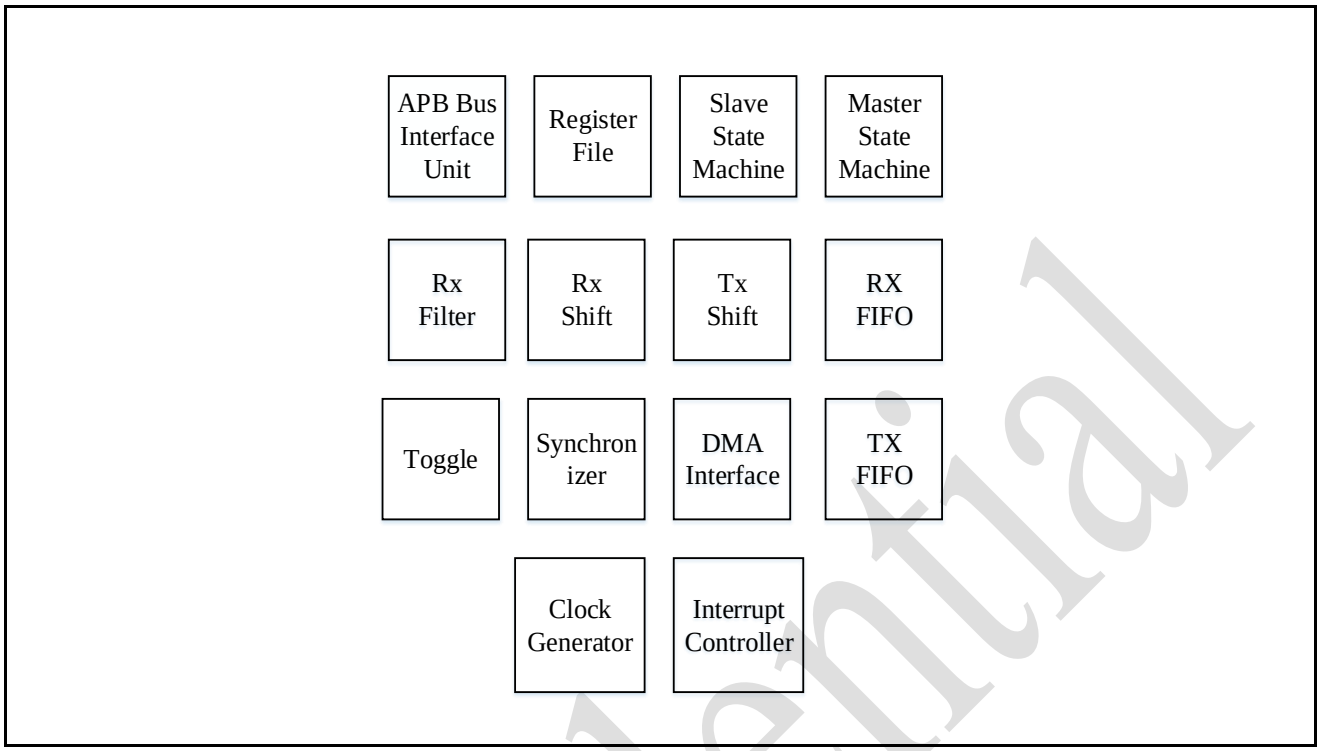


Figure 4-71 I2C Controller Block Diagram

4.10.4 Functional Description

This chapter describes the functional behavior of I2C in more detail.

4.10.4.1 I2C Behavior

On I2C bus, data is transferred between a Master and a Slave. The master is responsible for generating the clock and controlling the transfer of data. The slave is responsible for either transmitting or receiving data to/from the master. Data bits transfer on the SCL and SDA lines are synchronous on a byte-by-byte basis. There is one SCL clock pulse for each data bit with the MSB being transmitted first, and an acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (START or STOP).

Each slave has a unique address that is determined by the system designer. When a master wants to communicate with a slave, the master transmits a START/RESTART condition that is then followed by the slave's address and a control bit (R/W) to determine if the master wants to transmit data or receive data from the slave. The slave then sends an acknowledge (ACK) pulse after the address.

Moreover, the I2C protocol also allows multiple masters to reside on the I2C bus and uses an arbitration procedure to determine bus ownership.

The interface can operate in one of the four following modes:

- Slave transmitter
- Slave receiver

- Master transmitter
- Master receiver

If the master (master-transmitter) is writing to the slave (slave-receiver), the receiver gets one byte of data. This transaction continues until the master terminates the transmission with a STOP condition. If the master is reading from a slave (master-receiver), the slave transmits (slave-transmitter) a byte of data to the master, and the master then acknowledges the transaction with the ACK pulse. This behavior is illustrated in Figure 4-72.This transaction continues until the master terminates the transmission by not acknowledging (NAK) the transaction after the last byte is received, and then the master issues a STOP condition or addresses another slave after issuing a RESTART condition. This behavior is illustrated in Figure 4-73.

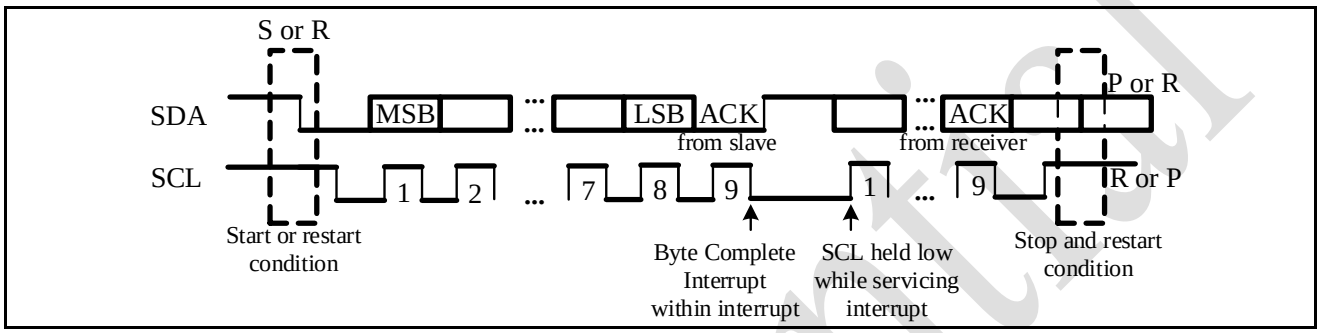


Figure 4-72 Data transfer on the I2C Bus for Master Transmitter

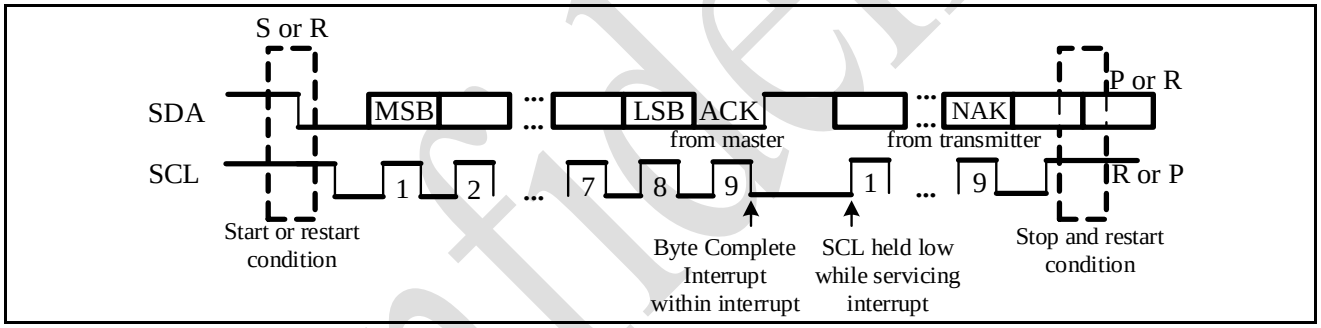


Figure 4-73 Data transfer on the I2C Bus for Master Receiver

The I2C is a synchronous serial interface. The SDA line is a bidirectional signal and changes only while the SCL line is low, except for STOP, START, and RESTART conditions. The output drivers are open-drain or open-collector to perform wire-AND functions on the bus. Data is transmitted in byte packages.

When operating as an I2C master, putting data into the transmit FIFO causes the I2C to generate a START condition on the I2C bus. Writing a 1 to IC_DATA_CMD[9] causes the I2C to generate a STOP condition on the I2C bus; a STOP condition is not issued if this bit is not set, even if the transmit FIFO is empty.

When operating as a slave, the I2C does not generate START and STOP conditions, as per the protocol. However, if a read request is made to the I2C, it holds the SCL line low until read data has been supplied to it. This stalls the I2C bus until read data is provided to the slave I2C module, or the I2C slave is disabled by writing a 0 to bit 0 of the IC_ENABLE register.

The I2C supports mixed read and write combined format transactions in both 7-bit and 10-bit addressing modes.

The I2C does not support mixed address and mixed address format—that is, a 7-bit address transaction followed by a 10-bit address transaction or vice versa—combined format transactions.

To initiate combined format transfers, IC_RESTART_EN should be set to 1. With this value set and operating as a master, when the I2C completes an I2C transfer, it checks the transmit FIFO and executes the next transfer. If the direction of this transfer differs from the previous transfer, the combined format is used to issue the transfer. If the transmit FIFO is empty when the current I2C transfer completes, IC_DATA_CMD[9] is checked:

- If set to 1, a STOP bit is issued.
- If set to 0, the SCL is held low until the next command is written to the transmit FIFO.

4.10.4.2 I2C Protocols

Normally, a standard communication consists of four parts:

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

4.10.4.2.1. START and STOP Conditions

When the bus is idle, both the SCL and SDA signals are pulled high through external pull-up resistors on the bus. When the master wants to start a transmission on the bus, the master issues a START condition. This is defined to be a high-to-low transition of the SDA signal while SCL is 1. When the master wants to terminate the transmission, the master issues a STOP condition. This is defined to be a low-to-high transition of the SDA line while SCL is 1. Figure 4-74 shows the timing of the START and STOP conditions. When data is being transmitted on the bus, the SDA line must be stable when SCL is 1.

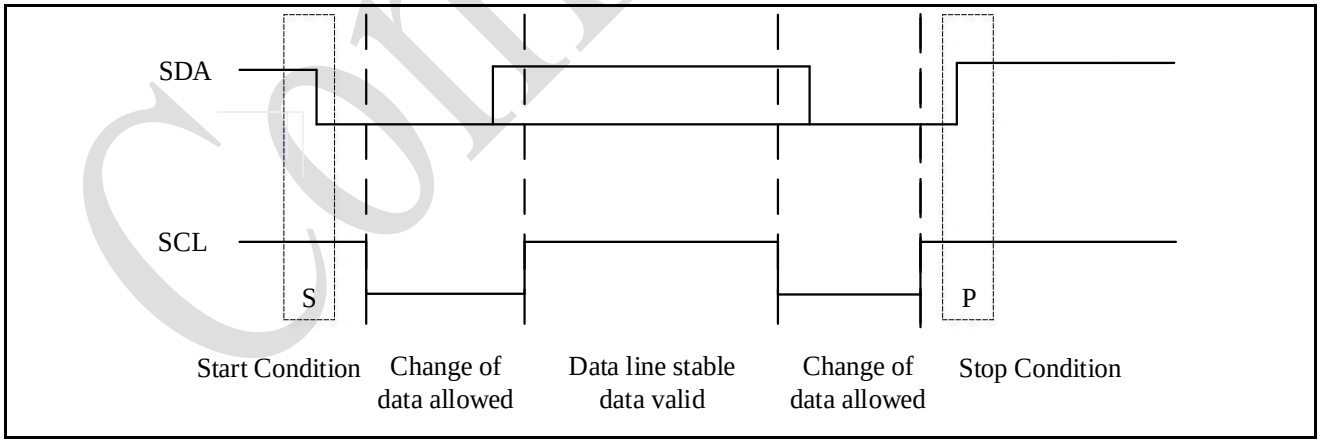


Figure 4-74 START and STOP Condition

4.10.4.2.2. Addressing Slave Protocol

There are two address formats: the 7-bit address format and the 10-bit address format.

- 7-bit Address Format:

During the 7-bit address format, the first seven bits (bits 7:1) of the first byte set the slave address and the LSB bit (bit 0) is the R/W bit as shown in Figure 4-75. When bit 0 (R/W) is set to 0, the master writes to the slave. When bit 0 (R/W) is set to 1, the master reads from the slave.

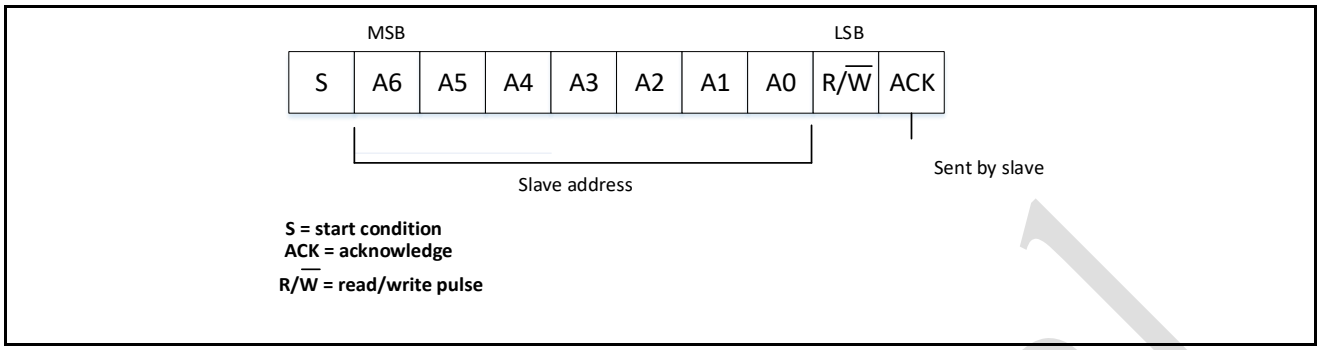


Figure 4-75 7-bit Address Format

■ 10-bit Address Format:

During 10-bit addressing, two bytes are transferred to set the 10-bit address. The transfer of the first byte contains the following bit definition. The first five bits (bits 7:3) notify the slaves that this is a 10-bit transfer followed by the next two bits (bits 2:1), which set the slaves address bits 9:8, and the LSB bit (bit 0) is the R/W bit. The second byte transferred sets bits 7:0 of the slave address. Figure 4-76 shows the 10-bit address format.

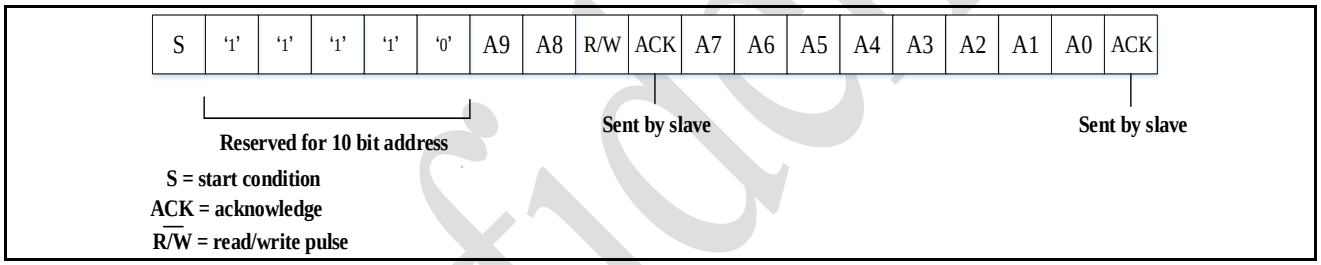


Figure 4-76 10-bit Address Format

Table 4-20 I2C Definition of Bits in First Byte

Slave Address	R/W Bit	Description
0000 000	0	General Call Address. I2C places the data in the receive buffer and issues a General Call interrupt.
0000 000	1	START Byte.
0000 001	X	CBUS address. Our I2C ignores these accesses.
0000 010	X	Reserved
0000 011	X	Reserved
0000 1XX	X	High-speed master code
1111 1XX	X	Reserved

1111 0×X	X	10-bit slave addressing
0001 000	X	Reserved
0001 100	X	Reserved
1100 001	X	Reserved

This I2C does not restrict you from using these reserved addresses. However, if you use these reserved addresses, you may run into incompatibilities with other I2C components.

4.10.4.2.3. Transmitting and Receiving Protocol

The master can initiate data transmission and reception to/from the bus, acting as either a master-transmitter or master-receiver. A slave responds to requests from the master to either transmit data or receive data to/from the bus, acting as either a slave-transmitter or slave-receiver, respectively

- Master-Transmitter and Slave-Receiver

All data is transmitted in byte format, with no limit on the number of bytes transferred per data transfer. After the master sends the address and R/W bit or the master transmits a byte of data to the slave, the slave-receiver must respond with the acknowledge signal (ACK). When a slave-receiver does not respond with an ACK pulse, the master aborts the transfer by issuing a STOP condition. The slave must leave the SDA line high so that the master can abort the transfer.

If the master-transmitter is transmitting data as shown in Figure 4-77, then the slave-receiver responds to the master-transmitter with an acknowledge pulse after every byte of data is received.

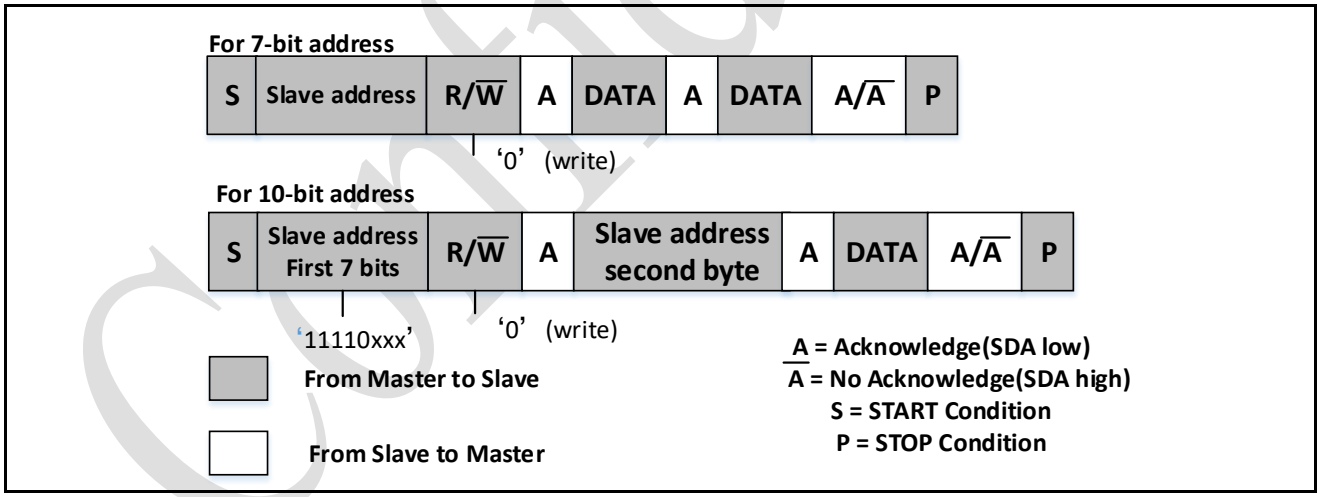


Figure 4-77 Master-Transmitter Protocol

- Master-Receiver and Slave-Transmitter

If the master is receiving data as shown in Figure 4-78, then the master responds to the slave-transmitter with an acknowledge pulse after a byte of data has been received, except for the last byte. This is the way the master-receiver notifies the slave-transmitter that this is the last byte. The slave-transmitter relinquishes the SDA line after detecting the No Acknowledge (NACK) so that the master can issue a STOP condition.

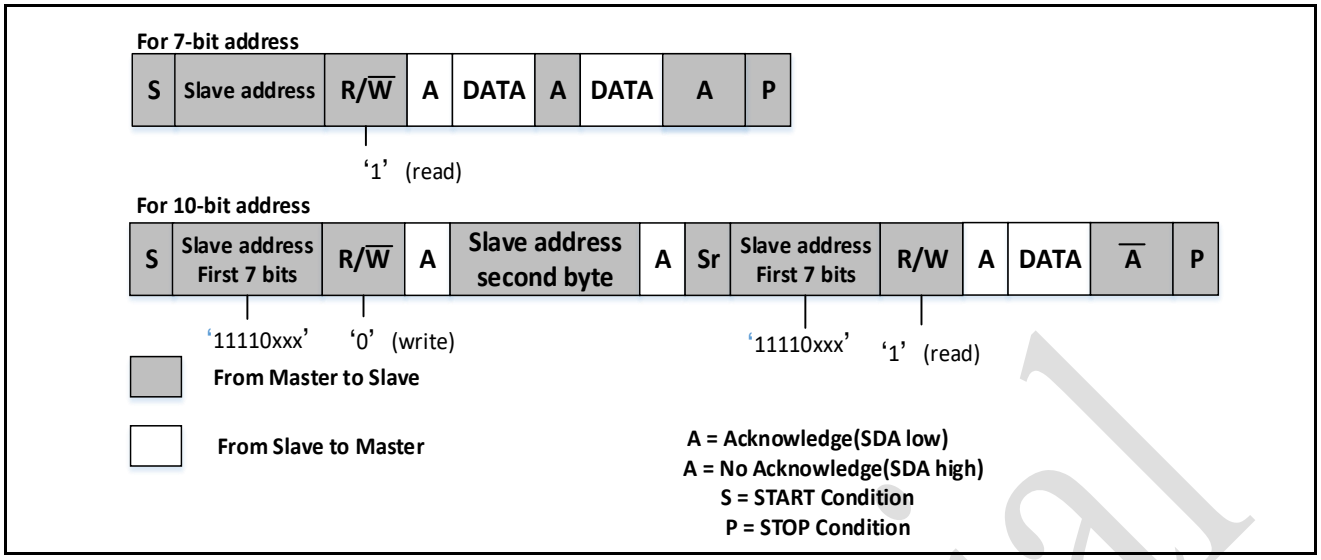


Figure 4-78 Master-Receiver Protocol

When a master does not want to relinquish the bus with a STOP condition, the master can issue a RESTART condition. This is identical to a START condition except it occurs after the ACK pulse. Operating in master mode, the I2C can then communicate with the same slave using a transfer of a different direction.

- START BYTE Transfer Protocol:

The START BYTE transfer protocol is set up for systems that do not have an on-board dedicated I2C hardware module. When the I2C is addressed as a slave, it always samples the I2C bus at the highest speed supported so that it never requires a START BYTE transfer. However, when I2C is a master, it supports the generation of START BYTE transfers at the beginning of every transfer in case a slave device requires it.

This protocol consists of seven zeros being transmitted followed by a 1, as illustrated in Figure 4-79. This allows the processor that is polling the bus to under-sample the address phase until 0 is detected. Once the microcontroller detects a 0, it switches from the under sampling rate to the correct rate of the master.

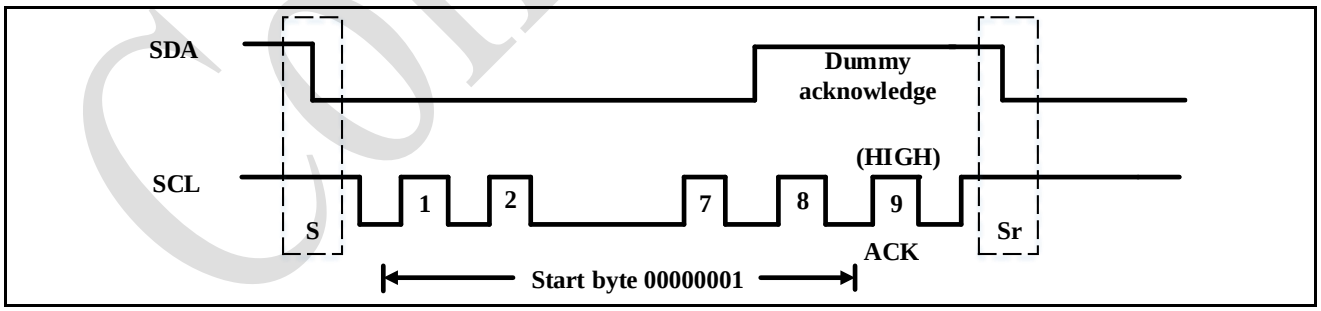


Figure 4-79 START BYTE Transfer

The START BYTE procedure is as follows:

1. Master generates a START condition.
2. Master transmits the START byte (0000 0001).
3. Master transmits the ACK clock pulse. (Present only to conform with the byte handling format used on the bus)
4. No slave sets the ACK signal to 0.
5. Master generates a RESTART (R) condition.

A hardware receiver does not respond to the START BYTE because it is a reserved address and resets after the RESTART condition is generated.

4.10.4.3 Tx FIFO Management and START, STOP and RESTART Generation

When I2C operates as a master and IC_EMPTYFIFO_HOLD_MASTER_EN = 1, the component generates a STOP on the bus whenever the Tx FIFO becomes empty. If RESTART generation capability is enabled, the component generates a RESTART when the direction of the transfer in the Tx FIFO commands changes from Read to Write or vice-versa.

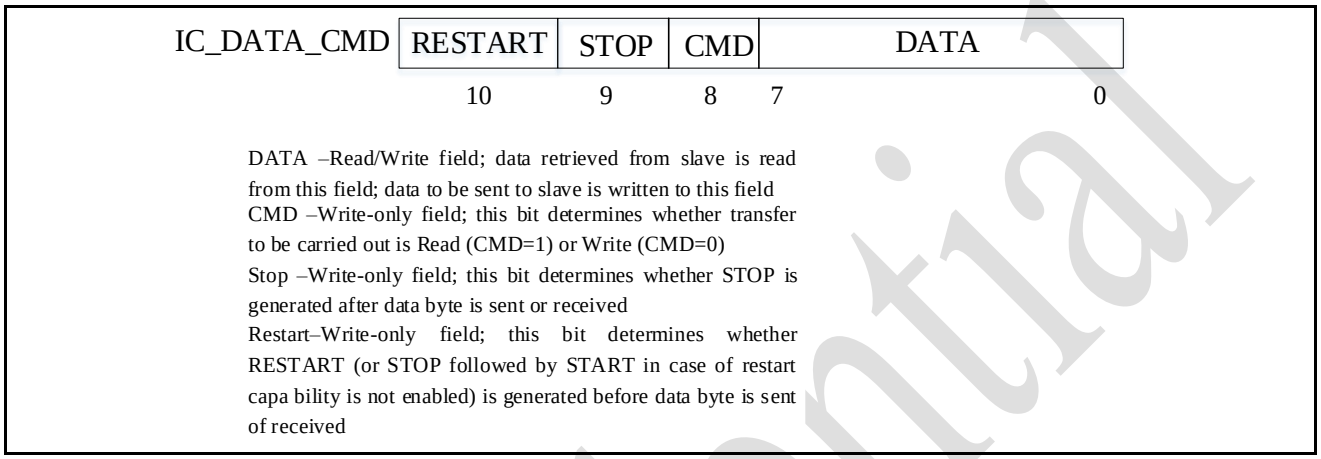


Figure 4-80 IC_DATA_CMD Register if IC_EMPTYFIFO_HOLD_MASTER_EN = 1

4.10.4.4 Multiple Master Arbitration

The I2C bus protocol allows multiple masters to reside on the same bus. If there are two masters on the same I2C-bus, there is an arbitration procedure if both try to take control of the bus at the same time by generating a START condition at the same time. Once a master (for example, a microcontroller) has control of the bus, no other master can take control until the first master sends a STOP condition and places the bus in an idle state.

Arbitration takes place on the SDA line, while the SCL line is 1. The master, which transmits a 1 while the other master transmits 0, loses arbitration and turns off its data output stage. The master that lost arbitration can continue to generate clocks until the end of the byte transfer. If both masters are addressing the same slave device, the arbitration could go into the data phase.

Upon detecting that it has lost arbitration to another master, the I2C will stop generating SCL.

Figure 4-81 illustrates the timing of when two masters are arbitrating on the bus.

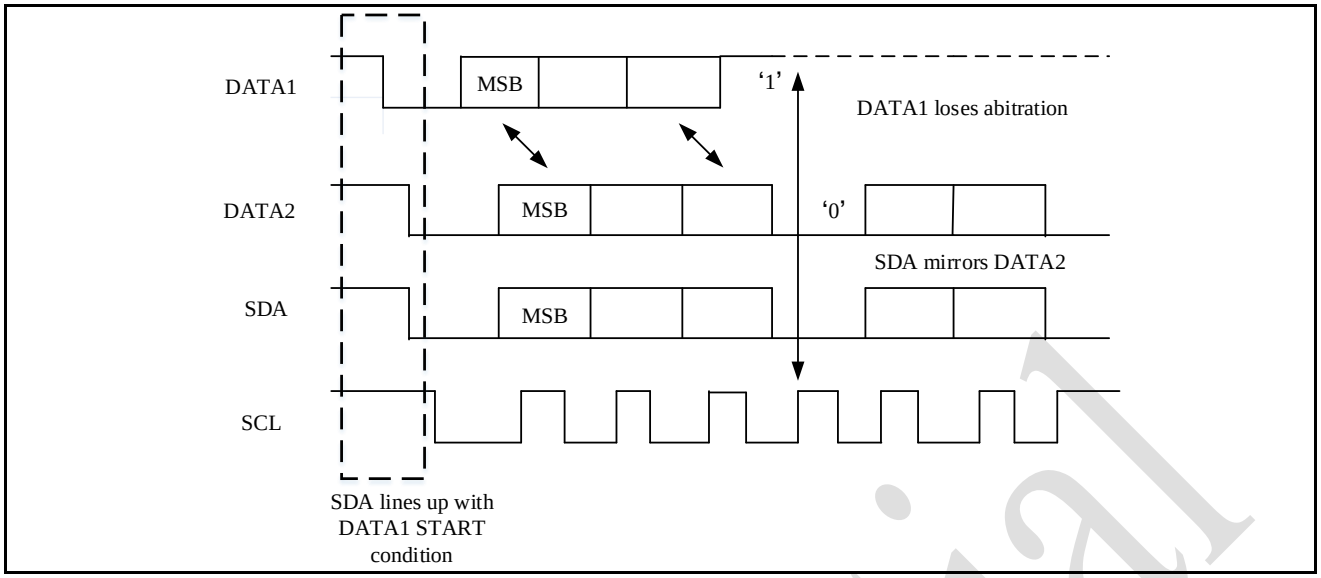


Figure 4-81 Multiple Master Arbitration

For high-speed mode, the arbitration cannot go into the data phase because each master is programmed with a unique high-speed master code. This 8-bit code is defined by the system designer and is set by writing to the High Speed Master Mode Code Address Register, IC_HS_MADDR. Because the codes are unique, only one master can win arbitration, which occurs by the end of the transmission of the high-speed master code.

Control of the bus is determined by address or master code and data sent by competing masters, so there is no central master nor any order of priority on the bus.

Arbitration is not allowed between the following conditions:

- A RESTART condition and a data bit
- A STOP condition and a data bit
- A RESTART condition and a STOP condition
- Slaves are not involved in the arbitration process.

4.10.4.5 Clock Synchronization

When two or more masters try to transfer information on the bus at the same time, they must arbitrate and synchronize the SCL clock. All masters generate their own clock to transfer messages. Data is valid only during the high period of SCL clock. Clock synchronization is performed using the wired-AND connection to the SCL signal. When the master transitions the SCL clock to 0, the master starts counting the low time of the SCL clock and transitions the SCL clock signal to 1 at the beginning of the next clock period. However, if another master is holding the SCL line to 0, then the master goes into a HIGH wait state until the SCL clock line transitions to 1.

All masters then count off their high time, and the master with the shortest high time transitions the SCL line to 0. The masters then counts out their low time and the one with the longest low time forces the other master into a HIGH wait state. Therefore, a synchronized SCL clock is generated, which is illustrated in Figure 4-82. Optionally, slaves may hold the SCL line low to slow down the timing on the I2C bus.

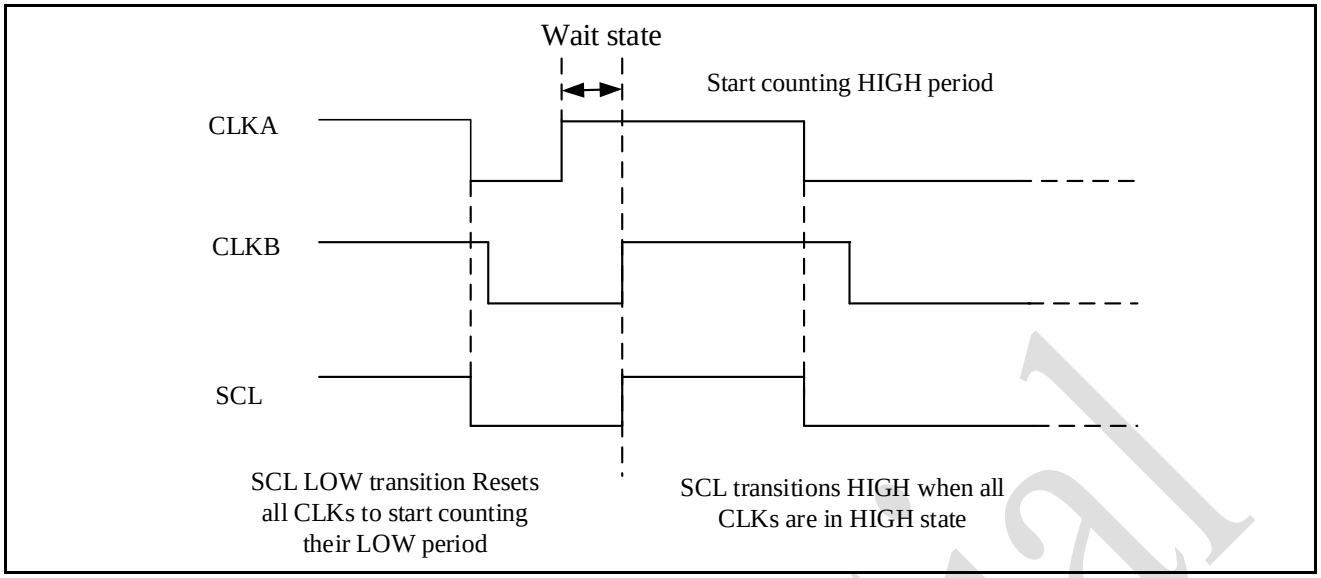


Figure 4-82 Multi-Master Clock Synchronization

4.10.4.6 Operation Modes

This section provides information on operation modes.

4.10.4.6.1. Slave Mode Operation

■ Initial Configuration

To use the I2C as a slave, perform the following steps:

1. Disable the I2C by writing a '0' to bit 0 of the IC_ENABLE register.

2. Write to the IC_SAR register (bits 9:0) to set the slave address. This is the address to which the I2C responds.

3. Write to the IC_CON register to specify which type of addressing is supported (7- or 10-bit by setting bit 3). Enable the I2C in slave-only mode by writing a '0' into bit 6 (IC_SLAVE_DISABLE) and a '0' to bit 0 (MASTER_MODE).

4. Enable the I2C by writing a '1' in bit 0 of the IC_ENABLE register.

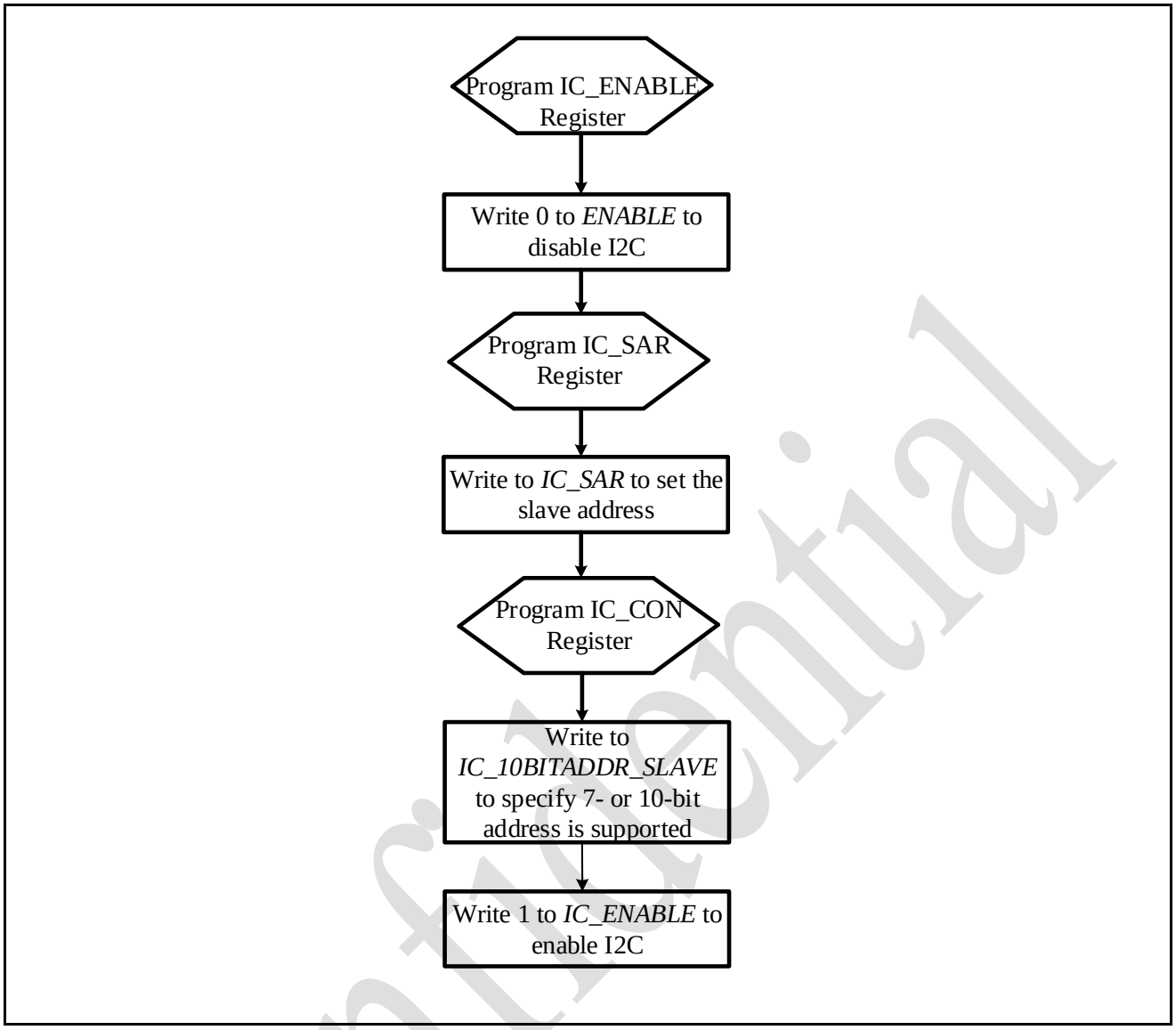


Figure 4-83 Initial Configuration

■ Slave-Transmitter Operation for a Single Byte

When another I2C master device on the bus addresses the I2C and requests data, the I2C acts as a slave-transmitter and the following steps occur:

1. The other I2C master device initiates an I2C transfer with an address that matches the slave address in the IC_SAR register of the I2C.

2. The I2C acknowledges the sent address and recognizes the direction of the transfer to indicate that it is acting as a slave-transmitter.

3. The I2C asserts the RD_REQ interrupt (bit 5 of the IC_RAW_INTR_STAT register) and holds the SCL line low. It is in a wait state until software responds. If the RD_REQ interrupt has been masked, due to IC_INTR_MASK[5] register (M_RD_REQ bit field) being set to 0, then it is recommended that a hardware and/or software timing routine be used to instruct the CPU to perform periodic reads of the IC_RAW_INTR_STAT register.

a. Reads that indicate IC_RAW_INTR_STAT[5] (R_RD_REQ bit field) being set to 1 must be treated as the equivalent of the RD_REQ interrupt being asserted.

b. Software must then act to satisfy the I2C transfer.

c. The timing interval used should be in the order of 10 times the fastest SCL clock period the I2C can handle. For example, for 400 kb/s, the timing interval is 25us.

4. If there is any data remaining in the Tx FIFO before receiving the read request, then the I2C asserts a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register) to flush the old data from the TX FIFO.

If the TX_ABRT interrupt has been masked, due to of IC_INTR_MASK[6] register (M_TX_ABRT bit field) being set to 0, then it is recommended that re-using the timing routine (described in the previous step), or a similar one, be used to read the IC_RAW_INTR_STAT register.

a. Reads that indicate bit 6 (R_TX_ABRT) being set to 1 must be treated as the equivalent of the TX_ABRT interrupt being asserted.

b. There is no further action required from software.

c. The timing interval used should be similar to that described in the previous step for the IC_RAW_INTR_STAT[5] register.

5. Software writes to the IC_DATA_CMD register with the data to be written (by writing a '0' in bit 8).

6. Software must clear the RD_REQ and TX_ABRT interrupts (bits 5 and 6, respectively) of the IC_RAW_INTR_STAT register before proceeding. If the RD_REQ and/or TX_ABRT interrupts have been masked, then clearing of the IC_RAW_INTR_STAT register will have already been performed when either the R_RD_REQ or R_TX_ABRT bit has been read as 1.

7. The I2C releases the SCL and transmits the byte.

8. The master may hold the I2C bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

- Slave-Receiver Operation for a Single Byte

When another I2C master device on the bus addresses the I2C and is sending data, the I2C acts as a slave-receiver and the following steps occur:

1. The other I2C master device initiates an I2C transfer with an address that matches the I2C's slave address in the IC_SAR register.

2. The I2C acknowledges the sent address and recognizes the direction of the transfer to indicate that the I2C is acting as a slave-receiver.

3. I2C receives the transmitted byte and places it in the receive buffer.

4. I2C asserts the RX_FULL interrupt (IC_RAW_INTR_STAT[2] register). If the RX_FULL interrupt has been masked, due to setting IC_INTR_MASK[2] register to 0 or setting IC_TX_TL to a value larger than 0, then it is recommended that a timing routine be implemented for periodic reads of the IC_STATUS register. Reads of the IC_STATUS register, with bit 3 (RFNE) set at 1, must then be treated by software as the equivalent of the RX_FULL interrupt being asserted.

5. Software may read the byte from the IC_DATA_CMD register (bits 7:0).

6. The other master device may hold the I2C bus by issuing a RESTART condition, or release the

bus by issuing a STOP condition.

- Slave-Transfer Operation For Bulk Transfers

In the standard I2C protocol, all transactions are single byte transactions and the programmer responds to a remote master read request by writing one byte into the slave's TX FIFO. When a slave (slave-transmitter) is issued with a read request (RD_REQ) from the remote master (master-receiver), at a minimum there should be at least one entry placed into the slave-transmitter's TX FIFO. I2C is designed to handle more data in the TX FIFO so that subsequent read requests can take that data without raising an interrupt to get more data. Ultimately, this eliminates the possibility of significant latencies being incurred between raising the interrupt for data each time had there been a restriction of having only one entry placed in the TX FIFO.

This mode only occurs when I2C is acting as a slave-transmitter. If the remote master acknowledges the data sent by the slave-transmitter and there is no data in the slave's TX FIFO, the I2C holds the I2C SCL line low while it raises the read request interrupt (RD_REQ) and waits for data to be written into the TX FIFO before it can be sent to the remote master.

If the RD_REQ interrupt is masked, due to bit 5 (M_RD_REQ) of the IC_INTR_STAT register being set to 0, then it is recommended that a timing routine be used to activate periodic reads of the IC_RAW_INTR_STAT register. Reads of IC_RAW_INTR_STAT that return bit 5 (R_RD_REQ) set to 1 must be treated as the equivalent of the RD_REQ interrupt referred to in this section. T

The RD_REQ interrupt is raised upon a read request, and like interrupts, must be cleared when exiting the interrupt service handling routine (ISR). The ISR allows you to either write 1 byte or more than 1 byte into the Tx FIFO. During the transmission of these bytes to the master, if the master acknowledges the last byte. then the slave must raise the RD_REQ again because the master is requesting for more data.

If the programmer knows in advance that the remote master is requesting a packet of n bytes, then when another master addresses I2C and requests data, the Tx FIFO could be written with n number bytes and the remote master receives it as a continuous stream of data. For example, the I2C slave continues to send data to the remote master as long as the remote master is acknowledging the data sent and there is data available in the Tx FIFO. There is no need to hold the SCL line low or to issue RD_REQ again.

If the remote master is to receive n bytes from the I2C but the programmer wrote a number of bytes larger than n to the Tx FIFO, then when the slave finishes sending the requested n bytes, it clears the Tx FIFO and ignores any excess bytes.

The the I2C generates a transmit abort (TX_ABRT) event to indicate the clearing of the Tx FIFO in this example. At the time an ACK/NACK is expected, if a NACK is received, then the remote master has all the data it wants. At this time, a flag is raised within the slave's state machine to clear the leftover data in the Tx FIFO. This flag is transferred to the processor bus clock domain where the FIFO exists and the contents of the Tx FIFO is cleared at that time.

4.10.4.6.2. Master Mode Operation

- Initial Configuration

The initial configuration procedure for Master Mode Operation depends on the configuration parameter I2C_DYNAMIC_TAR_UPDATE. When set to "Yes" (1), the target address and address format can be changed dynamically without having to disable I2C. This parameter only applies to when I2C is acting as a master because the slave requires the component to be disabled before any

changes can be made to the address.

The procedures are very similar and are only different with regard to where the IC_10BIT-ADDR_MASTER bit is set (either bit 4 of IC_CON register or bit 12 of IC_TAR register).

To use the I2C as a master when the I2C_DYNAMIC_TAR_UPDATE configuration parameter is set to “Yes” (1), perform the following steps:

1. Disable the I2C by writing 0 to bit 0 of the IC_ENABLE register.

2. Write to the IC_CON register to set the maximum speed mode supported for slave operation (bits2:1) and to specify whether the I2C starts its transfers in 7/10 bit addressing mode when thedevice is a slave (bit 3).

3. Write to the IC_TAR register the address of the I2C device to be addressed. It also indicates whether a General Call or a START BYTE command is going to be performed by I2C. The desired speed of the I2C master-initiated transfers, either 7-bit or 10-bit addressing, is controlled by the IC_10BITADDR_MASTER bit field (bit 12).

4. Only applicable for high-speed mode transfers. Write to the IC_HS_MADDR register the desired master code for the I2C. The master code is programmer-defined.

5. Enable the I2C by writing a 1 to bit 0 of the IC_ENABLE register.

6. Now write the transfer direction and data to be sent to the IC_DATA_CMD register. If the IC_DATA_CMD register is written before the I2C is enabled, the data and commands are lost as the buffers are kept cleared when I2C is not enabled.

- Dynamic IC_TAR or IC_10BITADDR_MASTER Update

The I2C supports dynamic updating of the IC_TAR (bits 9:0) and IC_10BITADDR_MASTER (bit 12) bit fields of the IC_TAR register. In order to perform a dynamic update of the IC_TAR register, the I2C_DYNAMIC_TAR_UPDATE configuration parameter must be set to Yes (1). You can dynamically write to the IC_TAR register provided the software ensures that there are no other commands in the Tx FIFO that use the existing TAR address. If the software does not ensure this, then IC_TAR should be re-programmed only if the following conditions are met:

  - I2C is not enabled (IC_ENABLE[0]=0);

  OR

  - I2C is enabled (IC_ENABLE[0]=1); AND
  - I2C is NOT engaged in any Master (tx, rx) operation (IC_STATUS[5]=0);

  AND

  - I2C is enabled to operate in Master mode (IC_CON[0]=1);

  AND

  - there are NO entries in the Tx FIFO (IC_STATUS[2]=1)

- Master Transmit and Master Receive

The I2C supports switching back and forth between reading and writing dynamically. To transmit data, write the data to be written to the lower byte of the I2C Rx/Tx Data Buffer and Command Register (IC_DATA_CMD). The CMD bit [8] should be written to 0 for I2C write operations. Subsequently, a read command may be issued by writing “don’t cares” to the lower byte of the IC_DATA_CMD register, and a 1 should be written to the CMD bit. The I2C master continues to initiate transfers as long as there are commands present in the transmit FIFO. If the transmit FIFO becomes empty—depending on the value of IC_EMPTYFIFO_HOLD_MASTER_EN, the master either inserts a STOP condition after completing the current transfers, or it checks to see if

IC_DATA_CMD[9] is set to 1.

- If set to 1, it issues a STOP condition after completing the current transfer.
- If set to 0, it holds SCL low until next command is written to the transmit FIFO.

4.10.4.6.3. Disabling I2C

The register IC_ENABLE_STATUS is added to allow software to unambiguously determine when the hardware has completely shut down in response to bit 0 of the IC_ENABLE register being set from 1 to 0.

Only one register is required to be monitored, as opposed to monitoring two registers (IC_STATUS and IC_RAW_INTR_STAT) which is a requirement for I2C versions 1.05a or earlier.

1. Define a timer interval (ti2c_poll) equal to the 10 times the signaling period for the highest I2C transfer speed used in the system and supported by I2C. For example, if the highest I2C transfer mode is 400 kb/s, then this ti2c_poll is 25us.

2. Define a maximum time-out parameter, MAX_T_POLL_COUNT, such that if any repeated polling operation exceeds this maximum value, an error is reported.

3. Execute a blocking thread/process/function that prevents any further I2C master transactions to be started by software, but allows any pending transfers to be completed.

4. The variable POLL_COUNT is initialized to zero.

5. Set bit 0 of the IC_ENABLE register to 0.

6. Read the IC_ENABLE_STATUS register and test the IC_EN bit (bit 0). Increment POLL_COUNT by one. If $POLL\_COUNT \geq MAX\_T\_POLL\_COUNT$, exit with the relevant error code.

7. If IC_ENABLE_STATUS[0] is 1, then sleep for ti2c_poll and proceed to the previous step. Otherwise, exit with a relevant success code.

4.10.4.6.4. Aborting I2C Transfers

The ABORT control bit of the IC_ENABLE register allows the software to relinquish the I2C bus before completing the issued transfer commands from the Tx FIFO. In response to an ABORT request, the controller issues the STOP condition over the I2C bus, followed by Tx FIFO flush. Aborting the transfer is allowed only in master mode of operation.

1. Stop filling the Tx FIFO (IC_DATA_CMD) with new commands.

2. When operating in DMAC mode, disable the transmit DMAC by setting TDMACE to 0.

3. Set bit 1 of the IC_ENABLE register (ABORT) to 1.

4. Wait for the M_TX_ABRT interrupt.

5. Read the IC_TX_ABRT_SOURCE register to identify the source as ABRT_USER_ABRT.

4.10.4.7 Spike Suppression

The I2C contains programmable spike suppression logic that match requirements imposed by the I2C Bus Specification for SS/FS and HS modes.

This logic is based on counters that monitor the input signals (SCL and SDA), checking if they remain stable for a predetermined amount of ic_clk cycles before they are sampled internally. There is one separate counter for each signal (SCL and SDA). The number of ic_clk cycles can be programmed by the user and should be calculated taking into account the frequency of ic_clk and the relevant spike length specification.

Each counter is started whenever its input signal changes its value. Depending on the behavior of the input signal, one of the following scenarios occurs:

- The input signal remains unchanged until the counter reaches its count limit value. When this happens, the internal version of the signal is updated with the input value, and the counter is reset and stopped. The counter is not restarted until a new change on the input signal is detected.
- The input signal changes again before the counter reaches its count limit value. When this happens, the counter is reset and stopped, but the internal version of the signal is not updated. The counter remains stopped until a new change on the input signal is detected.

The timing diagram in Figure 4-84 illustrates the behavior described above.

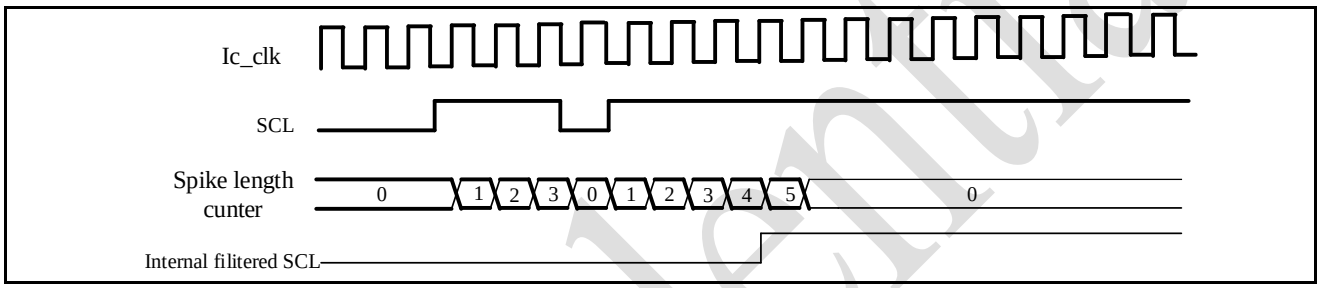


Figure 4-84 Spike Suppression Example

The count limit value used in this example is 5 and was calculated for a 10 ns ic_clk period and for SS/FS operation (50 ns spike suppression).

The I2C Bus Specification calls for different maximum spike lengths according to the operating mode—50 ns for SS and FS; 10 ns for HS, so two registers are required to store the values needed for each case:

- Register IC_FS_SPKLEN holds the maximum spike length for SS and FS/FM+ modes
- Register IC_HS_SPKLEN holds the maximum spike value for HS mode.

These registers are 8 bits wide and accessible through the APB interface for read and write purposes; however, they can be written to only when the I2C is disabled. The minimum value that can be programmed into these registers is 1; attempting to program a value smaller than 1 results in the value 1 being written.

4.10.4.8 Fast Mode Plus Operation

In fast mode plus, the I2C allows the fast mode operation to be extended to support speeds up to 1000 Kb/s. To enable the I2C for fast mode plus operation, perform the following steps before initiating any data transfer:

1. Configure the Maximum Speed mode of I2C Master or Slave to Fast Mode or High Speed mode.

2. ic_clk frequency should greater than or equal to 32 MHz

3. Program the IC_CON register [2:1] = 2’b10 for fast mode or fast mode plus.

4. Program IC_FS_SCL_LCNT and IC_FS_SCL_HCNT registers to meet the fast mode plus SCL.

5. Program the IC_FS_SPKLEN register to suppress the maximum spike of 50ns.

6. Program the IC_SDA_SETUP register to meet the minimum data setup time (tSU; DAT)

4.10.4.9 IC_CLK Frequency Configuration

When the I2C is configured as a Standard (SS), Fast (FS)/Fast-Mode Plus (FM+), or High Speed (HS) master, the *CNT registers must be set before any I2C bus transaction can take place in order to ensure proper I/O timing.

When the I2C operates as an I2C master, in both transmit and receive transfers:

- IC_SS_SCL_LCNT and IC_FS_SCL_LCNT register values must be larger than IC_FS_SPKLEN + 7.
- IC_SS_SCL_HCNT and IC_FS_SCL_HCNT register values must be larger than IC_FS_SPKLEN + 5.
- If the component is programmed to support HS, IC_HS_SCL_LCNT register value must be larger than IC_HS_SPKLEN + 7.
- If the component is programmed to support HS, IC_HS_SCL_HCNT register value must be larger than IC_HS_SPKLEN + 5.

Details regarding the I2C high and low counts are as follows:

- The minimum value of IC_*_SPKLEN + 7 for the *_LCNT registers is due to the time required for the I2C to drive SDA after a negative edge of SCL.
- The minimum value of IC_*_SPKLEN + 5 for the *_HCNT registers is due to the time required for the I2C to sample SDA during the high period of SCL.
- The I2C adds one cycle to the programmed *_LCNT value in order to generate the low period of the SCL clock; this is due to the counting logic for SCL low counting to (*_LCNT + 1).
- The I2C adds IC_*_SPKLEN + 7 cycles to the programmed *_HCNT value in order to generate the high period of the SCL clock; this is due to the following factors:
  - The counting logic for SCL high counts to (*_HCNT+1).
  - The digital filtering applied to the SCL line incurs a delay of SPKLEN + 2 ic_clk cycles, where SPKLEN is:
    - IC_FS_SPKLEN if the component is operating in SS or FS
    - IC_HS_SPKLEN if the component is operating in HS.
  - This filtering includes metastability removal and the programmable spike suppression on SDA and SCL edges.
  - Whenever SCL is driven 1 to 0 by the I2C—that is, completing the SCL high time—an internal logic latency of three ic_clk cycles is incurred. Consequently, the minimum SCL low time of which the I2C is capable is nine (9) ic_clk periods (7 + 1 + 1), while the minimum SCL high time is thirteen (13) ic_clk periods (6 + 1 + 3 + 3).

4.10.4.10 SDA Hold Time

The I2C protocol specification requires 300ns of hold time on the SDA signal (tHD;DAT) in standard mode and fast mode, and a hold time long enough to bridge the undefined part between logic 1 and logic 0 of the falling edge of SCL in high speed mode and fast mode plus.

The I2C contains a software programmable register (IC_SDA_HOLD) to enable dynamic adjustment of the SDA hold-time. The IC_SDA_HOLD register can be programmed only when the I2C

is disabled (IC_ENABLE[0] = 0).

4.10.4.10.1. SDA Hold Timings in Receiver

When I2C acts as a receiver, according to the I2C protocol, the device should internally hold the SDA line to bridge undefined gap between logic 1 and logic 0 of SCL.

IC_SDA_RX_HOLD can be used to alter the internal hold time which I2C applies to the incoming SDA line. Each value in the IC_SDA_RX_HOLD register represents a unit of one ic_clk period. The minimum value of IC_SDA_RX_HOLD is 0. This hold time is applicable only when SCL is HIGH. The receiver does not extend the SDA after SCL goes LOW internally.

Figure 4-85 shows the I2C as receiver with IC_SDA_RX_HOLD programmed to greater than or equal to 3.

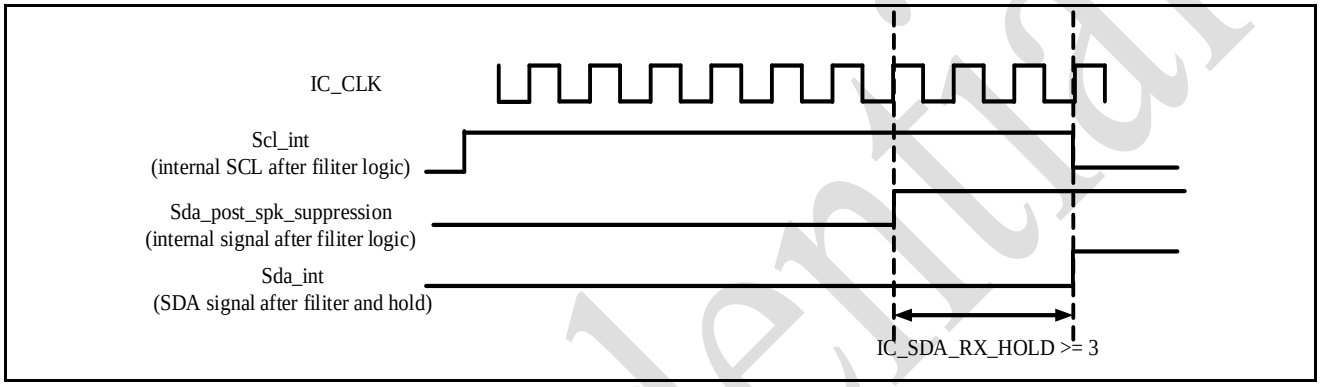


Figure 4-85 I2C receiver with IC_SDA_RX_HOLD

If IC_SDA_RX_HOLD is greater than 3, I2C does not hold SDA beyond 3 ic_clk cycles, because SCL goes LOW internally.

The maximum values of IC_SDA_RX_HOLD that can be programmed in the register for the respective speed modes are derived from the equations show in Table 4-21.

Table 4-21 Maximum Values for IC_SDA_RX_HOLD

Speed Mode	Maximum IC_SDA_RX_HOLD Value
Standard Mode	IC_SS_SCL_HCNT – IC_FS_SPKLEN – 3
Fast Mode or Fast Mode Plus	IC_FS_SCL_HCNT – IC_FS_SPKLEN – 3
High Speed	Min {IC_FS_SCL_HCNT – IC_FS_SPKLEN – 3 , IC_HS_SCL_LCNT – IC_HS_SPKLEN – 3}

4.10.4.10.2. SDA Hold Timings in Transmitter

The IC_SDA_TX_HOLD register can be used to alter the timing of the generated SDA signal by the I2C. Each value in the IC_SDA_TX_HOLD register represents a unit of one ic_clk period. When the I2C is operating in Master Mode, the minimum tHD:DAT timing is one ic_clk period. Therefore even when IC_SDA_TX_HOLD has a value of zero, the I2C will drive SDA one ic_clk cycle after driving SCL to logic 0. For all other values of IC_SDA_TX_HOLD, the following is true:

- Drive on SDA occurs IC_SDA_TX_HOLD ic_clk cycles after driving SCL to logic 0
  - When the I2C is operating in Slave Mode, the minimum tHD:DAT timing is SPKLEN + 7

ic_clk periods, where SPKLEN is:

- IC_FS_SPKLEN if the component is operating in standard mode, fast mode, or fast mode plus
- IC_HS_SPKLEN if the component is operating in high speed mode
  - This delay allows for synchronization and spike suppression on the SCL sample. Therefore, even when IC_SDA_TX_HOLD has a value less than SPKLEN + 7, the I2C drives SDA SPKLEN + 7 ic_clk cycles after SCL has transitioned to logic 0. For all other values of IC_SDA_TX_HOLD, the following is true:
- Drive on SDA occurs IC_SDA_TX_HOLD ic_clk cycles after SCL has transitioned to logic 0.
  - Figure 4-86 shows the tHD:DAT timing generated by the I2C operating in Master Mode when IC_SDA_TX_HOLD = 3.

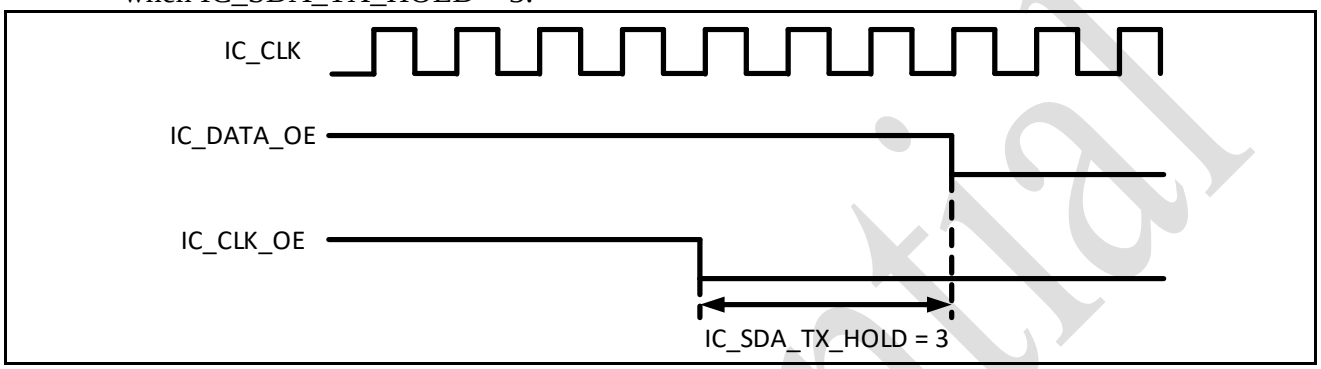


Figure 4-86 I2C Master Implementing tHD;DAT with IC_SDA_HOLD = 3

4.10.4.11 DMA Controller Interface

The I2C has an optional built-in DMA capability that can be selected at configuration time; it has a handshaking interface to a DMA Controller to request and control transfers. The APB bus is used to perform the data transfer to or from the DMA. The specific programing flowchart is shown in Figure 4-87.

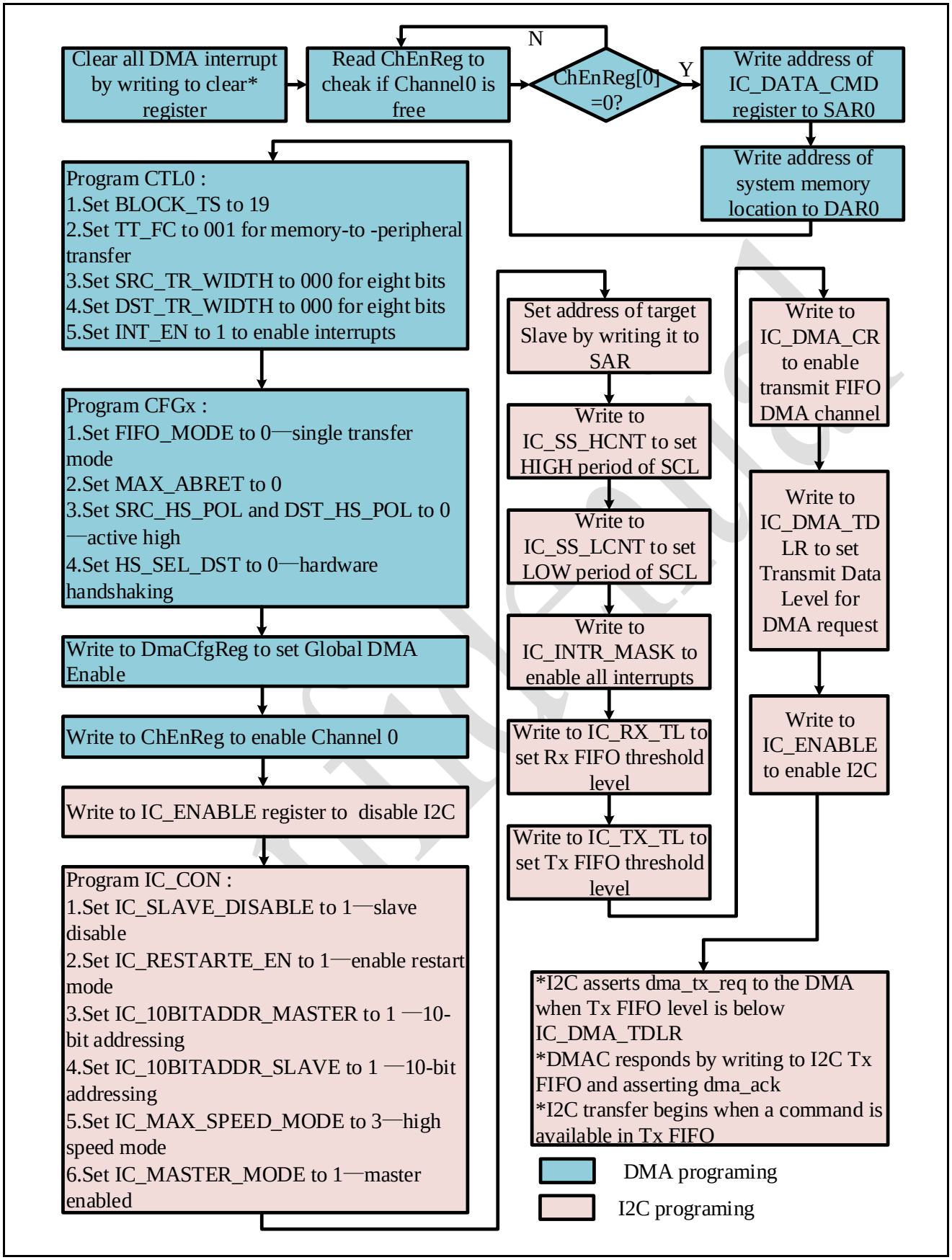


Figure 4-87 Flawchart for DMA and I2C Programing

4.10.5 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: **I2C0_BA = 0×4000_0000** **I2C1_BA = 0×4001_0000**				
IC_CON	I2C_BA+ 0×00	R/W	I2C Control Register	0×0000_007F
IC_TAR	I2C_BA+ 0×04	R/W	I2C Target Address Register	0×0000_1055
IC_SAR	I2C_BA+0×08	R/W	I2C Slave Address Register	0×0000_0055
IC_HS_MADDR	I2C_BA+0×0C	R/W	I2C High Speed Master Mode Code Address Register	0×0000_0001
IC_DATA_CMD	I2C_BA+0×10	R/W	I2C Rx/Tx Data Buffer and Command Register	0×0000_0000
IC_SS_SCL_HCNT	I2C_BA+0×14	R/W	Standard Speed I2C Clock SCL High Count Register	0×0000_0190
IC_SS_SCL_LCNT	I2C _BA+0×18	R/W	Standard Speed I2C Clock SCL Low Count Register	0×0000_01D6
IC_FS_SCL_HCNT	I2C_BA+0×1C	R/W	Fast Speed I2C Clock SCL High Count Register	0×0000_003C
IC_FS_SCL_LCNT	I2C_BA+0×20	R/W	Fast Speed I2C Clock SCL Low Count Register	0×0000_0082
IC_HS_SCL_HCNT	I2C_BA+0×24	R/W	High Speed I2C Clock SCL High Count Register	0×0000_0006
IC_HS_SCL_LCNT	I2C_BA+0×28	R/W	High Speed I2C Clock SCL Low Count Register	0×0000_0010
IC_INTR_STAT	I2C_BA+0×2C	R	I2C Interrupt Status Register	0×0000_0000
IC_INTR_MASK	I2C_BA+0×30	R/W	I2C Interrupt Mask Register	0×0000_08FF
IC_RAW_INTR_STAT	I2C_BA+0×34	R	I2C Raw Interrupt Status Register	0×0000_0000
IC_RX_TL	I2C_BA+0×38	R/W	I2C Receive FIFO Threshold Register	0×0000_0000
IC_TX_TL	I2C_BA+0×3C	R/W	I2C Transmit FIFO Threshold Register	0×0000_0000
IC_CLR_INTR	I2C_BA+ 0×40	R	Clear Combined and Individual Interrupt Register	0×0000_0000
IC_CLR_RX_UNDER	I2C_BA+0×44	R	Clear RX_UNDER Interrupt	0×0000_0000
IC_CLR_RX_OVER	I2C_BA+0×48	R	Clear RX_OVER Interrupt	0×0000_0000
IC_CLR_TX_OVER	I2C_BA+0×4C	R	Clear TX_OVER Interrupt	0×0000_0000
IC_CLR_RD_REQ	I2C_BA+0×50	R	Clear RD_REQ Interrupt	0×0000_0000
IC_CLR_TX_ABRT	I2C_BA+0×54	R	Clear TX_ABRT Interrupt	0×0000_0000
IC_CLR_RX_DONE	I2C_BA+0×58	R	Clear RX_DONE Interrupt	0×0000_0000
IC_CLR_ACTIVITY	I2C_BA+0×5C	R	Clear ACTIVITY Interrupt	0×0000_0000

IC_CLR_STOP_DET	I2C_BA+0×60	R	Clear STOP_DET Interrupt	0×0000_0000
IC_CLR_START_DET	I2C_BA+0×64	R	Clear START_DET Interrupt	0×0000_0000
IC_CLR_GEN_CALL	I2C_BA+0×68	R	Clear GEN_CALL Interrupt	0×0000_0000
IC_ENABLE	I2C_BA+0×6C	R/W	I2C Enable Register	0×0000_0000
IC_STATUS	I2C_BA+0×70	R	I2C Status Register	0×0000_0006
IC_TXFLR	I2C_BA+0×74	R	I2C Transmit Abort Status Register	0×0000_0000
IC_RXFLR	I2C_BA+0×78	R	Receive FIFO Level Register	0×0000_0000
IC_SDA_HOLD	I2C_BA+0×7C	R/W	SDA Hold Time Length Register	0×0000_0001
IC_TX_ABRT_SOURCE	I2C_BA+0×80	R	I2C Transmit Abort Status Register	0×0000_0000
IC_SLV_DATA_NACK_ONLY	I2C_BA+0×84	R/W	Generate SLV_DATA_NACK Register	0×0000_0000
IC_DMA_CR	I2C_BA+0×88	R/W	DMA Control Register for transmit and receive	0×0000_0000
IC_DMA_TDLR	I2C_BA+0×8C	R/W	DMA Transmit Data Level	0×0000_0000
IC_DMA_RDLR	I2C_BA+0×90	R/W	DMA Receive Data Level	0×0000_0000
IC_SDA_SETUP	I2C_BA+0×94	R/W	I2C SDA Setup Register	0×0000_0064
IC_ACK_GENERAL_CALL	I2C_BA+0×98	R/W	I2C ACK General Call Register	0×0000_0001
IC_ENABLE_STATUS	I2C_BA+0×9C	R	I2C Enable Status Register	0×0000_0000
IC_FS_SPKLEN	I2C_BA+0×A0	R/W	I2C SS and FS Spike Suppression Limit Register	0×0000_0005
IC_HS_SPKLEN	I2C_BA+0×A4	R/W	I2C HS Spike Suppression Limit Register	0×0000_0001
IC_CLR_RESTART_DET	I2Cx_BA+0×A8	R	Clear RESTART_DET Interrupt Register	0×0000_0000

4.10.6 Operation of Interrupt Registers

Table 4-22 lists the operation of the I2C interrupt registers and how they are set and cleared. Some bits are set by hardware and cleared by software, whereas other bits are set and cleared by hardware.

Table 4-22 Operation of the Interrupt Register

Interrupt Bit Fields	Set by Hardware/ Cleared by Software	Set and Cleared by Hardware
MST_ON_HOLD	×	√
RESTART_DET	√	×
GEN_CALL	√	×
START_DET	√	×
STOP_DET	√	×
ACTIVITY	√	×
RX_DONE	√	×
TX_ABRT	√	×

RD_REQ	√	×
TX_EMPTY	×	√
TX_OVER	√	×
RX_FULL	×	√
RX_OVER	√	×
RX_UNDER	√	×

4.10.7 I2C Register Description

This section describes the registers listed in Table 6-1. Registers are on the HCLK domain, but status bits reflect actions that occur in the ic_clk domain. Therefore, there is delay when the HCLK register reflects the activity that occurred on the ic_clk side.

Some registers may be written only when the I2C is disabled, programmed by the IC_ENABLE register. Software should not disable the I2C while it is active. If the I2C is in the process of transmitting when it is disabled, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. The slave continues receiving until the remote master aborts the transfer, in which case the I2C could be disabled. Registers that cannot be written to when the I2C is enabled are indicated in their descriptions.

4.10.7.1 I2C Control Register (IC_CON)

This register can be written only when the I2C is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect.

Register	Offset	R/W	Description	Reset Value
IC_CON	I2Cx_BA+0×00	R/W	I2C Control Register	0×0000_007F

Bits	Description	
[31:8]	Reserved	Reserved.
[7]	STOP_DET_IFADDRESSED	In slave mode: 1'b1 – issues the STOP_DET interrupt only when it is addressed. 1'b0 – issues the STOP_DET irrespective of whether it's addressed or not. Dependencies: This register bit value is applicable in the slave mode only (MASTER_MODE = 1'b0) Note: During a general call address, this slave does not issue the STOP_DET interrupt if STOP_DET_IF_ADDRESSED = 1'b1, even if the slaveresponds to the general call address by generating ACK. The STOP_DET interrupt is generated only when thetransmitted address matches the slave address (SAR).
[6]	IC_SLAVE_DISABLE	This bit controls whether I2C has its slave disabled, which means once the presetn signal is applied, then this bit takes on the value of the configuration parameter IC_SLAVE_DISABLE. You have the choice of having the slave enabled or disabled after reset is applied, which means software does not have to configure the slave. By default, theslave is always enabled (in reset state as well). If you

		need to disable it after reset, set this bit to 1. If this bit is set (slave is disabled), I2C functions only as a master and does not perform any action that requires aslave. 0 = slave is enabled 1 = slave is disabled Note: Software should ensure that if this bit is written with ‘0,’ then bit 0 should also be written with a ‘0’.
[5]	IC_RESTART_EN	Determines whether RESTART conditions may be sent whenacting as a master. Some older slaves do not supporthandling RESTART conditions; however, RE-START conditions are used in several I2C operations. 0 = disable 1 = enable When the RESTART is disabled, the I2C master is incapable of performing the following functions: Sending a START BYTE Performing any high-speed mode operation Performing direction changes in combined format mode Performing a read operation with a 10-bit address By replacing RESTART condition followed by a STOP and asubsequent START condition, split operations are brokendown into multiple I2C transfers. If the aboveoperations are performed, it will result in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register.
[4]	IC_10BITADDR_MASTER_rd_only	The function of this bit is handled by bit 12 of IC_TAR register, and becomes a read-only copy called IC_10BITADDR_MASTER_rd_only. 0 = 7-bit addressing 1 = 10-bit addressing
[3]	IC_10BITADDR_SLAVE	When acting as a slave, this bit controls whether the I2C responds to 7- or 10-bit addresses. 0 = 7-bit addressing. The I2C ignores transactionsthat involve 10-bit addressing; for 7-bit addressing, only thelower 7 bits of the IC_SAR register are compared. 1 = 10-bit addressing. The I2C responds to only10-bit addressing transfers that match the full 10 bits of theIC_SAR register.
[2:1]	SPEED	These bits control at which speed the I2C operates. Its setting is relevant only if one is operating the I2C in master mode. Hardware protects against illegal values being programmedby software. These bits must be programmed appropriately for slave mode also, as it is used to capture correct value of spike filter as per the speed mode. This register should be programmed only with a value in the range of 1 to 3; otherwise, hardware updates this register with the value of 3. 1 = standard mode (0 to 100 Kb/s) 2 = fast mode (≤ 400 Kb/s) or fast mode plus (≤ 1000 Kb/s) 3 = high speed mode (≤ 3.4 Mb/s)
[0]	MASTER_MODE	This bit controls whether the I2C master is enabled. 0 = master disabled

		1 = master enabled Note: Software should ensure that if this bit is written with ‘1’, then bit 6 should also be written with a ‘1’.

4.10.7.2 I2C Target Address Register (IC_TAR)

Register	Offset	R/W	Description	Reset Value
IC_TAR x= 0, 1	I2Cx_BA+0×04	R/W	I2C Target Address Register	0×0000_1055

Bits	Description	
[31:13]	Reserved	Reserved.
[12]	IC_10BIT-ADDR_MASTER	This bit controls whether the I2C starts its transfers in 7- or 10-bit addressing mode when acting as a master. 0 = 7-bit addressing 1 = 10-bit addressing
[11]	SPECIAL	This bit indicates whether software performs a Device-ID, General Call or START BYTE command. 0 = ignore bit 10 GC_OR_START and use IC_TAR normally 1 = perform special I2C command as specified in GC_OR_START bit
[10]	GC_OR_START	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a General Call or START BYTE command is to be performed by the I2C. 0 = General Call Address – after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The I2C remains in General Call mode until the SPECIAL bit value (bit 11) is cleared. 1 = START BYTE
[9:0]	IC_TAR	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits. If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only slave.

4.10.7.3 I2C Slave Address Register (IC_SAR)

Register	Offset	R/W	Description	Reset Value
IC_SAR x= 0, 1	I2Cx_BA+0×08	R/W	I2C Slave Address Register	0×0000_0055

Bits	Descriptions

[31:10]	Reserved	Reserved.
[9:0]	IC_SAR	The IC_SAR holds the slave address when the I2C is operating as a slave. For 7-bit addressing, only IC_SAR [6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE [0] being set to 0. Writes at other times have no effect. Note: The default values cannot be any of the reserved address locations: that is, 0×00 to 0×07, or 0×78 to 0×7F. The correct operation of the device is not guaranteed if you program the IC_SAR or IC_TAR to a reserved value.

4.10.7.4 I2C High Speed Master Mode Code Address Register (IC_HS_MADDR)

Register	Offset	R/W	Description	Reset Value
IC_HS_MADDR x= 0, 1	I2Cx_BA+0×0c	R/W	I2C High Speed Master Mode Code Address Register	0×0000_0001

Bits	Descriptions	
[31:3]	Reserved	Reserved.
[2:0]	IC_HS_MADDR	This bit field holds the value of the I2C HS mode master code. HS-mode master codes are reserved 8-bit codes (00001xxx) that are not used for slave addressing or other purposes. Each master has its unique master code; up to eight high speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect.

4.10.7.5 I2C Rx/Tx Data Buffer and Command Register (IC_DATA_CMD)

Register	Offset	R/W	Description	Reset Value
IC_DATA_CMD x= 0, 1	I2Cx_BA+0×10	R/W	I2C Rx/Tx Data Buffer and Command Register	0×0000_0000

Bits	Descriptions	
[31:11]	Reserved	Reserved.
[10]	RESTART	This bit controls whether a RESTART is issued before the byte is sent or received. (only writable) 1 = If IC_RESTART_EN is 1, a RESTART is issued before the data issent/received (according to the value of CMD), regardless of whether ornot the transfer direction is changing from the previous command. 0 = If IC_RESTART_EN is 1, a RESTART is issued only if the transfer direction is changing from the previous command.
[9]	STOP	This bit controls whether a STOP is issued after the byte is sent or received. (only

		writable) 1 = STOP is issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master immediatelytries to start a new transfer by issuing a START and arbitrating for thebus. 0 = STOP is not issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master continues thecurrent transfer by sending/receiving data bytes according to the value ofthe CMD bit. If the Tx FIFO is empty, the master holds the SCL line lowand stalls the bus until a new command is available in the Tx FIFO.
[8]	CMD	This bit controls whether a read or a write is performed. (only writable) This bit does not control the direction when the I2C acts as a slave. It controls only the direction when it acts as a master. 1 = Read 0 = Write When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave-receiver mode, this bit is a “don’t care” because writes to this register are not required. In slave-transmitter mode, a “0” indicates that the data in IC_DATA_CMD is to be transmitted. When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register), unless bit 11 (SPECIAL) in the IC_TAR register has been cleared. If a “1” is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs.
[7:0]	DAT	This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the I2C. However, when you read this register, these bits return the value of data received on the I2C interface.

4.10.7.6 Standard Speed I2C Clock SCL High Count Register (IC_SS_SCL_HCNT)

Register	Offset	R/W	Description	Reset Value
IC_SS_SCL_HCNT x= 0, 1	I2Cx_BA+0×14	R/W	Standard Speed I2C Clock SCL High Count Register	0×0000_0190

Bits	Descriptions	
[31:16]	Reserved	Reserved.
[15:0]	IC_SS_SCL_HCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed. For more information, refer to “IC_CLK Frequency Configuration”.

		This register can be written only when the I2C interface is disabled which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. **Note**: This register must not be programmed to a value higher than 65525,because I2C uses a 16-bit counter to flag an I2C bus idle conditionwhen this counter reaches a value of IC_SS_SCL_HCNT + 10.

4.10.7.7 Standard Speed I2C Clock SCL Low Count Register (IC_SS_SCL_LCNT)

Register	Offset	R/W	Description	Reset Value
IC_SS_SCL_LCNT x= 0, 1	I2Cx_BA+0×18	R/W	Standard Speed I2C Clock SCL Low Count Register	0×0000_01D6

Bits	Descriptions	
[31:16]	Reserved	Reserved.
[15:0]	IC_SS_SCL_LCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. For more information, refer to "IC_CLK Frequency Configuration". This register can be written only when the I2C interface is disabled which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted, results in 8 being set. .

4.10.7.8 Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register (IC_FS_SCL_HCNT)

Register	Offset	R/W	Description	Reset Value
IC_FS_SCL_HCNT x= 0, 1	I2Cx_BA+0×1C	R/W	Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register	0×0000_003C

Bits	Descriptions	
[31:16]	Reserved	Reserved.
[15:0]	IC_FS_SCL_HCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast mode or fast mode plus. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. For more information,refer to "IC_CLK Frequency Configuration". This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect.

		The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set.

4.10.7.9 Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register (IC_FS_SCL_LCNT)

Register	Offset	R/W	Description	Reset Value
IC_FS_SCL_LCNT x= 0, 1	I2Cx_BA+0×20	R/W	Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register	0×0000_0082

Bits	Descriptions	
[31:16]	Reserved	Reserved.
[15:0]	IC_FS_SCL_LCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast mode or fast mode plus. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. For more information, refer to “IC_CLK Frequency Configuration”. This register can be written only when the I2C interface is disabled, whichcorresponds to IC_ENABLE[0] being set to 0. Writes at other times have noeffect. The minimum valid value is 8; hardware prevents values less than this beingwritten, and if attempted results in 8 being set.

4.10.7.10 High Speed I2C Clock SCL High Count Register (IC_HS_SCL_HCNT)

Register	Offset	R/W	Description	Reset Value
IC_HS_SCL_HCNT x= 0, 1	I2Cx_BA+0×24	R/W	High Speed I2C Clock SCL High Count Register	0×0000_0006

Bits	Descriptions	
[31:16]	Reserved	Reserved.
[15:0]	IC_HS_SCL_HCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high period countfor high speed. For more information, refer to “IC_CLK FrequencyConfiguration”. The SCL High time depends on the loading of the bus. For 100pF loading,the SCL High time is 60ns; for 400pF loading, the SCL High time is 120ns. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set.

4.10.7.11 High Speed I2C Clock SCL Low Count Register (IC_HS_SCL_LCNT)

Register	Offset	R/W	Description	Reset Value
IC_HS_SCL_LCNT x= 0, 1	I2Cx_BA+0×28	R/W	High Speed I2C Clock SCL Low Count Register	0×0000_0010

Bits	Descriptions	
[31:16]	Reserved	Reserved.
[15:0]	IC_HS_SCL_LCNT	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period countfor high speed. For more information, refer to "IC_CLK FrequencyConfiguration". The SCL low time depends on the loading of the bus. For 100pF loading, theSCL low time is 160ns; for 400pF loading, the SCL low time is 320ns. This register can be written only when the I2C interface is disabled, whichcorresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. If the value is less than 8 then the count value gets changed to 8.

4.10.7.12 I2C Interrupt Status Register (IC_INTR_STAT)

Register	Offset	R/W	Description	Reset Value
IC_INTR_STAT x= 0, 1	I2Cx_BA+0×2C	R	I2C Interrupt Status Register	0×0000_0000

Bits	Descriptions	
[31:14]	Reserved	Reserved.
[13]	R_MST_ON_HOLD	See IC_RAW_INTR_STAT for a detailed description of this bit.
[12]	Reserved	Reserved
[11]	R_GEN_CALL	See IC_RAW_INTR_STAT for a detailed description of this bit.
[10]	R_START_DET	
[9]	R_STOP_DET	
[8]	R_ACTIVITY	
[7]	R_RX_DONE	
[6]	R_TX_ABRT	
[5]	R_RD_REQ	
[4]	R_TX_EMPTY	
[3]	R_TX_OVER	
[2]	R_RX_FULL	
[1]	R_RX_OVER	
[0]	R_RX_UNDER	

4.10.7.13 I2C Interrupt Mask Register (IC_INTR_MASK)

Register	Offset	R/W	Description	Reset Value
IC_INTR_MASK x= 0, 1	I2Cx_BA+0×30	R/W	I2C Interrupt Mask Register	0×0000_08FF

Bits	Descriptions	
[31:14]	Reserved	Reserved.
[13]	M_MST_ON_HOLD	This bit masks the R_MST_ON_HOLD interrupt bit in the IC_INTR_STAT register.(Read-Only)
[12]	Reserved	Reserved
[11]	M_GEN_CALL	These bits mask their corresponding interrupt status bits in the IC_INTR_STAT register.
[10]	M_START_DET	
[9]	M_STOP_DET	
[8]	M_ACTIVITY	
[7]	M_RX_DONE	
[6]	M_TX_ABRT	
[5]	M_RD_REQ	
[4]	M_TX_EMPTY	
[3]	M_TX_OVER	
[2]	M_RX_FULL	
[1]	M_RX_OVER	
[0]	M_RX_UNDER	

4.10.7.14 I2C Raw Interrupt Status Register (IC_RAW_INTR_STAT)

Register	Offset	R/W	Description	Reset Value
IC_RAW_INTR_STAT x= 0, 1	I2Cx_BA+0×34	R	I2C raw Interrupt Status Register	0×0000_0000

Bits	Descriptions	
[31:14]	Reserved	Reserved.
[13]	MST_ON_HOLD	Indicates whether a master is holding the bus and the Tx FIFO is empty.
[12]	Reserved	Reserved
[11]	GEN_CALL	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling I2C or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. I2C stores the received data in the Rx buffer.
[10]	START_DET	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode.
[9]	STOP_DET	Indicates whether a STOP condition has occurred on the I2C interface regardless of

		whether I2C is operating in slave or master mode. In Slave Mode: If IC_CON[7]=1'b1 (STOP_DET_IFADDRESSED), the STOP_DET interrupt is generated only if the slave is addressed. Note: During a general call address, this slave does not issue a STOP_DET interrupt if STOP_DET_IF_ADDRESSED=1'b1, even if the slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the slave address (SAR). If IC_CON[7]=1'b0 (STOP_DET_IFADDRESSED), the STOP_DET interrupt is issued irrespective of whether it is being addressed. In Master Mode: The STOP_DET interrupt is issued irrespective of whether the master is active.
[8]	ACTIVITY	This bit captures I2C activity and stays set until it is cleared. There are four ways to clear it: Disabling the I2C Reading the IC_CLR_ACTIVITY register Reading the IC_CLR_INTR register System reset Once this bit is set, it stays set unless one of the four methods is used to clearit. Even if the I2C module is idle, this bit remains set until cleared, indicating that there was activity on the bus.
[7]	RX_DONE	When the I2C is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.
[6]	TX_ABRT	This bit indicates if I2C, as an I2C transmitter, is unable to completethe intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a “transmit abort”.When this bit is set to 1, the IC_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. Note: The I2C flushes/resets/empties only the TX_FIFO whenever there is a transmit abort caused by any of the events tracked by the IC_TX_ABRT_SOURCE register. The Tx FIFO remains in this flushed state until the register IC_CLR_TX_ABRT is read. Once this read is performed, theTx FIFO is then ready to accept more data bytes from the APB interface. RX FIFO is also flushed.
[5]	RD_REQ	This bit is set to 1 when I2C is acting as a slave and another I2C master is attempting to read data from I2C. The I2C holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the IC_DATA_CMD register. This bit is set to 0 just after the processor reads the IC_CLR_RD_REQ register.
[4]	TX_EMPTY	The behavior of the TX_EMPTY interrupt status differs based on the TX_EMPTY_CTRL selection in the IC_CON register. When TX_EMPTY_CTRL = 0: This bit is set to 1 when the transmit buffer is at or below the threshold value set in

		the IC_TX_TL register. When TX_EMPTY_CTRL = 1: This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register and the transmission of the address/data from the internal shift register for the most recently popped command is completed. It is automatically cleared by hardware when the buffer level goes above the threshold. When IC_ENABLE[0] is set to 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer any activity, then with ic_en=0, this bit is set to 0.
[3]	TX_OVER	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
[2]	RX_FULL	Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (IC_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once IC_ENABLE[0] is set to 0, regardless of the activity that continues.
[1]	RX_OVER	Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The I2C acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.
[0]	RX_UNDER	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. If the module is disabled(IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared.

4.10.7.15 I2C Receive FIFO Threshold Register (IC_RX_TL)

Register	Offset	R/W	Description	Reset Value
IC_RX_TL x= 0, 1	I2Cx_BA+0×38	R/W	I2C Receive FIFO Threshold Register	0×0000_0000

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[7:0]	RX_TL	Receive FIFO Threshold Level Controls the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in IC_RAW_INTR_STAT register). The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the

		maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries.

4.10.7.16 I2C Transmit FIFO Threshold Register (IC_TX_TL)

Register	Offset	R/W	Description	Reset Value
IC_TX_TL x= 0, 1	I2Cx_BA+0×3C	R/W	I2C Transmit FIFO Threshold Register	0×0000_0000

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[7:0]	TX_TL	Transmit FIFO Threshold Level Controls the level of entries (or below) that trigger the TX_EMPTY interrupt (bit 4 in IC_RAW_INTR_STAT register). The valid range is 0-255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries.

4.10.7.17 Clear Combined and Individual Interrupt Register (IC_CLR_INTR)

Register	Offset	R/W	Description	Reset Value
IC_CLR_INTR x= 0, 1	I2Cx_BA+0×40	R	Clear Combined and Individual Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	DAT	Read this register to clear the combined interrupt, all individual interrupts, and the IC_TX_ABRT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE.

4.10.7.18 Clear RX_UNDER Interrupt Register (IC_CLR_RX_UNDER)

Register	Offset	R/W	Description	Reset Value
IC_CLR_RX_UNDER x= 0, 1	I2Cx_BA+0×44	R	Clear RX_UNDER Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	CLR_RX_UNDER	Read this register to clear the RX_UNDER interrupt (bit 0) of the

		IC_RAW_INTR_STAT register.

4.10.7.19 Clear RX_OVER Interrupt Register (IC_CLR_RX_OVER)

Register	Offset	R/W	Description	Reset Value
IC_CLR_RX_OVER x= 0, 1	I2Cx_BA+0×48	R	Clear RX_OVER Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	CLR_RX_OVER	Read this register to clear the RX_OVER interrupt (bit 1) of the IC_RAW_INTR_STAT register.

4.10.7.20 Clear TX_OVER Interrupt Register (IC_CLR_TX_OVER)

Register	Offset	R/W	Description	Reset Value
IC_CLR_TX_OVER x= 0, 1	I2Cx_BA+0×4C	R	Clear TX_OVER Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	CLR_TX_OVER	Read this register to clear the TX_OVER interrupt (bit 3) of the IC_RAW_INTR_STAT register.

4.10.7.21 Clear RD_REQ Interrupt Register (IC_CLR_RD_REQ)

Register	Offset	R/W	Description	Reset Value
IC_CLR_RD_REQ x= 0, 1	I2Cx_BA+0×50	R	Clear RD_REQ Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	CLR_RD_REQ	Read this register to clear the RD_REQ interrupt (bit 5) of the IC_RAW_INTR_STAT register.

4.10.7.22 Clear TX_ABRT Interrupt Register (IC_CLR_TX_ABRT)

Register	Offset	R/W	Description	Reset Value
IC_CLR_TX_ABRT x= 0, 1	I2Cx_BA+0×54	R	Clear TX_ABRT Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	CLR_TX_ABRT	Read this register to clear the TX_ABRT interrupt (bit 6) of the IC_RAW_INTR_STAT register, and the IC_TX_ABRT_SOURCE register. This also releases the Tx FIFO from the flushed/reset state, allowing more writes to the Tx FIFO. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE.

4.10.7.23 Clear RX_DONE Interrupt Register (IC_CLR_RX_DONE)

Register	Offset	R/W	Description	Reset Value
IC_CLR_RX_DONE x= 0, 1	I2Cx_BA+0×58	R	Clear RX_DONE Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	CLR_RX_DONE	Read this register to clear the RX_DONE interrupt (bit 7) of the IC_RAW_INTR_STAT register.

4.10.7.24 Clear ACTIVITY Interrupt Register (IC_CLR_ACTIVITY)

Register	Offset	R/W	Description	Reset Value
IC_CLR_ACTIVITY x= 0, 1	I2Cx_BA+0×5C	R	Clear ACTIVITY Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	CLR_ACTIVITY	Reading this register clears the ACTIVITY interrupt if the I2C is not active anymore. If the I2C module is still active on the bus, the ACTIVITY interrupt bit continues to be set. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the IC_RAW_INTR_STAT register.

4.10.7.25 Clear STOP_DET Interrupt Register (IC_CLR_STOP_DET)

Register	Offset	R/W	Description	Reset Value
IC_CLR_STOP_DET x= 0, 1	I2Cx_BA+0×60	R	Clear STOP_DET Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	CLR_STOP_DET	Read this register to clear the STOP_DET interrupt (bit 9) of the IC_RAW_INTR_STAT register.

4.10.7.26 Clear START_DET Interrupt Register (IC_CLR_START_DET)

Register	Offset	R/W	Description	Reset Value
IC_CLR_START_DET x= 0, 1	I2Cx_BA+0×64	R	Clear START_DET Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	CLR_START_DET	Read this register to clear the STOP_DET interrupt (bit 9) of the IC_RAW_INTR_STAT register.

4.10.7.27 Clear GEN_CALL Interrupt Register (IC_CLR_GEN_CALL)

Register	Offset	R/W	Description	Reset Value
IC_CLR_GEN_CALL x= 0, 1	I2Cx_BA+0×68	R	Clear GEN_CALL Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	CLR_GEN_CALL	Read this register to clear the GEN_CALL interrupt (bit 11) of IC_RAW_INTR_STAT register.

4.10.7.28 I2C Enable Register (IC_ENABLE)

Register	Offset	R/W	Description	Reset Value
IC_ENABLE x= 0, 1	I2Cx_BA+0×6C	R/W	I2C Enable Register	0×0000_0000

Bits	Descriptions	
[31:2]	Reserved	Reserved.
[1]	ABORT	When set, the controller initiates the transfer abort. 0 = ABORT not initiated or ABORT done 1 = ABORT operation in progress The software can abort the I2C transfer in master mode by setting this bit. The software can set this bit only when ENABLE is already set; otherwise, the controller ignores any write to ABORT bit. The software cannot clear the ABORT bit once set. In response to

		an ABORT, the controller issues a STOP and flushes the Tx FIFO after completing the current transfer, then sets the TX_ABORT interrupt after the abort operation. The ABORT bit is cleared automatically after the abort operation.
[0]	ENABLE	Controls whether the I2C is enabled. 0 = Disables I2C (TX and RX FIFOs are held in an erased state) 1 = Enables I2C Software can disable I2C while it is active. However, it is important that care be taken to ensure that I2C is disabled properly.. When I2C is disabled, the following occurs: The TX FIFO and RX FIFO get flushed. Status bits in the IC_INTR_STAT register are still active until I2C goes into IDLE state. If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the I2C stops the current transfer at the end of the current byte and does not acknowledge the transfer.

4.10.7.29 I2C Status Register (IC_STATUS)

Register	Offset	R/W	Description	Reset Value
IC_STATUS x= 0, 1	I2Cx_BA+0×70	R	I2C Status Register	0×0000_0006

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[6]	SLV_ACTIVITY	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0 = Slave FSM is in IDLE state so the Slave part of I2C is not Active 1 = Slave FSM is not in IDLE state so the Slave part ofI2C is Active
[5]	MST_ACTIVITY	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set. 0 = Master FSM is in IDLE state so the Master part of I2C is not Active 1 = Master FSM is not in IDLE state so the Master part ofI2C is Active Note: IC_STATUS[0]—that is, ACTIVITY bit—is the OR of SLV_ACTIVITY and MST_ACTIVITY bits.
[4]	RFF	Receive FIFO Completely Full. When the receive FIFO iscompletely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 = Receive FIFO is not full 1 = Receive FIFO is full
[3]	RFNE	Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty

[2]	TFE	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty
[1]	TFNF	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full
[0]	ACTIVITY	I2C Activity Status.

4.10.7.30 I2C Transmit FIFO Level Register (IC_TXFLR)

This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:

- The I2C is disabled
- There is a transmit abort—that is, TX_ABRT bit is set in the IC_RAW_INTR_STAT register
- The slave bulk transmit mode is aborted

The register increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.

Register	Offset	R/W	Description	Reset Value
IC_TXFLR x= 0, 1	I2Cx_BA+0×74	R	I2C Transmit FIFO Level Register	0×0000_0000

Bits	Descriptions	
[31:4]	Reserved	Reserved.
[3:0]	TXFLR	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.

4.10.7.31 I2C Receive FIFO Level Register (IC_RXFLR)

This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:

- The I2C is disabled
- Whenever there is a transmit abort caused by any of the events tracked in IC_TX_ABRT_SOURCE

The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.

Register	Offset	R/W	Description	Reset Value
IC_RXFLR x= 0, 1	I2Cx_BA+0×78	R	I2C Receive FIFO Level Register	0×0000_0000

Bits	Descriptions	
[31:4]	Reserved	Reserved.
[3:0]	RXFLR	Receive FIFO Level. Contains the number of valid data entries in thereceive FIFO.

4.10.7.32 I2C SDA Hold Time Length Register (IC_SDA_HOLD)

Writes to this register succeed only when IC_ENABLE[0]=0.

The values in this register are in units of ic_clk period. The value programmed in IC_SDA_TX_HOLD must be greater than the minimum hold time in each mode —one cycle in master mode, seven cycles in slave mode —for the value to be implemented.

The programmed SDA hold time during transmit (IC_SDA_TX_HOLD) cannot exceed at any time the duration of the low part of scl. Therefore the programmed value cannot be larger than N_SCL_LOW-2, where N_SCL_LOW is the duration of the low part of the scl period measured in ic_clk cycles.

Register	Offset	R/W	Description	Reset Value
IC_SDA_HOLD x= 0, 1	I2Cx_BA+0×7C	R/W	I2C SDA Hold Time Length Register	0×0001_0001

Bits	Descriptions	
[31:24]	Reserved	Reserved.
[23:16]	IC_SDA_RX_HOLD	Sets the required SDA hold time in units of ic_clk period, when I2C acts as a receiver. They are used to extend the SDA transition (if any) whenever SCL is HIGH in the receiver in either master or slave mode.
[15:0]	IC_SDA_TX_HOLD	Sets the required SDA hold time in units of ic_clk period, when I2C acts as a transmitter. They are used to control the hold time of SDA during transmit in both slave and master mode (after SCL goes from HIGH to LOW).

4.10.7.33 I2C Transmit Abort Source Register (IC_TX_ABRT_SOURCE)

This register has 32 bits that indicate the source of the TX_ABRT bit. Except for Bit 9, this register is cleared whenever the IC_CLR_TX_ABRT register or the IC_CLR_INTR register is read. To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]).

Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.

Register	Offset	R/W	Description	Reset Value
IC_TX_ABRT_SOURCE x= 0, 1	I2Cx_BA+0×80	R	I2C Transmit Abort Source Register	0×0000_0000

Bits	Descriptions	
[31]	TX_FLUSH_CNT	This field indicates the number of Tx FIFO data commands that are flushed due to TX_ABRT interrupt. It is cleared whenever I2C is disabled. Role of I2C: Master-Transmitter or Slave-Transmitter
[30:17]	Reserved	Reserved.
[16]	ABRT_USER_ABRT	This is a master-mode-only bit. Master has detected the transfer abort (IC_ENABLE[1]). Role of I2C: Master-Transmitter
[15]	ABRT_SLVRD_INTX	1 = When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register. Role of I2C:Slave-Transmitter
[14]	ABRT_SLV_ARBLOST	1 = Slave lost the bus while transmitting datato a remote master. IC_TX_ABRT_SOURCE[12] is set at the same time. Note: Even though the slave never “owns” the bus, something could go wrong on the bus.This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then I2C no longer own the bus. Role of I2C:Slave-Transmitter
[13]	ABRT_SLVFLUSH_TXFIFO	1 = Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO. Role of I2C:Slave-Transmitter
[12]	ARB_LOST	1 = Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration. Role of I2C:Slave-Transmitter Master-Transmitter
[11]	ABRT_MASTER_DIS	1 = User tries to initiate a Master operation with the Master mode disabled. Role of I2C:Master-Receiver Master-Transmitter
[10]	ABRT_10B_RD_NORSTRT	1 = The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the master sends a read command in10-bit addressing mode. Role of I2C:Master-Receiver
[9]	ABRT_SBYTE_NORSTRT	To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared(IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets re-asserted.

		1 = The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to send a START Byte. Role of I2C: Master
[8]	ABRT_HS_NORSTRT	1 = The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode. Role of I2C:Master-Transmitter Master-Receiver
[7]	ABRT_SBYTE_ACKDET	1 = Master has sent a START Byte and the START Byte was acknowledged (wrong behavior). Role of I2C:Master
[6]	ABRT_HS_ACKDET	1 = Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior). Role of I2C: Master
[5]	ABRT_GCALL_READ	1 = I2C in master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1). Role of I2C: Master-Transmitter
[4]	ABRT_GCALL_NOACK	1 = I2C in master mode sent a General Call and no slave on the bus acknowledged the General Call. Role of I2C: Master-Transmitter
[3]	ABRT_TXDATA_NOACK	1 = This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive anacknowledge from the remote slave(s). Role of I2C: Master-Transmitter
[2]	ABRT_10ADDR2_NOACK	1 = Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave. Role of I2C: Master-Transmitter Master-Receiver
[1]	ABRT_10ADDR1_NOACK	1 = Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. Role of I2C: Master-Transmitter Master-Receiver
[0]	ABRT_7B_ADDR_NOACK	1 = Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave. Role of I2C: Master-Transmitter Master-Receiver

4.10.7.34 Generate Slave Data NACK Register (IC_SLV_DATA_NACK_ONLY)

Register	Offset	R/W	Description	Reset Value
IC_SLV_DATA_NACK_ONLY x= 0, 1	I2Cx_BA+0×84	R/W	Generate Slave Data NACK Register	0×0000_0000

Bits	Description		
[31:1]	Reserved	N/A	Reserved.
[0]	NACK	R/W	Generate NACK. This NACK generation only occurs when I2C is a slave receiver. If this register is set to a value of 1, it can only generate a NACK after adata byte is received; hence, the data transfer is aborted and the data received isnot pushed to the receive buffer. When the register is set to a value of 0, it generates NACK/ACK, depending on normal criteria. 1 = generate NACK after data byte received 0 = generate NACK/ACK normally

4.10.7.35 DMA Control Register (IC_DMA_CR)

Register	Offset	R/W	Description	Reset Value
IC_DMA_CR x= 0, 1	I2Cx_BA+0×88	R/W	DMA Control Register	0×0000_0000

Bits	Descriptions	
[31:2]	Reserved	Reserved.
[1]	TDMAE	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled
[0]	RDMAE	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. 0 = Receive DMA disabled 1 = Receive DMA enabled

4.10.7.36 DMA Transmit Data Level Register (IC_DMA_TDLR)

Register	Offset	R/W	Description	Reset Value
IC_DMA_TDLR x= 0, 1	I2Cx_BA+0×8C	R/W	DMA Transmit Data Level Register	0×0000_0000

Bits	Descriptions	
[31:3]	Reserved	Reserved.
[2:0]	DMATDL	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMACE = 1.

4.10.7.37 DMA Receive Data Level Register (IC_DMA_RDLR)

Register	Offset	R/W	Description	Reset Value
IC_DMA_RDLR x= 0, 1	I2Cx_BA+0×90	R/W	DMA Receive Data Level Register	0×0000_0000

Bits	Descriptions	
[31:3]	Reserved	Reserved.
[2:0]	DMARDL	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE = 1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO.

4.10.7.38 I2C SDA Setup Register (IC_SDA_SETUP)

Register	Offset	R/W	Description	Reset Value
IC_SDA_SETUP x= 0, 1	I2Cx_BA+0×94	R/W	I2C SDA Setup Register	0×0000_0064

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[7:0]	SDA_SETUP	SDA Setup. It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz, IC_SDA_SETUP should be programmed to a value of 11. Note: Writes to this register succeed only when IC_ENABLE[0] = 0.

4.10.7.39 I2C ACK General Call Register (IC_ACK_GENERAL_CALL)

Register	Offset	R/W	Description	Reset Value
IC_ACK_GENERAL_CALL x= 0, 1	I2Cx_BA+0×98	R/W	I2C ACK General Call Register	0×0000_0001

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	ACK_GEN_CALL	ACK General Call. When set to 1, I2C responds with a ACK (by asserting ic_data_oe) when it receives a General Call. When set to 0, the I2C does not generate General Call interrupts. Note: This register is applicable only when the DW_apb_i2c is in the slave mode.

4.10.7.40 I2C Enable Status Register (IC_ENABLE_STATUS)

The register is used to report the I2C hardware status when IC_ENABLE[0] is set from 1 to 0; that is, when I2C is disabled.

- If IC_ENABLE[0] has been set to 1, bits 2:1 are forced to 0, and bit 0 is forced to 1.
- If IC_ENABLE[0] has been set to 0, bits 2:1 is only be valid as soon as bit 0 is read as '0'.

Register	Offset	R/W	Description	Reset Value
IC_ENABLE_STATUS x= 0, 1	I2Cx_BA+0×9C	R	I2C Enable Status Register	0×0000_0000

Bits	Descriptions	
[31:3]	Reserved	Reserved.
[2]	SLV_RX_DATA_LOST	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to setting IC_ENABLE[0] from 1 to 0. When read as 1, I2C is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. **Note**: If the remote I2C master terminates the transfer with a STOP condition before the I2C has a chance to NACK a transfer, and IC_ENABLE[0] has been set to 0, then this bit is also set to 1. When read as 0, I2C is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. **Note**: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.
[1]	SLV_DISABLED_WHILE_BUSY	Slave Disabled While Busy (Transmit, Receive). This bit indicates ifa potential or active Slave operation has been aborted due tosetting bit 0 of the IC_ENABLE register from 1 to 0. This bit is setwhen the CPU writes a 0 to bit 0 of IC_ENABLE while: ■ I2C is receiving the address byte of the Slave-Transmitteroperation from a remote master; OR, ■ Address and data bytes ofthe Slave-Receiver operation from a remote master. When read as 1, I2C is deemed to have forced a NACKduring any part of an I2C transfer, irrespective of whether the I2Caddress matches the slave address set in I2C (IC_SARregister) OR if the transfer is completed before bit 0 of IC_ENABLEis set to 0, but has not taken effect. **Note**: If the remote I2C master terminates the transfer with a STOP condition before the I2C has a chance to NACK a transfer, and bit 0 of IC_ENABLE has been set to 0, then this bit will also be set to 1. When read as 0, I2C is deemed to have been disabled when there is master activity, or when the I2C bus is idle. **Note**: The CPU can safely read this bit when *IC_EN* (bit 0) is read as 0.

[0]	IC_EN	ic_en Status. This bit always reflects the value driven on the output port ic_en. ■ When read as 1, I2C is deemed to be in an enabled state. ■ When read as 0, I2C is deemed completely inactive. **Note:** The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read *SLV_RX_DATA_LOST* (bit 2) and *SLV_DISABLED_WHILE_BUSY* (bit 1).

4.10.7.41 I2C SS and FS Spike Suppression Limit Register (IC_FS_SPKLEN)

Register	Offset	R/W	Description	Reset Value
IC_FS_SPKLEN x= 0, 1	I2Cx_BA+0×A0	R/W	I2C SS and FS Spike Suppression Limit Register	0×0000_0005

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[7:0]	IC_FS_SPKLEN	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that are filtered out by the spike suppression logic; for more information, refer to “Spike Suppression”. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted, results in 1 being set.

4.10.7.42 I2C HS Spike Suppression Limit Register (IC_HS_SPKLEN)

Register	Offset	R/W	Description	Reset Value
IC_HS_SPKLEN x= 0, 1	I2Cx_BA+0×A4	R/W	I2C HS Spike Suppression Limit Register	0×0000_0001

Bits	Descriptions	
[31:8]	Reserved	Reserved.
[7:0]	IC_HS_SPKLEN	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that are filtered out by the spike suppression logic; for more information, refer to “Spike Suppression”. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted, results in 1 being set.

4.10.7.43 Clear RESTART_DET Interrupt Register (IC_CLR_RESTART_DET)

Register	Offset	R/W	Description	Reset Value
IC_CLR_RESTART_DET x= 0, 1	I2Cx_BA+0×A8	R	Clear RESTART_DET Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved.
[0]	RESTART	Read this register to clear the RESTART_DET interrupt (bit 12) of the IC_RAW_INTR_STAT register.

4.10.7.44 Clear SCL Stuck at Low Detect Interrupt Register (IC_CLR_SCL_STUCK_DET)

Register	Offset	R/W	Description	Reset Value
IC_CLR_SCL_STUCK_DET	I2Cx_BA+0×B4	R	Clear SCL Stuck at Low Detect Interrupt Register	0×0000_0000

Bits	Descriptions	
[31:1]	Reserved	Reserved
0	IC_SDA_STUCK_AT_LOW_TIMEOUT	Read this register to clear the SCL_STUCK_DET interrupt (bit 14) of the IC_RAW_INTR_STAT register.

4.10.7.45 Component Parameter Register 1 (IC_COMP_PARAM_1)

Register	Offset	R/W	Description	Reset Value
IC_COMP_PARAM_1	I2Cx_BA+0×F4	R	Component Parameter Register 1	0×0007_07EE

Bits	Descriptions	
[31:24]	Reserved	Reserved.
[23:16]	TX_BUFFER_DEPTH	Depth of the transmit buffer. 0×00 = Reserved 0×01 = 2 0×02 = 3 … 0×FF = 256
[15:8]	RX_BUFFER_DEPTH	Depth of receive buffer. 0×00 = Reserved 0×01 = 2 0×02 = 3

		… 0×FF = 256
7	ADD_EN-CODED_PARAMS	Reading 1 in this bit means that the capability of reading these encoded parameters via software has been included. Otherwise, the entire register is 0 regardless of the setting of any other parameters that are encoded in the bits. 0 = False 1 = True
6	HAS_DMA	When checked, includes the DMA handshaking interface signals at the top-level I/O. 0 = False 1 = True
5	INTR_IO	If unchecked, each interrupt source has its own output. If checked, all interrupt sources are combined into a single output. 0 = Individual 1 = Combined
4	HC_COUNT_VAL-UES	By checking this parameter, the *CNT registers are set to read only. Unchecking this parameter (default setting) allows the *CNT registers to be writable. 0 = False 1 = True
[3:2]	MAX_SPEED_MODE	Maximum I2C mode supported. 0×0 = Reserved 0×1 = Standard 0×2 = Fast 0×3 = High
[1:0]	APB_DATA_WIDTH	Width of the APB data bus. 0×0 = 8 bits 0×1 = 16 bits 0×2 = 32 bits 0×3 = Reserved

4.11 Serial Peripheral Interface (SPI)

4.11.1 Overview

In order for the SPI to connect to a serial-master or serial-slave peripheral device, the peripheral have the following interface: Motorola Serial Peripheral Interface (SPI) – A four-wire, full-duplex serial protocol from Motorola.There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of the slave select signal or the first edge of the serial clock. The slave select line is held high when the SPI is idle or disabled.

4.11.2 Features

- Supports APB data bus widths of 32 bits.
- SPI 0,1 only can be serial-master, SPI 2,3 only can be serial-slave.
- Serial-master or serial-slave operation – Enables serial communication with serial-master or serial-slave peripheral devices.
- Bypass of meta-stability flip-flops for synchronous clocks – When the APB clock (HCLK) and the SPI serial clock (ssi_clk) are synchronous, meta-stable flip-flops are not used when transferring control signals across these clock domains.
- DMA Controller Interface – Enables the SPI to interface to a DMA controller over the bus using a handshaking interface for transfer requests.
- Independent masking of interrupts
- Configurable features:
  - The FIFO depth of the transmit and receive FIFO buffer is 16
  - 1 slave select output
  - Combined interrupt lines
  - Serial clock polarity and clock phase

4.11.3 Block Diagram

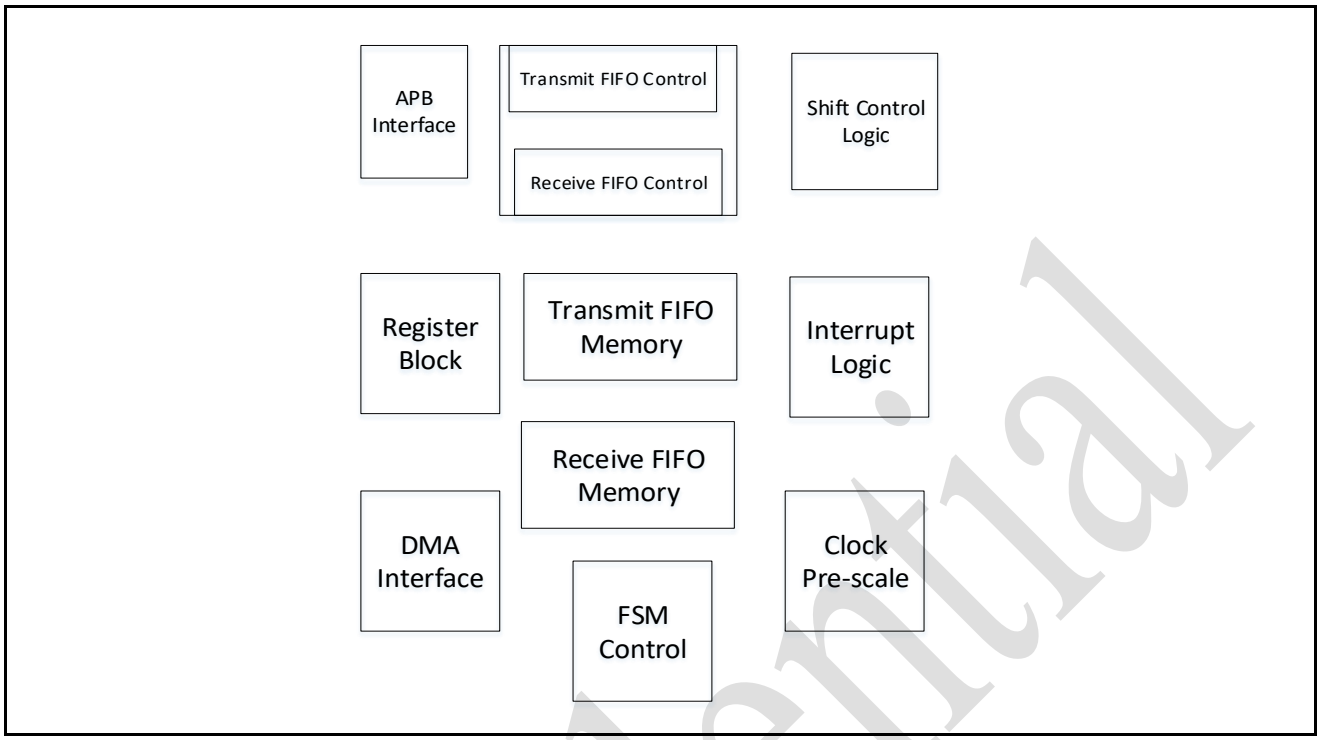


Figure 4-88 SPI Block Diagram

4.11.4 Functional Description

4.11.4.1 Motorola Serial Peripheral Interface

The SPI can connect to any serial-master or serial-slave peripheral device using Motorola Serial Peripheral interfaces. In this circumstance, data transmission begins on the rising edge of the slave select signal. The first data bit is captured by the master and slave peripherals on the first edge of the serial clock. In PAN1020, the clock polarity (SCPOL) and serial clock phase (SCPH) are both set to 0, and the maximum transfer size (SSI_MAX_XFER_SIZE) is set to 32 bits. When SSI_SCPH0_SSTOGGLE is configured as True, the SPI toggles the slave select signal between frames and the serial clock is held to its default value while the slave select signal is active; this operating mode is illustrated in Figure 4-89.

To reduce the synchronization delay, the synchronization scheme uses two flip flops: one works on the positive edge of ssi_clk; and other works on the negative edge of ssi_clk. These flip flops reduce the synchronization delay to one ssi_clk cycle and enable SPI to work on lower clock ratios. When SSI_ENH_CLK_RATIO=1, the SPI slave device minimum frequency of ssi_clk is 4 times the maximum expected frequency of the bit-rate clock (sclk_in).

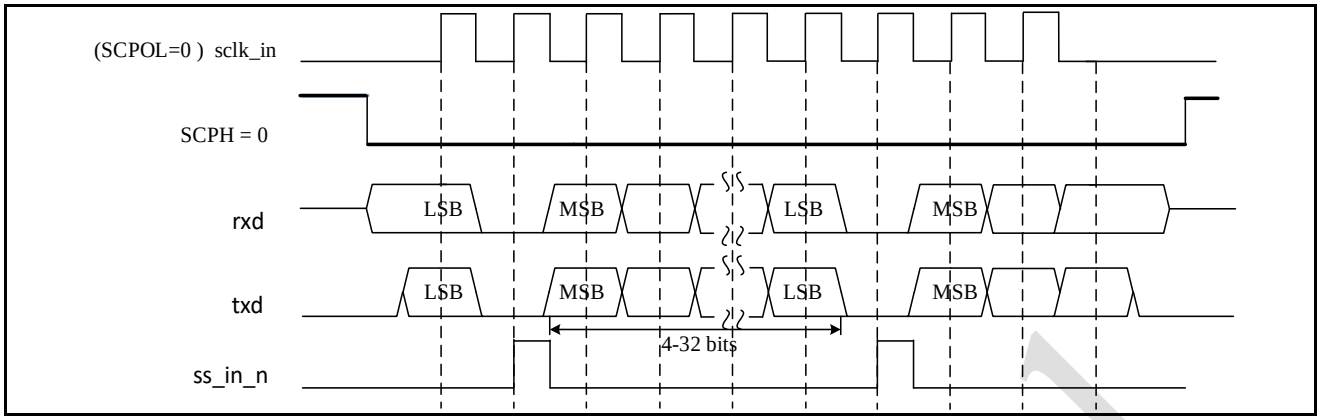


Figure 4-89 Frame Format for Motorola Serial Peripheral Interface

4.11.4.2 SPI Interrupts

The SPI only supports combined interrupt requests (ssi_intr) which is an OR'ed result of all interrupt requests after masking. To mask this interrupt signal, you must mask all other SPI interrupt requests. The polarity level of SPI is active-high. The specific interrupt signals are described in Table 4-23.

Table 4-23 Interrupt Signals

Port Name	I/O	Description
ssi_txe_intr	O	Transmit FIFO Empty Interrupt. Set when the transmit FIFO is equal to or below its threshold value and requires service to prevent an under-run.
ssi_txo_intr	O	Transmit FIFO Overflow Interrupt. Set when an APB access attempts to write into the transmit FIFO after it has been completely filled.
ssi_rxf_intr	O	Receive FIFO Full Interrupt. Set when the receive FIFO is equal to or above its threshold value plus 1 and requires service to prevent an overflow.
ssi_rxo_intr	O	Receive FIFO Overflow Interrupt. Set when the receive logic attempts to place data into the receive FIFO after it has been completely filled.
ssi_rxu_intr	O	Receive FIFO Underflow Interrupt. Set when an APB access attempts to read from the receive FIFO when it is empty.
ssi_mst_intr	O	Multi-Master Contention Interrupt. Set when another serial master on the serial bus selects the SPI master as a serial-slave device and is actively transferring data.
ssi_intr	O	Combined Interrupt Request. OR'ed result of all the above interrupt requests after masking.

4.11.4.3 Transfer Modes

When transferring data on the serial bus, the SPI operates in the modes discussed in this section. The transfer mode (TMOD) is set by writing to control register 0 (CTRLR0)

- Transmit and Receive

When TMOD = 2‘b00, both transmit and receive logic are valid. The data transfer occurs as normal according to the selected frame format (serial protocol). Transmit data are popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. The

receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame.

- Transmit Only

When TMOD = 2‘b01, the receive data are invalid and should not be stored in the receive FIFO. The data transfer occurs as normal, according to the selected frame format (serial protocol). Transmit data are popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. At the end of the data frame, the receive shift register does not load its newly received data into the receive FIFO. The data in the receive shift register is overwritten by the next transfer. You should mask interrupts originating from the receive logic when this mode is entered.

- Receive Only

When TMOD = 2‘b10, the transmit data are invalid. When configured as a slave, the transmit FIFO is never popped in Receive Only mode. The txd output remains at a constant logic level during the transmission.The data transfer occurs as normal according to the selected frame format (serial protocol). The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame. You should mask interrupts originating from the transmit logic when this mode is entered.

4.11.4.4 Operation Modes

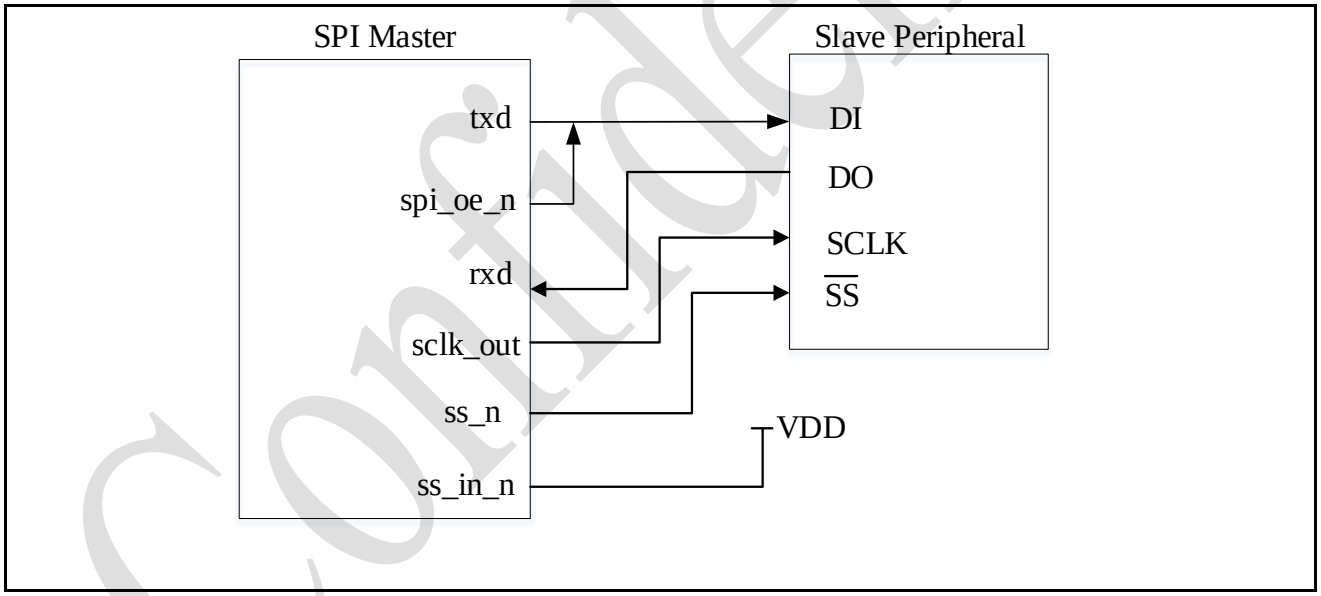


Figure 4.11-4 SPI Configured as Master Device

When the transfer mode is “transmit and receive” or “transmit only” (TMOD = 2'b00 or TMOD = 2'b01, respectively), transfers are terminated by the shift control logic when the transmit FIFO is empty. For continuous data transfers, you must ensure that the transmit FIFO buffer does not become empty before all the data have been transmitted. The transmit FIFO threshold level (TXFTLR) can be used to early interrupt (ssi_txe_intr) the processor indicating that the transmit FIFO buffer is nearly empty.

When a DMAC is used for APB accesses, the transmit data level (DMATDLR) can be used to early request (dma_tx_req) the DMA Controller, indicating that the transmit FIFO is nearly empty. The FIFO can then be refilled with data to continue the serial transfer. The user may also write a block

of data (at least two FIFO entries) into the transmit FIFO before enabling a serial slave. This ensures that serial transmission does not begin until the number of data-frames that make up the continuous transfer are present in the transmit FIFO.

When the transfer mode is “receive only” (TMOD = 2'b10), a serial transfer is started by writing one“dummy” data word into the transmit FIFO when a serial slave is selected. The txd output from the SPI is held at a constant logic level for the duration of the serial transfer. The transmit FIFO is popped only once at the beginning and may remain empty for the duration of the serial transfer. The end of the serial transfer is controlled by the “number of data frames” (NDF) field in control register 1 (CTRLR1).

If, for example, you want to receive 24 data frames from a serial-slave peripheral, you should program the NDF field with the value 23; the receive logic terminates the serial transfer when the number of frames received is equal to the NDF value + 1. This transfer mode increases the bandwidth of the APB bus as the transmit FIFO never needs to be serviced during the transfer. The receive FIFO buffer should be read each time the receive FIFO generates a FIFO full interrupt request to prevent an overflow.

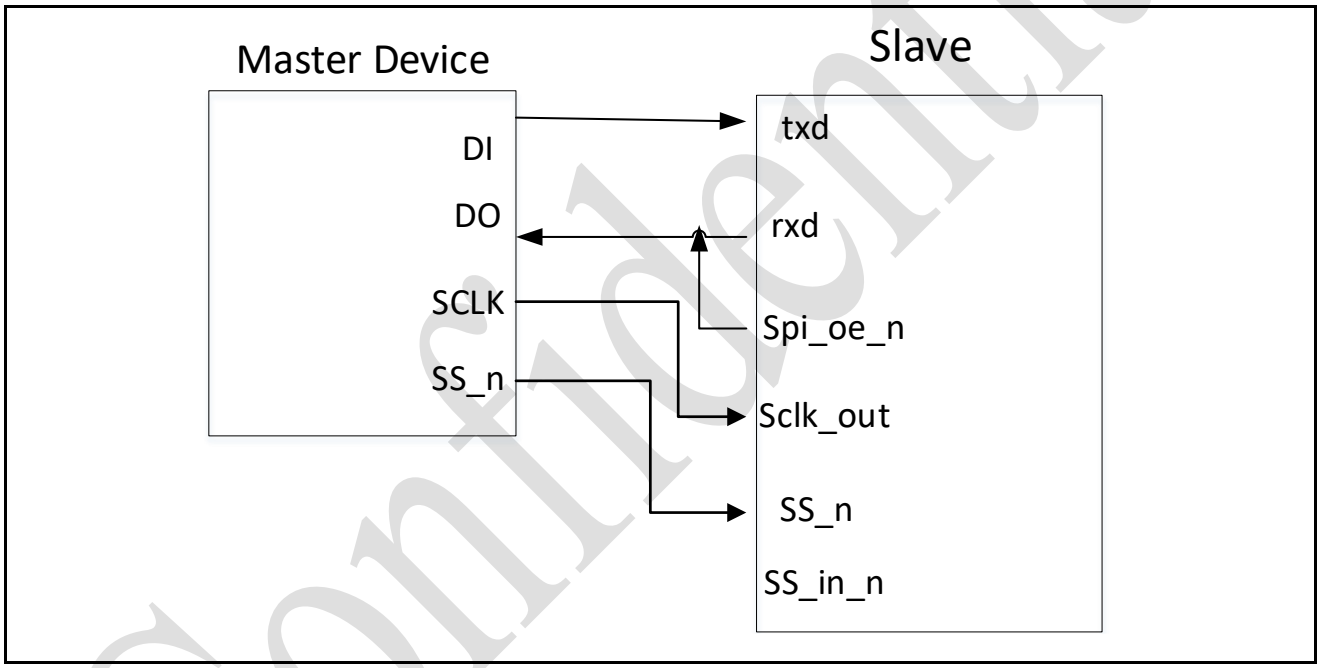


Figure 4.11-5 SPI Configured as Slave Device

If the SPI slave is receive only (TMOD=10), the transmit FIFO need not contain valid data because the data currently in the transmit shift register is resent each time the slave device is selected. The TXE error flag in the status register (SR) is not set when TMOD=01. You should mask the transmit FIFO empty interrupt when this mode is used. If the SPI slave transmits data to the master, you must ensure that data exists in the transmit FIFO before a transfer is initiated by the serial-master device.

If the master initiates a transfer to the SPI slave when no data exists in the transmit FIFO, an error flag (TXE) is set in the SPI status register, and the previously transmitted data frame is resent on txd. For continuous data transfers, you must ensure that the transmit FIFO buffer does not become empty before all the data have been transmitted. The transmit FIFO threshold level register (TXFTLR) can be used to early interrupt (ssi_txe_intr) the processor, indicating that the transmit FIFO buffer is nearly empty. When a DMA Controller is used for APB accesses, the DMAC transmit data level register (DMATDLR) can be used to early request (dma_tx_req) the DMA Controller,

indicating that the transmit FIFO is nearly empty. The FIFO can then be refilled with data to continue the serial transfer.

The receive FIFO buffer should be read each time the receive FIFO generates a FIFO full interrupt request to prevent an overflow. The receive FIFO threshold level register (RXFTLR) can be used to give early indication that the receive FIFO is nearly full. When a DMA Controller is used for APB accesses, the DMAC receive data level register (DMARDLR) can be used to early request (dma_rx_req) the DMAC controller, indicating that the receive FIFO is nearly full.

4.11.4.5 DMA Controller Interface

The SPI supports a built-in DMA capability; it has a handshaking interface to a DMA Controller to request and control transfers. The APB bus is used to perform the data transfer to or from the DMAC. The settings of the DMAC that are relevant to the operation of the SPI are discussed here, mainly bit fields in the DMAC channel control register, CTLx, where x is the channel number.

- Transmit Watermark Level and Transmit FIFO Underflow

During SPI serial transfers, transmit FIFO requests are made to the DMAC whenever the number of entries in the transmit FIFO is less than or equal to the DMA Transmit Data Level Register (DMATDLR) value; this is known as the watermark level. The DMAC responds by writing a burst of data to the transmit FIFO buffer, of length CTLx.DEST_MSIZE.Data should be fetched from the DMAC often enough for the transmit FIFO to perform serial transfers continuously; that is, when the FIFO begins to empty another DMA request should be triggered. Otherwise the FIFO will run out of data (underflow). To prevent this condition, the user must set the watermark level correctly.

- Receive Watermark Level and Receive FIFO Overflow

During SPI serial transfers, receive FIFO requests are made to the DMAC whenever the number of entries in the receive FIFO is at or above the DMA Receive Data Level Register; that is, DMARDLR+1. This is known as the watermark level. The DMAC responds by fetching a burst of data from the receive FIFO buffer of length CTLx.SRC_MSIZE.Data should be fetched by the DMAC often enough for the receive FIFO to accept serial transfers continuously; that is, when the FIFO begins to fill, another DMAC transfer is requested. Otherwise, the FIFO will fill with data (overflow). To prevent this condition, the user must correctly set the watermark level.

- Handshaking Interface Operation
  - dma_tx_req,dma_rx_req：

The request signals for source and destination, dma_tx_req and dma_rx_req, are activated when their corresponding FIFOs reach the watermark levels as discussed earlier.

The DMAC uses rising-edge detection of the dma_tx_req signal/dma_rx_req to identify a request on the channel. Upon reception of the dma_tx_ack/dma_rx_ack signal from the DMAC to indicate the burst transaction is complete, the SPI de-asserts the burst request signals, dma_tx_req/dma_rx_req, until dma_tx_ack/dma_rx_ack is de-asserted by the DMAC.

When the SPI samples that dma_tx_ack/dma_rx_ack is de-asserted, it can re-assert the dma_tx_req/dma_rx_req of the request line if their corresponding FIFOs exceed their watermark levels (back-to-back burst transaction). If this is not the case, the DMA request lines remain de-asserted.

Two things to keep in mind:

1. The burst request lines, dma_tx_req /dma_rx_req signal, once asserted remain asserted until their corresponding dma_tx_ack/dma_rx_ack signal is received even if the respective FIFO's drop below their watermark levels during the burst transaction.
2. The dma_tx_req/dma_rx_req signals are de-asserted when their corresponding dma_tx_ack/dma_rx_ack signals are asserted, even if the respective FIFOs exceed their water-mark levels.

- dma_tx_single, dma_rx_single:

The dma_tx_single signal is asserted when there is at least one free entry in the transmit FIFO, and is clearer when the dma_tx_ack signal is active. The dma_tx_single signal will be re-asserted when the dma_tx_ack signal is de-asserted, if the condition for setting still holds true.

The dma_rx_single signal is asserted when there is at least one valid data entry in the receive FIFO, and is cleared when the dma_rx_ack signal is active. The dma_rx_single signal will be re-asserted when the dma_rx_ack signal is de-asserted, if the condition for setting still holds true

4.11.4.6 APB Interface

The host processor accesses data, control, and status information on the SPI through the APB interface. The SPI supports APB data bus widths of 32 bits. Control and status registers within the SPI are byte-addressable. The maximum width of the control or status register in the SPI is 16 bits. All read and write operations to the SPI control and status registers require only one APB access.

The data register (DR) within the SPI is 32-bits wide. and the SPI DR can be written/read in one APB access. This is the optimal configuration as the full bandwidth of the bus is used when accessing this peripheral.

4.11.5 Register Map

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

Register	Offset	R/W	Description	Reset Value
SPI Base Address: **SPI0_BA =0x4010_4000** **SPI1_BA =0x4010_5000** **SPI2_BA =0x4010_6000** **SPI3_BA =0x4010_7000**				
CTRLR0	0x00	R/W	Control Register 0	0x0007_0000
CTRLR1	0x04	R/W	Control Register 1	0x0000_0000
SSIENR	0x08	R/W	SPI Enable Register	0x0000_0000
SER	0x10	R/W	Slave Enable Register	0x0000_0000
BAUDR	0x14	R/W	Baud Rate Select	0x0000_0000
TXFTLR	0x18	R/W	Transmit FIFO Threshold Level	0x0000_0000
RXFTLR	0x1C	R/W	Receive FIFO Threshold Level	0x0000_0000
TXFLR	0x20	R	Transmit FIFO Level Register	0x0000_0000
RXFLR	0x24	R	Receive FIFO Level Register	0x0000_0000
SR	0x28	R	Status Register	0x0000_0006

IMR	0x2C	R/W	Interrupt Mask Register	0x0000003F(master)/0x0000001F(slave)
ISR	0x30	R	Interrupt Status Register	0x0000_0000
RISR	0x34	R	Raw Interrupt Status Register	0x0000_0000
TXOICR	0x38	R	Transmit FIFO Overflow Interrupt Clear Register	0x0000_0000
RXOICR	0x3C	R	Receive FIFO Overflow Interrupt Clear Register	0x0000_0000
RXUICR	0x40	R	Receive FIFO Underflow Interrupt Clear Register	0x0000_0000
MSTICR	0x44	R	Multi-Master Interrupt Clear Register	0x0000_0000
ICR	0x48	R	Interrupt Clear Register	0x0000_0000
DMACR	0x4C	R/W	DMA Control Register	0x0000_0000
DMATDLR	0x50	R/W	DMA Transmit Data Level	0x0000_0000
DMARDLR	0x54	R/W	DMAC Receive Data Level	0x0000_0000
IDR	0x58	R	Identification Register	N/A
DR	0x60-0xEC	R/W	Data Register	0x0000_0000
RSVD_1	0xF8	R/W	Reserved location for future use	0x0000_0000
RSVD_2	0xFC	R/W	Reserved location for future use	0x0000_0000

4.11.6 Register Description

4.11.6.1 Control Register 0 (CTRLR0)

Register	Offset	R/W	Description	Reset Value
CTRLR0 x = 0,1,2,3	SPIx_BA +0x00	R/W	Control Register 0	0x0007_0000

Bits	Description	
[31-21]	Reserved	Reserved
[20:16]	DFS_32	Data Frame Size in 32-bit mode Used to select the data frame size in 32-bit mode.

DFS_32	Description
00000	Reserved
00001	Reserved
00010	Reserved
00011	4-bit serial data transfer
00100	5-bit serial data transfer
00101	6-bit serial data transfer
…	…
11101	30-bit serial data transfer

		11110 — 31-bit serial data transfer 11111 — 32-bit serial data transfer
[15:12]	Reserved	Reserved.
[11]	SRL	Shift Register Loop. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. Can be used in both serial-slave and serial-master modes. 0 – Normal Mode Operation 1 – Test Mode Operation When the SPI is configured as a slave in loopback mode, the ss_in_n and ssi_clk signals must be provided by an external source. In this mode, the slave cannot generate these signals because there is nothing to which to loop back.
[10]	SLV_OE	Slave Output Enable. Relevant only when the SPI is configured as a serial-slave device. When configured as a serial master, this bit field has no functionality. This bit enables or disables the setting of the ssi_oe_n output from the SPI serial slave. When SLV_OE = 1, the ssi_oe_n output can never be active. When the ssi_oe_n output controls the tri-state buffer on the txd output from the slave, a high impedance state is always present on the slave txd output when SLV_OE = 1. This is useful when the master transmits in broadcast mode (master transmits data to all slave devices). Only one slave may respond with data on the master rxd line. This bit is enabled after reset and must be disabled by software (when broadcast mode is used), if you do not want this device to respond with data. 0 – Slave txd is enabled 1 – Slave txd is disabled
[9:8]	TMOD	Transfer Mode. Selects the mode of transfer for serial communication. This field does not affect the transfer duplicity. Only indicates whether the receive or transmit data are valid. The specific mode can refer to “DMA Controller Interface". 00 – Transmit & Receive 01 – Transmit Only 10 – Receive Only 11 – Reserved
[7]	SCPOL	Serial Clock Polarity. Used to select the polarity of the inactive serial clock, which is held inactive when the SPI master is not actively transferring data on the serial bus. 0 – Inactive state of serial clock is low 1 – Inactive state of serial clock is high
[6]	SCPH	Serial Clock Phase. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is

		activated, and data are captured on the second edge of the serial clock. 0: Serial clock toggles in middle of first data bit 1: Serial clock toggles at start of first data bit.
[5:4]	FRF	Frame Format. Motorola SPI protocol transfers the data.
[3:0]	Reserved	Reserved

4.11.6.2 Control Register 1 (CTRLR1)

Register	Offset	R/W	Description	Reset Value
CTRLR1 x = 0,1,2,3	SPIx_BA +0x04	R/W	Control Register 1	0x0000_0000

Bits	Description	
[31-16]	Reserved	Reserved
[15-0]	NDF	Number of Data Frames. When TMOD = 10 or TMOD = 11, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of datain a continuous transfer. Note: This register exists only when the SPI is configured as a master device. When the SPI is configured as a serial slave, writing to this location has no effect; reading from this location returns 0.

4.11.6.3 SPI Enable Register(SSIENR)

Register	Offset	R/W	Description	Reset Value
SSIENR x = 0,1,2,3	SPIx_BA +0x08	R/W	SPI Enable Register	0x0000_0000

Bits	Description	
[31:1]	Reserved	Reserved
[0]	SSI_EN	SSI Enable. Enables and disables all DW_apb_ssi operations. When disabled, all serialtransfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the DW_apb_ssi control registers when enabled. When disabled, the ssi_sleep output is set (after delay) to inform the system that it is safe to remove the ssi_clk, thus saving power consumption in the system.

4.11.6.4 Slave Enable Register (SER)

Register	Offset	R/W	Description	Reset Value
SER x = 0,1,2,3	SPIx_BA +0x10	R/W	Slave Enable Register	0x0000_0000

Bits	Description	
[31:1]	Reserved	Reserved
[0]	SER	Slave Select Enable Flag. When a bit in this register is set (1), the corresponding slave select line from the master is activated when a serial transfer begins. It should be noted that setting or clearing bits in this register have no effect on the corresponding slave select outputs until a transfer is started. Before beginning a transfer, you should enable the bit in this register that corresponds to the slave device with which the master wants to communicate. When not operating in broadcast mode, only one bit in this field should be set. 1: Selected 0: Not Selected

4.11.6.5 Baud Rate Select Register (BAUDR)

Register	Offset	R/W	Description	Reset Value
BAUDR x = 0,1,2,3	SPIx_BA+0x14	R/W	Baud Rate SelectRegister	0x0000_0000

Bits	Description	
[31:16]	Reserved	Reserved
[15:0]	SCKDV	SPI Clock Divider. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: $F_{sclk\_out} = F_{ssi\_clk}/SCKDV$ where SCKDV is any even value between 2 and 65534. For example: for F_{ssi_clk} = 3.6864MHz and SCKDV =2 F_{sclk_out} = 3.6864/2 = 1.8432MHz Note: This register exists only when the SPI is configured as a master device. When the SPI is configured as a serial slave, writing to this location has no effect; reading from this location returns 0.

4.11.6.6 Transmit FIFO Threshold Level Register (TXFTLR)

Register	Offset	R/W	Description	Reset Value
TXFTLR x = 0,1,2,3	SPIx_BA +0x18	R/W	Transmit FIFO Threshold Level	0x0000_0000

Bits	Description	
[31: 4]	Reserved	Reserved
[3:0]	TFT	Transmit FIFO Threshold. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable as 10; this register is sized to the number of address bits needed to access the FIFO. If you attempt to set bits [4:0] of this register to a value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered. For field decode, refer to the following table.

TFT Value	Description
0000	ssi_txe_intr is asserted when 0 data entries are present in transmit FIFO
0001	ssi_txe_intr is asserted when 1 or less data entry is present in transmit FIFO
0010	ssi_txe_intr is asserted when 2 or less data entries are present in transmit FIFO
0011	ssi_txe_intr is asserted when 3 or less data entries are present in transmit FIFO
…	…
1100	ssi_txe_intr is asserted when12 or less data entries are present in transmit FIFO
1101	ssi_txe_intr is asserted when 13 or less data entries are present in transmit FIFO
1110	ssi_txe_intr is asserted when 14 or less data entries are present in transmit FIFO
1111	ssi_txe_intr is asserted when 15 or less data entries are present in transmit FIFO

4.11.6.7 Receive FIFO Threshold Level Register (RXFTLR)

Register	Offset	R/W	Description	Reset Value
RXFTLR x = 0,1,2,3	SPIx_BA +0x1C	R/W	Receive FIFO Threshold Level	0x0000_0000

Bits	Description	
[31: 4]	Reserved	Reserved

[3:0]	RFT	Receive FIFO Threshold. Controls the level of entries (or above) atwhich the receive FIFO controller triggers an interrupt. The FIFOdepth is configurable as 10. This register is sized tothe number of address bits needed to access the FIFO. If youattempt to set this value greater than the depth of the FIFO, thisfield is not written and retains its current value.

RFT Value	Description
0000	ssi_rxf_intr is asserted when 1 or more data entry is present in receive FIFO
0001	ssi_rxf_intr is asserted when 2 or more data entries are present in receive FIFO
0010	ssi_rxf_intr is asserted when 3 or more data entries are present in receive FIFO
0011	ssi_rxf_intr is asserted when 4 or more data entries are present in receive FIFO
…	…
1100	ssi_rxf_intr is asserted when 13 or more data entries are present in receive FIFO
1101	ssi_rxf_intr is asserted when 14 or more data entries are present in receive FIFO
1110	ssi_rxf_intr is asserted when 15 or more data entries are present in receive FIFO
1111	ssi_rxf_intr is asserted when 16 data entries are present in receive FIFO

4.11.6.8 Transmit FIFO Level Register (TXFLR)

Register	Offset	R/W	Description	Reset Value
TXFLR x = 0,1,2,3	SPIx_BA +0x20	R	Transmit FIFO Level Register	0x0000_0000

Bits	Description	
[31:5]	Reserved	Reserved
[4:0]	TXTFL	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.

4.11.6.9 Receive FIFO Level Register (RXFLR)

Register	Offset	R/W	Description	Reset Value
RXFLR x = 0,1,2,3	SPIx_BA +0x24	R	Receive FIFO Level Register	0x0000_0000

Bits	Description	
[31:5]	Reserved	Reserved

[4:0]	RXTFL	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.

4.11.6.10 Status Register (SR)

Register	Offset	R/W	Description	Reset Value
SR x = 0,1,2,3	SPIx_BA +0x28	R	Status Register	0x0000_0006

Bits	Description	
[31: 7]	Reserved	Reserved
[6]	DCOL	Data Collision Error. Relevant only when the SPI is configured as a master device. This bit is set if the ss_in_n input is asserted by another master, while the SPI master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. 0 – No error 1 – Transmit data collision error
[5]	TXE	Transmission Error. Set if the transmit FIFO is empty when a transfer is started. This bit can be set only when the SPI is configured as a slave device. Data from the previous transmission is resent on the txd line. This bit is cleared when read. 0 – No error 1 – Transmission error
[4]	RFF	Receive FIFO Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 – Receive FIFO is not full 1 – Receive FIFO is full
[3]	RFNE	Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty
[2]	TFE	Transmit FIFO Empty. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty
[1]	TFNF	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full

[0]	BUSY	SPI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. 0 – SPI is idle or disabled 1 – SPI is actively transferring data

4.11.6.11 Interrupt Mask Register (IMR)

Register	Offset	R/W	Description	Reset Value
IMR x = 0,1,2,3	SPIx_BA +0x2C	R/W	Interrupt Mask Register	0x0000_003F(master)/0x0000_001F(slave)

Bits	Description	
[31: 6]	Reserved	Reserved
[5]	MSTIM	Multi-Master Contention Interrupt Mask. This bit field is not present if the SPI is configured as a serial-slave device. 0 – ssi_mst_intr interrupt is masked 1 – ssi_mst_intr interrupt is not masked
[4]	RXFIM	Receive FIFO Full Interrupt Mask 0 – ssi_rxf_intr interrupt is masked 1 – ssi_rxf_intr interrupt is not masked
[3]	RXOIM	Receive FIFO Overflow Interrupt Mask 0 – ssi_rxo_intr interrupt is masked 1 – ssi_rxo_intr interrupt is not masked
[2]	RXUIM	Receive FIFO Underflow Interrupt Mask 0 – ssi_rxu_intr interrupt is masked 1 – ssi_rxu_intr interrupt is not masked
[1]	TXOIM	Transmit FIFO Overflow Interrupt Mask 0 – ssi_txo_intr interrupt is masked 1 – ssi_txo_intr interrupt is not masked
[0]	TFNE	Transmit FIFO Empty Interrupt Mask 0 – ssi_txe_intr interrupt is masked 1 – ssi_txe_intr interrupt is not masked

4.11.6.12 Interrupt Status Register (SPI_ISR)

Register	Offset	R/W	Description	Reset Value
ISR x = 0,1,2,3	SPIx_BA +0x30	R	Interrupt Status Register	0x0000_0000

Bits	Description	
[31: 6]	Reserved	Reserved
[5]	MSTIS	Multi-Master Contention Interrupt Status. This bit field is not present if the SPI is configured as a serial-slave device.

		0 = ssi_mst_intr interrupt not active after masking 1 = ssi_mst_intr interrupt is active after masking
[4]	RXFIS	Receive FIFO Full Interrupt Status 0 = ssi_rxf_intr interrupt is not active after masking 1 = ssi_rxf_intr interrupt is full after masking
[3]	RXOIS	Receive FIFO Overflow Interrupt Status 0 = ssi_rxo_intr interrupt is not active after masking 1 = ssi_rxo_intr interrupt is active after masking
[2]	RXUIS	Receive FIFO Underflow Interrupt Status 0 = ssi_rxu_intr interrupt is not active after masking 1 = ssi_rxu_intr interrupt is active after masking
[1]	TXOIS	Transmit FIFO Overflow Interrupt Status 0 = ssi_txo_intr interrupt is not active after masking 1 = ssi_txo_intr interrupt is active after masking
[0]	TXEIS	Transmit FIFO Empty Interrupt Status 0 = ssi_txe_intr interrupt is not active after masking 1 = ssi_txe_intr interrupt is active after masking

4.11.6.13 Raw Interrupt Status Register (RISR)

Register	Offset	R/W	Description	Reset Value
RISR x = 0,1,2,3	SPIx_BA +0x34	R	Raw Interrupt Status Register	0x0000_0000

Bits	Description	
[31: 6]	Reserved	Reserved
[5]	MSTIR	Multi-Master Contention Raw Interrupt Status. This bit field is not present if the SPI is configured as a serial-slave device. 0 = ssi_mst_intr interrupt is not active prior to masking 1 = ssi_mst_intr interrupt is active prior masking
[4]	RXFIR	Receive FIFO Full Raw Interrupt Status 0 = ssi_rxf_intr interrupt is not active prior to masking 1 = ssi_rxf_intr interrupt is active prior to masking
[3]	RXOIR	Receive FIFO Overflow Raw Interrupt Status 0 = ssi_rxo_intr interrupt is not active prior to masking 1 = ssi_rxo_intr interrupt is active prior masking
[2]	RXUIR	Receive FIFO Underflow Raw Interrupt Status 0 = ssi_rxu_intr interrupt is not active prior to masking 1 = ssi_rxu_intr interrupt is active prior to masking
[1]	TXOIR	Transmit FIFO Overflow Raw Interrupt Status 0 = ssi_txo_intr interrupt is not active prior to masking 1 = ssi_txo_intr interrupt is active prior masking
[0]	TXEIR	Transmit FIFO Empty Raw Interrupt Status

		0 = ssi_txe_intr interrupt is not active prior to masking 1 = ssi_txe_intr interrupt is active prior masking

4.11.6.14 Transmit FIFO Overflow Interrupt Clear Register (TXOICR)

Register	Offset	R/W	Description	Reset Value
TXOICR x = 0,1,2,3	SPIx_BA +0x38	R	Transmit FIFO Overflow Interrupt Clear Register	0x0000_0000

Bits	Description	
[31: 1]	Reserved	Reserved
[0]	TXOICR	Clear Transmit FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr interrupt; writing has no effect

4.11.6.15 Receive FIFO Overflow Interrupt Clear Register (RXOICR)

Register	Offset	R/W	Description	Reset Value
RXOICR x = 0,1,2,3	SPIx_BA +0x3C	R	Receive FIFO Overflow Interrupt Clear Register	0x0000_0000

Bits	Description	
[31: 1]	Reserved	Reserved
[0]	RXOICR	Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect.

4.11.6.16 Receive FIFO Underflow Interrupt Clear Register (RXUICR)

Register	Offset	R/W	Description	Reset Value
RXUICR x = 0,1,2,3	SPIx_BA +0x40	R	Receive FIFO Underflow Interrupt Clear Register	0x0000_0000

Bits	Description	
[31: 1]	Reserved	Reserved
[0]	RXUICR	Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect.

4.11.6.17 Multi-Master Interrupt Clear Register (MSTICR)

Register	Offset	R/W	Description	Reset Value
MSTICR	SPIx_BA +0x44	R	Multi-Master Interrupt Clear Register	0x0000_0000

x = 0,1,2,3				

Bits	Description	
[31: 1]	Reserved	Reserved
[0]	MSTICR	Clear Multi-Master Contention Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect.

4.11.6.18 Interrupt Clear Register (ICR)

Register	Offset	R/W	Description	Reset Value
ICR x = 0,1,2,3	SPIx_BA +0x48	R	Interrupt Clear Register	0x0000_0000

Bits	Description	
[31: 1]	Reserved	Reserved
[0]	ICR	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect..

4.11.6.19 DMA Control Register (DMACR)

Register	Offset	R/W	Description	Reset Value
DMACR x = 0,1,2,3	SPIx_BA +0x4C	R/W	DMA Control Register	0x0000_0000

Bits	Description	
[31: 2]	Reserved	Reserved
[1]	TDMAE	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled
[0]	RDMAE	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel 0 = Receive DMA disabled 1 = Receive DMA enabled

4.11.6.20 DMA Transmit Data Level Register(DMATDLR)

Register	Offset	R/W	Description	Reset Value
DMATDLR x = 0,1,2,3	SPIx_BA +0x50	R/W	DMA Transmit Data Level Register	0x0000_0000

Bits	Description	
[31: 4]	Reserved	Reserved
[3:0]	DMATDL	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1. Refer to table below for the field decode.

DMATDL Value	Description
0000	dma_tx_req is asserted when 0 data entries are present in the transmit FIFO
0001	dma_tx_req is asserted when 1 or less data entry is present in the transmit FIFO
0010	dma_tx_req is asserted when 2 or less data entries are present in the transmit FIFO
0011	dma_tx_req is asserted when 3 or less data entries are present in the transmit FIFO
…	…
1100	dma_tx_req is asserted when 12 or less data entries are present in the transmit FIFO
1101	ssi_rxf_intr is asserted when 13 or more data entries are present in receive FIFO
1110	dma_tx_req is asserted when 14 or less data entries are present in the transmit FIFO
1111	dma_tx_req is asserted when 15 or less data entries are present in the transmit FIFO

4.11.6.21 DMA Receive Data Level Register(DMARDLR)

Register	Offset	R/W	Description	Reset Value
DMARDLR x = 0,1,2,3	SPIx_BA +0x54	R/W	DMA Receive Data Level Register	0x0000_0000

Bits	Description	
[31: 4]	Reserved	Reserved
[3:0]	DMARDL	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMACE=1. Refer to Table below for the field decode

DMARDL Value	Description
0000	dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO
0001	dma_rx_req is asserted when 2 or more data entries are present in the receive FIFO
0010	dma_rx_req is asserted when 3 or more data entries are present in the receive FIFO
0011	dma_rx_req is asserted when 4 or more valid data entries are present in the receive FIFO
…	…
1100	dma_rx_req is asserted when 13 or more data entries are present in the receive FIFO
1101	dma_rx_req is asserted when 14 or more data entries are present in the receive FIFO
1110	dma_rx_req is asserted when 15 or more data entries are present in the receive FIFO
1111	dma_rx_req is asserted when 16 data or more data entries are present in the receive FIFO

4.11.6.22 Identification Register (IDR)

Register	Offset	R/W	Description	Reset Value
IDR x = 0,1,2,3	SPIx_BA +0x58	R	Identification Register	N/A

Bits	Description	
[31: 0]	IDCODE	Identification Code. This register contains the peripherals identification code, which is written into the register at configuration time using core Consultant.

4.11.6.23 Data Register (DR)

Register	Offset	R/W	Description	Reset Value
DR x = 0,1,2,3	SPIx_BA +0x60 - 0xEC	R/W	Data Register.	0x0000_0000

Bits	Description	
[31: 0]	DR	Data Register When writing to this register, you must right-justify the data. Read data are automatically right-justified. Read = Receive FIFO buffer Write = Transmit FIFO buffer

		Note: The DR register in the DW_apb_ssi occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus.

4.11.6.24 Reserved location for future use (RSVD_1)

Register	Offset	R/W	Description	Reset Value
Reserved location for future use Register x = 0,1,2,3	SPIx_BA +0xF8	N/A	Reserved location for future use	0x0000_0000

Bits	Description	
[31:0]	Reserved	Reserved

4.11.6.25 Reserved location for future use (RSVD_2)

Register	Offset	R/W	Description	Reset Value
Reserved location for future usex = 0,1,2,3	SPIx_BA +0xFC	N/A	Reserved location for future use	0x0000_0000

Bits	Description	
[31:0]	Reserved	Reserved

4.12 DMA Serial Interface Controller (DMA)

4.12.1 Overview

The DMA connects to the Advanced High-performance Bus (AHB). It can realize direct transmission of data without the participation of CPU, which can reduce the latency on the bus, improves system performance, and reduce power consumption greatly.

4.12.2 Features

- Up to 3 channels, one per source and destination pair
- One AHB master interface
- DMA to or from APB peripherals through the APB bridge
- Handshaking interfaces for source and destination peripherals (up to 16)
- DMA flow control
- Support for memory-to-memory, memory-to-peripheral, peripheral-to-memory, and peripheral-to-peripheral
- DMA transfers
- Little Endian

4.12.3 Block Diagram

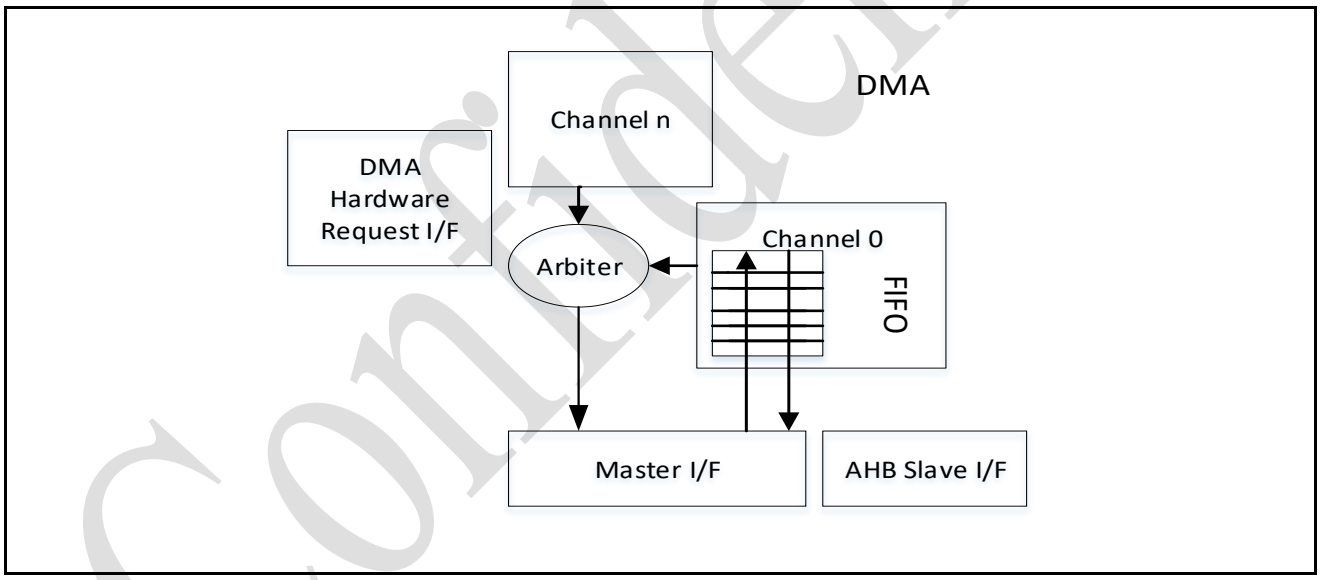


Figure 4-90 DMA Controller Block Diagram

4.12.4 Functional Description

4.12.4.1 AHB master interfsce

The DMA contains one full AHB masters.Figure 4-91 describes s a block diagram that shows the master connected into a system. This enables, for example, the DMA to transfer data directly from the memory connected to AHB port 1 to any AHB peripheral connected to AHB port 2. It also enables transactions between the DMA and any APB peripheral to occur independently of transactions on AHB bus 1.PAN1020 enables the peripherals, including UART, I2C and SPI, to interface to a DMA controller over the bus using a handshaking interface for transfer requests.

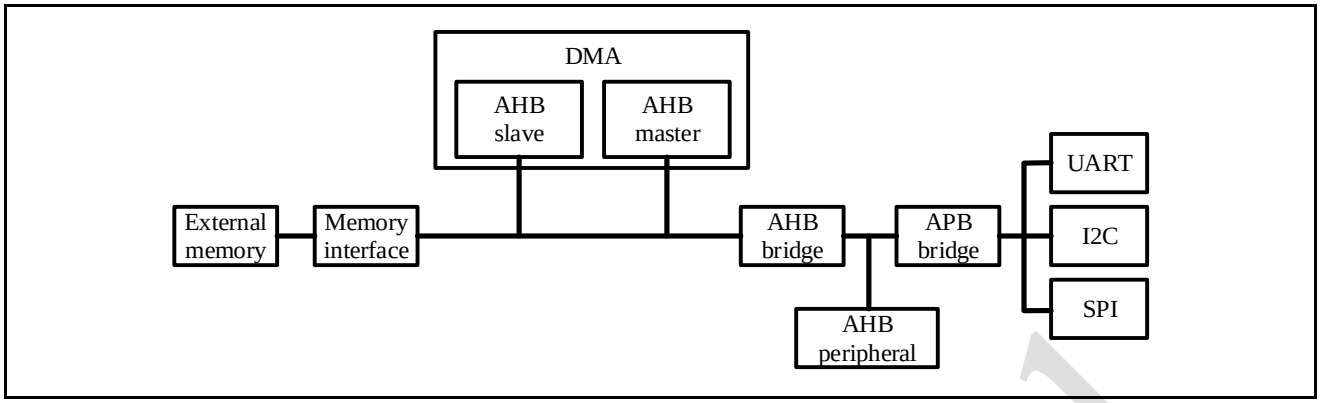


Figure 4-91 DMA Transfer Hierarchy

4.12.4.2 Handshaking Interface

Handshaking interfaces are used at the transaction level to control the flow of single or burst transactions. When the DMA is the flow controller, the DMA tries to efficiently transfer the data using as little of the bus bandwidth as possible. Generally, the DMA tries to transfer the data using burst transactions and, where possible, fill or empty the channel FIFO in single bursts – provided that the software has not limited the burst length.

A non-memory peripheral can request a DMA transfer through the DMA controller using one of two types of handshaking interfaces:

- Hardware handshaking

Figure 4-92 illustrates the hardware handshaking interface between a peripheral – whether a destination or source – and the DMA when the DMAC is the flow controller.

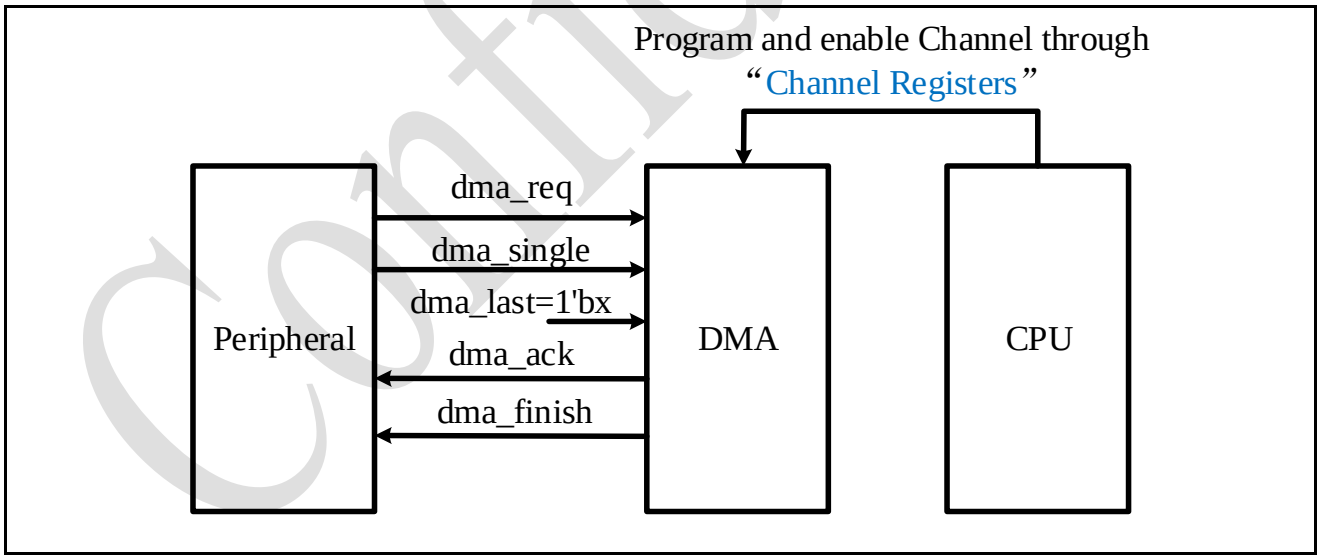


Figure 4-92 Hardware Handshaking Interface

- Software handshaking

When the slave peripheral requires the DMA controller to perform a DMA transaction, it communicates this request by sending an interrupt to the CPU or interrupt controller. The interrupt service routine then uses the software register, which shows in the Figure 4-93.

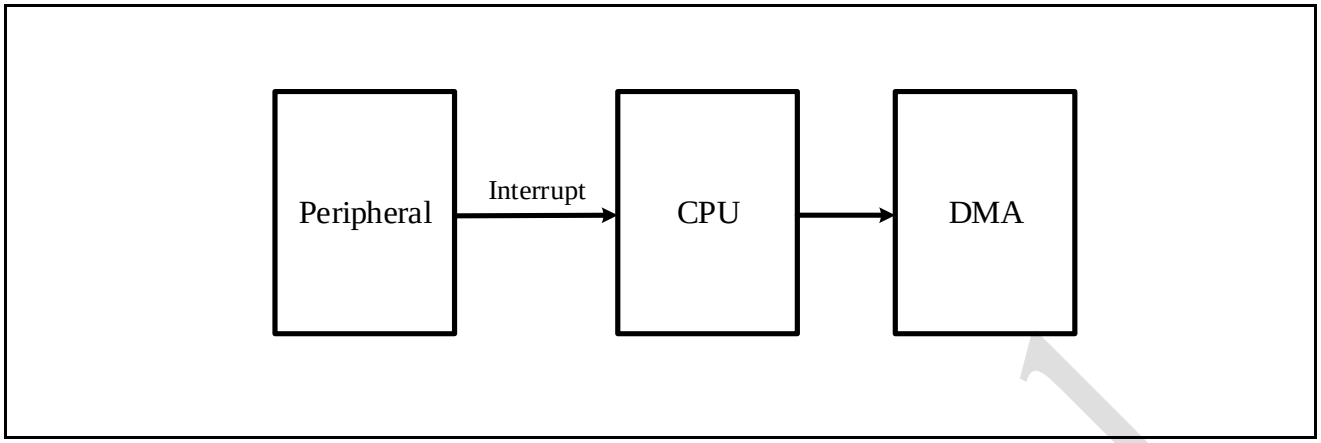


Figure 4-93 Software Controlled DMA Transfers

On completion of the transaction – single or burst – the relevant channel bit in the ReqSrcReg/ReqDstReg register is cleared by hardware. Software can therefore poll this bit in order to determine when the requested transaction has completed. Alternatively, the IntSrcTran or IntDstTran interrupts can be enabledand unmasked in order to generate an interrupt when the requested transaction – single or burst – has completed.

Software selects between the hardware or software handshaking interface on a per-channel basis. Software handshaking is accomplished through memory-mapped registers, while hardware handshaking is accomplished using a dedicated handshaking interface.

4.12.4.3 Block Flow Controller and Transfer Type

The device that controls the length of a block is known as the flow controller. Either the DMA, the source peripheral, or the destination peripheral must be assigned as the flow controller. The PAN1020 surpports only the DMA to be the flow controller.

The following transfer types are supported:

- Peripheral to Peripheral
- Memory to Memory
- Memory to Peripheral
- Peripheral to Memory

The transfer types can be set through decoding the CTLx.TT_FC. Table 4-24 lists the decoding for CTLx.TT_FC field.

Table 4-24 CTLx.TT_FC Field Decoding

CTL*x*.TT_FC Field	Transfer Type
000	Memory to Memory
001	Memory to Peripheral
010	Peripheral to Memory
011	Peripheral to Peripheral
others	-

- Peripheral to Peripheral DMAC transfer

Figure 4-94 illustrates a peripheral-to-peripheral DMAC transfer, where peripheral A (source) uses a hardware handshaking interface, and peripheral B (destination) uses a software handshaking

interface. For example, the request to send data to peripheral B is originated by the CPU, while writing to peripheral B is handled by the DMAC. The channel source and destination arbitrate independently for the AHB master, which are described in Arbitration for AHB Master Interface in detail.

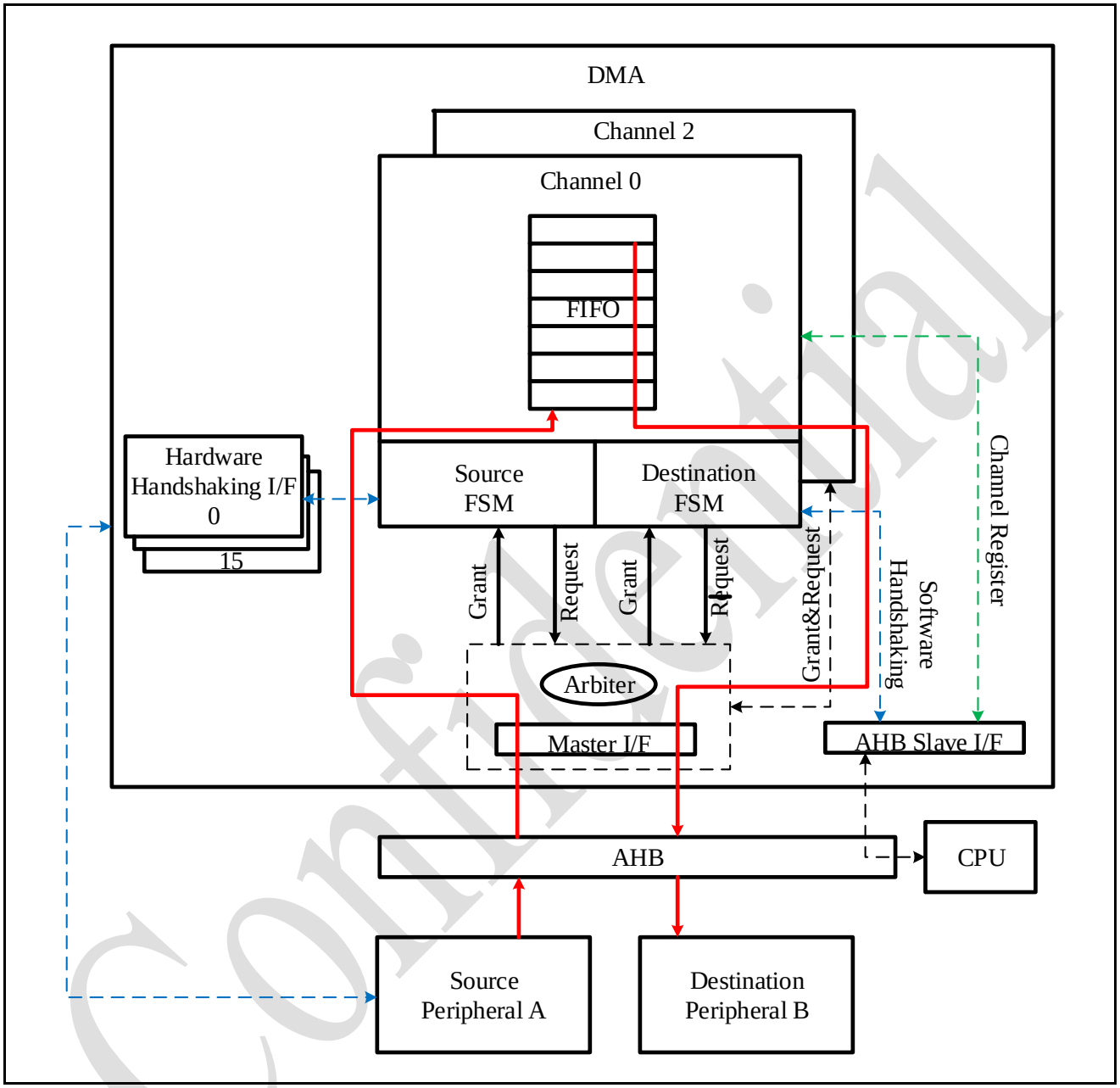


Figure 4-94 Peripheral-to-Peripheral DMA Transfer

4.12.4.4 Basic Interface Definitions

The following definitions are used in this chapter:

- Source single transaction size in bytes

  src_single_size_bytes = CTLx.SRC_TR_WIDTH/8 (1)

- Source burst transaction size in bytes

  *src_burst_size_bytes = CTLx.SRC_MSIZE * src_single_size_bytes* (2)

- Destination single transaction size in bytes

  dst_single_size_bytes = CTLx.DST_TR_WIDTH/8 (3)

- Destination burst transaction size in bytes

  *dst_burst_size_bytes = CTLx.DEST_MSIZE * dst_single_size_bytes* (4)

- Block size in bytes:

  - DMAC is flow controller – With the DMAC as the flow controller, the processor programs the DMAC with the number of data items (block size) of source transfer width (CTLx.SRC_TR_WIDTH) to be transferred by the DMAC in a block transfer; this is programmed into the CTLx.BLOCK_TS field. Therefore, the total number of bytes to be transferred in a block is:

    *blk_size_bytes_DMAC = CTLx.BLOCK_TS * src_single_size_bytes* (5)

4.12.4.5 Single-block Transfer

The PAN1020 only supports single-block transfer. As shown in Figure 4-95, a single block is made up of numerous transactions – single and burst – which are in turn composed of AHB transfers. A peripheral requests a transaction through the handshaking interface to the DMAC.

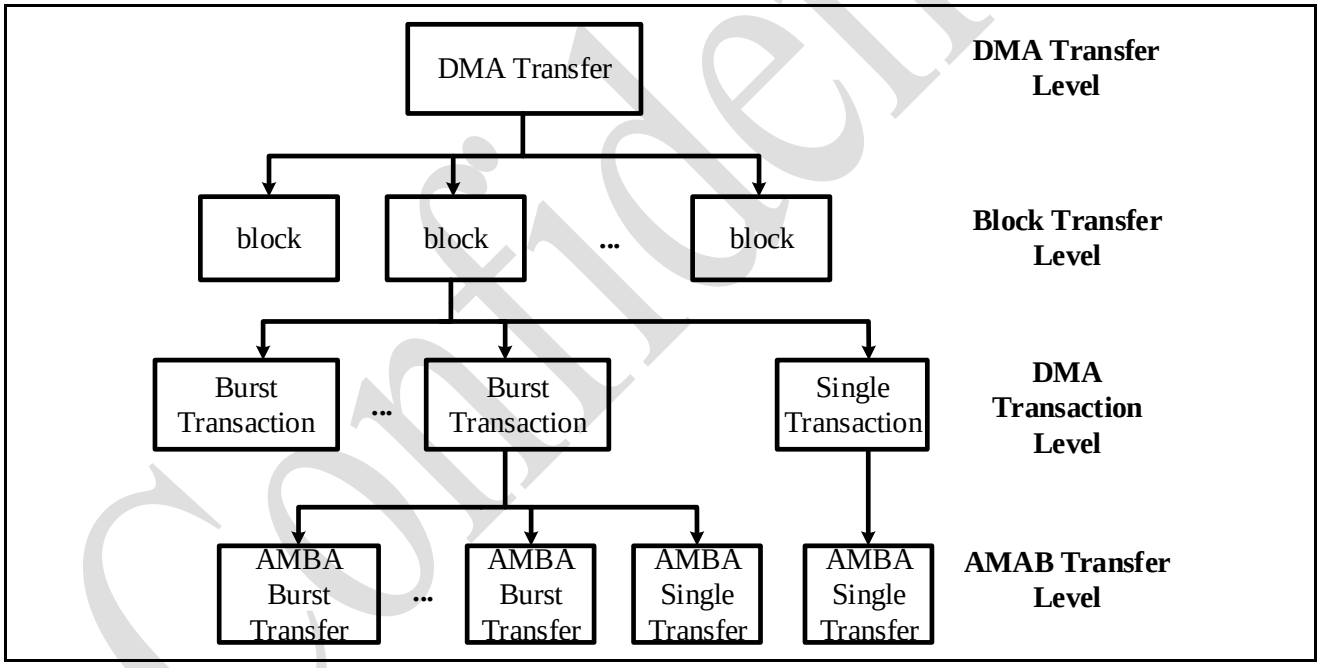


Figure 4-95 DMA Transfer Hierarchy

The programing processes are shown as follow:

1. Read the Channel Enable register to choose a free (disabled) channel; refer to "ChEnReg".
2. Clear any pending interrupts on the channel from the previous DMA transfer by writing to the Interrupt Clear registers: ClearTfr, ClearBlock, ClearSrcTran, ClearDstTran, and ClearErr. Reading the Interrupt Raw Status and Interrupt Status registers confirms that all interrupts have been cleared.
3. Program the following channel registers:
   a. Write the starting source address in the SARx register for channel x
   b. Write the starting destination address in the DARx register for channel x
   c. Program CTLx and CFGx.

d. Write the control information for the DMA transfer in the CTLx register for channel x. For example, in the register, you can program the following:

i. Set up the transfer type (memory or non-memory peripheral for source and destination) and flow control device by programming the TT_FC of the CTLx register.

ii. Set up the transfer characteristics, such as:

- Transfer width for the source in the SRC_TR_WIDTH field.
- Transfer width for the destination in the DST_TR_WIDTH field.
- Incrementing/decrementing or fixed address for the source in the SINC field.
- Incrementing/decrementing or fixed address for the destination in the DINC field.

e. Write the channel configuration information into the CFGx register for channel x (see page 180).

i. Designate the handshaking interface type (hardware or software) for the source and destination peripherals; this is not required for memory.

This step requires programming the HS_SEL_SRC/HS_SEL_DST bits, respectively. Writing a 0 activates the hardware handshaking interface to handle source/destination requests.Writing a 1 activates the software handshaking interface to handle source and destination requests.

ii. If the hardware handshaking interface is activated for the source or destination peripheral, assign a handshaking interface to the source and destination peripheral; this requires programming the SRC_PER and DEST_PER bits, respectively.

4. Ensure that bit 0 of the DMACfgReg register is enabled before writing to ChEnReg.

5. Source and destination request single and burst DMA transactions in order to transfer the block of data (assuming non-memory peripherals). The DMAC acknowledges at the completion of every transaction (burst and single) in the block and carries out the block transfer.

6. Once the transfer completes, hardware sets the interrupts and disables the channel. At this time, you can respond to either the Block Complete or Transfer Complete interrupts, or poll for the transfer complete raw interrupt status register (RawTfr[n], n = channel number) until it is set by hardware, in order to detect when the transfer is complete. Note that if this polling is used, the software must ensure that the transfer complete interrupt is cleared by writing to the Interrupt Clear register, ClearTfr[n], before the channel is enabled.

The flow diagram in Figure 4-96 shows an overview of programming the DMA.

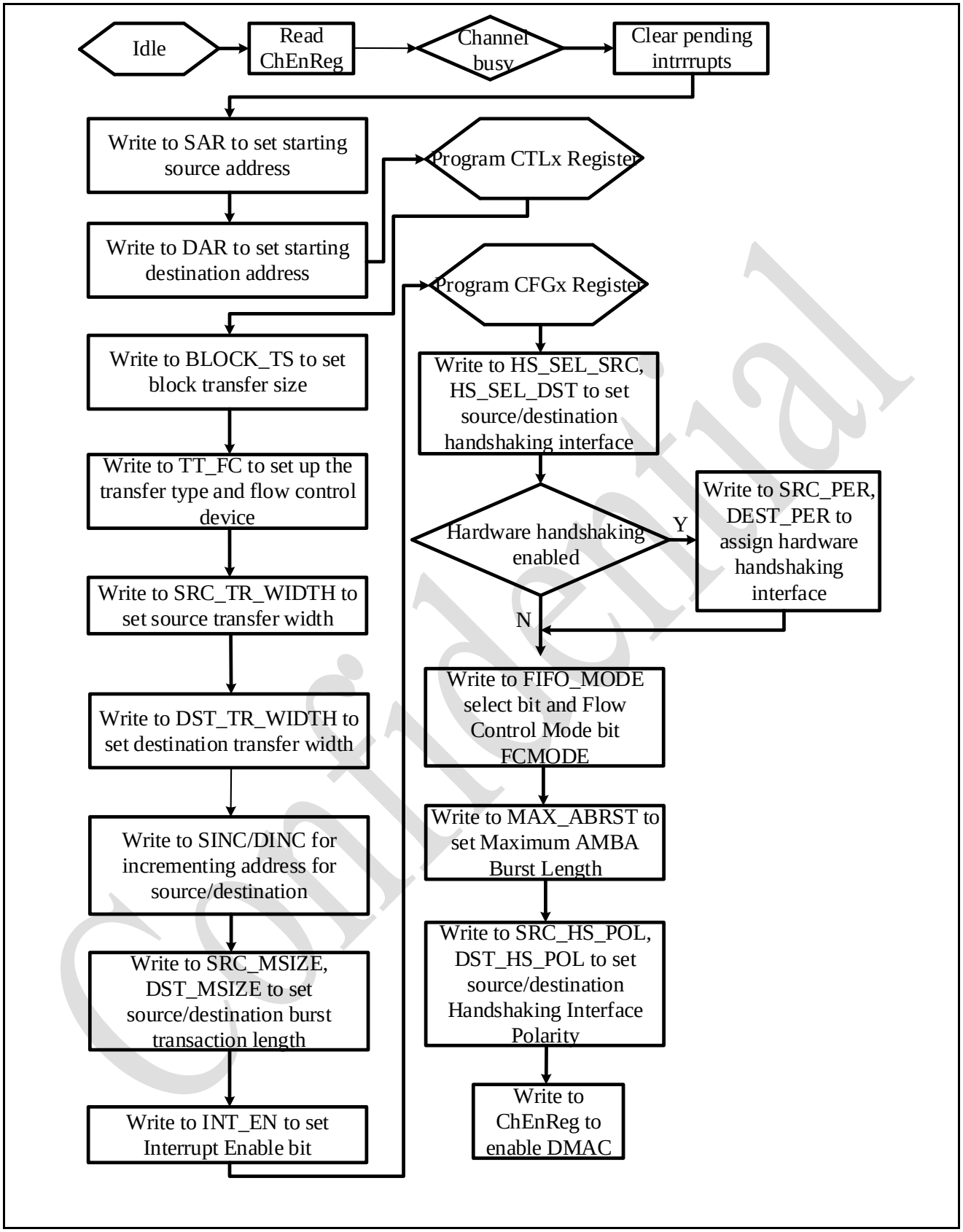


Figure 4-96 Flowchart for single-block DMAC Programming

4.12.4.6 Peripheral Burst Transaction Requests

For a source FIFO, an active edge is triggered on dma_req when the source FIFO exceeds some watermark level. For a destination FIFO, an active edge is triggered on dma_req when the destination FIFO drops below some watermark level.

This section investigates the optimal settings of these watermark levels on the source and destination peripherals and their relationship to, respectively:

- Source transaction length, CTLx.SRC_MSIZE
- Destination transaction length, CTLx.DEST_MSIZE

- Transmit Watermark Level and Transmit FIFO Underflow

During SPI serial transfers, SPI transmit FIFO requests are made to the DMAC whenever the number of entries in the SPI transmit FIFO is less than or equal to the SPI Transmit Data Level Register (SSI.DMATDLR) value. This is known as the watermark level. The DMAC responds by writing a burst of data to the SPI transmit FIFO buffer, of length DMAC.CTLx.DEST_MSIZE.

Data should be fetched from the DMAC often enough for the SPI transmit FIFO to continuously perform serial transfers; that is, when the SPI transmit FIFO begins to empty, another burst transaction request should be triggered. Otherwise the SPI transmit FIFO runs out of data (underflow). To prevent this condition, the user must set the watermark level correctly.

- Receive Watermark Level and Receive FIFO Overflow

During SPI serial transfers, SPI receive FIFO requests are made to the DMAC whenever the number of entries in the SPI receive FIFO is at or above the DMA Receive Data Level Register; that is, SSI.DMARDLR+1. This is known as the watermark level. The DMAC responds by reading a burst of data from the receive FIFO buffer of length DMA.CTLx.SRC_MSIZE.

Data should be fetched by the DMAC often enough for the SPI receive FIFO to accept serial transfers continuously; that is, when the SPI receive FIFO begins to fill, another burst transaction request should be triggered. Otherwise, the SPI receive FIFO fills with data (overflow). To prevent this condition, the user must correctly set the watermark level.

4.12.4.7 Generating Requests for the AHB Master Bus Interface

Each channel has a source state machine and destination state machine running in parallel. These state machines generate the request inputs to the arbiter, which arbitrates for the master bus interface (one arbiterper master bus interface).

When the source/destination state machine is granted control of the master bus interface, and when the master bus interface is granted control of the external AHB bus, then AHB transfers between the peripheral and the DMAC (on behalf of the granted state machine) can take place.

AHB transfers from the source peripheral or to the destination peripheral cannot proceed until the channel FIFO is ready. For burst transaction requests and for transfers involving memory peripherals, the criterion for “FIFO readiness” is controlled by the FIFO_MODE field of the CFGx register.

The definition of FIFO readiness is the same for:

- Single transactions
- Burst transactions, where CFGx.FIFO_MODE = 0
- Transfers involving memory peripherals, where CFGx.FIFO_MODE = 0

The channel FIFO is deemed ready when the space/data available is sufficient to complete a single AHB transfer of the specified transfer width. FIFO readiness for source transfers occurs when the channel FIFO contains enough room to accept at least a single transfer of CTLx.SRC_TR_WIDTH width. FIFO readiness for destination transfers occurs when the channel FIFO contains data to form at least a single transfer of CTLx.DST_TR_WIDTH width.

When *CFGx.FIFO_MODE = 1*, then the criteria for FIFO readiness for burst transaction requests and transfers involving memory peripherals are as follows:

- A FIFO is ready for a source burst transfer when the FIFO is less than half empty.
- A FIFO is ready for a destination burst transfer when the FIFO is greater than or equal to half full.

Exceptions to this "readiness" occur. During these exceptions, a value of *CTLx.FIFO_MODE = 0* is assumed.

The following are the exceptions:

- Near the end of a burst transaction or block transfer – The channel source state machine does not wait for the channel FIFO to be less than half empty if the number of source data items left to complete thesource burst transaction or source block transfer is less than *DMAH_CHx_FIFO_DEPTH/2*. Similarly, the channel destination state machine does not wait for the channel FIFO to be greater than or equal to half full, if the number of destination data items left to complete the destination bursttransaction or destination block transfer is less than *DMAH_CHx_FIFO_DEPTH/2*.
- In FIFO flush mode
- When a channel is suspended – The destination state machine does not wait for the FIFO to become half empty to flush the FIFO, regardless of the value of the FIFO_MODE field.
- When the source/destination peripheral is not memory, the source/destination state machine waits for a single/burst transaction request. Upon receipt of a transaction request and only if the channel FIFO is"ready" for source/destination AHB transfers, a request for the master bus interface is made by the source/destination state machine.

4.12.4.8 Interrupt

There are five interrupt registers in the DMA:

- IntBlock – Block Transfer Complete Interrupt
- This interrupt is generated on DMA block transfer completion to the destination peripheral.
- IntDstTran – Destination Transaction Complete Interrupt
- This interrupt is generated after completion of the last AHB transfer of the requested single/burst transaction from the handshaking interface (either the hardware or software handshaking interface) on the destination side.
- IntErr – Error Interrupt
- This interrupt is generated when an ERROR response is received from an AHB slave on the HRESP bus during a DMA transfer. In addition, the DMA transfer is cancelled and the channel is disabled.
- IntSrcTran – Source Transaction Complete Interrupt
- This interrupt is generated after completion of the last AHB transfer of the requested single/burst transaction from the handshaking interface (either the hardware or software handshaking interface) on the source side.
- IntTfr – DMA Transfer Complete Interrupt

This interrupt is generated on DMA transfer completion to the destination peripheral.

When a channel has been enabled to generate interrupts, the following is true:

- Interrupt events are stored in the Raw Status registers.
- The contents of the Raw Status registers are masked with the contents of the Mask registers.
- The masked interrupts are stored in the Status registers.
- The contents of the Status registers are used to drive the int_* port signals.
- Writing to the appropriate bit in the Clear registers clears an interrupt in the Raw Status registers and the Status registers on the same clock cycle.

Upon occurrence of an error in an AHB transfer, the following occurs:

- DMA transfer in progress stops immediately
- Relevant channel is disabled,
- An interrupt is issued (if not masked)

If multiple channels are enabled, only the one where the AHB error was detected is disabled.

The contents of the FIFO are not cleared, but they become inaccessible and are overwritten once the channel is re-enabled to start a new sequence.

There is no support for automatically resuming the transfer from the point where the error occurred, and the full block transfer has to be re-initiated in order to be successfully completed.

The DMA does not use the hardware handshaking interface to signal the error occurrence in any way, nor does it signal the end of a transfer. In practice, this means that if a request from a peripheral is active when the error occurs—dma_req is high if peripheral is the flow controller; dma_req or dma_single are high if peripheral is not the flow controller—the channel is disabled without the DMA ever asserting dma_ack (or dma_finish).

The hardware handshaking interface on the peripheral side has to be re-initiated by the CPU upon detection of the error interrupt. The dma_req signal needs to be brought low before the channel is re-enabled and thenbrought high when the channel has been enabled.

4.12.4.9 Arbitration for AHB Master Interface

Each DMAC channel has two request lines that request ownership of a particular master bus interface: channel source and channel destination request lines.

Source and destination arbitrate separately for the bus. Once a source/destination state machine gains ownership of the master bus interface and the master bus interface has ownership of the AHB bus, then AHB transfers can proceed between the peripheral and DMAC. Figure 4-97 illustrates the arbitration flow of the master bus interface.

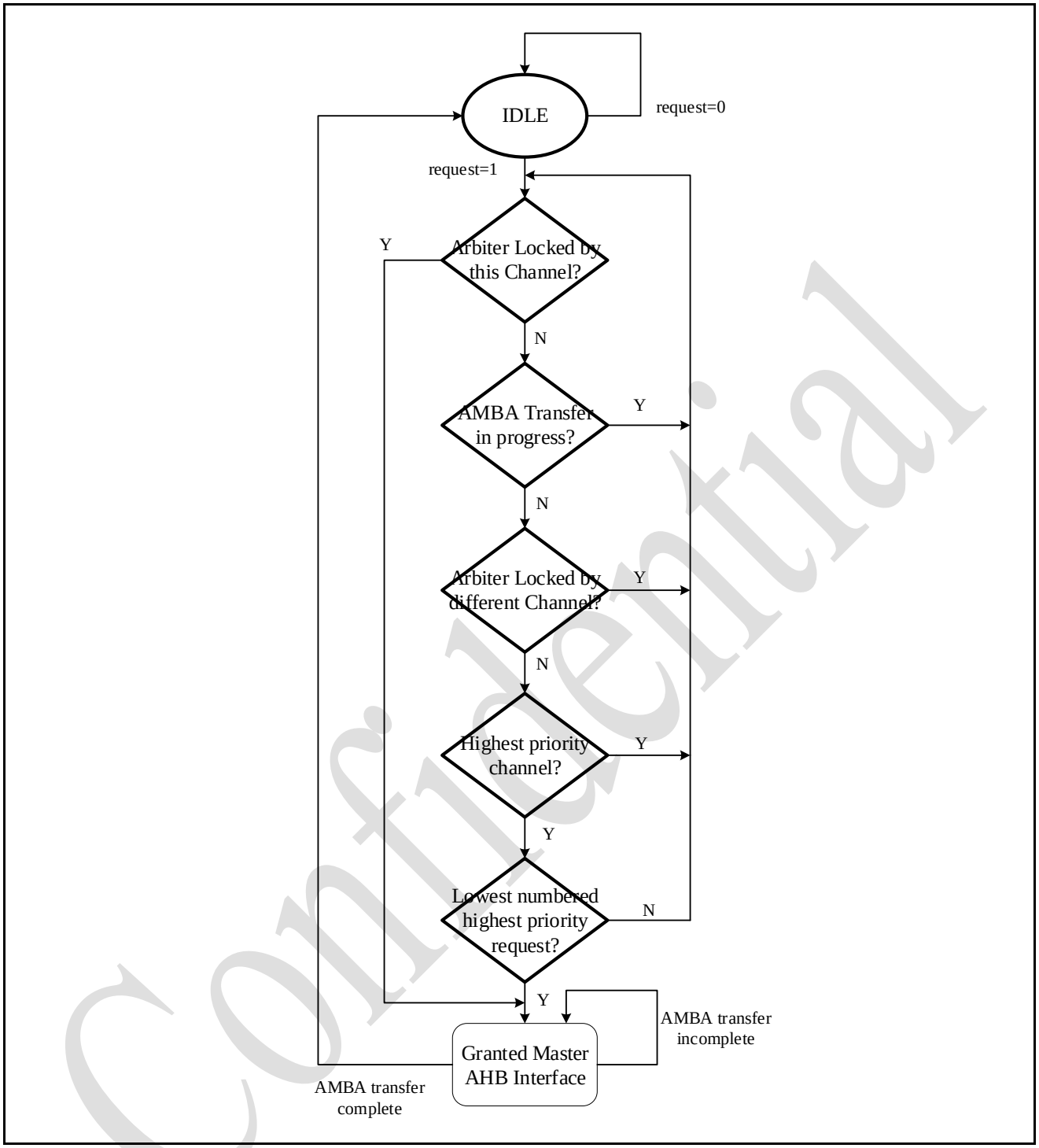


Figure 4-97 Arbitration Flow for Master Bus Interface

An arbitration scheme is used to decide which of the request lines (2 * DMAH_NUM_CHANNELS) is granted the particular master bus interface. Each channel has a programmable priority. A request for the master bus interface can be made at any time, but is granted only after the current AHB transfer (burst or single) has completed. Therefore, if the master interface is transferring data for a lower priority channel and a higher priority channel requests service, then the master interface will complete the current burst for the lower priority channel before switching to transfer data for the higher priority channel.

To prevent a channel from saturating the master bus interface, it can be given a maximum AMBA

burst length (MAX_ABRST field in CFGx register) at channel setup time. This also prevents the master bus interface from saturating the AHB bus where the system arbiter cannot change the grant lines until the end of an undefined length burst.

4.12.5 Register Map

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

Register	Offset	R/W	Description	Reset Value
DMA Base Address: **DMA_BA = 0x5000_2000**				
SAR0	DMA_BA+0x000	R/W	Source Address Register for Channel 0	0x0
DAR0	DMA_BA+0x008	R/W	Destination Address Register for Channel 0	0x0
CTL0	DMA_BA+0x018	R/W	Control Register for Channel 0	Configuration dependent
CFG0	DMA_BA+ 0x040	R/W	Configuration Register for Channel 0	0x0
SAR1	DMA_BA+0x058	R/W	Source Address Register for Channel 1	0x0
DAR1	DMA_BA+0x060	R/W	Destination Address Register for Channel 1	0x0
CTL1	DMA_BA+0x070	R/W	Control Register for Channel 1	Configuration dependent
CFG1	DMA_BA+ 0x098	R/W	Configuration Register for Channel 1	0x0
SAR2	DMA_BA+0x0B0	R/W	Source Address Register for Channel 2	0x0
DAR2	DMA_BA+0x0B8	R/W	Destination Address Register for Channel 2	0x0
CTL2	DMA_BA+0x0C8	R/W	Control Register for Channel 2	Configuration dependent
CFG2	DMA_BA+ 0x0F0	R/W	Configuration Register for Channel 2	0x0
RawTfr	DMA_BA+0x2C0	R/W	Interrupt Raw Status Registers	0x0
RawBlock	DMA_BA+0x2C8	R/W	Interrupt Raw Status Registers	0x0
RawSrcTran	DMA_BA+ 0x2D0	R/W	Interrupt Raw Status Registers	0x0
RawDstTran	DMA_BA+0x2D8	R/W	Interrupt Raw Status Registers	0x0
RawErr	DMA_BA+0x2E0	R/W	Interrupt Raw Status Registers	0x0
StatusTfr	DMA_BA+0x2E8	R	Interrupt Status Registers	0x0
StatusBlock	DMA_BA+0x2F0	R	Interrupt Status Registers	0x0
StatusSrcTran	DMA_BA+0x2F8	R	Interrupt Status Registers	0x0
StatusDstTran	DMA_BA+0x300	R	Interrupt Status Registers	0x0
StatusErr	DMA_BA+0x308	R	Interrupt Status Registers	0x0
MaskTfr	DMA_BA+0x310	R/W	Interrupt Mask Registers	0x0
MaskBlock	DMA_BA+0x318	R/W	Interrupt Mask Registers	0x0
MaskSrcTran	DMA_BA+0x320	R/W	Interrupt Mask Registers	0x0
MaskDstTran	DMA_BA+0x328	R/W	Interrupt Mask Registers	0x0
MaskErr	DMA_BA+0x330	R/W	Interrupt Mask Registers	0x0
ClearTfr	DMA_BA+0x338	W	Interrupt Clear Registers	0x0
ClearBlock	DMA_BA+0x340	W	Interrupt Clear Registers	0x0

ClearSrcTran	DMA_BA+0x348	W	Interrupt Clear Registers	0x0
ClearDstTran	DMA_BA+0x350	W	Interrupt Clear Registers	0x0
ClearErr	DMA_BA+0x358	W	Interrupt Clear Registers	0x0
Statusint	DMA_BA+0x360	R	Combined Interrupt Status Register	0x0
ReqSrcReg	DMA_BA+0x368	R/W	Source Software Transaction Request Register	0x0
ReqDstReg	DMA_BA+0x370	R/W	DestinationSoftware Transaction Request Register	0x0
SglReqSrcReg	DMA_BA+0x378	R/W	Destination Software Transaction Request Register	0x0
SglReqDstReg	DMA_BA+0x380	R/W	Destination Software Transaction Request Register	0x0
LstSrcReg	DMA_BA+0x388	R/W	Last Source Transaction Request Register	0x0
LstDstReg	DMA_BA+0x390	R/W	Last Destination Transaction Request Register	0x0
DmaCfgReg	DMA_BA+0x398	R	DMA Configuration Register	0x0
ChEnReg	DMA_BA+0x3A0	R/W	DMA Channel Enable Register	0x0
DmaIdReg	DMA_BA+0x3A8	R/W	DMA ID Register	0x0
DMA_COMP_PARAMS_3	DMA_BA+0x3E0	R	DMA Component Parameters Register 3	0x1001000010010000
DMA_COMP_PARAMS_2	DMA_BA+0x3E8	R	DMA Component Parameters Register 2	0x1001_0000
DMA_COMP_PARAMS_1	DMA_BA+0x3F0	R	DMA Component Parameters Register 1	0x2100_0204_0000_0888
DCIR	DMA_BA+0x3F8	R	DMA Component ID Register	0x4457_1110

4.12.6 Register Description

4.12.6.1 Configuration and Channel Enable Register

4.12.6.1.1. DMA Configuration Register (DmaCfgReg)

Register	Offset	R/W	Description	Reset Value
DmaCfgReg	DMA_BA+0x398	R/W	DMA Configuration Register	0x0

Bits	Description	
[63:1]	Reserved	Reserved.
[0]	DMA_EN	DMA Enable bit. 0 = DMA Disabled 1 = DMA Enabled.

4.12.6.1.2. DMA Channel Enable Register (ChEnReg)

Register	Offset	R/W	Description	Reset Value
ChEnReg	DMA_BA+0x3A0	R/W	DMA Channel Enable Register	0x0

Bits	Description	
[63:11]	Reserved	Reserved.
[10:8]	CH_EN_WE	Channel enable write enable(only writable)
[7:3]	Reserved	Reserved
[2:0]	CH_EN	Enables/Disables the channel. Setting this bit enables a channel; clearing this bit disables the channel. 0 = Disable the Channel 1 = Enable the Channel The *CH_EN* bit is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer. **Note**: All bits of this register are cleared to 0 when the global DMA channel enable bit, DmaCfgReg[0], is 0.

4.12.6.2 Channel Register

4.12.6.2.1. Source Address Register for Channel x Register (SARx)

Register	Offset	R/W	Description	Reset Value
SARx x=[0,1,2] SAR0 – 0x000 SAR1 – 0x058 SAR2 – 0x0B0	DMA_BA+SARx	R/W	Source Address Register for Channel x	0x0

Bits	Description	
[63:32]	Reserved	Reserved.
[31:0]	SAR	Current Source Address of DMA transfer. Updated after each source transfer. The SINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every source transfer throughout the block transfer.

4.12.6.2.2. Destination Address Register for Channel x(DARx)

Register	Offset	R/W	Description	Reset Value
DAR*x* x=[0,1,2] DAR0 – 0x008 DAR1 – 0x060	DMA_BA+ DARx	R/W	Destination Address Register for Channel x	0x0

DAR2 – 0x0B8				

Bits	Description	
[63:32]	Reserved	Reserved.
[31:0]	DAR	Current Destination address of DMA transfer. Updated after each destination transfer. The DINC field in the CTLx register determines whether the address increments, decrements, or is left unchanged on every destination transfer throughout the block transfer.

4.12.6.2.3. Control Register for Channel x(CTLx)

Register	Offset	R/W	Description	Reset Value
CTL*x* x=[0,1,2] CTL0–0x018 CTL1– 0x070 CTL2– 0x0C8	DMA_BA+CTLx	R/W	Control Register for Channel x	0x0

Bits	Description	
[63:45]	Reserved	Reserved.
[44]	DONE	Done bit If status write-back is enabled, the upper word of the control register, CTLx[63:32], is written to the control register location of the Linked List Item (LLI) in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTLx.DONE bit to see when a block transfer is complete. The LLI CTLx.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel. LLI accesses are always 32-bit accesses (Hsize = 2) aligned to 32-bit boundaries and cannot be changed or programmed to anything other than 32-bit.
[43:42]	Reserved	Reserved.
[41:32]	BLOCK_TS	When the DMA is the flow controller, the user writes this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_TS indicates the total number of single transactions to perform for every block transfer; a single transaction is mapped to a single AMBA beat. **Width**: The width of the single transaction is determined by SRC_TR_WIDTH. Once the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of what is the flow controller. When the source or destination peripheral is assigned as the flow controller, then the maximum block size that can be read back saturates at 1023, but the actual block size can be greater.
[31:23]	Reserved	Reserved
[22:20]	TT_FC	Transfer Type and Flow Control. The following transfer types are supported.

<table>
<tr><td></td><td></td><td>

Memory to Memory
Memory to Peripheral
Peripheral to Memory
Peripheral to Peripheral

Flow Control can be assigned to the DMA, the source peripheral, or the destination peripheral.
<table>
<tr><th>CTLx.TT_FC Field</th><th>Transfer Type</th><th>Flow Controller</th></tr>
<tr><td>000</td><td>Memory to Memory</td><td>DMA</td></tr>
<tr><td>001</td><td>Memory to Peripheral</td><td>DMA</td></tr>
<tr><td>010</td><td>Peripheral to Memory</td><td>DMA</td></tr>
<tr><td>011</td><td>Peripheral to Peripheral</td><td>DMA</td></tr>
<tr><td>100</td><td>Peripheral to Memory</td><td>Peripheral</td></tr>
<tr><td>101</td><td>Peripheral to Peripheral</td><td>Source Peripheral</td></tr>
<tr><td>110</td><td>Memory to Peripheral</td><td>Peripheral</td></tr>
<tr><td>111</td><td>Peripheral to Peripheral</td><td>Destination Peripheral</td></tr>
</table>
</td></tr>
<tr><td>[19:17]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[16:14]</td><td>SRC_MSIZE</td><td>Source Burst Transaction Length.
Number of data items, each of width SRC_TR_WIDTH, to be read from the source every time a source burst transaction request is made from either the corresponding hardware or software handshaking interface.</td></tr>
<tr><td>[13:11]</td><td>DEST_MSIZE</td><td>Destination Burst Transaction Length.
Number of data items, each of width DST_TR_WIDTH, to be written to the destination every time a destination burst transaction request is made from either the corresponding hardware or software handshaking interface.
<table>
<tr><th>CTLx.SRC_MSIZE / CTLx.DEST_MSIZE</th><th>Number of data items to be transferred (of width CTLx.SRC_TR_WIDTH or CTLx.DST_TR_WIDTH)</th></tr>
<tr><td>000</td><td>1</td></tr>
<tr><td>001</td><td>4</td></tr>
<tr><td>010</td><td>8</td></tr>
<tr><td>011</td><td>16</td></tr>
<tr><td>100</td><td>32</td></tr>
<tr><td>101</td><td>64</td></tr>
<tr><td>110</td><td>128</td></tr>
<tr><td>111</td><td>256</td></tr>
</table>
</td></tr>
<tr><td>[10:9]</td><td>SINC</td><td>Source Address Increment.
Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to “No change.”
00 = Increment
01 = Decrement
1x = No change</td></tr>
</table>

[8:7]	DINC	Destination Address Increment. Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to "No change." 00 = Increment 01 = Decrement 1x = No change
[6:4]	SRC_TR_WIDTH	Source Transfer Width. Mapped to AHB bus "hsize." For a non-memory peripheral, typically the peripheral (source) FIFO width. This value must be less than or equal to 32.
[3:1]	DST_TR_WIDTH	Destination Transfer Width. For a non-memory peripheral, typically rgw peripheral (destination) FIFO width. This value must be less than or equal to 32.

CTLx.SRC_TR_WIDTH / CTLx.DST_TR_WIDTH	Size (bits)
000	8
001	16
010	32
011	64
100	128
101	256
11X	256

[0]	INT_EN	Interrupt Enable Bit. If set, then all interrupt-generating sources are enabled. Functions as a global mask bit for all interrupts for the channel; raw* interrupt registers still assert if INT_EN = 0.

4.12.6.2.4. Configuration Register for Channel x(CFGx)

Register	Offset	R/W	Description	Reset Value
CFG*x* x =[0,1,2] CFG0 – 0x040 CFG1 – 0x098 CFG2 – 0x0f0	DMA_BA+ CFGx	R/W	Configuration Register for Channel x	0x0

Bits	Description	
[63:47]	Reserved	Reserved.
[46:43]	DEST_PER	Assigns a hardware handshaking interface (0 - 15) to the destination of channel x if the *HS_SEL_DST* field is 0; otherwise, this field is ignored. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. **Note**: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.

[42:39]	SRC_PER	Assigns a hardware handshaking interface (0 - 15) to the source of channel x if the *HS_SEL_SRC* field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. **Note**: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
[38:37]	Reserved	Reserved
[36:34]	PROTCTL	Protection Control bits used to drive the AHB HPROT[3:1] bus. The AMBA Specification recommends that the default value of HPROT indicates a non-cached, non-buffered, privileged data access. The reset value is used to indicate such an access. HPROT[0] is tied high because all transfers are data accesses, as there are no opcode fetches. There is a one-to-one mapping of these register bits to the HPROT[3:1] master interface signals.

1'b1	HPROT[0]
CFGx.PROTCTL[1]	HPROT[1]
CFGx.PROTCTL[2]->	HPROT[2]
CFGx.PROTCTL[3]->	HPROT[3]

[33]	FIFO_MODE	FIFO Mode Select. Determines how much space or data needs to be available in the FIFO before a burst transaction request is serviced. 0 = Space/data available for single AHB transfer of the specified transfer width. 1 = Data available is greater than or equal to half the FIFO depth for destination transfers and space available is greater than half the fifo depth for source transfers. The exceptions are at the end of a burst transaction request or at the end of a block transfer.
[32]	FCMODE	Flow Control Mode. Determines when source transaction requests are serviced when the Destination Peripheral is the flow controller. 0 = Source transaction requests are serviced when they occur. Data pre-fetching is enabled. 1 = Source transaction requests are not serviced until a destination transaction request occurs. In this mode, the amount of data transferred from the source is limited so that it is guaranteed to be transferred to the destination prior to block termination by the destination. Data pre-fetching is disabled.
[31:20]	Reserved	Reserved
[19]	SRC_HS_POL	Source Handshaking Interface Polarity. 0 = Active high 1 = Active low
[18]	DST_HS_POL	Destination Handshaking Interface Polarity. 0 = Active high 1 = Active low
[17:12]	Reserved	Reserved

[11]	HS_SEL_SRC	Source Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces – hardware or software – is active for source requests on this channel. 0 = Hardware handshaking interface. Software-initiated transaction requests are ignored. 1 = Software handshaking interface. Hardware-initiated transaction requests are ignored. If the source peripheral is memory, then this bit is ignored.
[10]	HS_SEL_DST	Destination Software or Hardware Handshaking Select. This register selects which of the handshaking interfaces – hardware or software – is active for destination requests on this channel. 0 = Hardware handshaking interface. Software-initiated transaction requests are ignored. 1 = Software handshaking interface. Hardware- initiated transaction requests are ignored. If the destination peripheral is memory, then this bit is ignored.
[9]	FIFO_EMPTY	Indicates if there is data left in the channel FIFO. Can be used inconjunction with CFGx.CH_SUSP to cleanly disable a channel. For more information. 1 = Channel FIFO empty 0 = Channel FIFO not empty
[8]	CH_SUSP	Channel Suspend. Suspends all DMAC data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. Can also be used in conjunction with *FIFO_EMPTY* to cleanly disable a channel without losing any data. 0 = Not suspended. 1 = Suspend DMAC transfer from the source
[7:5]	CH_PRIOR	Channel priority. A priority of 7 is the highest priority, and 0 is the lowest. This field must be programmed within the following range: 0~2 A programmed value outside this range will cause erroneous behavior.
[4:0]	Reserved	Reserved

4.12.6.3 Interrupt Registers

4.12.6.3.1. Interrupt Raw Status Registers (IRSR)

Each bit in these registers is cleared by writing a 1 to the corresponding location in the ClearTfr, ClearBlock, ClearSrcTran, ClearDstTran, ClearErr registers.

Register	Offset	R/W	Description	Reset Value
RawBlock, RawDstTran, RawErr, RawSrcTran, RawTfr RawTfr_offset – 0x2C0 RawBlock_ offset– 0x2C8	DMA_BA+x_offset	R/W	Interrupt Raw Status Registers	0x0

RawSrcTran– 0x2D0 RawDstTran_ offset– 0x2D8 RawErr_ offset – 0x2E0				

Bits	Description	
[63:3]	Reserved	Reserved.
[2:0]	RAW	Raw interrupt status.

4.12.6.3.2. Interrupt Status Registers(ISR)

All interrupt events from all channels are stored in these Interrupt Status registers after masking: StatusBlock, StatusDstTran, StatusErr, StatusSrcTran, and StatusTfr.

Register	Offset	R/W	Description	Reset Value
StatusBlock, StatusDstTran, StatusErr, StatusSrcTran, StatusTfr StatusTfr_ offset – 0x2E8 StatusBlock_ offset – 0x2F0 StatusSrcTran_offset – 0x2F8 StatusDstTran_offset– 0x300 StatusErr_ offset – 0x308	DMA_BA+x_offset	R	Interrupt Status Registers	0x0

Bits	Description	
[63:3]	Reserved	Reserved.
[2:0]	STATUS	Interrupt status.

4.12.6.3.3. Interrupt Mask Registers (IMR)

Register	Offset	R/W	Description	Reset Value
MaskBlock, MaskDstTran, MaskErr, MaskSrcTran, MaskTfr MaskTfr_offset – 0x310 MaskBlock_offset – 0x318 MaskSrcTran_offset – 0x320 MaskDstTran_offset – 0x328 MaskErr_offset – 0x330	DMA_BA+x_offset	R/W	Interrupt Mask Registers	0x0

Bits	Description	
[63:11]	Reserved	Reserved.
[10:8]	INT_MASK_WE	Interrupt Mask Write Enable 0 = write disabled 1 = write enabled
[7:3]	Reserved	Reserved

[2:0]	INT_MASK	Interrupt Mask 0 = masked 1 = unmasked **Note**: A channel *INT_MASK* bit will be written only if the corresponding mask write enable bit in the *INT_MASK_WE* field is asserted on the same AHB write transfer. This allows software to set a mask bit without performing a read-modified write operation.

4.12.6.3.4. Interrupt Clear Registers (ICR)

Each bit in the Raw Status and Status registers is cleared on the same cycle by writing a 1 to the corresponding location in the Clear registers: ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, and ClearTfr.

Register	Offset	R/W	Description	Reset Value
ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, ClearTfr ClearTfr – 0x338 ClearBlock – 0x340 ClearSrcTran – 0x348 ClearDstTran – 0x350 ClearErr – 0x358	DMA_BA+x_offset	W	Interrupt Clear Registers	0x0

Bits	Description	
[63:3]	Reserved	Reserved.
[2:0]	CLEAR	Interrupt clear. 0 = no effect 1 = clear interrupt

4.12.6.3.5. Combined Interrupt Status Register (Statusint)

Register	Offset	R/W	Description	Reset Value
Statusint	DMA_BA+0x360	R	Combined Interrupt Status Register	0x0

Bits	Description	
[63:5]	Reserved	Reserved.
[4]	ERR	OR of the contents of StatusErr register
[3]	DSTT	OR of the contents of StatusDst register.
[2]	SRCT	OR of the contents of StatusSrcTran register.
[1]	BLOCK	OR of the contents of StatusBlock register.
[0]	TFR	OR of the contents of StatusTfr register.

4.12.6.4 Software Handshaking Registers

4.12.6.4.1. Source Software Transaction Request Register (ReqSrcReg)

Register	Offset	R/W	Description	Reset Value
ReqSrcReg	DMA_BA+0x368	R/W	Source Software Transaction Request Register	0x0

Bits	Description	
[63:11]	Reserved	Reserved.
[10:8]	SRC_REQ_WE	Source request write enable(only writable) 0 = write disabled 1 = write enabled
[7:3]	Reserved	Reserved
[2:0]	SRC_REQ	Source request **Note**: A channel *SRC_REQ* bit is written only if the corresponding channel write enable bit in the *SRC_REQ_WE* field is asserted on the same AHB write transfer, and if the channel is enabled in the ChEnReg register.

4.12.6.4.2. Destination Software Transaction Request Register (ReqDstReg)

Register	Offset	R/W	Description	Reset Value
ReqDstReg	DMA_BA+0x370	R/W	Destination Software Transaction Request Register	0x0

Bits	Description	
[63:11]	Reserved	Reserved.
[10:8]	DST_REQ_WE	Destination request write enable(only writable) 0 = write disabled 1 = write enabled
[7:3]	Reserved	Reserved
[2:0]	DST_REQ	Destination request **Note**: A channel *DTS_REQ* bit is written only if the corresponding channel write enable bit in the *DTS_REQ_WE* field is asserted on the same AHB write transfer, and if the channel is enabled in the ChEnReg register.

4.12.6.4.3. Single Source Transaction Request Register (SglReqSrcReg)

Register	Offset	R/W	Description	Reset Value
SglReqSrcReg	DMA_BA+0x378	R/W	Destination Software Transaction Request Register	0x0

Bits	Description	
[63:11]	Reserved	Reserved.
[10:8]	SRC_SGLREQ_WE	Single write enable(only writable) 0 = write disabled

		1 = write enabled
[7:3]	Reserved	Reserved
[2:0]	SRC_SGLREQ	Source single request **Note**: A channel *SRC_SGLREQ* bit is written only if the corresponding channel write enable bit in the *SRC_SGLREQ_WE* field is asserted on the same AHB write transfer, and if the channel is enabled in the ChEnReg register.

4.12.6.4.4. Single Destination Transaction Request Register (SglReqDstReg)

Register	Offset	R/W	Description	Reset Value
SglReqDstReg	DMA_BA+0x380	R/W	Destination Software Transaction Request Register	0x0

Bits	Description	
[63:11]	Reserved	Reserved.
[10:8]	DST_SGLREQ_WE	Destination write enable(only writable) 0 = write disabled 1 = write enabled
[7:3]	Reserved	Reserved
[2:0]	DST_SGLREQ	Destination single or burst request **Note**: A channel *DTS_SGLREQ* bit is written only if the corresponding channel write enable bit in the *DTS_SGLREQ_WE* field is asserted on the same AHB write transfer, and if the channel is enabled in the ChEnReg register.

4.12.6.4.5. Last Source Transaction Request Register(LstSrcReg)

Register	Offset	R/W	Description	Reset Value
LstSrcReg	DMA_BA+0x388	R/W	Last Source Transaction Request Register	0x0

Bits	Description	
[63:11]	Reserved	Reserved.
[10:8]	LSTSRC_WE	Source last transaction request write enable(only writable) 0 = write disabled 1 = write enabled
[7:3]	Reserved	Reserved
[2:0]	LSTSRC	Source last transaction request 0 = Not last transaction in current block 1 = Last transaction in current block

4.12.6.4.6. Last Destination Transaction Request Register(LstDstReg)

Register	Offset	R/W	Description	Reset Value
LstDstReg	DMA_BA+0x390	R/W	Last Destination Transaction Request Register	0x0

Bits	Description	
[63:11]	Reserved	Reserved.
[10:8]	LSTDST_WE	Destination last transaction request write enable(only writable) 0 = write disabled 1 = write enabled
[7:3]	Reserved	Reserved
[2:0]	LSTDST	Destination last transaction request 0 = Not last transaction in current block 1 = Last transaction in current block

4.12.6.5 Miscellaneous DMA Registers

4.12.6.5.1. DMA ID Register (DmaIdReg)

Register	Offset	R/W	Description	Reset Value
DmaIdReg	DMA_BA+0x3A8	R	DMA ID Register	0x0

Bits	Description	
[63:32]	Reserved	Reserved.
[31:0]	DMA_ID	Hardcoded DMA Peripheral ID

4.12.6.5.2. DMA Test Register (DmaTestReg)

Register	Offset	R/W	Description	Reset Value
DmaTestReg	DMA_BA+0x3B0	R/W	DMA Test Register	0x0

Bits	Description	
[63:1]	Reserved	Reserved.
[0]	TEST_SLV_IF	Puts the AHB slave interface into test mode. In this mode, the readback value of the writable registers always matches the value written. This bit does not allow writing to read-only registers. 0 = Normal mode 1 = Test mode

4.12.6.5.3. DMA Component Parameters Register 3(DMA_COMP_PARAMS_3)

Register	Offset	R/W	Description	Reset Value
DMA_COMP_PARAMS_3	DMA_BA+0x3E0	R	DMA Component Parameters Register 3	0x1001_0000 _1001_0000

Bits	Description	
[63]	Reserved	Reserved.
[62:60]	CH1_FIFO_DEPTH	Channel 1 FIFO depth in bytes.

		0x0 = 8 0x1 = 16 0x2 = 32 0x3 = 64 0x4 = 128 **Default value**: 0x1
[59:51]	Reserved	Reserved.
[50:48]	CH1_MAX_MULT_SIZE	Maximum value of burst transaction 0x0 = 4 size that can be programmed for channel 1. 0x0 = 4 0x1 = 8 0x2 = 16 0x3 = 32 0x4 = 64 0x5 = 128 0x6 = 256 0x7 = reserved **Default value**: 0x1
[47:46]	CH1_FC	Hardcode the flow control peripheral for channel 1. 0x0 = DMA 0x1 = SRC 0x2 = DST 0x3 = ANY **Default value**: 0x0
[45:44]	Reserved	Reserved
[43]	CH1_MULTI_BLK_EN	Include or exclude logic to enable multi-block DMA transfers on channel 1. 0x0 = FALSE 0x1 = TRUE **Default value**: 0x0
[42]	CH1_LOCK_EN	Include or exclude logic to enable channel or bus level locking on channel 1. 0x0 = FALSE 0x1 = TRUE **Default value**: 0x0
[41:38]	Reserved	Reserved
[37:35]	CH1_STW	Hardcode the source transfer width for transfers from the source of channel 1. 0x0 = NO_HARDCODE 0x1 = 8 0x2 = 16 0x3 = 32 0x4 = 64 0x5 = 128 0x6 = 256 0x7 = reserved **Default value**: 0x0

[34:32]	CH1_DTW	Hardcode the destination transfer width for transfers from the destination of channel 1. 0x0 = NO_HARDCODE 0x1 = 8 0x2 = 16 0x3 = 32 0x4 = 64 0x5 = 128 0x6 = 256 0x7 = reserved **Default value**: 0x0
[31]	Reserved	Reserved
[30:28]	CH2_FIFO_DEPTH	Channel 2 FIFO depth in bytes. 0x0 = 8 0x1 = 16 0x2 = 32 0x3 = 64 0x4 = 128 **Default value**: 0x1
[27:19]	Reserved	Reserved
[18:16]	CH2_MAX_MULT_SIZE	Maximum value of burst transaction 0x0 = 4 size that can be programmed for channel 2. 0x0 = 4 0x1 = 8 0x2 = 16 0x3 = 32 0x4 = 64 0x5 = 128 0x6 = 256 0x7 = reserved **Default value**: 0x1
[15:14]	CH2_FC	Hardcode the flow control peripheral for channel 2. 0x0 = DMA 0x1 = SRC 0x2 = DST 0x3 = ANY **Default value**: 0x0
[13:12]	Reserved	Reserved
[11]	CH2_MULTI_BLK_EN	Include or exclude logic to enable multi-block DMA transfers on channel 2. 0x0 = FALSE 0x1 = TRUE **Default value**: 0x0
[10]	CH2_LOCK_EN	Include or exclude logic to enable channel or bus level locking on channel 2. 0x0 = FALSE

		0x1 = TRUE **Default value**: 0x0
[9:6]	Reserved	Reserved
[5:3]	CH2_STW	Hardcode the source transfer width for transfers from the source of channel 2. 0x0 = NO_HARDCODE 0x1 = 8 0x2 = 16 0x3 = 32 0x4 = 64 0x5 = 128 0x6 = 256 0x7 = reserved **Default value**: 0x0
[2:0]	CH2_DTW	Hardcode the destination transfer width for transfers from the destination of channel 2. 0x0 = NO_HARDCODE 0x1 = 8 0x2 = 16 0x3 = 32 0x4 = 64 0x5 = 128 0x6 = 256 0x7 = reserved **Default value**: 0x0

4.12.6.5.4. DMA Component Parameters Register 2(DMA_COMP_PARAMS_2)

Register	Offset	R/W	Description	Reset Value
DMA_COMP_PARAMS_2	DMA_BA+0x3E8	R	DMA Component Parameters Register 2	0x1001_0000

Bits	Description	
[63:44]	Reserved	Reserved
[43:40]	CH2_MULTI_BLK_TYPE	Hardcode the type of multi-block transfers that the DMA can perform for channel x. 0x0 = NO_HARDCODE 0x1 = CONT_RELOAD 0x2 = RELOAD_CONT 0x3 = RELOAD_RELOAD 0x4 = CONT_LLP 0x5 = RELOAD_LLP 0x6 = LLP_CONT 0x7 = LLP_RELOAD 0x8 = LLP_LLP **Default value**: 0x0
[39:36]	CH1_MULTI_BLK_TYPE	
[35:32]	CH0_MULTI_BLK_TYPE	

[31]	Reserved	Reserved
[30:28]	CH0_FIFO_DEPTH	Channel 0 FIFO depth in bytes. 0x0 = 8 0x1 = 16 0x2 = 32 0x3 = 64 0x4 = 128 **Default value**: 0x1
[27:19]	Reserved	Reserved
[18:16]	CH0_MAX_MULT_SIZE	Maximum value of burst transaction 0x0 = 4 size that can be programmed for channel 0. 0x0 = 4 0x1 = 8 0x2 = 16 0x3 = 32 0x4 = 64 0x5 = 128 0x6 = 256 0x7 = reserved **Default value**: 0x1
[15:14]	CH0_FC	Hardcode the flow control peripheral for channel 0. 0x0 = DMA 0x1 = SRC 0x2 = DST 0x3 = ANY **Default value**: 0x0
[13:12]	Reserved	Reserved
[11]	CH0_MULTI_BLK_EN	Include or exclude logic to enable multi-block DMA transfers on channel 0. 0x0 = FALSE 0x1 = TRUE **Default value**: 0x0
[10]	CH0_LOCK_EN	Include or exclude logic to enable channel or bus level locking on channel 0. 0x0 = FALSE 0x1 = TRUE **Default value**: 0x0
[9:6]	Reserved	Reserved
[5:3]	CH0_STW	Hardcode the source transfer width for transfers from the source of channel 0. 0x0 = NO_HARDCODE 0x1 = 8 0x2 = 16 0x3 = 32 0x4 = 64 0x5 = 128

		0x6 = 256 0x7 = reserved **Default value**: 0x0
[2:0]	CH0_DTW	Hardcode the destination transfer width for transfers from the destination of channel 0. 0x0 = NO_HARDCODE 0x1 = 8 0x2 = 16 0x3 = 32 0x4 = 64 0x5 = 128 0x6 = 256 0x7 = reserved **Default value**: 0x0

4.12.6.5.5. DMA Component Parameters Register 1(DMA_COMP_PARAMS_1)

Register	Offset	R/W	Description	Reset Value
DMA_COMP_PARAMS_1	DMA_BA+0x3F0	R	DMA Component Parameters Register 1	0x2100_0204_0000_0888

Bits	Description	
[63:62]	Reserved	Reserved
[61]	STATIC_ENDIAN_SELECT	The endian scheme of the DMA can be configured statically through coreConsultant or dynamically via pins on the I/O. 0 = FALSE 1 = TRUE **Default value**: 0x1
[60]	ADD_ENCODED_PARAMS	Adding the encoded parameters gives firmware an easy and quick way of identifying the DesignWare component within an I/O memory map. 0 = FALSE 1 = TRUE **Default value**: 0x0
[59:55]	NUM_HS_INT	Number of handshaking interfaces. 0x00 = 0 to 0x10 = 16 **Default value**: 0x10
[54:53]	M1_HDATA_WIDTH	Master 1 interface data bus width 0x0 = 32 bits 0x1 = 64 bits 0x2 = 128 bits 0x3 = 256 bits **Default value**: 0x0

[52:47]	Reserved	Reserved
[46:45]	S_HDATA_WIDTH	Slave interface data bus width. 0x0 = 32 bits 0x1 = 64 bits 0x2 = 128 bits 0x3 = 256 bits **Default value**: 0x0
[44:43]	NUM_MASTER_INT	Number of AHB master interfaces. 0x0 = 1 to 0x3 = 4 **Default value**: 0x0
[42:40]	NUM_CHANNELS	Number Of DMA channels. Each channel is uni-directional and transfers data from the channel's source to the channel's destination. 0x0 = 1 to 0x7 = 8 **Default value**: 0x2
[39:36]	Reserved	Reserved
[35]	MABRST	Selecting this box will allow software to limit the maximum AMBA burst length to a value under software control. If this box is not selected then software cannot limit the the maximum AMBA burst length. 0 = FALSE 1 = TRUE **Default value**: 0x0
[34:33]	INTR_IO	Selects which interrupt related pins appear as outputs of the design. 0x0 = ALL. 0x1 = TYPE 0x2 = COMBINED 0x3 = reserved **Default value**: 0x2
[32]	BIG_ENDIAN	The AHB master and slave interfaces can be configured to exist in either big or little endian system. When set to 0, all master interfaces and the slave interface are set to little endian. 0 = FALSE 1 = TRUE **Default value**: 0x0
[31:12]	Reserved	Reserved
[11:8]	CH2_MAX_BLK_SIZE	Maximum block size, in multiples of source transfer width, that can be programmed for channel x. 0x0 = 3 0x1 = 7 0x2 =15
[7:4]	CH1_MAX_BLK_SIZE	
[3:0]	CH0_MAX_BLK_SIZE	

		0x3 = 31 0x4 = 63 0x5 = 127 0x6 = 255 0x7 = 511 0x8 = 1023 0x9 = 2047 0xa = 4095 **Default value**: 0x8

4.12.6.5.6. DMA Component ID Register(DCIR)

Register	Offset	R/W	Description	Reset Value
DCIR	DMA_BA+0x3F8	R	DMA Component ID Register	0x4457_1110

Bits	Description	
[63:32]	DMA_COMP_VERSION	Version of the component.
[31:0]	DMA_COMP_TYPE	Component Type number = 0x4457_1110

4.13 Analog-to-Digital Converter (ADC)

4.13.1 Overview

The PAN1020 contains one 12-bit successive approximation analog-to-digital converters (SAR A/D converter) with nine input channels. The A/D converters can be started by software, external pin (STADC/P3.2) or PWM trigger.

4.13.2 Features

- Analog input voltage range: 0 ~ Analog Supply Voltage from AVDD
- 10-bit resolution and 12-bit accuracy is guaranteed
- Up to eight single-end analog input channels, one band-gap input channel, one temperature input channel and one voltage input channel.
- Maximum ADC clock frequency is 16 MHz, and 14 ADC clocks per sample
- Two operating modes
  - Single mode: A/D conversion is performed one time on a specified channel
  - PWM sequence mode: When PWM trigger, two of three ADC channels from 0 to 2 will automatically convert analog data in the sequence of channel [0,1] or channel[1,2] or channel[0,2] defined by MODESEL (ADC_SEQCTL[3:2])
- An A/D conversion can be started by
  - Software write 1 to SWTRG bit
  - External pin STADC
  - PWM trigger with optional start delay period
- Each Conversion result is held in data register with valid and overrun indicators
- Conversion results can be compared with specified value and user can select whether to generate an interrupt when conversion result matches the compare register setting

4.13.3 Block Diagram

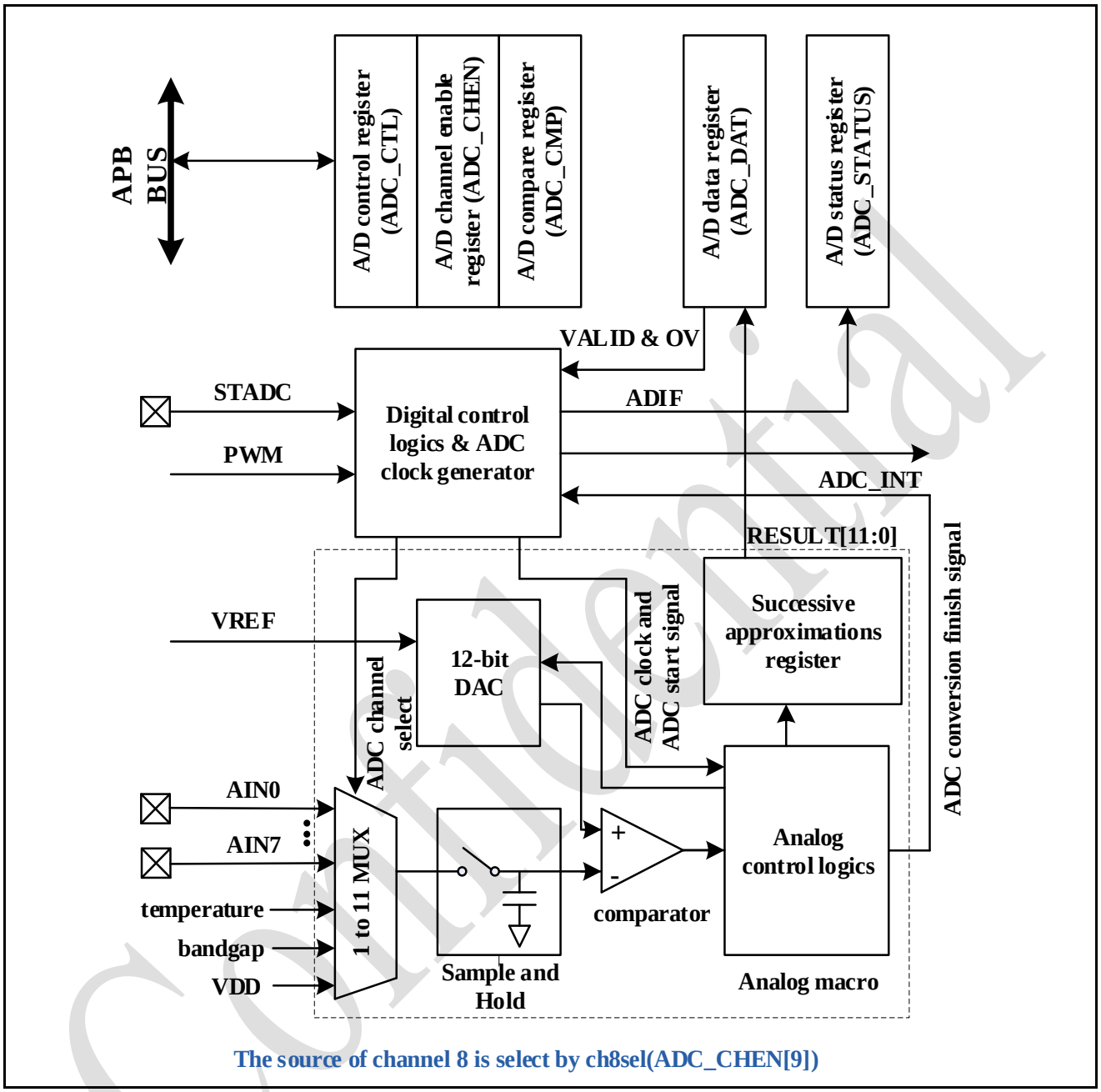


Figure 4-98 AD Controller Block Diagram

4.13.4 Basic Configuration

The ADC pin functions are configured in SYS_P1_MFP, SYS_P2_MFP and SYS_P3_MFP register. It is recommended to disable the digital input path of the analog input pins to avoid the leakage current. User can disable the digital input path by configuring P1_DINOFF, P2_DINOFF and P3_DINOFF register.

The ADC peripheral clock can be enabled in ADCCKEN (CLK_APBCLK[28]). The ADC peripheral clock source is selected by ADCSEL (CLK_CLKSEL1[3:2]).

4.13.5 Functional Description

The A/D converter operates by successive approximation with 12-bit resolution. When changing the analog input channel is enabled, in order to prevent incorrect operation, software must clear SWTRG bit to 0 in the ADC_CTL register. The A/D converter discards the current conversion immediately and enters idle state while SWTRG bit is cleared.

4.13.5.1 ADC Peripheral Clock Generator

The ADC engine is always from HCLK. The ADC clock peripheral frequency is divided by an 3-bit prescaler with the following formula:

ADC peripheral *clock frequency = (ADC peripheral clock source frequency) / (ADCDIV+1)*; where the 3-bit ADCDIV is located in registrer ADC_CTL2[18:16].

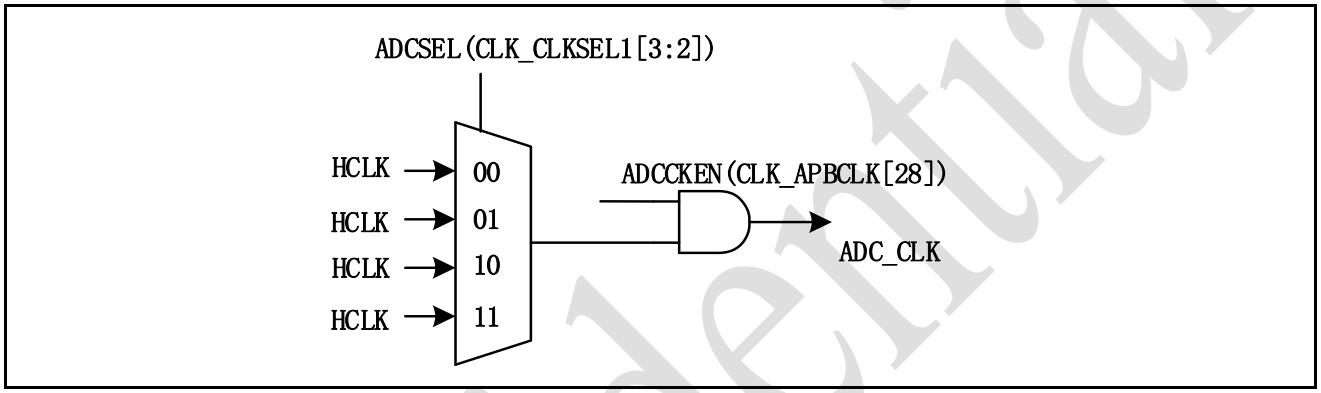


Figure 4-99 ADC Peripheral Clock Control

4.13.5.2 ADC Operation

A/D conversion is performed only once on the specified single channel. The operation is as follows:

1. A/D conversion will be started when the SWTRG bit of ADC_CTL is set to 1 by software or external trigger input.
2. When A/D conversion is finished, the result is stored in the A/D data register.
3. The ADIF bit of ADC_STATUS register will be set to 1. If the ADCIEN bit of ADC_CTL register is set to 1, the ADC interrupt will be asserted.
4. The SWTRG bit remains 1 during A/D conversion. When A/D conversion ends, the SWTRG bit is automatically cleared to 0 and the A/D converter enters idle state.
5. Figure 4-100 shows an example timing diagram for Single mode.

Note: If software enables more than one channel, the channel with the smallest number will be selected and the other enabled channels will be ignored.

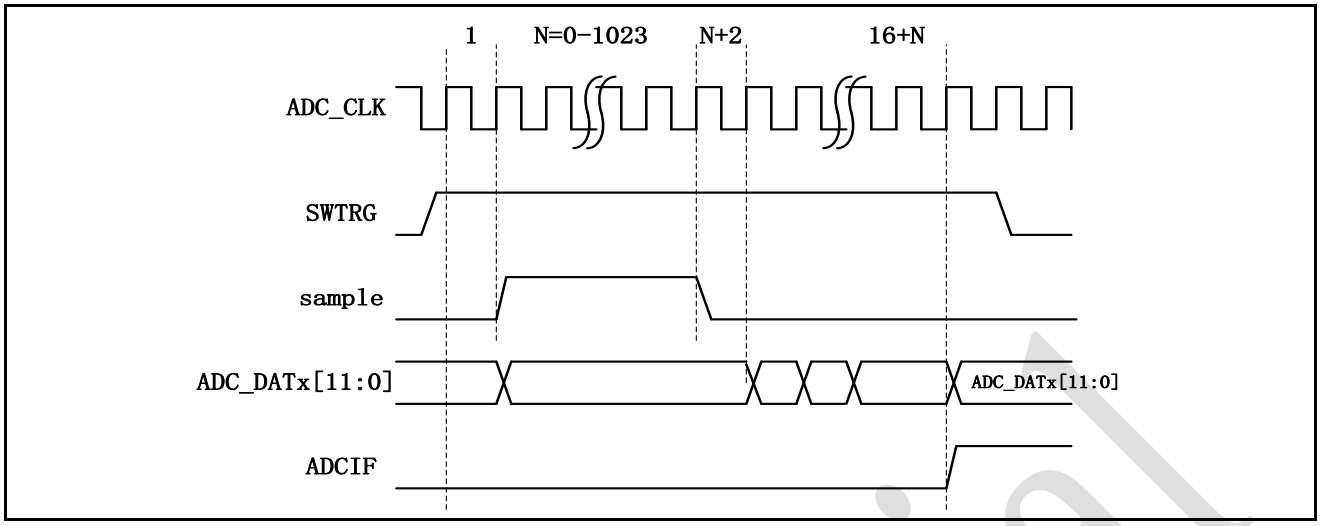


Figure 4-100 Single Mode Conversion Timing Diagram

4.13.5.3 External Trigger Input Sampling and A/D Conversion Time

A/D conversion can be triggered by external pin request. When the HWTRGEN (ADC_CTL[8]) bit is set to 1 to enable ADC external trigger function, setting the HWTRGSEL (ADC_CTL[5:4]) bits to 00b is to select external trigger input from the STADC pin. Software can set HWTRGCOND to select trigger condition between falling or rising edge. An 8-bit sampling counter is used to deglitch. If edge trigger condition is selected, the high and low state must be kept at least 4 HCLKs. Pulse that is shorter than this specification will be ignored.

4.13.5.4 PWM Trigger

A/D conversion can also be triggered by PWM request. When the HWTRGEN is set to high to enable ADC external hardware trigger function, setting the HWTRGSEL (ADC_CTL[5:4]) bits to 11b is to select external hardware trigger input source from PWM trigger. When PWM trigger is enabled, setting DELAY (ADC_TRGDLY[7:0]) bits can insert a delay time between PWM trigger condition and ADC start conversion.

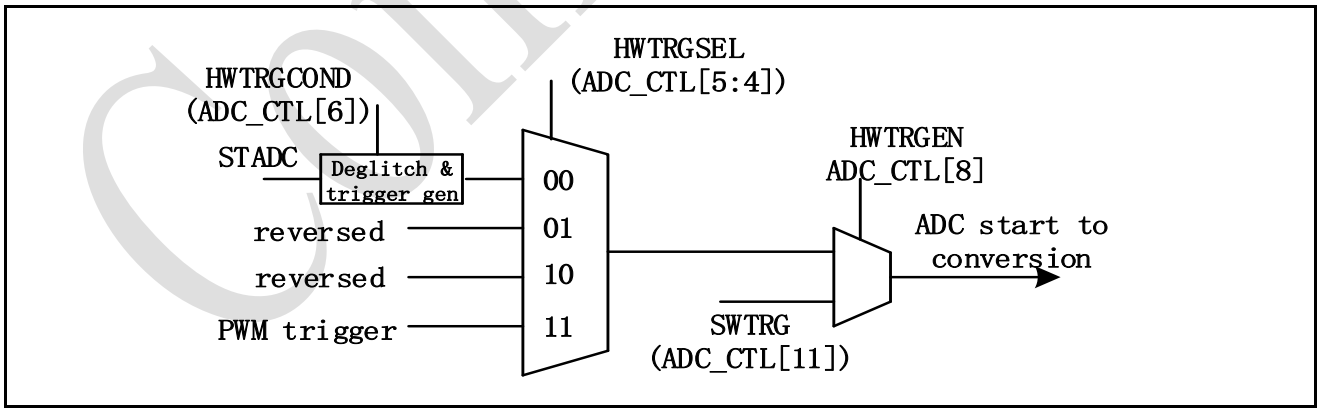


Figure 4-101 ADC Start Conversion Conditions

4.13.5.5 Conversion Result Monitor by Compare Mode Function

The PAN1020 ADC controller provides two compare registers, ADC_CMP0 and ADC_CMP1, to monitor maximum two specified channels. Software can select which channel to be monitored by setting CMPCH (ADC_CMPx[5:0]). CMPCOND bit is used to determine the compare condition. If

CMPCOND bit is cleared to 0, the internal match counter will increase one when the conversion result is less than the value specified in CMPDAT (ADC_CMPx[25:16]) ; if CMPCOND bit is set to 1, the internal match counter will increase one when the conversion result is greater than or equal to the value specified in CMPDAT[11:0]. When the conversion of the channel specified by CMPCH is completed, the comparing action will be triggered one time automatically. When the compare result meets the setting, compare match counter will increase 1, otherwise, the compare match counter will be clear to 0. When the match counter reaches the setting of (CMPMCNT+1) then ADCMPIF bit will be set to 1, if ADCMPIE bit is set then an ADC_INT interrupt request is generated. Software can use it to monitor the external analog input pin voltage transition in scan mode without imposing a load on software. Figure 4-102 show detailed logic diagram.

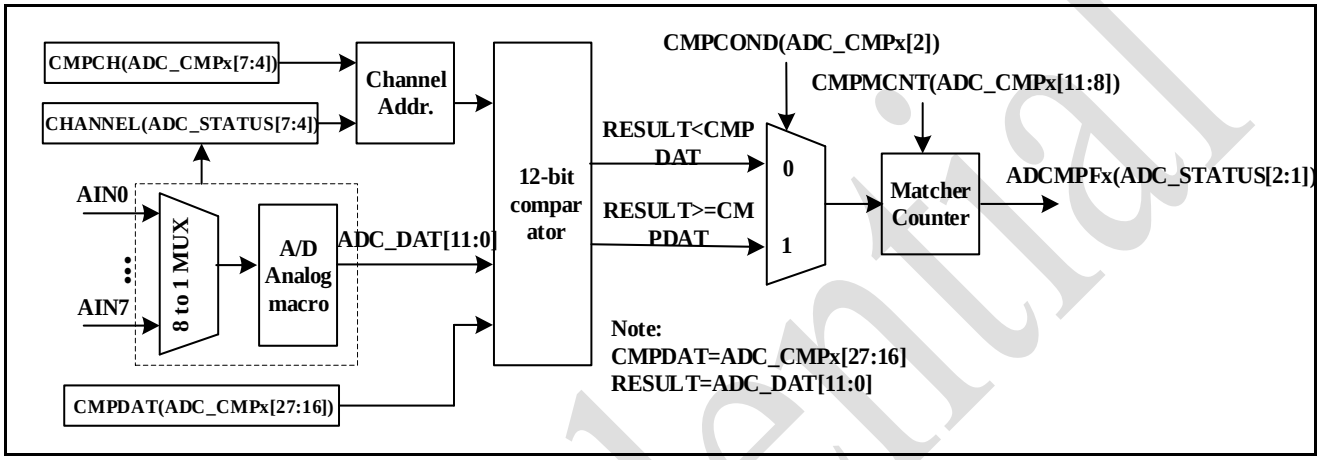


Figure 4-102 A/D Conversion Result Monitor Logics Diagram

4.13.5.6 Interrupt Sources

There are three interrupt sources of ADC interrupt. When an ADC operation mode finishes its conversion, the A/D conversion end flag, ADIF, will be set to 1. The ADCMPIF0 and ADCMPIF1 are the compare flags of compare function. When the conversion result meets the settings of ADC_CMP0/1, the corresponding flag will be set to 1. When one of the flags, ADIF, ADCMPIF0 and ADCMPIF1, is set to 1 and the corresponding interrupt enable bit, ADCIEN of ADC_CTL and ADCMPIE of ADC_CMP0/1, is set to 1, the ADC interrupt will be asserted. Software can clear these flags to revoke the interrupt request.

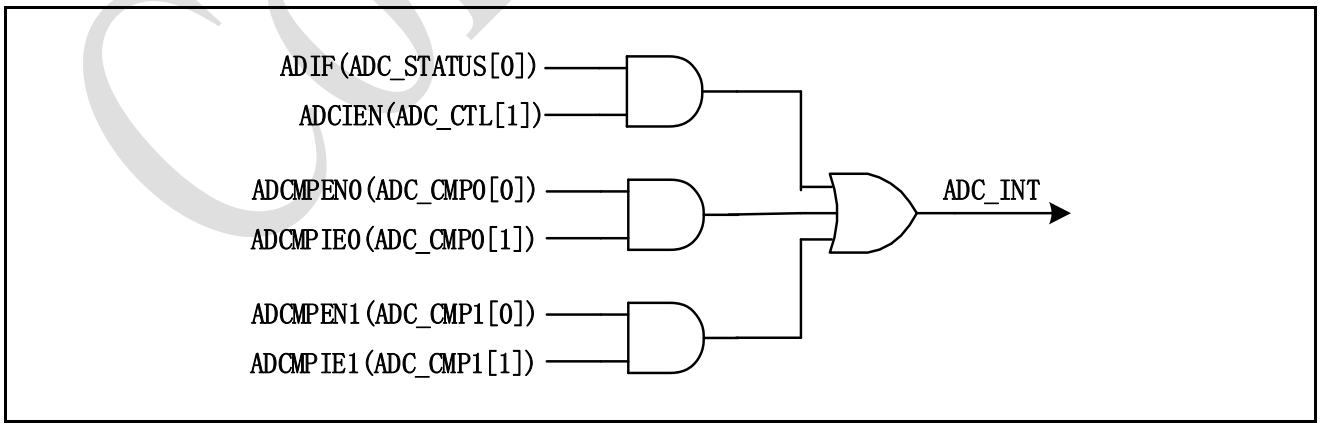


Figure 4-103 A/D Controller Interrupt

4.13.5.7 Flag Sources

There are three flag sources of ADC conversion. When an ADC operation mode finishes its conversion, the A/D conversion end flag, ADCF, will be set to 1. The ADCMPF0 and ADCMPF1 are the compare flags of compare function. When the conversion result meets the settings of ADC_CMP0/1, the corresponding flag will be set to 1.

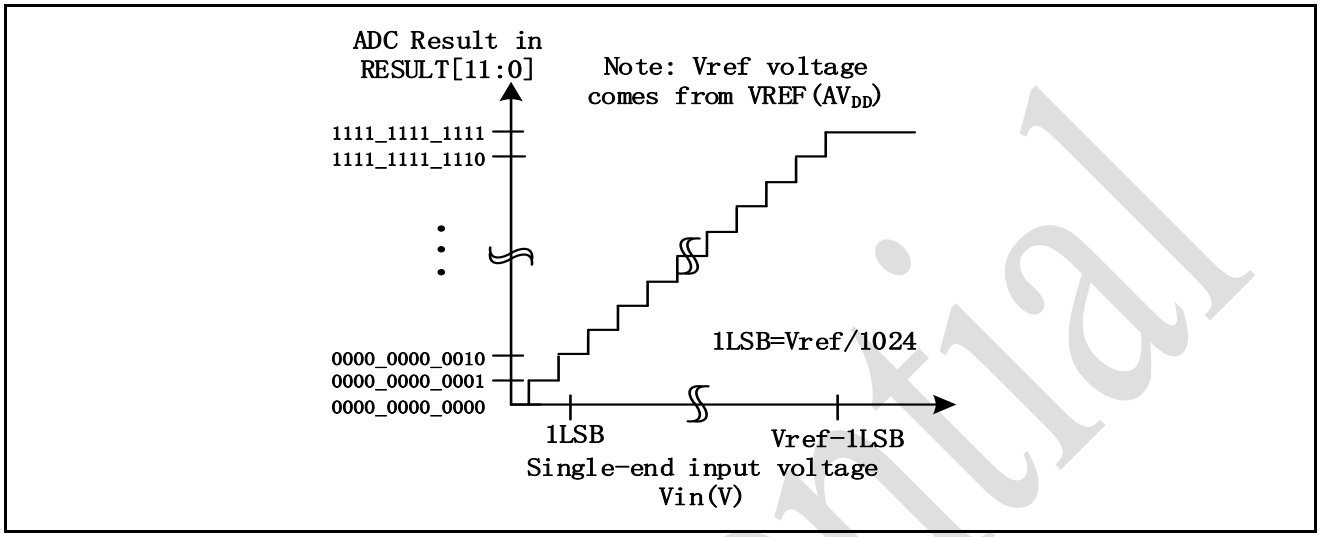


Figure 4-104 Conversion Result Mapping Diagram of ADC Single-end Input

4.13.5.8 PWM Sequential

Support sequential mode for 2 channels to reduce half interrupt frequency. When the SEQEN (ADC_SEQCTL[0]) is set to high to enable A/D PWM sequential function, setting the TRG1CTL (ADC_SEQCTL[11:8]) and TRG2CTL (ADC_SEQCTL[19:16]) are to select external trigger input from the PWM channel 0/2/4, type can be rising/center/falling/period, When ADC sequential mode is enabled, two of three ADC channels from 0 to 2 will automatically convert analog data in the sequence of channel [0, 1] or channel[1,2] or channel[0,2] defined by MODESEL (ADC_SEQCTL[3:2]). By the way, if SEQTYPE (ADC_SEQCTL[1]) is set to low, ADC delay time is only inserted before the first conversion. The second conversion starts immediately after the first conversion is completed. (for 2/3- shunt type), if SEQTYPE (ADC_SEQCTL[1]) is set to high, ADC delay time is inserted before each conversion. (for 1-shunt type), By the way, Valid ADC channel are CH0~2 in 2/3-shunt type, Valid ADC channel are CH0~7 in 1-shunt type, Figure 4-105 and Figure 4-106 show the function diagram.

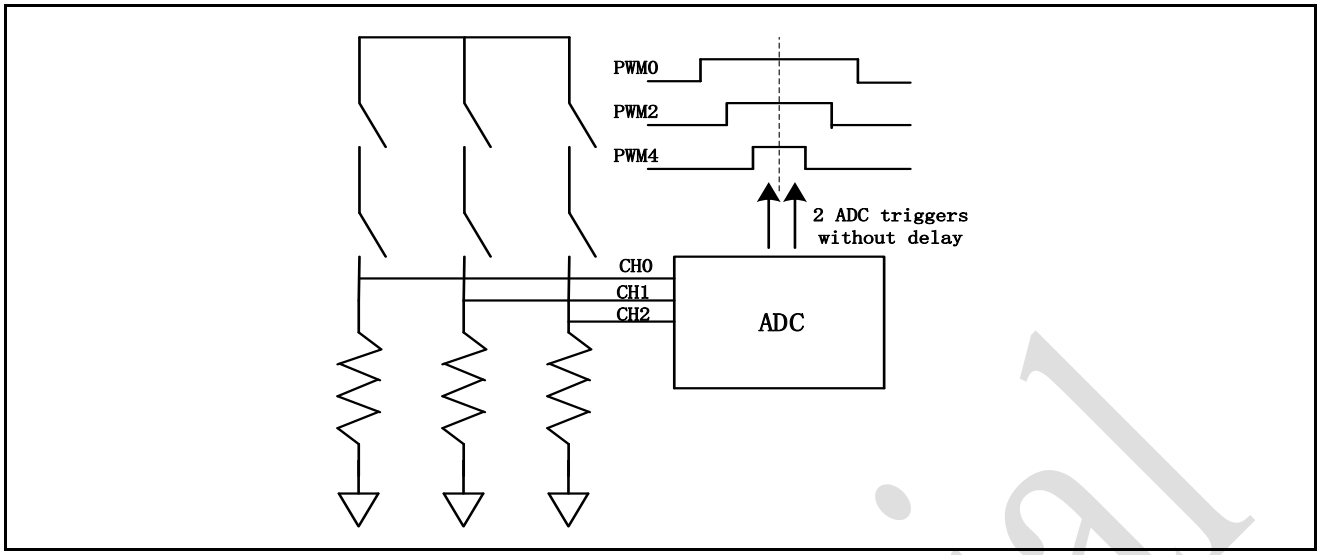


Figure 4-105 ADC Sequential Mode Type is 2/3-shunt

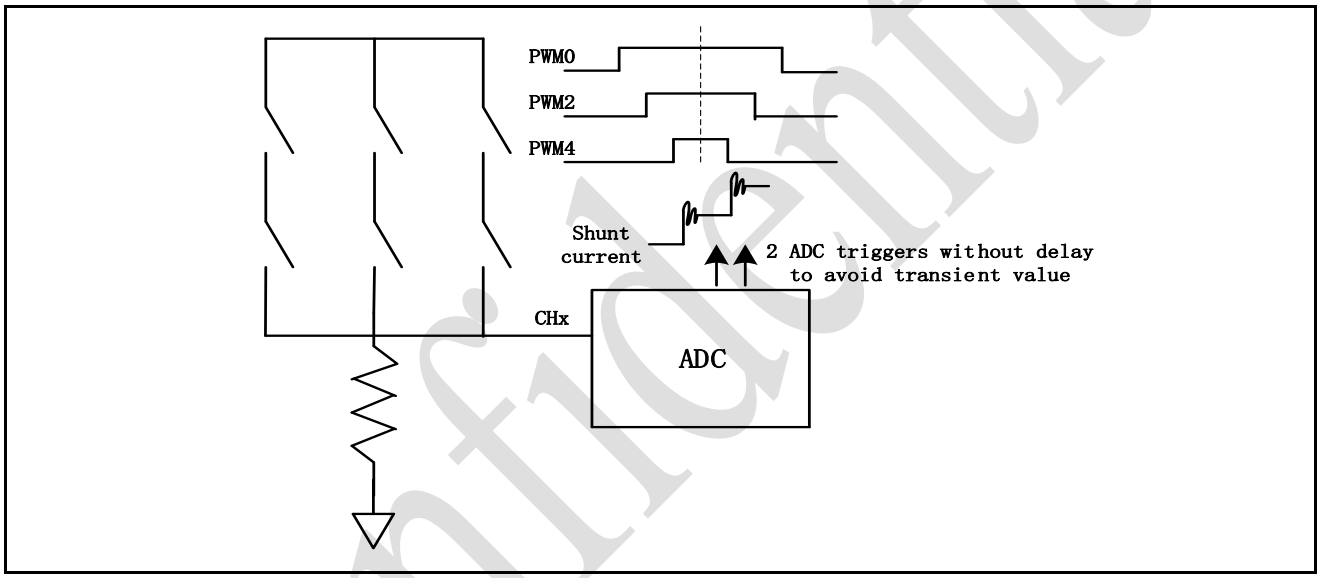


Figure 4-106 ADC Sequential Mode Type is 1-shunt

4.13.6 Register Map

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

Register	Offset	R/W	Description	Reset Value
ADC Base Address: **ADC_BA = 0x400E_0000**				
ADC_DAT	ADC_BA+0x00	R	A/D Data Register	0x0000_0000
ADC_CTL	ADC_BA+0x20	R/W	A/D Control Register	0x0000_0000
ADC_CHEN	ADC_BA+0x24	R/W	A/D Channel Enable Register	0x0000_0000
ADC_CMP0	ADC_BA+0x28	R/W	A/D Compare Register 0	0x0000_0000
ADC_CMP1	ADC_BA+0x2C	R/W	A/D Compare Register 1	0x0000_0000
ADC_STATUS	ADC_BA+0x30	R/W	A/D Status Register	0x0000_0000
ADC_TRGDLY	ADC_BA+0x44	R/W	A/D Trigger Delay Control Register	0x0000_0000

ADC_EXTSMPT	ADC_BA+0x48	R/W	A/D Sampling Time Counter Register	0x0000_0002
ADC_SEQCTL	ADC_BA+0x4C	R/W	A/D PWM Sequential Mode Control Register	0x0000_0000
ADC_SEQDAT1	ADC_BA+0x50	R	A/D PWM Sequential Mode First Result Register1	0x0000_0000
ADC_SEQDAT2	ADC_BA+0x54	R	A/D PWM Sequential Mode Second Result Register1	0x0000_0000
ADC_CTL2	ADC_BA+0x58	R/W	A/D Control Register 2	0x0101_0000

4.13.7 Register Description

4.13.7.1 ADC Data Register (ADC_DAT)

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

Bits	Description	
[31:18]	Reserved	Reserved.
[17]	VALID	Valid Flag This bit is set to 1 when ADC conversion is completed and cleared by hardware after the ADC_DAT register is read. 0 = Data in RESULT[11:0] bits not valid. 1 = Data in RESULT[11:0] bits valid.
[16]	OV	Over Run Flag If converted data in RESULT[11:0] has not been read before the new conversion result is loaded to this register, OV is set to 1. It is cleared by hardware after the ADC_DAT register is read. 0 = Data in RESULT[11:0] is recent conversion result. 1 = Data in RESULT[11:0] overwrote.
[15:12]	Reserved	Reserved.
[11:0]	RESULT	A/D Conversion Result This field contains conversion result of ADC.

4.13.7.2 ADC Control Register (ADC_CTL)

Register	Offset	R/W	Description	Reset Value
ADC_CTL	ADC_BA+0x20	R/W	A/D Control Register	0x0000_0000

Bits	Description	
[31:12]	Reserved	Reserved.
[11]	SWTRG	Software Trigger A/D Conversion Start SWTRG bit can be set to 1 from two sources: software and external pin STADC. SWTRG will be cleared to 0 by hardware automatically after conversion complete. 0 = Conversion stopped and A/D converter entered idle state. 1 = Conversion start.
[10:9]	Reserved	Reserved.
[8]	HWTRGEN	Hardware External Trigger Enable Bit Enable or disable triggering of A/D conversion by external STADC pin. If external trigger is

		enabled, the SWTRG bit can be set to 1 by the selected hardware trigger source. 0= External trigger Disabled. 1= External trigger Enabled.
[7]	Reserved	Reserved.
[6]	HWTRGCOND	Hardware External Trigger Condition This bit decides whether the external pin STADC trigger event is falling or raising edge. The signal must be kept at stable state at least 4 PCLKs at high and low state for edge trigger. 0 = Falling edge. 1 = Raising edge.
[5:4]	HWTRGSEL	Hardware Trigger Source Select Bit 00 = A/D conversion is started by external STADC pin. 11 = A/D conversion is started by PWM trigger. Others = Reserved. **Note**: Software should disable TRGEN and SWTRG before change TRGS.
[3:2]	Reserved	Reserved.
[1]	ADCIEN	A/D Interrupt Enable Bit A/D conversion end interrupt request is generated if ADCIEN bit is set to 1. 0 = A/D interrupt function Disabled. 1 = A/D interrupt function Enabled.
[0]	ADCEN	A/D Converter Enable Bit 0 = A/D Converter Disabled. 1 = A/D Converter Enabled. **Note**: Before starting A/D conversion function, this bit should be set to 1. Clear it to 0 to disable A/D converter analog circuit to save power consumption.

4.13.7.3 ADC Channel Enable Register (ADC_CHEN)

Register	Offset	R/W	Description	Reset Value
ADC_CHEN	ADC_BA+0x24	R/W	A/D Channel Enable Register	0x0000_0000

Bits	Description	
[31:11]	Reserved	Reserved.
[10]	CHEN10	Analog Input Channel 10 Enable Bit 0 = Channel 10 Disabled. 1 = Channel 10 Enabled.
[9]	CHEN9	Analog Input Channel 9 Enable Bit 0 = Channel 9 Disabled. 1 = Channel 9 Enabled.
[8]	CHEN8	Analog Input Channel 8 Enable Bit 0 = Channel 8 Disabled. 1 = Channel 8 Enabled.
[7]	CHEN7	Analog Input Channel 7 Enable Bit 0 = Channel 7 Disabled.

		1 = Channel 7 Enabled.
[6]	CHEN6	Analog Input Channel 6 Enable Bit 0 = Channel 6 Disabled. 1 = Channel 6 Enabled.
[5]	CHEN5	Analog Input Channel 5 Enable Bit 0 = Channel 5 Disabled. 1 = Channel 5 Enabled.
[4]	CHEN4	Analog Input Channel 4 Enable Bit 0 = Channel 4 Disabled. 1 = Channel 4 Enabled.
[3]	CHEN3	Analog Input Channel 3 Enable Bit 0 = Channel 3 Disabled. 1 = Channel 3 Enabled.
[2]	CHEN2	Analog Input Channel 2 Enable Bit 0 = Channel 2 Disabled. 1 = Channel 2 Enabled.
[1]	CHEN1	Analog Input Channel 1 Enable Bit 0 = Channel 1 Disabled. 1 = Channel 1 Enabled.
[0]	CHEN0	Analog Input Channel 0 Enable Bit 0 = Channel 0 Disabled. 1 = Channel 0 Enabled. **Note**: If software enables more than one channel, the channel with the smallest number will be selected and the other enabled channels will be ignored.

4.13.7.4 A/D Compare Register 0/1 (ADC_CMP0/1)

Register	Offset	R/W	Description	Reset Value
ADC_CMP0	ADC_BA+0x28	R/W	A/D Compare Register 0	0x0000_0000
ADC_CMP1	ADC_BA+0x2C	R/W	A/D Compare Register 1	0x0000_0000

Bits	Description	
[31:28]	Reserved	Reserved.
[27:16]	CMPDAT	Comparison Data The 12-bit data is used to compare with conversion result of specified channel.
[15:12]	Reserved	Reserved.
[11:8]	CMPMCNT	Compare Match Count When the specified A/D channel analog conversion result matches the compare condition defined by CMPCOND[2], the internal match counter will increase 1. When the internal counter reaches the value to (CMPMCNT+1), the ADCMPFx bit will be set.
[7:4]	CMPCH	Compare Channel Selection Set this field to select which channel's result to be compared. **Note**: Valid setting of this field is channel 0~7.

[3]	Reserved	Reserved.
[2]	CMPCOND	Compare Condition 0 = Set the compare condition as that when a 12-bit A/D conversion result is less than the 12-bit CMPDAT (ADC_CMPx[25:16]), the internal match counter will increase one. 1 = Set the compare condition as that when a 12-bit A/D conversion result is greater or equal to the 12-bit CMPDAT (ADC_CMPx[25:16]), the internal match counter will increase one. **Note**: When the internal counter reaches the value to (CMPMCNT+1), the ADCMPFx bit will be set.
[1]	ADCMPIE	A/D Compare Interrupt Enable Bit If the compare function is enabled and the compare condition matches the setting of CMPCOND and CMPMCNT, ADCMPIE bit will be asserted, in the meanwhile, if ADCMPIE is set to 1, a compare interrupt request is generated. 0 = Compare function interrupt Disabled. 1 = Compare function interrupt Enabled.
[0]	ADCMPEN	A/D Compare Enable Bit Set 1 to this bit to enable comparing CMPDAT (ADC_CMPx[25:16]) with specified channel conversion results when converted data is loaded into the ADC_DAT register. 0 = Compare function Disabled. 1 = Compare function Enabled.

4.13.7.5 A/D Status Register (ADC_STATUS)

Register	Offset	R/W	Description	Reset Value
ADC_STATUS	ADC_BA+0x30	R/W	A/D Status Register	0x0000_0000

Bits	Description	
[31:27]	Reserved	Reserved.
[26]	ADCMPF1	A/D Compare Flag 1 When the selected channel A/D conversion result meets the setting condition in ADC_CMP1, this bit is set to 1. 0 = Conversion result in ADC_DAT does not meet the ADC_CMP1 setting. 1 = Conversion result in ADC_DAT meets the ADC_CMP1 setting. **Note**: This bit can be cleared to 0 by software writing 1.
[25]	ADCMPF0	A/D Compare Flag 0 When the selected channel A/D conversion result meets the setting condition in ADC_CMP0, this bit is set to 1. 0 = Conversion result in ADC_DAT does not meet the ADC_CMP0 setting. 1 = Conversion result in ADC_DAT meets the ADC_CMP0 setting. **Note**: This bit can be cleared to 0 by software writing 1.
[24]	ADCF	A/D Conversion End Flag A status flag that indicates the end of A/D conversion. ADIF is set to 1 When A/D conversion ends. **Note**: This bit can be cleared to 0 by software writing 1.
[23:17]	Reserved	Reserved.

[16]	OV	Overrun Flag (Read Only) It is a mirror to OV bit in ADC_DAT register.
[15:9]	Reserved	Reserved.
[8]	VALID	Data Valid Flag (Read Only) It is a mirror of VALID bit in ADC_DAT register.
[7:4]	CHANNEL	Current Conversion Channel (Read Only) This filed reflects the current conversion channel when BUSY=1. When BUSY=0, it shows the number of the next converted channel.
[3]	BUSY	BUSY/IDLE (Read Only) This bit is mirror of as SWTRG bit in ADC_CTL 0 = A/D converter is in idle state. 1 = A/D converter is busy at conversion.
[2]	ADCMPIF1	A/D Compare Interrupt Flag 1 When the selected channel A/D conversion result meets the setting condition in ADC_CMP1, this bit is set to 1. 0 = Conversion result in ADC_DAT does not meet the ADC_CMP1 setting. 1 = Conversion result in ADC_DAT meets the ADC_CMP1 setting. **Note**: This bit can be cleared to 0 by software writing 1.
[1]	ADCMPIF0	A/D Compare Interrupt Flag 0 When the selected channel A/D conversion result meets the setting condition in ADC_CMP0, this bit is set to 1. 0 = Conversion result in ADC_DAT does not meet the ADC_CMP0 setting. 1 = Conversion result in ADC_DAT meets the ADC_CMP0 setting. **Note**: This bit can be cleared to 0 by software writing 1.
[0]	ADIF	A/D Conversion End Interrupt Flag A status flag that indicates the end of A/D conversion. ADIF is set to 1 When A/D conversion ends. **Note**: This bit can be cleared to 0 by software writing 1.

4.13.7.6 A/D Trigger Delay Controller Register (ADC_TRGDLY)

Register	Offset	R/W	Description	Reset Value
ADC_TRGDLY	ADC_BA+0x44	R/W	A/D Trigger Delay Control Register	0x0000_0000

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

4.13.7.7 A/D Sampling Register (ADC_EXTSMPT)

Register	Offset	R/W	Description	Reset Value
ADC_EXTSMPT	ADC_BA+0x48	R/W	A/D Sampling Time Counter Register	0x0000_0002

Bits	Description	
[31:10]	Reserved	Reserved.
[3:0]	EXTSMPT	Additional ADC Sample Clock If the ADC input is unstable, user can set this register to increase the sampling time to get a stable ADC input signal. The default sampling time is 2 ADC clocks. 1-16 ADC clock number will be inserted to lengthen the sampling clock. Set 0 will disable ADC

4.13.7.8 A/D PWM Sequential Register (ADC_SEQCTL)

Register	Offset	R/W	Description	Reset Value
ADC_SEQCTL	ADC_BA+0x4C	R/W	A/D PWM Sequential Mode Control Register	0x0000_0000

Bits	Description	
[31:20]	Reserved	Reserved.
[19:16]	TRG2CTL	PWM Trigger Source Selection For TRG2CTL[3:2] 00 = PWM Trigger source is PWM0. 01 = PWM Trigger source is PWM2. 10 = PWM Trigger source is PWM4. 11 = PWM Trigger source is PWM6. PWM Trigger Type Selection for TRG2CTL[1:0] 00 = Rising of the selected PWM. 01 = Center of the selected PWM. 10 = Falling of the selected PWM. 11 = Period of the selected PWM. **Note**: PWM trigger source is valid for 1-shunt type and 2/3-shunt type.
[15:12]	Reserved	Reserved.
[11:8]	TRG1CTL	PWM Trigger Source Selection For TRG1CTL[3:2] 00 = PWM Trigger source is PWM0. 01 = PWM Trigger source is PWM2. 10 = PWM Trigger source is PWM4. 11 = PWM Trigger source is PWM6. PWM Trigger Type Selection for TRG1CTL[1:0] 00 = Rising of the selected PWM. 01 = Center of the selected PWM. 10 = Falling of the selected PWM. 11 = Period of the selected PWM. **Note**: PWM trigger source is valid for 1-shunt and 2/3-shunt type.
[7:6]	Reserved	Reserved.
[5]	TRG_SEL	TRG1CTL or TRG2CTL select for 1-shunt sequential mode. 0 = Using TRG1CTL to trigger sequential conversion;

		1 = Using TRG2CTL to trigger sequential conversion;
[4]	DELAY_EN	ADC delay time inserted before 2nd conversion. 0 = ADC delay time only inserted before the 1st conversion; 1 = ADC delay time inserted before each conversion.
[3:2]	MODESEL	ADC Sequential Mode Selection 00 = Issue ADC_INT after Channel 0 then Channel 1 conversion finishes when SEQEN =1. 01 = Issue ADC_INT after Channel 1 then Channel 2 conversion finishes when SEQEN =1. 10 = Issue ADC_INT after Channel 0 then Channel 2 conversion finishes when SEQEN =1. 11 = Reserved.
[1]	SEQTYPE	ADC Sequential Mode Type 0 = 2/3-shunt type 1 = 1-shunt type
[0]	SEQEN	ADC Sequential Mode Enable Bit When ADC sequential mode is enabled, two of three ADC channels from 0 to 2 will automatically convert analog data in the sequence of channel [0, 1] or channel[1, 2] or channel[0, 2] defined by MODESEL (ADC_SEQCTL[3:2]). 0 = ADC sequential mode Disabled. 1 = ADC sequential mode Enabled.

4.13.7.9 A/D PWM Sequential Mode Result Register (ADC_SEQDAT1/2)

Register	Offset	R/W	Description	Reset Value
ADC_SEQDAT1	ADC_BA+0x50	R	A/D PWM Sequential Mode First Result Register1	0x0000_0000
ADC_SEQDAT2	ADC_BA+0x54	R	A/D PWM Sequential Mode Second Result Register1	0x0000_0000

Bits	Description	
[31:18]	Reserved	Reserved.
[17]	VALID	Valid Flag This bit is set to 1 when ADC conversion is completed and cleared by hardware after the ADC_SEQDATx register is read. 0 = Data in RESULT[11:0] bits not valid. 1 = Data in RESULT[11:0] bits valid.
[16]	OV	Over Run Flag If converted data in RESULT[11:0] has not been read before the new conversion result is loaded to this register, OV is set to 1. It is cleared by hardware after the ADC_SEQDATx register is read. 0 = Data in RESULT[11:0] is recent conversion result. 1 = Data in RESULT[11:0] overwritten.
[15:12]	Reserved	Reserved.
[11:0]	RESULT	A/D PWM Sequential Mode Conversion Result This field contains conversion result of ADC.

4.14 Analog Control (ANAC)

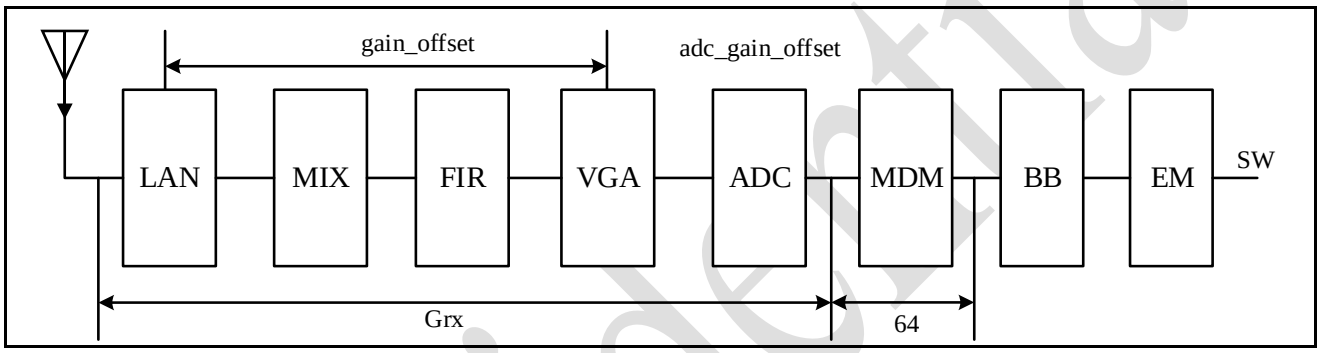
4.14.1 Overview

The PAN1020 ANAC contain following function:

- RSSI calculate, AGC of RX chain;
- Timing control FSM of RF;
- Control register of RF and dig2;
- SPI test mode;
- Digital to analog control signals mux;

4.14.2 Features

RSSI(dBm) = Pmodem – Grx – 64;



The AGC algorithm please reference。

The current Grx table：

Rssi_index	Grx
0	9
1	11
2	13
3	15
4	17
5	19
6	21
7	23
8	25
9	27
10	29
11	31

The next gain setting table:

6dB/ step	Input signal(dBm)		RF Selection (dB)					Output signal (dBm)		
Rssi_ index	from	to	LNA(dB) /1bit*NOTE	MIX(dB) /1bit	FIR(dB) /4bit	VGA(dB) /4bit	Total(dB) Gain	from	to	Control bit
0	-20	-10	0/00	0/0	4/0001	10/0010	14	-6	4	00_0_0001_0010
1	-26	-20	0/00	0/0	10/1101	10/0010	20	-6	0	00_0_1101_0010
2	-32	-26	0/00	0/0	2/0100	24/1000	26	-6	0	00_0_0100_1000
3	-38	-32	0/00	0/0	0/0000	32/1001	32	-6	0	00_0_0000_1001
4	-44	-38	0/00	0/0	8/1001	30/0111	38	-6	0	00_0_1001_0111
5	-50	-44	26/11	0/0	6/1000	12/0011	44	-6	0	11_0_1000_0011
6	-56	-50	26/11	0/0	0/0000	24/1000	50	-6	0	11_0_0000_1000
7	-62	-56	26/11	6/1	0/0000	24/1000	56	-6	0	11_1_0000_1000
8	-68	-62	26/11	0/0	12/1010	14/0101	62	-6	0	11_0_1010_0101
9	-74	-68	26/11	6/1	6/1000	36/1110	68	-6	0	11_1_1000_1110
10	-80	-74	26/11	6/1	14/1111	28/0110	74	-6	0	11_1_1111_0110
11	-95	-80	26/11	6/1	14/1111	34/1111	80	-15	0	11_1_1111_1111

*NOTE: The LNA bit can only be set to 00 or 11.

4.14.2.1 ADC Control Register 2 (ADC_CTL2)

Register	Offset	R/W	Description	Reset Value
ADC_CTL2	ADC_BA+0x58	R/W	A/D Control Register 2	0x0AA2_0000

Bits	Description	
[31:29]	Reserved	Reserved.
[28]	CMP_CTL_IB	ADC_CMP_CTL_IB_O, Default : 0
[27:26]	ICTL_CMP	ADC_ICTL_CMP_O, Default : 2
[25:24]	ICTL_VCM	ADC_ICTL_VCM_O, Default : 2
[23:22]	ICTL_VREF	ADC_ICTL_VREF_O, Default : 2
[21]	SEL_VREF	ADC_SEL_VREF_O, Default : 1
[20]	EN_BUFTST	EN_BUFTST, Default : 0
[19]	Reserved	Reserved
[18:16]	ADCDIV	ADC Clock Divider Could not be set as 0. Default : 2
[15:3]	Reserved	Reserved.
[2]	DMA_EN	ADC DMA enable, default:0
[1]	SH_SEL	1: External Sample Hold (SH) width with HOLD2(Internal Signal) 0: SH is SH. Default : 0
[0]	TEST_MODE	ADC Test enable

4.14.3 Register Map

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

Register	Offset	R/W	Description	Reset Value
ANAC Base Address: **ANAC_BA = 0x4007_0000**				
ANAC_AGCTB0	ANAC_BA+0x00	R/W	AGC Setting Table part 0	0x910A_2044
ANAC_AGCTB1	ANAC_BA+0x04	R/W	AGC Setting Table part 1	0xD132_2344
ANAC_AGCTB2	ANAC_BA+0x08	R/W	AGC Setting Table part 2	0xFD3E_A754
ANAC_AGCTB3	ANAC_BA+0x0C	R/W	AGC Setting Table part 3	0xFEFF_CFF8
ANAC_GAIN0	ANAC_BA+0x10	R/W	AGC Gain Table part 0	0x0283_4380
ANAC_GAIN1	ANAC_BA+0x14	R/W	AGC Gain Table part 1	0x062A_919C
ANAC_GAIN2	ANAC_BA+0x18	R/W	AGC Gain Table part 2	0x07EF_1CB6
ANAC_AGCCTL	ANAC_BA+0x1C	R/W	AGC control register	0x700B_8000
ANAC_AGCADC	ANAC_BA+0x20	R/W	AGC-ADC control register	0x0000_0000
ANAC_MCURF	ANAC_BA+0x24	R/W	MCU to RF control register	0x0000_0000
ANAC_FSMRF	ANAC_BA+0x28	R/W	FSM to RF control register	0x0A0A_6464
ANAC_PLLCTL	ANAC_BA+0x2C	R/W	PLL control register	0x0001_0000
ANAC_SDCTL	ANAC_BA+0x30	R/W	Sigma-Delta modulation control register	0x7F80_6020
ANAC_GSCTL	ANAC_BA+0x34	R/W	Gauss Filter control register	0x0800_1F1F
ANAC_TPCTL	ANAC_BA+0x38	R/W	Two Point Calibration control register	0x0008_0000
ANAC_TPSTS	ANAC_BA+0x3C	R/W	Two Point Calibration status register	0x0000_0000
ANAC_VCOCTL	ANAC_BA+0x40	R/W	VCO Calibration control register	0x0003_F085
ANAC_RXCTL	ANAC_BA+0x44	R/W	RF RX control register	0x0001_FF27
ANAC_TXCTL	ANAC_BA+0x48	R/W	RF TX control register	0x0072_7008
ANAC_RFPLL	ANAC_BA+0x4C	R/W	RFPLL control register	0x006A_D790
ANAC_RFPLL2	ANAC_BA+0x50	R/W	RFPLL control register	0x0028_A900
ANAC_LDOCTL	ANAC_BA+0x54	R/W	Xtal/RCO/DPLL/LDO control register	0x4EBA_0208
ANAC_RCCCTL	ANAC_BA+0x58	R/W	RC Calibration control register	0x0684_2120
ANAC_FRACTL	ANAC_BA+0x5C	R/W	Fra_spi_in register	0x0001_6AAA
ANAC_LDO2CTL	ANAC_BA+0x60	R/W	Xtal/RCO/DPLL/LDO control register2	0x8000_0044
ANAC_DRIFTCTL	ANAC_BA+0x64	R/W	XTAL drift control register	0x0000_0000
ANAC_3VCTL	ANAC_BA+0x68	R/W	Low power 3V logic control register	0x0003_5BFC

4.14.4 Register Description

4.14.4.1 AGC Setting Table part 0 Register (ANAC_AGCTB0)

Register	Offset	R/W	Description	Reset Value
ANAC_AGCTB0	ANAC_BA+0x00	R/W	AGC Setting Table part 0 Register	0x910A_2044

Bits	Description	Default

[31:22]	TB0H[9:0]	AGC Setting Table part 0 High word, Table[31:22]	10_0100_0100
[21:11]	TB0M[10:0]	AGC Setting Table part 0 Middle word, Table[21:11]	001_0100_0100
[10:0]	TB0L[10:0]	AGC Setting Table part 0 Low word, Table[10:0]	000_0100_0100

4.14.4.2 AGC Setting Table part 1 Register (ANAC_AGCTB1)

Register	Offset	R/W	Description	Reset Value
ANAC_AGCTB1	ANAC_BA+0x04	R/W	AGC Setting Table part 1 Register	0xD132_2344

Bits	Description		Default
[31:22]	TB1H[9:0]	AGC Setting Table part 1 High word, Table[64:55]	11_0100_0100
[21:11]	TB1M[10:0]	AGC Setting Table part 1 Middle word, Table[54:44]	110_0100_0100
[10:0]	TB1L[10:0]	AGC Setting Table part 1 Low word, Table[43:33]	011_0100_0100

4.14.4.3 AGC Setting Table part 2 Register (ANAC_AGCTB2)

Register	Offset	R/W	Description	Reset Value
ANAC_AGCTB2	ANAC_BA+0x08	R/W	AGC Setting Table part 2 Register	0xFD3E_A754

Bits	Description		Default
[31:22]	TB2H[9:0]	AGC Setting Table part 2 High word, Table[97:88]	11_1111_0100
[21:11]	TB2M[10:0]	AGC Setting Table part 2 Middle word, Table[87:77]	111_1101_0100
[10:0]	TB2L[10:0]	AGC Setting Table part 2 Low word, Table[76:66]	111_0101_0100

4.14.4.4 AGC Setting Table part 3 Register (ANAC_AGCTB3)

Register	Offset	R/W	Description	Reset Value
ANAC_AGCTB3	ANAC_BA+0x0C	R/W	AGC Setting Table part 3 Register	0xFEFF_CFF8

Bits	Description		Default
[31:22]	TB3H[9:0]	AGC Setting Table part 3 High word, Table[130:121]	11_1111_1011
[21:11]	TB3M[10:0]	AGC Setting Table part 3 Middle word, Table[120:110]	111_1111_1001
[10:0]	TB3L[10:0]	AGC Setting Table part 3 Low word, Table[109:99]	111_1111_1000

4.14.4.5 AGC Gain Table part 0 Register (ANAC_GAIN0)

Register	Offset	R/W	Description	Reset Value
ANAC_GAIN0	ANAC_BA+0x10	R/W	AGC Gain Table part 0 Register	0x0283_4380

Bits	Description		Default
[31:28]	Reserved	Reserved	0000
[27:21]	GTB0H	AGC Current gain table part 0 High word, Table[27:21]	0010_100

[20:14]	GTB0MH	AGC Current gain table part 0 Middle High word, Table[20:14]	0_0011_01
[13:7]	GTB0ML	AGC Current gain table part 0 Middle Low word, Table[13:7]	00_0011_1
[6:0]	GTB0L	AGC Current gain table part 0 Low word, Table[6:0]	000_0000

4.14.4.6 AGC Gain Table part 1 Register (ANAC_GAIN1)

Register	Offset	R/W	Description	Reset Value
ANAC_GAIN1	ANAC_BA+0x14	R/W	AGC Gain Table part 1 Register	0x062A_919C

Bits	Description		Default
[31:28]	Reserved	Reserved	0000
[27:21]	GTB1H	AGC Current gain table part 1 High word, Table[55:49]	0110_001
[20:14]	GTB1MH	AGC Current gain table part 1 Middle High word, Table[48:42]	0_1010_10
[13:7]	GTB1ML	AGC Current gain table part 1 Middle Low word, Table[41:35]	01_0001_1
[6:0]	GTB1L	AGC Current gain table part 1 Low word, Table[34:28]	001_1100

4.14.4.7 AGC Gain Table part 2 Register (ANAC_GAIN2)

Register	Offset	R/W	Description	Reset Value
ANAC_GAIN2	ANAC_BA+0x18	R/W	AGC Gain Table part 2 Register	0x07EF_1CB6

Bits	Description		Default
[31:28]	Reserved	Reserved	0000
[27:21]	GTB2H	AGC Current gain table part 2 High word, Table[83:77]	0111_111
[20:14]	GTB2MH	AGC Current gain table part 2 Middle High word, Table[76:70]	0_1111_00
[13:7]	GTB2ML	AGC Current gain table part 2 Middle Low word, Table[69:63]	01_1100_1
[6:0]	GTB2L	AGC Current gain table part 2 Low word, Table[62:56]	011_0110

4.14.4.8 AGC Control Register (ANAC_AGCCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_AGCCTL	ANAC_BA+0x1C	R/W	AGC Control Register	0x700B_8000

Bits	Description		SPI-Reg	Default
[31]	TB0H[10]	AGC_TB0H highest bit		0
[30]	TB1H[10]	AGC_TB1H highest bit		1
[29]	TB2H[10]	AGC_TB2H highest bit		1
[28]	TB3H[10]	AGC_TB3H highest bit		1
[27:23]	Reserved	Reserved	Reserved	Reserved
[22:19]	RSSI_IN-DEX	Read only, test for AGC.		0001 （ Read Only）
[18:15]	GOFFSET	gain_offset[3:0], -8~7dB, 1db/step, 4'd7 – gain_offset, default 4'd7		0111

[14]	AGCTH2	Threshold of the gain setting: 0 = 2dB; 1 = 4dB		0
[13]	AGCTH1	Threshold of the adjusting of RSSI: 0 = 0dB; 1 = 1dB		0
[12:11]	AGCMODE	RX GAIN control mode select: 00/11 = AGC; 01 = MCU; 10 = RF-SPI.		00
[10:0]	GAINOVRD	RX gain setting when AGCMODE = MCU.	{reg_02[7:0] , reg_01[7:5]}	000_0000_0000

4.14.4.9 AGC-ADCControl Register (ANAC_AGCADC)

Register	Offset	R/W	Description	Reset Value
ANAC_AGCADC	ANAC_BA+0x20	R/W	ADCControl Register in AGC module.	0x0000_0000

Bits	Description		Default
[31]	IQ_CHANGE	RX ADC_CODE_I/Q change or not: 1: I/Q change 0: I/Q not change	0
[30]	IQSMPLEDGE	0: ADC code sampled by negative edge 1: ADC code sampled by positive edge	0
[29:24]	ADCGOS	adc_gain_offset, used for RSSI calculate. It's a signed number and from -31 to +31 dB. The MSB is sign bit. 011111 = +31dB; 011110 = +30dB; 011101 = +29dB; …… 000000 = 0dB; 111111 = -1dB; 111110 = -2dB; 111101 = -3dB; …… 100001 = -31dB;	00_0000
[23:18]	Reserved	Reserved	Reserved
[17:16]	ADCBO	adc_backoff, used for rssi_index: 00 = -1; 01 = 0; 10 = +1; 11 = +2.	00
[15:14]	Reserved	Reserved	Reserved

[13:8]	PWRSAT	power_sat, ADC Saturation threshold setting: adcsat = rxpwr > PWRSAT.	00_0000
[7]	ADCFSCL	Adc fullscale selection signal, used for rssi_index: 0 = Rin is 26KOm; 1 = Rin is 6.5KOm.	0
[6:0]	ADCSATBO	adc_sat_backoff, back off when ADC sat, used for rssi_index.	000_0000

4.14.4.10 MCU to RF control register (ANAC_MCURF)

Register	Offset	R/W	Description	Reset Value
ANAC_MCURF	ANAC_BA+0x24	R/W	MCU to RF control register	0x0000_0000

Bits	Description		SPI-Reg	Default
[31:12]	Reserved	Reserved	Reserved	Reserved
[11]	RFMODE	RF Control signals and dig2 Control signals mode select, it's different with Flash-Ver: 0 = MCU(Normal work); 1 = SPI(RF TEST).	Reserved	0
[10:8]	RFGCMODE	RF General Control Signals(4 signals) mode select: 000 = MCU(power up); 001 = FSM(active); 010 = Timing Gen(deep sleep); 011 = SPI(RF test); 100 = TPCAL(two point calibration).	Reserved	000
[7:4]	Reserved			
[3]	ENRX	MCU to RF control signal when RFGCMODE = MCU.	reg_01[3]	0
[2]	ENTX	MCU to RF control signal when RFGCMODE = MCU.	reg_01[2]	0
[1]	ENRXSYN	MCU to RF control signal when RFGCMODE = MCU.	reg_01[1]	0
[0]	ENTXSYN	MCU to RF control signal when RFGCMODE = MCU.	reg_01[0]	0

4.14.4.11 FSM to RF control register (ANAC_FSMRF)

Register	Offset	R/W	Description	Reset Value
ANAC_FSMRF	ANAC_BA+0x28	R/W	FSM to RF control register	0x0A0A_6464

Bits	Description		Default
[31:29]	Reserved	Reserved	Reserved
[28:24]	RXRAMUP	RX RAMP UP time(about 10us), default is 10 us. The range is [0, 31].	0_1010
[23:21]	Reserved	Reserved	Reserved
[20:16]	TXRAMUP	TX RAMP UP time(about 10us), default is 10 us. The range is [0,31].	0_1010
[15]	Reserved	Reserved	Reserved

[14:8]	RXPLLUP	PLL lock time in RX mode, about <100us, default is 100 us. The range is [0,127].	110_0100
[7]	Reserved	Reserved	Reserved
[6:0]	TXPLLUP	PLL lock time in TX mode, about <100us, default is 100 us. The range is [0,127].	110_0100

4.14.4.12 PLL control register (ANAC_PLLCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_PLLCTL	ANAC_BA+0x2C	R/W	Dig2 PLL control register	0x0001_0000

Bits	Description		SPI-Reg	Default
[31:18]	Reserved	Reserved	Reserved	Reserved
[17]	INTMDEN	Int_mode_en, Dig2 PLL control signal.	reg_13[0]	0
[16]	FASTLKEN	Fast_lock_en, Dig2 PLL control signal.	reg_15[7]	1
[15:14]	DATARATE	Data_rate, Dig2 PLL control signal.	reg_14[7:6]	00
[13:8]	DFSEL	DF_SEL, Dig2 PLL control signal.	reg_14[5:0]	000000
[7:4]	Reserved	Reserved	Reserved	Reserved
[3:2]	SYNCBPS	Sync_bypass, Dig2 PLL control signal.	reg_15[6:5]	00
[1]	Reserved	Reserved	Reserved	Reserved
[0]	PHASEADJ	Phase_adj, Dig2 PLL control signal.	reg_15[3]	0

4.14.4.13 Sigma-Delta modulation control register (ANAC_SDCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_SDCTL	ANAC_BA+0x30	R/W	Dig2 Sigma-Delta modulation control register	0x7F80_6020

Bits	Description		SPI-Reg	Default
[31]	Reserved	Reserved	Reserved	0
[30:23]	RESERVEDH	Reserved High	reg_37[7:0]	11111111
[22:15]	PA_GC	PA_GC[0] => ANAC_SDCTL [20] PA_GC[1] => ANAC_SDCTL [18] PA_GC[2] => ANAC_SDCTL [17] PA_GC[3] => ANAC_SDCTL [16] PA_GC[4] => ANAC_SDCTL [15]	reg_38[7:0]	00000000
[14:13]	RESERVEDH3V	Reserved High	reg_39[7:6]	11
[12:11]	RESERVEDL3V	Reserved Low	reg_39[5:4]	00
[10:8]	CTLDLSB	Ctl_dither_lsb, Dig2 Sigma-Delta control signal.	reg_15[2:0]	000
[7]	DACMODE	Dac_mode, Dig2 Sigma-Delta control signal.	reg_22[2]	0
[6]	CTLDSHAPE	Ctl_dither_shape, Dig2 Sigma-Delta control signal.	reg_16[7]	0
[5]	CLKEN	Clk_en, Dig2 Sigma-Delta control signal.	reg_16[6]	1
[4]	DIV2EN	Div2_en, Dig2 Sigma-Delta control signal.	reg_16[5]	0

[3]	DSSHIFT	Ds_shift, Dig2 Sigma-Delta control signal.	reg_16[4]	0
[2]	INVCLKEN	Inv_clk_en, Dig2 Sigma-Delta control signal.	reg_16[3]	0
[1]	MASH2MD	Mash2_mode, Dig2 Sigma-Delta control signal.	reg_16[2]	0
[0]	SHIFTOS	Shift_offset, Dig2 Sigma-Delta control signal.	reg_16[1]	0

4.14.4.14 Gauss Filter control register (ANAC_GSCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_GSCTL	ANAC_BA+0x34	R/W	Dig2 Gauss Filter control register	0x0800_1F1F

Bits	Description		SPI-Reg	Default
[31:28]	Reserved	Reserved	Reserved	0000
[27:24]	GSSCALE	Gauss_scale, Dig2 Gauss Filter control signal.	reg_17[7:4]	1000
[23:19]	INBDDLY	Inband_delay, Dig2 Gauss Filter control signal.	reg_18[7:3]	0_0000
[18:14]	OUTBDDLY	Outband_delay, Dig2 Gauss Filter control signal.	reg_19[7:3]	0_0000
[13:8]	DBINBD	Dac_basal_inband, Dig2 Gauss Filter control signal.	reg_20[7:2]	01_1111
[7:6]	GSCTRL	Gauss_ctrl, Dig2 Gauss Filter control signal.	reg_20[1:0]	00
[5:0]	DBOUTBD	Dac_basal_outband, Dig2 Gauss Filter control signal.	reg_21[7:2]	01_1111

4.14.4.15 Two Point Calibration control register (ANAC_TPCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_TPCTL	ANAC_BA+0x38	R/W	Dig2 Two Point Calibration control register	0x0008_0000

Bits	Description		SPI-Reg	Default
[31:28]	Reserved			
[27]	TPENRX	RF control signal when RFGCMODE = TPCal.	Reserved	0
[26]	TPENTX	RF control signal when RFGCMODE = TPCal.	Reserved	0
[25]	TPENRXSYN	RF control signal when RFGCMODE = TPCal.	Reserved	0
[24]	TPENTXSYN	RF control signal when RFGCMODE = TPCal.	Reserved	0
[23:20]	Reserved	Reserved	Reserved	Reserved
[19:16]	TPCODEOUT	Two Point Calibration results, Read Only.	read_01[7:4]	
[15:10]	Reserved	Reserved	Reserved	Reserved
[9]	TPCALIE	Two Point Calibration Interrupt Enable bit: 0 = Interrupt function Disabled. 1 = Interrupt function Enabled.	Reserved	0
[8]	ENTPCAL	Two Point Calibration Enable bit: 0 = Disabled. 1 = Enabled.	reg_16[0]	0
[7]	TPSPITRIG	Two Point Calibration SPI_TRIG signal: 0 = Disabled. 1 = Enabled.	reg_21[1]	0

[6:4]	CODEOFSET	CODE OFFSET	reg_18[2:0]	000
[3:0]	TPMCODEIN	Two Point Calibration manual code in.	reg_17[3:0]	0000

4.14.4.16 Two Point Calibration status register (ANAC_TPSTS)

Register	Offset	R/W	Description	Reset Value
ANAC_TPSTS	ANAC_BA+0x3C	R/W	Two Point Calibration status register	0x0000_0000

Bits	Description		Default
[31:2]	Reserved	Reserved	
[1]	TPF	Two Point Calibration flag This bit is set by hardware when the Two Point Calibration done. 0 = Two Point Calibration not Done 1 = Two Point Calibration Done **Note**: this bit can be cleared to 0 by software writing 1.	0
[0]	TPIF	Two Point Calibration Interrupt flag This bit is set by hardware when the Two Point Calibration done. This will generate an interrupt if TPCALIE(ANAC_TPCTL[9]) = 1. 0 = Two Point Calibration not Done 1 = Two Point Calibration Done **Note**: this bit can be cleared to 0 by software writing 1.	0

4.14.4.17 VCO Calibration control register (ANAC_VCOCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_VCOCTL	ANAC_BA+0x40	R/W	Dig2 VCO Calibration control register	0x0003_F085

Bits	Description		SPI-Reg	Default
[31:20]	Reserved	Reserved	Reserved	000000000000
[19]	ADCCALSEL	ADC calibration mode select 0：self-calibration 1：manual calibration	reg_03[7]	0
[18:13]	ADCCALMANUAL	ADC manual calibration configure	reg_11[5:0]	01_1111
[12:8]	VCODERX	Manual code setting of RX when ENVCOCAL = 0; When ENVCOCAL = 1, vcoderx auto refresh when vco calibration done.	Reserved	1_0000
[7:3]	VCODETX	Manual code setting of TX when ENVCOCAL = 0; When ENVCOCAL = 1, vcodetx auto refresh when vco calibration done.	Reserved	1_0000
[2]	VCOCFT	Vco cal full trig: 0 = disable; 1 = enable.	reg_19[2]	1

[1]	SPICALTRIG	Vco cal spi cal trig: 0 = disable; 1 = enable.	reg_19[1]	0
[0]	ENVCOCAL	Vco calibration enable: 0 = disable; 1 = enable.	reg_19[0]	1

4.14.4.18 RF RX Chain control register (ANAC_RXCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_RXCTL	ANAC_BA+0x44	R/W	RF RX Chain control register	0x0001_FF27

Bits	Description		SPI-Reg	Default
[31]	BPFIQSEL	BPF IQ swap	Reg33[7]	0
[30]	ADCPHASESEL	ADC clock phase selection	Reg33[6]	0
[29]	ADCSEL0P75M	ADC center frequency 0.75MHz	Reg33[5]	0
[28:21]	RXRSV	RX Reserved control signals.	reg_24[7:0]	00000000
[20:17]	Reserved	Reserved		0000
[16]	ADCBPEN	ADC bandpass mode enable: 0 = disable; 1 = enable.	Reg_26[7]	1
[15:12]	ADCGC	ADC gain: 1111 = 6dB; 0111 = 0dB; 0011 = -6dB; 0001 = -12dB; 0000 = -18dB.	Reg_25[3:0]	1111
[11]	ADCIQSEL	ADC IQ swap: 0 = not swap; 1 = swap.	Reg_26[6]	1
[10]	ADCEN	ADC enable: 0 = disable; 1 = enable.	Reg_26[5]	1
[9]	LNAEN	LNA enable: 0 = disable; 1 = enable.	reg_03[6]	1
[8]	RXMIXEN	RX Mixer enable: 0 = disable; 1 = enable.	reg_03[5]	1
[7]	IBLNA	LNA bias current: 0 = 20uA; 1 = 30uA.	reg_03[4]	0
[6]	IBRMDIV2	RXMIX Divider bias current:	reg_03[3]	0

		0 = 50uA; 1 = 100uA.		
[5:3]	LNACAP	LNA load frequency tuning: 111 = Lowest frequency; 000 = Highest frequency.	reg_03[2:0]	100
[2]	BPFEN	RX BPF enable: 0 = disable; 1 = enable.	reg_04[7]	1
[1]	RXVGAEN	RX VGA enable: 0 = disable; 1 = enable.	reg_04[6]	1
[0]	IBBPF	BPF bias current: 0 = 1uA; 1 = 2uA.	reg_04[5]	1

4.14.4.19 RF TX Chain control register (ANAC_TXCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_TXCTL	ANAC_BA+0x48	R/W	RF TX Chain control register	0x0072_7008

Bits	Description		SPI-Reg	Default
[31]	Reserved	Reserved		
[30:23]	TXRSV	TX Reserved control signal.	reg_27[7:0]	00000000
[22]	TXDACEN	DAC enable: 0 = disable; 1 = enable.	reg_04[4]	1
[21]	DALPFBW	DAC LPF BW control: 0 = low BW; 1 = High BW.	reg_04[3]	1
[20:18]	DAVREFMB	DAC upper reference select: 000 = lowest; 111 = highest.	reg_04[2:0]	100
[17:15]	DAVREFLB	DAC lower reference select: 000 = highest; 111 = lowest.	reg_05[7:5]	100
[14]	DAGAIN	DAC gain select: 0 = low gain; 1 = high gain.	reg_05[4]	1
[13]	PAEN	TX PA enable: 0 = disable; 1 = enable.	reg_05[3]	1
[12:10]	BLCTRIM	Balun frequency tuning: 000 = highest;	reg_05[2:0]	100

		111 = lowest.		
[9]	ENTXDATST	TX DAC test enable: 0 = disable; 1 = enable.	reg_06[2]	0
[8:4]	Reserved	Reserved	Reserved	00000
[3:0]	PABIAS	PA bias setting: 1111 = 155uA; 1000 = 85uA; 0000 = 5uA.	reg_07[7:4]	1000

4.14.4.20 RF PLL control register (ANAC_RFPLL)

Register	Offset	R/W	Description	Reset Value
ANAC_RFPLL	ANAC_BA+0x4C	R/W	RF PLL control register	0x006A_D790

Bits	Description		SPI-Reg	Default
[31:23]	Reserved	Reserved	Reserved	000000000
[22]	ENPLLBIAS	RFPLL bias enable: 0 = disable; 1 = enable.	reg_07[3]	1
[21]	ENPFD	RFPLL pfd/cp enable: 0 = disable; 1 = enable.	reg_08[7]	1
[20:19]	PFDDLY	RFPLL pfd delay setting: 00 = shortest; 11 = longest.	reg_06[1:0]	01
[18:16]	CPI	RFPLL cp setting: 100 = 10uA; 010 = 5uA; 001 = 2.5uA.	reg_07[2:0]	010
[15]	VREFVS	VCO control voltage select in calibration mode: 0 = disable; 1 = enable.	reg_08[6]	1
[14]	ENVCO	RFVCO enable: 0 = disable; 1 = enable.	reg_08[5]	1
[13:12]	VCOCT	RF VCO coarse tuning: 00 = highest; 11 = lowest.	reg_08[4:3]	01
[11:10]	VCOBIAS	RF VCO bias setting: 00 = 100uA; 01 = 150uA; 10 = 200uA;	reg_08[2:1]	01

		11 = 250uA.		
[9]	ENVCOBUF	RFVCO buffer enable: 0 = disable; 1 = enable.	reg_08[0]	1
[8]	ENDIV2	RFPLL divide by 2 enable: 0 = disable; 1 = enable.	reg_09[7]	1
[7]	ENDIV89	RFPLL divide by 8/9 enable: 0 = disable; 1 = enable.	reg_09[6]	1
[6]	ITDIV2	RFPLL divide by 2 bias setting: 0 = 50uA; 1 = 100uA.	reg_09[5]	0
[5]	FBDOE	RFPLL feedback divider output test enable: (Reserved??) 0 = disable; 1 = enable.	reg_09[4]	0
[4]	ENLPF	RFPLL loop filter enable: 0 = disable; 1 = enable.	reg_09[3]	1
[3]	SPICPO	Charge pump output test enable: 0 = disable; 1 = enable.	reg_09[2]	0
[2]	SPIVCTRL	VCO control test enable: 0 = disable; 1 = enable.	reg_10[7]	0
[1:0]	LPFRESSEL	RFPLL loop filter 3rd pole select	reg_09[1:0]	00

4.14.4.21 RF PLL control register2(ANAC_RFPLL2)

Register	Offset	R/W	Description	Reset Value
ANAC_RFPLL2	ANAC_BA+0x50	R/W	RF PLL control register 2	0x0028_A900

Bits	Description		SPI-Reg	Default
[31:22]	Reserved	Reserved	Reserved	0000000000
[21:18]	MODTST	Test Mode selection 0000 = Mux Mixer output to test PAD(A/B/C/D), with internal buffer 0001 = Mux BPF output to test PAD(A/B/C/D), with internal buffer 0010 = Mux VGA output to test PAD(A/B/C/D), with internal buffer 0011 = Mux ADC input to test PAD(A/B/C/D)	reg_40[7:4]	1010

		0100 = Mux Mixer output to test PAD(A/B/C/D), without internal buffer 0101 = Mux BPF output to test PAD(A/B/C/D), without internal buffer 0110 = Mux VGA output to test PAD(A/B/C/D), without internal buffer 0111 = Mux ADC input to test PAD(A/B/C/D) 1000 = Mux Voltage test to test PAD TEST_VCON to A TEST_CP_OUT to B DPLL_VCTL to C TX_DA to D 1001 = Mux Voltage test to test PAD VREF_ANA to A VREF_VCO to B VDD_IF to C VDD_VCO to D 1010 = Mux Voltage test to test PAD VREF_1p0 to A VREF_1p2 to B IB_10u_TST to C Vref_Low_Power to D 1011 = Mux Voltage test to test PAD VCM_TST to A VOM_TST to B VOP_TST to C Temp_Out to D Other = disable test mode		
[17:14]	ENVOTSTPAD	Test PAD voltage output enable	reg_40[3:0]	0010
[13]	ENCPNSHIFT	RFPLL charge pump constant current enable: 0 = disable; 1 = enable.	reg_28[7]	1
[12:11]	ICPNS	RFPLL charge pump constant current setting	reg_28[6:5]	01
[10:9]	Reserved	Reserved		00
[8]	ENTSTMOD	Enable analog test Mux.	reg_28[2]	1
[7:0]	PLLRSV	PLL Reserved control signal.	reg_29[7:0]	00000000

4.14.4.22 RF XTal/LDO/RCO/DPLL control register (ANAC_LDOCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_LDOCTL	ANAC_BA+0x54	R/W	RF LDO control register	0x4EBA_0208

Bits	Description		SPI-Reg	Default
[31]	Reserved	Reserved	Reserved	0

[30:29]	CAPTRIMRCO	32kHz RC OSC coase tuning	reg_39[3:2]	10
[28]	LDOFLASH	flash 1.8V voltage control 0 = 1.8V 1 = 1.9V	reg_39[1]	0
[27]	ENFLASHLDO	flash 1.8V voltage enable control flash 1.2V power gate control	reg_39[0]	1
[26:24]	VSELLDO0P7V		reg_41[7:5]	110
[23]	ENLDO0P7V		reg_28[3]	1
[22]	Reserved	Reserved	reg_30[0]	0
[21]	OSCIC	Xtal bias setting	reg_11[7]	1
[20]	EN32KRCO	1、32K RCO clock enable control	reg_41[4]	1
[19]	ENLOWPOWER-BIAS	Low-power LDO and low-power 32K(RC&XTAL) bias voltage enable control	reg_41[3]	1
[18:17]	DPLLDLY	DPLL pfd delay setting	reg_12[5:4]	01
[16:10]	Reserved	Reserved	reg_10[6:0]	0000000
[9:5]	VBGTRIM[4:0]	Bandgap voltage trimming bits	reg_42[4:0]	10000
[4:3]	IBGTRIM	Iref trimming bits	reg_30[5:4]	01
[2:0]	Reserved	Reserved		000

4.14.4.23 RF RC Calibration control register (ANAC_RCCCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_RCCCTL	ANAC_BA+0x58	R/W	RF RC Calibration control register	0x0684_2120

<table>
<tr><th>Bits</th><th>Description</th><th colspan="4"></th><th>SPI-Reg</th><th>Default</th></tr>
<tr><td>[31:29]</td><td>Reserved</td><td colspan="4">Reserved</td><td>Reserved</td><td>000</td></tr>
<tr><td>[28]</td><td>RCCLKSEL</td><td colspan="4">RC Calibr clock source select
0 = 4M
1 = HXT clock</td><td>reg_30[7]</td><td>0</td></tr>
<tr><td>[27]</td><td>ENLVR</td><td colspan="4">LVR enable
LVR active：1.77V</td><td>reg_30[6]</td><td>0</td></tr>
<tr><td>[26:25]</td><td>BODVL</td><td colspan="4"><table><tr><th>BODEN</th><th>BODVL</th><th>voltage(1->0)</th><th>voltage(0->1)</th></tr><tr><td>1</td><td>00</td><td>2.157V</td><td>2.276V</td></tr><tr><td>1</td><td>01</td><td>2.306V</td><td>2.43V</td></tr><tr><td>1</td><td>10</td><td>2.477V</td><td>2.616V</td></tr><tr><td>1</td><td>11</td><td>2.907V</td><td>3.067V</td></tr></table></td><td>reg_41[1:0]</td><td>11</td></tr>
<tr><td>[24]</td><td>BODEN</td><td colspan="4">BOD enable bit.</td><td>reg_41[2]</td><td>0</td></tr>
<tr><td>[23:22]</td><td>GAINTEMP</td><td colspan="4">The gain control of temperature detection circuit</td><td>reg_42[7:6]</td><td>10</td></tr>
<tr><td>[21]</td><td>ENTEMP</td><td colspan="4">Temperature detection enable control</td><td>reg_42[5]</td><td>0</td></tr>
<tr><td>[20:19]</td><td>XTALMOD</td><td colspan="4">XTAL frequency select
00 = 16M
01 = 12M</td><td>reg_43[2:1]</td><td>00</td></tr>
</table>

		10 = 24M 11 = forbid		
[18]	ENDPLL	Digital DPLL enable control	reg_43[0]	1
[17]	RFSEL32KXTAL	32K XTAL feedback resistance select 0 = 10M res provided by chip 1 = res provided by user	reg_44[7]	0
[16:15]	RDSEL32KXTAL	32K XTAL power consumption resistance select 00 = 0K 01 = 25K 10 = 50K 11 = 75K	reg_43[4:3]	00
[14:12]	ISEL32KXTAL	32K XTAL bias current select 000 = 200nA 001 = 400nA …. 111 = 1600nA	reg_43[7:5]	010
[11]	EN32KXTAL	32K XTAL clock enable control	reg_44[6]	0
[10]	RCCSEL	RC code select: 0 = use auto calibration result; 1 = use manual setting.	reg_12[3]	0
[9]	RCCCLKI	CLK phase select	reg_12[2]	0
[8]	RCCALEN	Power down RCCAL: 0 = power up; 1 = power down.	reg_12[1]	1
[7]	RCCRSTN	RCCAL reset	reg_12[0]	0
[6]	RCCALST	RCCAL start	reg_13[7]	0
[5:0]	RCCALIN	Manual RC code	reg_13[6:1]	10_0000

4.14.4.24 Fra_spi_in register (ANAC_FRACTL)

Register	Offset	R/W	Description	Reset Value
ANAC_FRACTL	ANAC_BA+0x5C	R/W	FRA_SPI_IN register	0x0001_6AAA

Bits	Description		SPI-Reg	Default
[31:28]	Reserved	Reserved		
[27]	PLL_RSTN_SEL	PLL_RSTN_FPGA，PLL_RSTN_SEL 0 = PLL_RSTN is provided by chip；default 1 = PLL_RSTN is provided by input PLL_RSTN_FPGA	reg_10[3]	0
[26]	CAL_DONE_SEL	CAL_DONE_FPGA，CAL_DONE_SEL 0 = cal_done is provided by chip；default 1 = cal_done is provided by input CAL_DONE_FPGA	reg_10[2]	0
[25]	DA_IN_SEL	DA_IN_FPGA[5:0]，DA_IN_SEL 0 = da_in is provided by chip；default	reg_10[1]	0

		1 = da_in is provided by input DA_IN_FPGA		
[24]	FRACN_OUT_SEL	FRACN_OUT_FPGA[9:0]，FRACN_OUT_SEL 0 = fracn_out is provided by chip；default 1 = fracn_out is provided by input FRACN_OUT_FPGA	reg_10[0]	0
[23:18]	RCCALVAL	For RCCAL test identifier. Read only	rf_read03[5:0]	xxxxxx
[17]	VCONODLY	vco_delay select	reg_44[5]	0
[16]	INTDLY	sigma_delta integer process select	reg_44[4]	1
[15]	CH7_REG	Decimal de-bounce en， 0 = default 1 = enable	reg_31[7]	0
[14]	FRA_EN	Decimal 0.75/0.5/0.25 de-bounce enable	reg_31[6]	1
[13:12]	VCODLY_SEL	VCO_delay select control 00 = 3us 01 = 6us 10 = 9us 11 = 12us.	reg_31[5:4]	10
[11:0]	FRA_IN	Optimization of decimal de-bounce	{reg_31[3:0], reg_32[7:0]}	12'hAAA

4.14.4.25 Xtal/RCO/DPLL/LDO control register2(ANAC_LDO2CTL)

Register	Offset	R/W	Description	Reset Value
ANAC_LDO2CTL	ANAC_BA+0x60	R/W	RF LDO control register2	0x8000_0044

Bits	Description		SPI-Reg	Default
[31:26]	LPOSEL[5:0]	32kHz RC OSC tuning	reg_45[5:0]	100000
[25:23]	LDORXADC	[2]:RXADC/LO LDO bypass [1:0] RXADC/LO LDO voltage select	reg_44[2:0]	000
[22:20]	LDOPLL	[2]:RFPLL/LO LDO bypass [1:0] RFPLL/LO LDO voltage select	Reg33[4:2]	000
[19:17]	LDORF	[2]RF LDO bypass [1:0] RF LDO voltage select	Reg33[1] Reg34[7:6]	000
[16:14]	LDOIF	[2]Analog LDO bypass [1:0] Analog LDO voltage select	Reg33[0] Reg34[5:4]	000
[13:11]	LDODPLL	[2]DPLL LDO bypass [1:0] DPLL LDO voltage select	Reg34[3:1]	000
[10:8]	LDOVCO	[2]RFVCO LDO bypass [1:0] RFVCO LDO voltage select	Reg34[0] Reg35[7:6]	000
[7:5]	LDODVDD	[2]DVDD LDO bypass [1:0] DVDD LDO voltage select	Reg35[5:3]	010
[4:2]	LDOV18	[2]1.8V LDO bypass (Reserved) [1:0] 1.8V LDO voltage select	Reg35[2:0]	001
[1]	reserved		Reg36[7]	0

[0]	DPLLPD	DPLL power down (Reserved)	Reg36[6]	0

4.14.4.26 DRIFT CTRL REGISTER (ANAC_DRIFTCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_DRIFTCTL	ANAC_BA+0x64	R/W	XTAL DRIFT control register	0x0000_0000

Bits	Description		SPI-Reg	Default
[31:22]	Reserved			0
[21]	DRIFTFIX-SEL	HXT temperature drift fix enable control	reg_44[3]	0
[20:0]	DRIFTFIX	HXT temperature drift fix value	{reg_46[7:0],reg_47[7:0],reg_48[4:0]}	00000000000000000000

4.14.4.27 3V LOGIC CTRL REGISTER (ANAC_3VCTL)

Register	Offset	R/W	Description	Reset Value
ANAC_3VCTL	ANAC_BA+0x68	R/W	Low power 3V logic control register	0x0003_5BFC

Bits	Description		SPI-Reg	Default
[31:18]	Reserved			
[17]	en_ ibias_sleep_en	Enable Power Down IBIAS in sleep mode 0 = do not power down IBIAS in sleep mode 1 = power down IBIAS in sleep mode **Note**: It's active after settingCONFIG_ACT to 1 and re-turning to 0 automatically	N/A	1
[16]	EN_IBIAS_1P2V	0 = disable DPLL current bias 1 = enable DPLL current bias **Note**: It's active after settingCONFIG_ACT to 1 and re-turning to 0 automatically	N/A	1
[15]	XTAL_NRCO_32K_1P2V	32K clock source selection. 0 = 32k using internal oscilltor(LIRC) 1 = 32k using external xtal(LXT)	N/A	0
[14]	en_32k_xtal_sleep_en	set 1 to enable EN_32K_XTAL goto 0 automaticlly in sleep mode	N/A	1
[13]	en_32k_rco_sleep_en	set 1 to enable EN_32K_RCO goto 0 automaticlly in sleep mode	N/A	0
[12]	osc_ic_sleep_en	set 1 to enable OSC_IC goto 0 automaticlly in sleep mode	N/A	1
[11]	en_flashldo_sleep_en	set 1 to enable EN_FLASHLDO goto 0 automaticlly in sleep mode	N/A	1
[10]	en_ldo_0p7v_sleep_en	set 1 to enable EN_LDO_0P7V goto 0 automaticlly in sleep mode	N/A	0
[9]	ldo_en_sleep_en	set 1 to enable LDO_EN goto 0 automaticlly in sleep mode	N/A	1

[8]	en_pm_sleep_en	set 1 to enable EN_PM goto 0 automaticlly in sleep mode	N/A	1
[7]	ce_int_sleep_en	set 1 to enable CE_INT goto 0 automaticlly in sleep mode	N/A	1
[6]	pwr_up_sleep_en	set 1 to enable PWR_UP goto 0 automaticlly in sleep mode	N/A	1
[5]	LDO_EN_1P2V	Firmware control LDO_EN, active after setting config_act to 1 and return to 0 automaticlly	N/A	1
[4]	CE_INT_1P2V	Firmware control CE_INT, active after setting config_act to 1 and return to 0 automaticlly	N/A	1
[3]	EN_PM_1P2V	Firmware control EN_PM, active after setting config_act to 1 and return to 0 automaticlly	N/A	1
[2]	PWR_UP_1P2V	Firmware control PWR_UP, active after setting config_act to 1 and return to 0 automaticlly	N/A	1
[1]	configact32k	Configuration complete flag (delay two 32k cycles from config_act) READ ONLY	N/A	0
[0]	configact	Set to 1 to sync the configuration to 3.3V power domain, reuturn to 0 after config complete	N/A	0

5 Welding Process Recommendations

5.1 Lead-free reflow process parameters

The lead-free reflow soldering process curve is shown in Figure 5-1

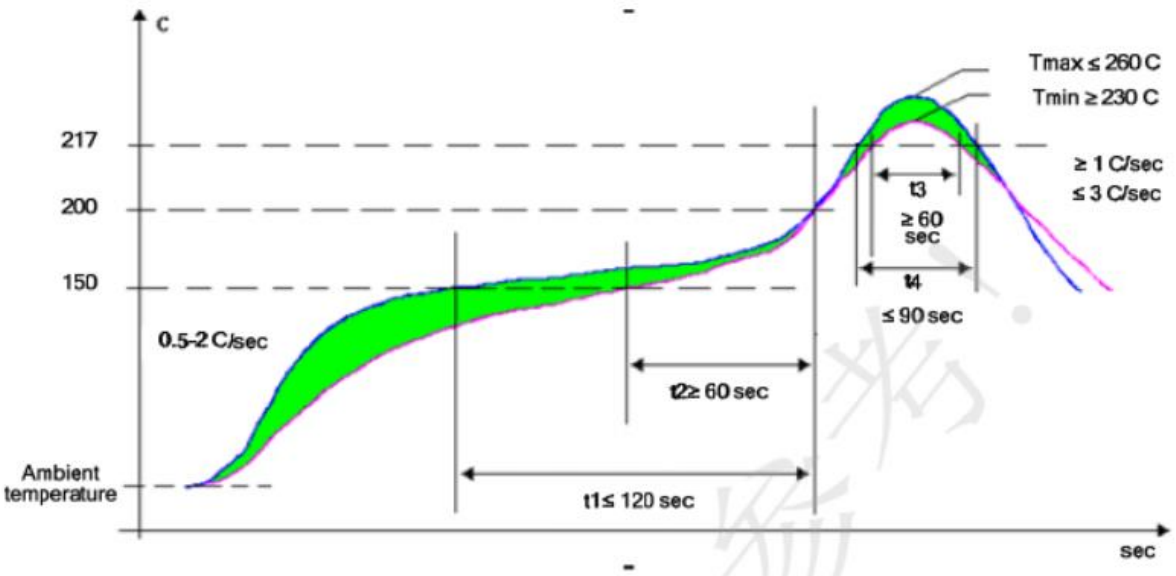


Figure 5-1 Lead-free reflow soldering process curve

Lead-free reflow process parameters are shown in Table 5-1.

Table 5-1 Lead-free reflow process parameters

Zone	Time	Heating rate	Peak temperature	Cooling rate
Preheating zone (40~150℃)	60~150s	≤2.0℃/s	-	-
Temperature zone(150~200℃)	60~120s	<1.0℃/s		-
Recirculation zone(>217℃)	60~90s	-	230-260℃	-
Cooling zone（Tmax~180℃）	-	-	-	1.0 ℃ ≤ Slope ≤ 4.0℃/s

Description

- Preheating zone: The temperature is from 40℃ ~ 150℃, the temperature rising rate is controlled at about 2℃/ s, and the temperature is 60 ~ 150 s.
- Average temperature zone: The temperature is from 150℃ ~ 200℃, the temperature stably and slowly rises. The temperature rise rate is less than 1℃/s, and the time is controlled in 60 ~120s. (Note: this area must be heated slowly, otherwise it will lead to poor welding).
- Recirculation zone: The temperature is from 217℃ ~ Tmax ~ 217℃, and the whole interval is controlled at 60 ~ 90 s.

- Cooling zone: The temperature is from Tmax ~ 180℃, and the temperature drop rate cannot exceed 4℃/s.
- The temperature should be raised from room temperature 25℃ ~250℃ time should not exceed 6 minutes.
- The reflow profile is only the recommended value, and the client needs to be adjusted accordingly to the actual production situation.
- The reflow time is targeted at 60 to 90 s. For some boards with large heat capacity and time requirements, the reflow time can be relaxed to 120s. Package temperature resistance standard reference IPC/JEDEC J-STD-020D standard, package temperature measurement method reference JEP 140 standard.

IPC/JEDEC J-STD-020D standard, package temperature measurement method reference JEP 140 standard: The temperature resistance standards for lead-free device packages in the IPC/JEDEC 020D are shown in Table 5-2.

Table 5-2 Lead-free device package temperature resistance standard in IPC/JEDEC 020D

Package Thickness	Volume mm^3 <350	Volume mm^3 350~2000	Volume mm^3 >2000
<1.6mm	260℃	260℃	260℃
1.6mm~2.5mm	260℃	250℃	245℃
>2.5mm	250℃	245℃	245℃

The solder joints (bumps, pins) and external heat sinks are not included in the volume calculation.

Reflow soldering process curve measurement method:

JEP140 Recommendation: For devices with smaller thickness, when directly measuring the temperature of the package, place the thermocouple directly on the device table. Due to the requirement of the thickness of the device, it is recommended to use a method of embedding a thermocouple in the surface of the package (except for thin devices, which cannot be drilled), as shown in Figure 5-2.

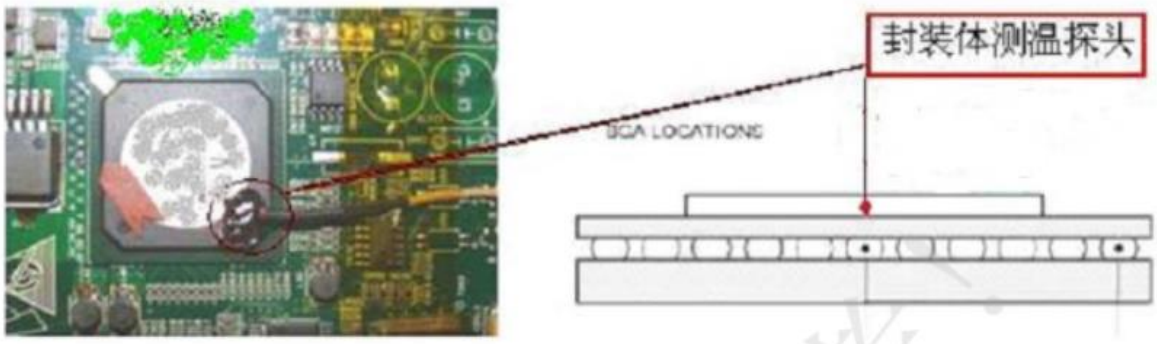


Figure 5-2 Package temperature measurement diagram

Description

If it is a QFP package chip, just place the temperature probe on the pin.

5.2 Mixed reflow process parameters

During reflow soldering, if device mixing occurs, the normal soldering of lead-free devices should be ensured first. The specific requirements are shown in Table 5-3:

Table 5-3 Mixed reflow soldering process parameter table

Numerical requirements		Leaded BGA	Lead-free BGA	Other devices
Preheating zone (40~150℃)	Time	60-150Sec		
	Temperature ramp	<2.5℃/Sec		
Temperature zone (150~183℃)	Time	30-90Sec		
	Temperature ramp	<1.0℃/Sec		
Recirculation zone (>183℃)	Peak temperature	210~240℃	220~240℃	210~245℃
	Time	30~120s	60~120s	30~120s
Cooling zone（Tmax~150℃）	Cooling slope	1.0℃/Sec≤Slope≤4.0℃/Sec		

Attention

The above process parameters are all for the solder joint temperature. The hottest and coldest spots of the solder joints on the board must meet the above specifications.

In the curve modulation, it is also necessary to meet the temperature resistance requirements of the package of the components on the single board. The temperature standard of the package is in accordance with

IPC/JEDEC J-STD-020D standard, package temperature measurement method according to JEP 140 standard.

IPC/JEDEC J-STD-020D standard, package temperature measurement method according to JEP 140 standard.

The temperature resistance standards for leaded device packages in IPC/JEDEC 020D are shown in Table 5-4:

Table 5-4 Leaded device package temperature standard

Package Thickness	Temperature 1 (Package Volume <350 mm^3 or 0.02 $in.^3$)	Temperature 1 (Package Volume ≥350 mm^3 or 0.02 $in.^3$)
< 2.5 mm (0.10 in.)	235°C (455°F)	220°C (428°F)
≥2.5 mm (0.10 in.)	220°C (428°F)	220°C (428°F)

The solder joints (bumps, pins) and external heat sinks are not included in the volume calculation.

The JEP140 standard stipulates that the method of measuring the package temperature is the same as the lead-free process. Please refer to 5.1 Lead-free reflow process parameters.

6 Moisture Sensitivity Index

6.1 Hisilicon products moisture-proof packaging

6.1.1 Packaging Information

Dry vacuum packaging materials include:

- Humidity Indicator Card (HIC)
- Moisture Barrier Bag (MBB)
- Desiccant

Figure 6-1 Dry vacuum packaging material schematic

6.1.2 Moisture sensitive product feed inspection

After opening the vacuum bag before production use (SMT) :

- If the maximum indication point of HIC has changed (not blue or khaki), the product must refer to Table 6-2.
- If the 10% RH dot in the HIC is blue or khaki, the product is very dry and can be vacuum packaged after only replacing the moisture barrier.
- If the 10% RH dot in HIC is not blue or khaki, the 5% RH dot has turned red or light green, indicating that the product has been wet. Refer to Table 6-2 for rebake.

6.2 Storage and use

【Storage environment】

It is recommended that the product be stored in a vacuum package and stored at <30℃/ 60% RH.

【Shelf life】

Storage environment <30℃/ 60% RH, vacuum packaging, shelf life (storage period)≦12 months.

【Floor life】

Under ambient conditions <30℃/ 60%, floor life is shown in Table 6-1 below.

Table 6-1 Floor life reference table

MSL	Floor life(out of bag) at factory ambient≦30℃/60% RH or as stated
1	Unlimited at ≦30℃/85% RH
2	1 year
2a	4 weeks
3	168 hours
4	72 hours
5	48 hours
5a	24 hours
6	Mandatory bake before use, must be reflowed within the time limit specified on the label

【Use of moisture sensitive products】

- The product is exposed continuously or cumulatively for more than 2 hours at≦30℃/60% RH. It is recommended to carry out rebake and vacuum dry the package.
- The product is exposed at≦30℃/60% RH for no more than 2 hours. It is not necessary to use rebake, but a new desiccant should be replaced and vacuum dried.

For the storage and usage principles not mentioned in this document, please refer to JEDEC J-STD-033A directly.

6.3 Rebake

【Scope of use】

ICs that needs to be re-baked (moisture -sensitive products)

【Rebake reference table】

Table 6-2 Rebake reference table

Body thickness	Level	bake@125℃	bake@90℃≦5% RH	bake@40℃≦5% RH
≦1.4mm	2a	3 hours	11 hours	5 days

	3	7 hours	23 hours	9 days
	4	7 hours	23 hours	9 days
	5	7 hours	24 hours	10 days
	5a	10 hours	24 hours	10 days
≦2.0mm	2a	16 hours	2 days	22 days
	3	17 hours	2 days	23 days
	4	20 hours	3 days	28 days
	5	25 hours	4 days	35 days
	5a	40 hours	6 days	56 days
≦4.5mm	2a	48 hours	7 days	67 days
	3	48 hours	8 days	67 days
	4	48 hours	10 days	67 days
	5	48 hours	10 days	67 days
	5a	48 hours	10 days	67 days

Description

- Table 6-2 shows the minimum necessary baking time after damp.
- Re-baking prefers low-temperature baking.
- Please refer to JEDEC for details.

7 Electrical Characteristics

All the parameters are accurate to the one decimal place.

7.1 Absolute Maximum Ratings

Table 7-1 Absolute maximum ratings

Symbol	Description	Parameter			Unit
		Min	Typ	Max	
VDD	VDD1/VDD2	-0.3	-	3.6	V
V_I	Input voltage	-0.3	-	VDD	V
V_O	Output voltage	VSS	-	VDD	V
T_{OP}	Operating Temperature	-40	-	85	℃
T_{STG}	Storage Temperature	-40	-	125	℃

Note: Exceeding one or more of the limiting values may cause permanent damage to PAN1020.

Caution: Electrostatic sensitive device, comply with protection rules when operating.

7.2 DC Electrical Characteristics

Table 7-2 Voltage and current

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
VDD1/VDD2	Power Supply	2.2	3	3.6	V	TA=25℃
VSS	Ground	-	0	-	V	-
$I_{DP\_SLP\_PAD}$	Deep sleep current	1.5	2	2.5	uA	MCU power down, SRAM maintain, HCLK and 32K RC off, wake up by GPIO or RE-SET
$I_{DP\_SLP\_RC}$	Deep sleep current	2	3	5	uA	MCU power down, SRAM maintain, HCLK off, 32K RC on
$I_{TX,0dBm}$	Operating Current of TX mode	-	17	-	mA	0dBm output power
$I_{TX,8dBm}$	Operating Current of TX mode	-	31	-	mA	8dBm output power
$I_{TX,10dBm}$	Operating Current of TX mode	-	41	-	mA	10dBm output power
I_{RX}	Operating Current of RX mode	-	16	-	mA	Maximum LNA Gain
V_{OH}	Output high level voltage	VDD-0.3	-	VDD	V	-
V_{OL}	Output low level voltage	VSS	-	VSS+0.3	V	-
V_{IH}	Input high level voltage	2.0	3	3.6	V	-
V_{IL}	Input low level voltage	VSS	-	VSS+0.3	V	-

7.3 16 MHz Crystal Oscillator Characteristics

Table 7-3 16M RC oscillator

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$F_{XTAL(16M)}$	crystal oscillator frequency	-	-	16	-	MHz
ESR(16M)	equivalent series resistance	-	-	-	80	Ω
$\Delta f_{XTAL(16M)}$	crystal frequency tolerance	-	-20	-	20	ppm
$V_{CLK(EXT)(16M)}$	external clock voltage	-	0.1	0.8	-	V
$\varphi N_{(EXTERNAL)16M}$	phase noise	fC = 50 kHz in case of an external reference clock	-	-	-130	dBc/Hz

7.4 32 KHz Crystal Oscillator Characteristics

Table 7-4 32K RC oscillator

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{CLK(EXT)(32K)}$	external clock voltage	peak-peak voltage of external clock at XTAL32Kp, pin XTAL32Km floating. note: XTAL32Kp is internally AC coupled	0.1	0.2	1.5	V
$f_{XTAL(32k)}$	crystal oscillator frequency	frequency range for an external clock (for a crystal, use either 32.000 kHz or 32.768 kHz)	TBD	32.768	TBD	KHz
ESR(32k)	equivalent series resistance	-	-	-	100	KΩ
$\Delta f_{XTAL(32k)}$	crystal frequency tolerance (including aging)	Timing accuracy is dominated by crystal accuracy. A much smaller value is preferred	-250	-	250	ppm

7.5 Stable Low Frequency RCX Oscillator Characteristics

Table 7-5 Stable Low Frequency RCX Oscillator

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$f_{RC(RCX)}$	RCX oscillator frequency	default setting,	-	32	-	Khz
$\Delta f_{RC(RCX)}$	RCX oscillator frequency drift	-	-500	-	500	ppm

7.6 Digital Input/Output Characteristics

Table 7-6 Digital Input/Output Characteristics

Symbol	Paramter	Conditions	Min	Type	Max	Unit
RSTN pin						
V_{IH}	HIGH level input voltage	2.2≤VDD<3.6	0.4xVDD	-	VDD	V
V_{IL}	LOW level input voltage	2.2≤VDD<3.6	0	-	0.35xVDD	V

V_{hys}	Hysteresis voltage	2.2≤VDD<3.6	0.05xVDD	-	-	V
Standard I/O pins						
Input characteristics						
V_{IH}	HIGH level input voltage	2.2≤VDD<3.6	0.6xVDD	-	VDD	V
V_{IL}	LOW level input voltage	2.2≤VDD<3.6	0	-	0.4xVDD	V
V_{hys}	Hysteresis voltage	2.2≤VDD<3.6	0.05xVDD	-	-	V
Output characteristics						
V_O	Output voltage	Output active	0	-	VDD	V
V_{OH}	HIGH level output voltage	2.2≤VDD<3.6	VDD	-	-	V
V_{OL}	LOW level output voltage	2.2≤VDD<3.6	-	-	0	V
I_{OH}	HIGH level output current	VDD=3.3	-	20	-	mA
I_{OL}	LOW level output current	VDD=3.3	-	35	-	mA
R_{PU}	Pull up resistor	2.2≤VDD<3.6	-	50	-	kΩ

7.7 AC Electrical Characteristics

Table 7-7 RF

Symbol	Condition	Min	Typ	Max	Unit
General frequency					
Fop	Operating frequency	2400	-	2483	MHz
PLLres	PLL Programming resolution	-	1	-	MHz
Fxtal	Crystal frequency	-	16	-	MHz
DR	Data rate	-	1	-	Mbps
Transmitter					
PRF	Output power	2	8	13	dBm
PRFC	Output Power Range	-16	-	13	dBm
PBW	20dB Bandwidth for Modulated Carrier at 1Mbps	950	-	1100	MHz
Spur2M	In-band 2M Spurious Emission	-	-	-26	dBm
Spur≥3M	In-band 3M or greater Spurious Emission	-	-	-36	dBm
MDR	Maximum drift rate	-	-	13	KHz/50us
FD	Frequency Deviation	225	-	275	KHz
Receiver					
RXmax	Maximum received signal at <0.1% BER	-	0	-	dBm
RXSENS	Sensitivity (0.1%BER) @1Mbps	-	-90	-	dBm
C/ICO	C/I Co-channel interference	-	11	-	dBc
C/I1M	Adjacent 1MHz interference	-	-2	-	dBc
C/I2M	Adjacent 2MHz interference	-	-22	-	dBc
C/I≥3M	Adjacent ≥ 3MHz interference	-	-38	-	dBc
C/Iimage	Image frequency interference	-	-12	-	dBc
C/Iimage±1M	Adjacent（1MHz）interference to in-band image frequency	-	-35	-	dBc
P_IMD	Intermodulation interference	-	-45	-	dBm
P_Blocking	Out-of-band Blocking interference	-30	-	-	dBm

Table 7-8 DPLL

Symbol	Parameter	Min	Typ	Max	Unit	Notes
VDD2	Power Supply	2.2	-	3.6	V	-
T_A	Temperature	-40	-	85	℃	-
F_{in}	Input Clock frequency	-	12	-	MHz	-
		-	16	-	MHz	-
		-	24	-	MHz	-
F_{DPLL}	Clock frequency	-	26	-	MHz	-

Table 7-9 ADC

Symbol	Parameter	Min	Typ	Max	Unit	Notes
-	Resolution	-	12	-	Bit	-
	Significant Bits		10		Bit	
VDD2	Power Supply	2.5(for VTOP=2.4V) 2.2(for VTOP=1.4V)	-	3.6	VDDA	-
I_{TOT}	Operation Current	880	-	1600	uA	-
INL	Integral Nonlinearity Error	-	-	±2	LSB	
PCLK	System Clock	13	-	26	MHz	-
F_S	Sample Rate	-	-	PCLK/30	MHz	-
T_s	Sample Time	2	-	128	PCLK	-
T_h	Compare Time	24	-	96	PCLK	-
TCONV	Data Output cycle	30	50	234	PCLK	-
N	S-H counter	1	2	7	-	-
V_{in}	Analog input voltage	0.4 0.4	- -	2.4 1.4	V	-
C_{in}	Input Capacitance	-	10	-	pF	-
R_{in}	Input resistance	0.66	10	100	KΩ	See Note
V_{ref}	ADC reference voltage	-	VBG（1.2V）	-	V	-
DATA	ADC Output	000	-	FFF	HEX	-
SFDR	Spurious Free Dynamic range	-	64	-	dB	-

Note:

Sample time formula:

Ts= (EXTSMPT+1) *(ADCDIV +1)*T0

Continuous mode period formula:

T=(EXTSMPT+14) *(ADCDIV +1)* T0+2T0

T0: System clock, the maximum to 1/26M, or chosen as 1/13M.

Maximum resistance formula:

$$Rin < \frac{Ts}{Csample\ x\ ln(2^{N+1})} - R_{adc}$$

Ts: Sample Time, see the Table 7-9 for specific ranges. It is required to be stable within the DAC establishment period.

Csample：Sample capacitance =10pF

N：ADC bit defaults as 12. If the accuracy requirement is not high, it can be taken as 11,10,9,8, which can reduce the requirement of input impedance.

Radc：Sample switch resistance, 100Ω~300Ω

Table 7-10 Rin Maximum under Different Conditions

ADC significant bit	PCLK(Mhz)	Ts(cycles)	Ts(us)	Rinmax(KΩ)
12	26	2	0.08	0.753
12	26	8	0.31	3.314
12	26	32	1.23	13.558
12	26	64	2.46	27.217
12	26	128	4.92	54.534
12	13	2	0.15	1.607
12	13	8	0.62	6.729
12	13	32	2.46	27.217
12	13	64	4.92	54.534
12	13	128	9.85	109.169
9	13	2	0.15	2.119
9	13	8	0.62	8.778
9	13	32	2.46	35.412
9	13	64	4.92	70.924
9	13	128	9.85	141.949
9	13	2	0.08	1.009
9	13	8	0.31	4.339
9	13	32	1.23	17.656
9	13	64	2.46	35.412
9	13	128	4.92	70.924

Table 7-11 LVR

Symbol	Parameter	Min	Typ	Max	Unit	Notes
VDD2	Power Supply	2.2	3	3.6	V	-
V_{LVR}	Threshold Voltage	1.6	1.7	1.8	V	-

Table 7-12 BOD

Symbol	Parameter	Vout(V) 1→0	Vout(V) 0→1	Test Conditions	Notes
V_{BOD}	Brown-Out Detector	1.93	2.06	BODEN=1 BODVL<1:0>=00	-
		2.20	2.34	BODEN=1 BODVL<1:0>=01	-
		2.55	2.72	BODEN=1 BODVL<1:0>=10	-
		2.82	2.87	BODEN=1 BODVL<1:0>=11	-

8 Application Reference Design

8.1 QFN32 Application Reference Circuit

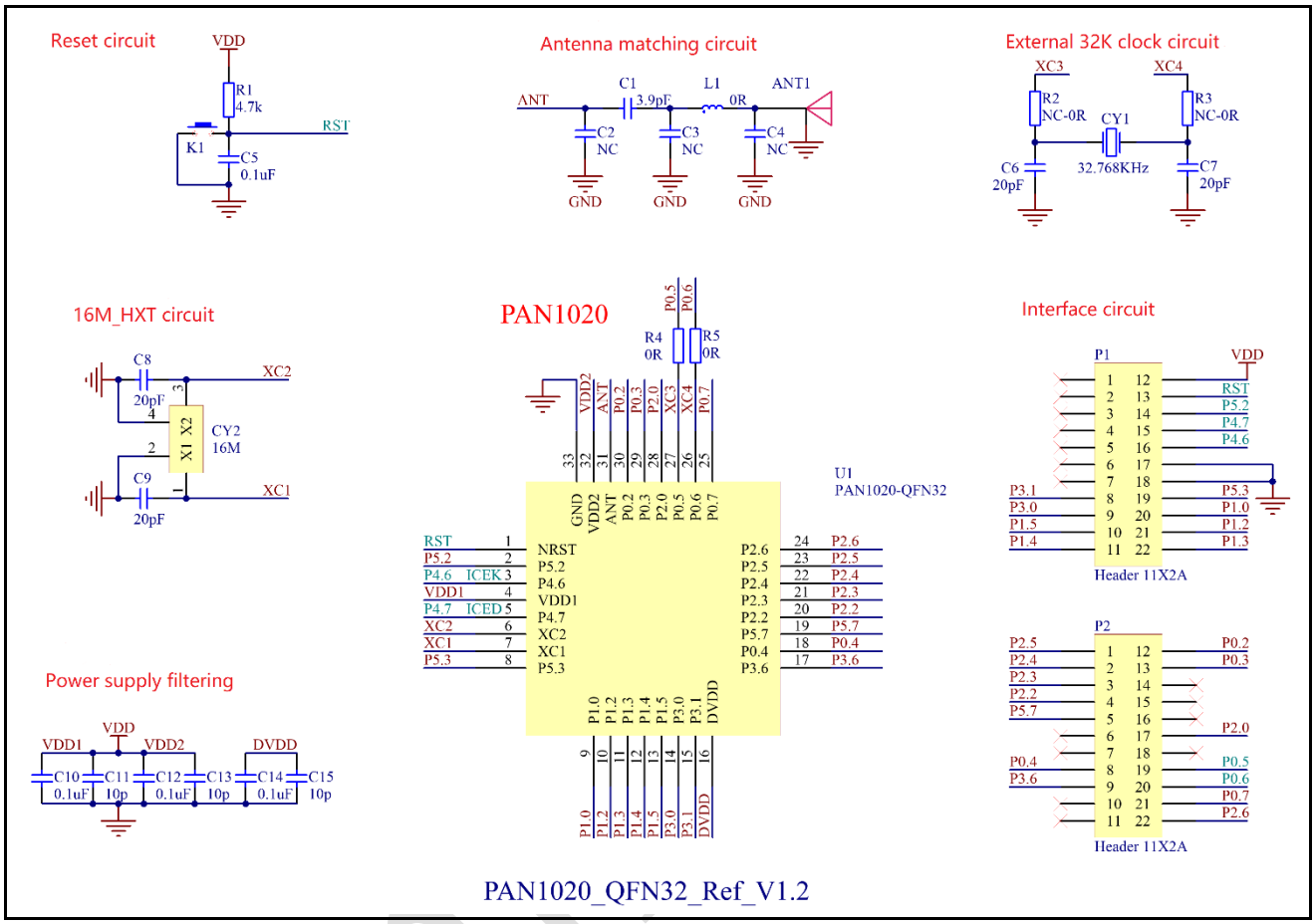


Figure 8-1 Application Reference Circuit for QFN32

8.2 QFN48 Application Reference Circuit

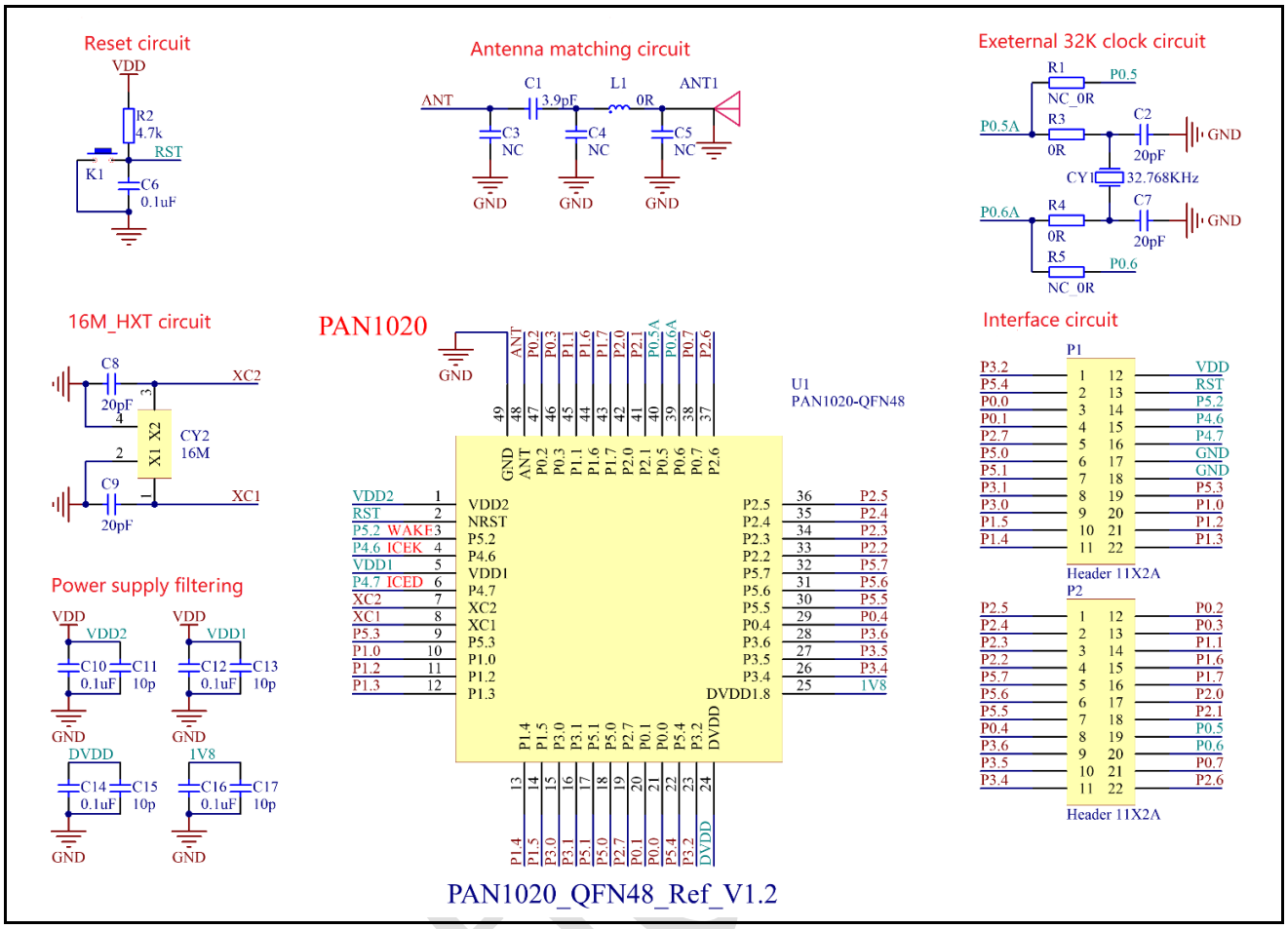


Figure 8-2 Application Reference Circuit for QFN48

Note: P1.1 of PAN1020DY has no actual function, P1.1 of PAN1020BY has actual function.

8.3 QSOP24 Application Reference Circuit

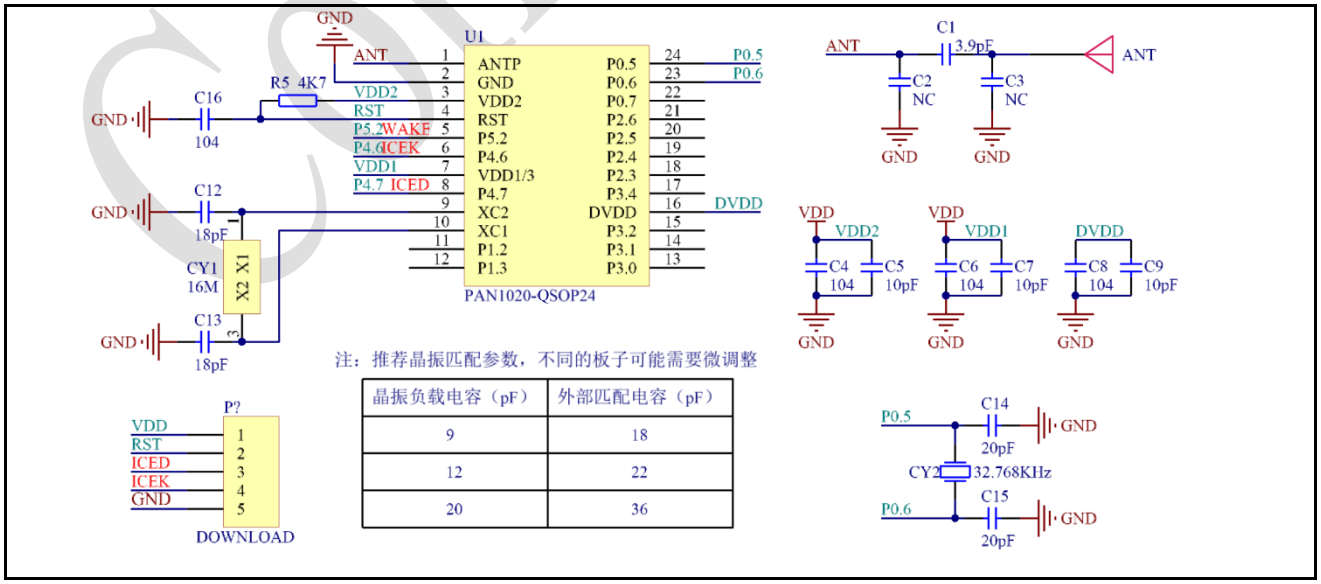


Figure 8-3 Application Reference Circuit for QSOP24

9 Pakage Dimensions

9.1 QFN32 Package Dimensions

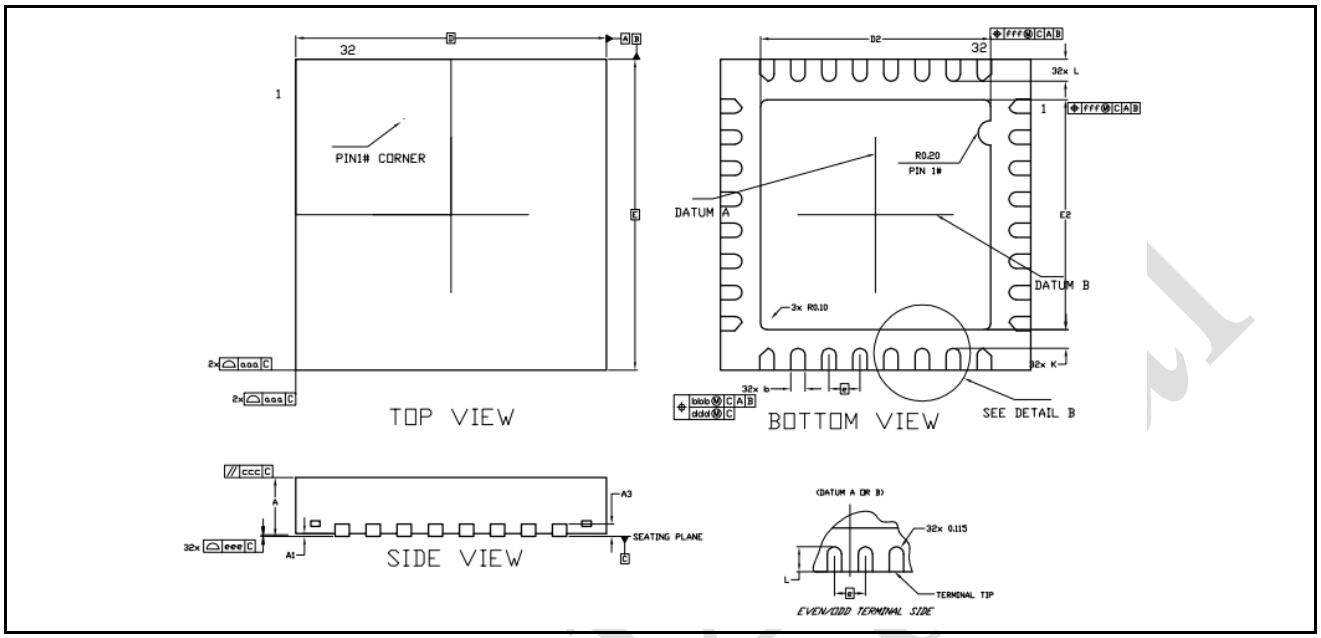


Figure 9-1 QFN32 Package View

Table 9-1 QFN32 Package Detail Parameters

DIM / SYMBOL	MIN.(mm)	NOM.(mm)	MAX.(mm)
A	0.70	0.75	0.80
	0.85	0.90	0.95
A1	0	0.02	0.05
A3	-	0.20 REF	-
b	0.18	0.23	0.28
D	5.00BSC		
E	5.00BSC		
D2	3.55	3.65	3.75
E2	3.55	3.65	3.75
e	0.50BSC		
L	0.30	0.35	0.40
K	0.20	-	-
aaa	0.15		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

9.2 QFN48 Package Dimensions

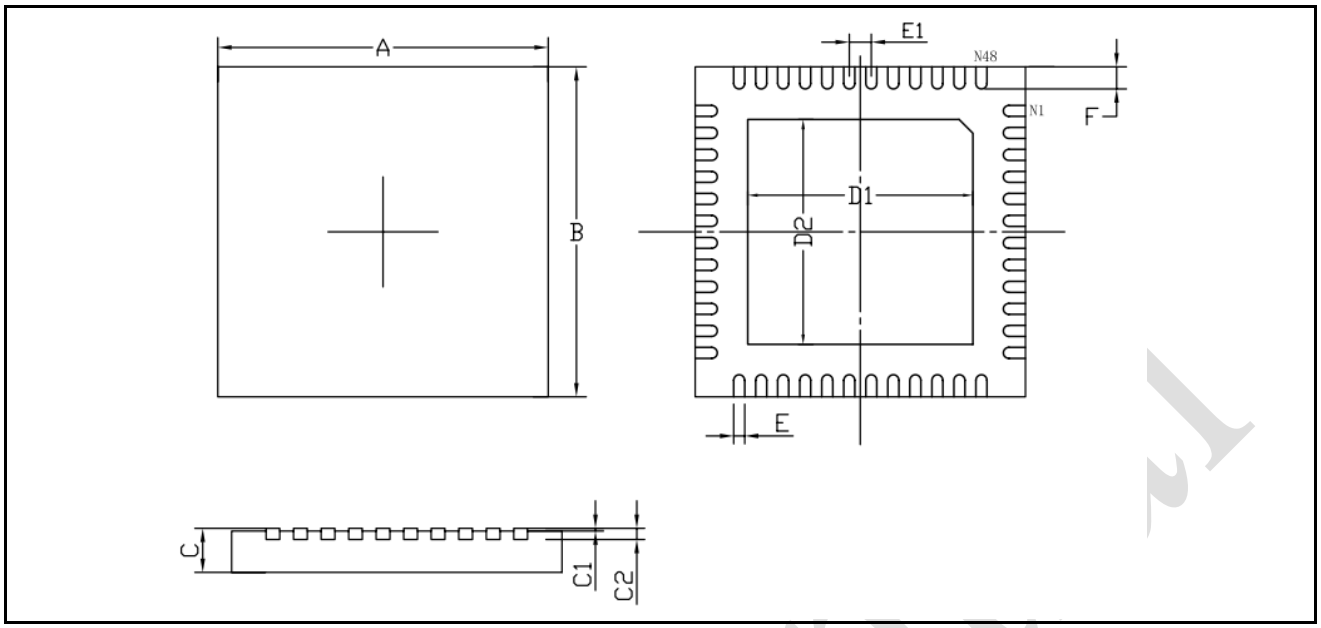


Figure 9-2 QFN48 Package View

Table 9-2 QFN48 Package Detail Parameters

DIM / SYMBOL	MIN.(mm)	MAX.(mm)
A	6.0±0.1	
B	6.0±0.1	
C	0.70	0.80
C1	0~0.050	
C2	0.203TYP	
D1	4.05TYP	
D2	4.05TYP	
E	0.200TYP	
E1	0.400TYP	
F	0.400TYP	

9.3 QSOP24 Package Dimensions

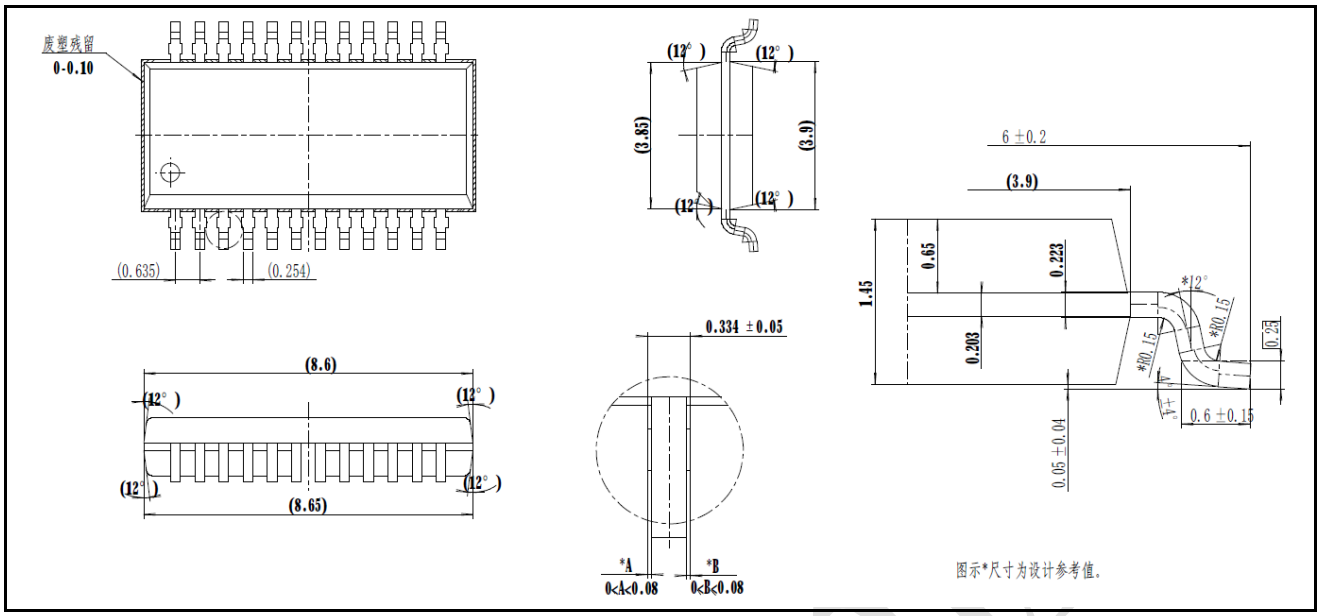


Figure 9-3 QSOP24 Package Views

10 Precautions

1) This product is a CMOS device and should be protected against static electricity during storage, transportation and use.
2) Grounding when device is in use.
3) Reflow temperature can not exceed 260℃.

11 Storage Conditions

1) Products should be stored in sealed packages: when the temperature is less than 30 degrees and the humidity is less than 90%, it can last for 12 months.

2) After the package is opened, the components will be used in the reflow process or other high-temperature processes. The following conditions must be met:

a) Completed within 72 hours and the factory environment is less than 30℃≤ 60% RH.
b) Stored in 10% RH environment.
c) Exhaust at 125℃ for 24 hours to remove internal water vapor before used.