ISD Cortex™-M0 ChipCorder

ISD93xx Series

Design Guide

The information described in this document is the exclusive intellectual property of
Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of ISD ChipCorder microcontroller
based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

June 1, 2014
Revision V1.0

Table of Contents-

1   GENERAL DESCRIPTION

The ISD93xx series are system-on-chip products optimized for low power, audio record and playback with an embedded ARM® Cortex™-M0 32-bit microcontroller core.

The series embeds a Cortex™-M0 core running up to 98 MHz with 145K-byte of non-volatile flash memory and 16K-byte of embedded SRAM. It also comes equipped with a variety of peripheral devices, such as Timers, Watchdog Timer (WDT), Real-time Clock (RTC), Peripheral Direct Memory Access (PDMA), a variety of serial interfaces (UART, SPI/SSP, I^{2}C, I^{2}S), PWM modulators, GPIO, Analog Comparator, Low Voltage Detector and Brown-out detector.

The ISD93xx series comes equipped with a rich set of power saving modes including a Deep Power Down (DPD) mode drawing less than 1μA. A micro-power 16KHz oscillator can periodically wake up the device from deep power down to check for other events. A Standby Power Down (SPD) mode can maintain a real time clock function at less than 10 μA.

For audio functionality the ISD93xx series includes a Sigma-Delta ADC with 90dB SNR performance coupled with a Programmable Gain Amplifier (PGA) capable of a maximum gain of 61dB to enable direct connection of a microphone. Audio output is provided by a Differential Class D amplifier (DPWM) that can deliver 1W of power to an 8Ω speaker.

The ISD93xx series provides sixteen analog enabled general purpose IO pins (GPIO). These pins can be configured to connect to an analog comparator, can be configured as analog current sources or can be routed to the SDADC for analog conversion. They can also be used as a relaxation oscillator to perform capacitive sensing.

2 FEATURES

- Core
  - ARM® Cortex™-M0 core running up to 98MHz.
  - One 24-bit System tick timer for operating system support.
  - Supports a variety of low power sleep and power down modes.
  - Single-cycle 32-bit hardware multiplier.
  - NVIC (Nested Vector Interrupt Controller) for 32 interrupt inputs, each with 4-levels of priority.
  - Serial Wire Debug (SWD) support with 2 watch points/4 breakpoints.

- Power Management
  - Wide operating voltage range from 2.4V to 5.5V.
  - Power management Unit (PMU) providing four levels of power control.
  - Deep Power Down (DPD) mode with sub micro-amp leakage (<1µA).
  - Wakeup from Deep Power Down via dedicated WAKEUP pin or timed operation from internal low power16kHz oscillator.
  - Standby mode with limited RAM retention and RTC operation (<10µA).
  - Wakeup from Standby can be from any GPIO interrupt, RTC or BOD.
  - Sleep mode with minimal dynamic power consumption.
  - 3V LDO for operation of external 3V devices such as serial flash.

- Flash EPROM Memory
  - 145K bytes Flash EPROM for program code and data storage.
  - Mini-cache to maintain near zero-wait state memory access.
  - 4KB of flash can be configured as boot sector for ISP loader.
  - Support In-system program(ISP) and In-circuit program(ICP) application code update
  - 1K byte page erase for flash
  - Configurable boundary to delineate code and data flash.
  - Support 2 wire In-circuit Programming (ICP) update from SWD ICE interface

- SRAM Memory
  - 16K bytes embedded SRAM.

- Clock Control
  - One high speed and two low speed oscillators providing flexible selection for different applications. No external components necessary.
  - Built-in trimmable oscillator with range of 16-50MHz. Factory trimmed within 1% to settings of 49.152MHz 36.824MHz and 32.768MHz. User trimmable with in-built frequency measurement block (OSCFM) using reference clock of 32kHz crystal or external reference source.
  - Ultra-low power (<1uA) 16kHz oscillator for watchdog and wakeup from power-down or sleep operation.
  - External 32kHz crystal input for RTC function and low power system operation.

- GPIO
  - Four I/O modes:
    - Quasi bi-direction
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - TTL/Schmitt trigger input selectable.
  - I/O pin can be configured as interrupt source with edge/level setting.
  - Switchable pull-up.

- Audio Analog to Digital converter
  - Sigma Delta ADC with configurable decimation filter and 16 bit output.
  - 90dB Signal-to-Noise (SNR) performance.
  - Programmable gain amplifier with 32 steps from -12 to 35.25dB in 0.75dB steps.
  - Boost gain stage of 26dB, giving maximum total gain of 61dB.

- Input selectable from dedicated MIC pins or analog enabled GPIO.
- Programmable bi-quad filter to support multiple sample rates from 8-32kHz.
- DMA support for minimal CPU intervention.

- Differential Audio PWM Output (DPWM)
  - Direct connection of speaker
  - 1W drive capability into 8Ω load.
  - High efficiency 88%
  - Configurable up-sampling to support sample rates from 8-32kHz.
  - DMA support for minimal CPU intervention.

- Timers
  - Two timers with 8-bit pre-scaler and 24-bit resolution.
  - Counter auto reload.

- Watch Dog Timer
  - Default ON/OFF by configuration setting
  - Multiple clock sources
  - 8 selectable time out period from micro seconds to seconds (depending on clock source)
  - WDT can wake up power down/sleep.
  - Interrupt or reset selectable on watchdog time-out.

- RTC
  - Real Time Clock counter (second, minute, hour) and calendar counter (day, month, year)
  - Alarm registers (second, minute, hour, day, month, year)
  - Selectable 12-hour or 24-hour mode
  - Automatic leap year recognition
  - Time tick and alarm interrupts.
  - Device wake up function.
  - Supports software compensation of crystal frequency by compensation register (FCR)

- PWM/Capture
  - Six 16-bit PWM generators provide six single ended PWM outputs or three complementary paired PWM outputs.
  - The PWM generator equipped with a clock source selector, a clock divider, an 8-bit pre-scaler and Dead-Zone generator for complementary paired PWM.
  - PWM interrupt synchronous to PWM period.
  - 16-bit digital Capture timers (shared with PWM timers) provide rising/falling capture inputs.
  - Support Capture interrupt

- UART
  - UART ports with flow control (TX, RX, CTS and RTS)
  - 8-byte FIFO.
  - Support IrDA (SIR) and LIN function
  - Programmable baud-rate generator up to 1/16 of system clock.

- SPI
  - Master up to 20 Mbps /Slave up to 10 Mbps.
  - Support MICROWIRE/SPI master/slave mode (SSP)
  - Full duplex synchronous serial data transfer
  - Variable length of transfer data from 1 to 4 bytes
  - MSB or LSB first data transfer
  - 2 slave/device select lines when used in master mode.
  - Hardware CRC calculation module available for CRC calculation of data stream.
  - DMA support.
  - Quad/Dual SPI support.

- I2C

- Master/Slave up to 1Mbit/s
- Bidirectional data transfer between masters and slaves
- Multi-master bus (no central master).
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus.
- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer.
- Programmable clock allowing versatile rate control.
- I2C-bus controller supports multiple address recognition.

- I^2S
  - Interface with external audio CODEC.
  - Operate as either master or slave.
  - Capable of handling 8, 16, 24 and 32 bit word sizes
  - Mono and stereo audio data supported
  - I^2S and MSB justified data format supported
  - Two 8 word FIFO data buffers are provided, one for transmit and one for receive
  - Generates interrupt requests when buffer levels cross a programmable boundary
  - Supports DMA requests, for transmit and receive

- Brown-out detector
  - With 16levels
  - Supports time-multiplex operation to minimize power consumption.
  - Supports Brownout Interrupt and Reset option

- Built in Low Dropout Voltage Regulator (LDO)
  - Capable of delivering 30mA load current.
  - Configurable for output voltage of 1.8V, 2.4V, 3.0V and 3.3V
  - Eight GPIO (GPIOA<7:0>) operate from LDO voltage domain allowing direct interface to, for example, 3V SPI Flash.
  - Can be bypassed and voltage domain supplied directly from system power.

- Additional Features
  - Over temperature alarm. Can generate interrupt if device exceeds safe operating temperature.
  - Temperature proportional voltage source which can be routed to ADC for temperature measurements.
  - Digital Microphone interface.

- Operating Temperature: -40C~85C

- Package:
  - All Green package (RoHS)
    - LQFP 64-pin

3 PIN CONFIGURATION

3.1 ISD93xx LQFP 64 pin
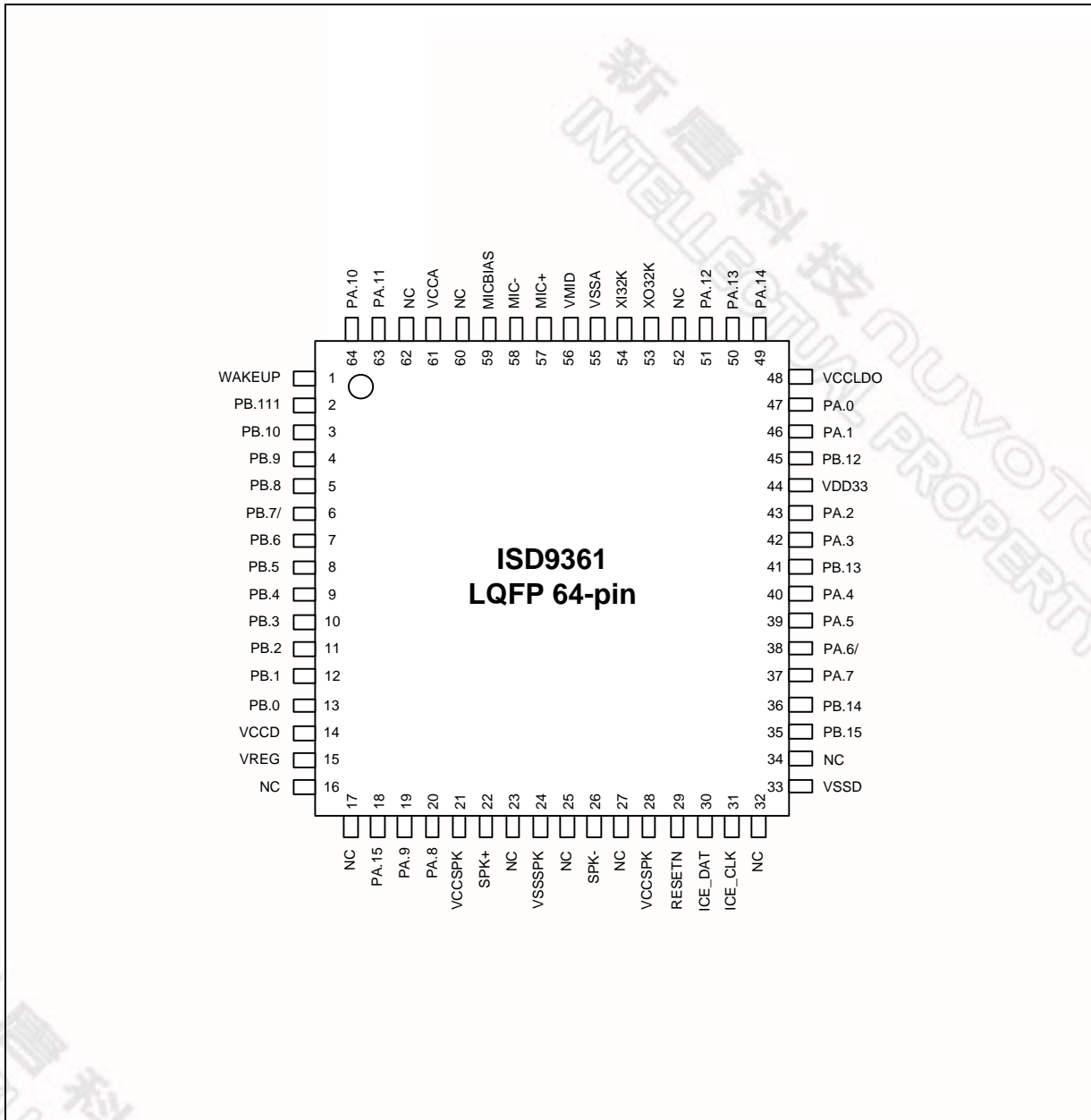
3.2 Pin Description

The ISD93xx is a low pin count device where many pins are configurable to alternative functions. All General Purpose Input/Output (GPIO) pins can be configured to alternate functions as described in the table below and also in Table 5-7and Table 5-8.

Pin No. LQFP 64	Pin Name	Pin Type	Alt CFG	Description
1	WAKEUP	I		Pull low to wake part from deep power down
2	PB.11	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 11
	I2S_SDO	O	1	I2S Serial Data out
	CMP11	AIO	2	Configure as relaxation oscillator for CapSense
3	PB.10	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 10
	CMP10	AIO	2	Configure as relaxation oscillator for CapSense
4	PB.9	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 9
	I2S_BCLK	O	1	I2S Bit Clock (master mode only)
	CMP9	AIO	2	Configure as relaxation oscillator for CapSense
5	PB.8	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 8
	I2S_FS	O	1	I2S Frame Sync (master mode only)
	CMP8	AIO	2	Configure as relaxation oscillator for CapSense
6	PB.7	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 7
	I2S_SDO	O	1	Serial Data Output for I2S interface
	CMP7	AIO	2	Configure as relaxation oscillator for CapSense
7	PB.6	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 6
	I2S_SDI	I	1	Serial Data Input for I2S interface
	CMP6	AIO	2	Configure as relaxation oscillator for CapSense
	SPI_MOSI1	O	3	Master Out, Slave In channel 1 for SPI interface
8	PB.5	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 5
	PWM1B	O	1	PWM channel 1 complementary output pin
	CMP5	AIO	2	Configure as relaxation oscillator for CapSense
	SPI_MISO1	I	3	Master In, Slave Out channel 1 for SPI interface
9	PB.4	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 4
	PWM0B	O	1	PWM channel 0 complementary output pin
	CMP4	AIO	2	Configure as relaxation oscillator for CapSense
	SPI_MOSI0	O	3	Master Out, Slave In channel 0 for SPI interface
10	PB.3	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 3

Pin No. LQFP 64	Pin Name	Pin Type	Alt CFG	Description
	I2C_SDA	I/O	1	Serial Data for I2C interface
	CMP3	AIO	2	Configure as relaxation oscillator for CapSense
	SPI_MISO0	I	3	Master In, Slave Out channel 0 for SPI interface
11	PB.2	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 2
	I2C_SCL	I/O	1	Serial Clock for I2C interface
	CMP2	AIO	2	Configure as relaxation oscillator for CapSense
	SPI_SCLK	I/O	3	Serial Clock for SPI interface
12	PB.1	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 1. Triggers external interrupt 1 (EINT1/IRQ3)
	MCLK	O	1	Master clock output for synchronizing external device
	CMP1	AIO	2	Configure as relaxation oscillator for CapSense
	SPI_SSB1	O	3	Slave Select Bar 1 for SPI interface
13	PB.0	A/I/O	0	General purpose input/output pin, analog capable; Port B, bit 0. Triggers external interrupt 0 (EINT0/IRQ2)
	SPI_SSB1	O	3	Slave Select Bar 1 for SPI interface
	CMP0	AIO	2	Configure as relaxation oscillator for CapSense
	SPI_SSB0	I/O	3	Slave Select Bar 0 for SPI interface
14	VCCD	P		Main Digital Supply for Chip. Supplies all IO except analog, Speaker Driver and PA<7:0> and PB<12:13>
15	VREG	P		Logic regulator output decoupling pin. A 1µF capacitor returning to VSSD must be placed on this pin.
16	NC			Should remain unconnected.
17	NC			Should remain unconnected.
18	PA.15	I/O	0	General purpose input/output pin; Port A, bit 15
	TM1	I	1	External input to Timer 1
	SDIN	I	2	Sigma Delta bit stream input for digital MIC mode
	PWM5	O	3	Output of PWM5
19	PA.9	I/O	0	General purpose input/output pin; Port A, bit 9
	UART_RX	I	1	Receive channel of UART
	CMP13	AIO	2	Configure as relaxation oscillator for CapSense
	PWM3	O	3	Output of PWM3
20	PA.8	I/O	0	General purpose input/output pin; Port A, bit 8
	UART_TX	O	1	Transmit channel of UART

Pin No. LQFP 64	Pin Name	Pin Type	Alt CFG	Description
	CMP12	**AIO**	2	Configure as relaxation oscillator for CapSense
	PWM2	**O**	3	Output of PWM2
21	VCCSPK	**P**		Power Supply for PWM Speaker Driver
22	SPK+	**O**		Positive Speaker Driver Output
23	NC			Should remain unconnected.
24	VSSSPK	**P**		Ground for PWM Speaker Driver
25	NC			Should remain unconnected.
26	SPK-	**O**		Negative Speaker Driver Output
27	NC			Should remain unconnected.
28	VCCSPK	**P**		Power Supply for PWM Speaker Driver
29	RESETN	**I**		External reset input. Pull this pin low to reset device to initial state. Has internal weak pull-up.
30	ICE_DAT	**I/O**		Serial Wire Debug port data pin. Has internal weak pull-up.
31	ICE_CLK	**I**		Serial Wire Debug port clock pin. Has internal weak pull-up.
32	NC			Should remain unconnected.
33	VSSD	**P**		Digital Ground.
34	NC			Should remain unconnected.
35	PB.15	**I/O**	0	General purpose input/output pin; Port B, bit 15
	PWM3	**O**	1	Output of PWM3
	PWM2B	**O**	3	Complementary output of PWM2
36	PB.14	**I/O**	0	General purpose input/output pin; Port B, bit 14
	PWM2	**O**	1	Output of PWM2
	UART_TX	**O**	2	UART transmit
	PWM3B	**O**	3	Complementary output of PWM3
37	PA.7	**I/O**	0	General purpose input/output pin; Port A, bit 7
	I2S_SDO	**O**	1	Serial Data Out for I2S interface
38	PA.6	**I/O**	0	General purpose input/output pin; Port A, bit 6
	I2S_SDI	**I**	1	Serial Data In for I2S interface
	UART_TX	**O**	2	UART transmit
39	PA.5	**I/O**	0	General purpose input/output pin; Port A, bit 5
	I2S_BCLK	**I/O**	1	Bit Clock for I2S interface

Pin No.	Pin Name	Pin Type	Alt CFG	Description
LQFP 64				
	SPI_SSB1	**I/O**	3	SPI Slave Select 1
40	PA.4	**I/O**	0	General purpose input/output pin; Port A, bit 4
	I2S_FS	**I/O**	1	Frame Sync Clock for I2S interface
41	PB.13	**I/O**	0	General purpose input/output pin; Port B, bit 13
	SPI_MOSI1	**I/O**	1	Master Out, Slave In channel 1 for SPI interface
	PWM5	**O**	2	Output of PWM5
	PWM4B	**O**	3	Complementary output of PWM4
42	PA.3	**I/O**	0	General purpose input/output pin; Port A, bit 3
	SPI_MISO0	**I**	1	Master In, Slave Out channel 0 for SPI interface
	I2C_SDA	**I/O**	2	Serial Data for I2C interface
43	PA.2	**I/O**	0	General purpose input/output pin; Port A, bit 2
	SPI_SSB0	**I/O**	1	Slave Select Bar 0 for SPI interface
44	VDD33	**P**		LDO Regulator Output. If used, a 1μF capacitor must be placed to ground. If not used then tie to VCCD.
45	PB.12	**I/O**	0	General purpose input/output pin; Port B, bit 12
	SPI_MISO1	**I/O**	1	Master In, Slave Out channel 1 for SPI interface
	PWM5	**O**	2	Output of PWM4
	PWM4B	**O**	3	Complementary output of PWM5
46	PA.1	**I/O**	0	General purpose input/output pin; Port A, bit 1
	SPI_SCLK	**I/O**	1	Serial Clock for SPI interface
	I2C_SCL	**I/O**	2	Serial Clock for I2C interface
47	PA.0	**I/O**	0	General purpose input/output pin; Port A, bit 2
	SPI_MOSI0	**O**	1	Master Out, Slave In channel 0 for SPI interface
	MCLK	**O**	2	Master clock output.
48	VCCLDO	**P**		Power Supply for LDO, should be connected to VCCD
49	PA.14	**I/O**	0	General purpose input/output pin; Port A, bit 14
	TM0	**I**	1	External input to Timer 0
	SDCLKn	**O**	2	Inverse Clock output for digital microphone mode.
	PWM4	**O**		PWM4 Output
50	PA.13	**I/O**	0	General purpose input/output pin; Port A, bit 13
	PWM1	**O**	1	PWM1 Output.

Pin No. LQFP 64	Pin Name	Pin Type	Alt CFG	Description
	SPKM	O	2	Equivalent to SPK-.
	I2S_BCLK	I/O	3	Bit Clock for I2S interface
51	PA.12	I/O	0	General purpose input/output pin; Port A, bit 12
	PWM0	O	1	PWM0 Output.
	SPKP	O	2	Equivalent to SPK+
	I2S_FS	I/O	3	Frame Sync Clock for I2S interface
52	NC			Should remain unconnected.
53	XO32K	O		32.768kHz Crystal Oscillator Output
54	XI32K	I		32.768kHz Crystal Oscillator Input. Max Voltage 1.8V
55	VSSA	AP		Ground for analog circuitry.
56	VMID	O		Mid rail reference. Connect 4.7µF to VSSA.
57	MIC+	AI		Positive microphone input.
58	MIC-	AI		Negative microphone input.
59	MICBIAS	AO		Microphone bias output.
60	NC			Should remain unconnected.
61	VCCA	AP		Analog power supply.
62	NC			Should remain unconnected.
63	PA.11	A/I/O	0	General purpose input/output pin, analog capable; Port A, bit 11
	I2C_SCL	I/O	1	Serial Clock for I2C interface
	CMP15	AIO	2	Configure as relaxation oscillator for CapSense
	UART_CTSn	I	3	UART Clear to Send Input.
64	PA.10	A/I/O	0	General purpose input/output pin, analog capable; Port A, bit 10
	I2C_SDA	I/O	1	Serial Data for I2C interface
	CMP14	I	2	Configure as relaxation oscillator for CapSense
	UART_RTSn	O	3	UART Request to Send Output.

Note:
1.   Pin Type I=Digital Input, O=Digital Output; AI=Analog Input; P=Power Pin; AP=Analog Power

4 BLOCK DIAGRAM
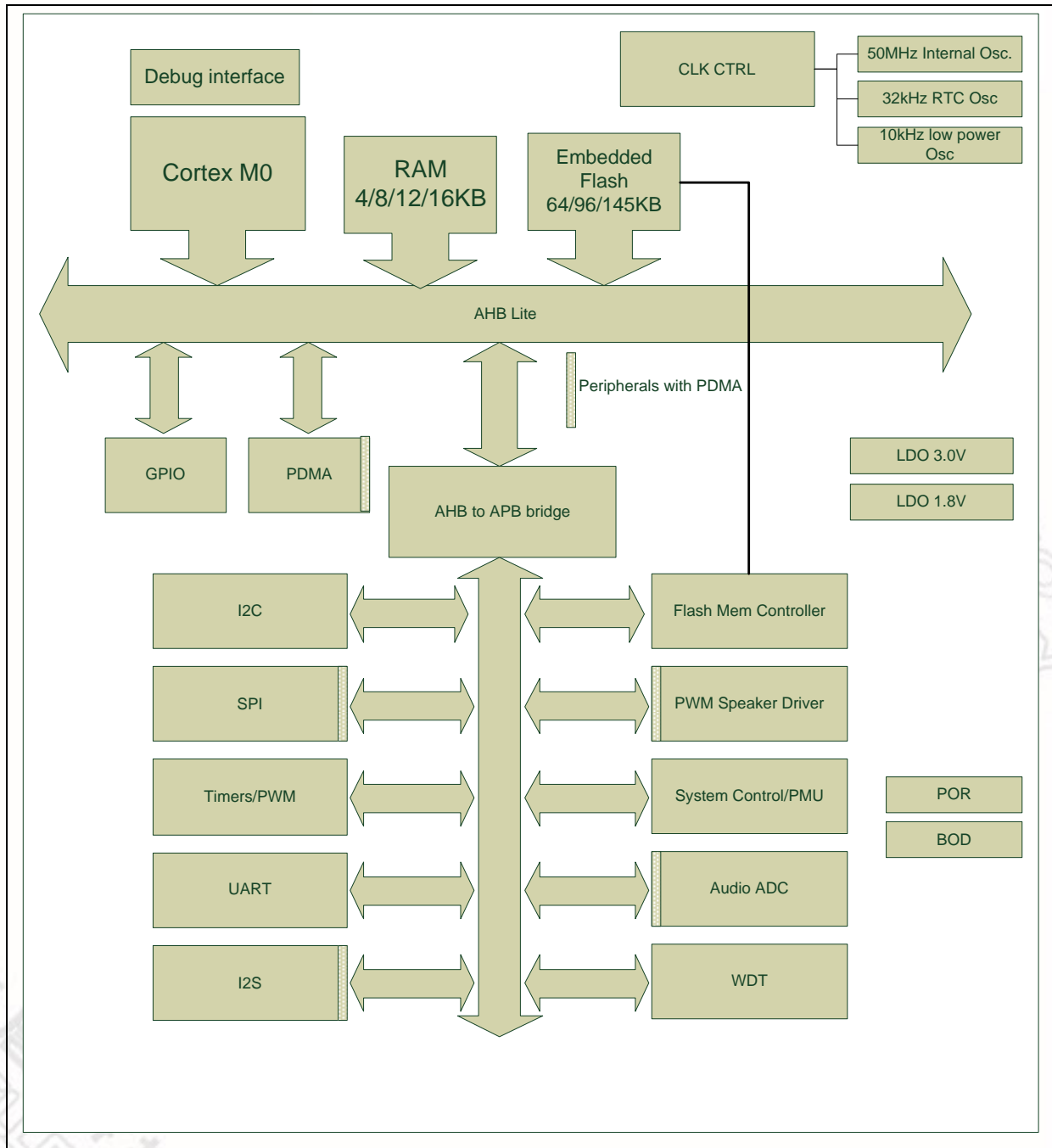
Figure 4-1ISD93xxBlock Diagram

5 FUNCTIONAL DESCRIPTION

5.1 ARM® Cortex™-M0 core

The Cortex™-M0 processor is a multistage, 32-bit RISC processor. It has an AMBA AHB-Lite interface and includes an NVIC component. It also has hardware debug functionality. The processor can execute Thumb code and is compatible with other Cortex-M profile processor.

Figure 5-1 shows the functional blocks of processor.
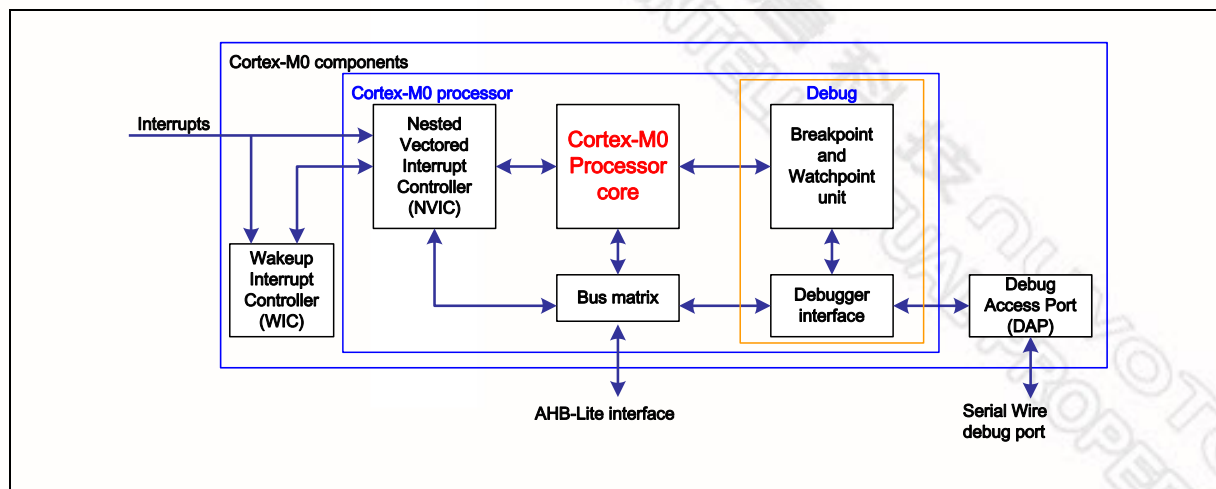
Figure 5-1 Functional Block Diagram

The implemented device provides:

- A low gate count processor that features:
  - The ARMv6-M Thumb® instruction set.
  - Thumb-2 technology.
  - ARMv6-M compliant 24-bit SysTick timer.
  - A 32-bit hardware multiplier.
  - The system interface supports little-endian data accesses.
  - The ability to have deterministic, fixed-latency, interrupt handling.
  - Load/store-multiples that can be abandoned and restarted to facilitate rapid interrupt handling.
  - C Application Binary Interface compliant exception model.
    This is the ARMv6-M, C Application Binary Interface (C-ABI) compliant exception model that enables the use of pure C functions as interrupt handlers.
  - Low power sleep-mode entry using Wait For Interrupt (WFI), Wait For Event (WFE) instructions, or the return from interrupt sleep-on-exit feature.

- NVIC that features:
  - 32 external interrupt inputs, each with four levels of priority.
  - Dedicated non-Maskable Interrupt (NMI) input.
  - Support for both level-sensitive and pulse-sensitive interrupt lines
  - Wake-up Interrupt Controller (WIC), providing ultra-low power sleep mode support.

- Debug support
  - Four hardware breakpoints.
  - Two watch points.
  - Program Counter Sampling Register (PCSR) for non-intrusive code profiling.

‒ Single step and vector catch capabilities.

- Bus interfaces:
  - ‒ Single 32-bit AMBA-3 AHB-Lite system interface that provides simple integration to all system peripherals and memory.
  - ‒ Single 32-bit slave port that supports the DAP (Debug Access Port).

5.2 System Manager

5.2.1 Overview

The following functions are included in system manager section

- System Memory Map

- System Timer (SysTick)

- Nested Vectored Interrupt Controller (NVIC)

- System management registers for product ID

- System management registers for chip and module functional reset and multi-function pin control

- Brown-Out and chip miscellaneous Control Register

- Combined peripheral interrupt source identify

5.2.2 System Reset

The system reset includes one of the list below event occurs. For these reset event flags can be read by **RSTSRC** register.

- The Power-On Reset

- The low level on the RESETN pin

- Watchdog Time Out Reset

- Low Voltage Reset

- Cortex-M0 MCU Reset

- PMU Reset – for details of wakeup events, also examine **PWRCON** register.

- SWD Debug interface.

A power-on reset (POR) will occur if the main external supply rail ramps from 0V or the voltage of the main supply drops below reset threshold. A low voltage reset monitors the regulated core logic (1.8V) supply and will assert if the voltage on this rail drops below reliable logic threshold.

5.2.3 System Power Distribution

The ISD93xx implements several power domains:

• Analog power from VCCA and VSSA provides the power for analog module operation.

• Digital power from VCCD and VSSD supplies the power to the IO ring and the internal regulator which provides 1.8V power for digital operation.

• VCCLDO supplies the LDO regulator whose output is available on pin VDD33. This supply powers the IO ring for GPIOA<7:0>.

• An internal Standby reference (SB REG) generates a 1.8V rail to part of the logic including the IO ring, Standby RAM and RTC during standby mode for low power operation.

The outputs of internal voltage regulators: VREG and VDD33, require external decoupling capacitors which should be located close to the corresponding pin. The following diagram shows the power distribution of this device.
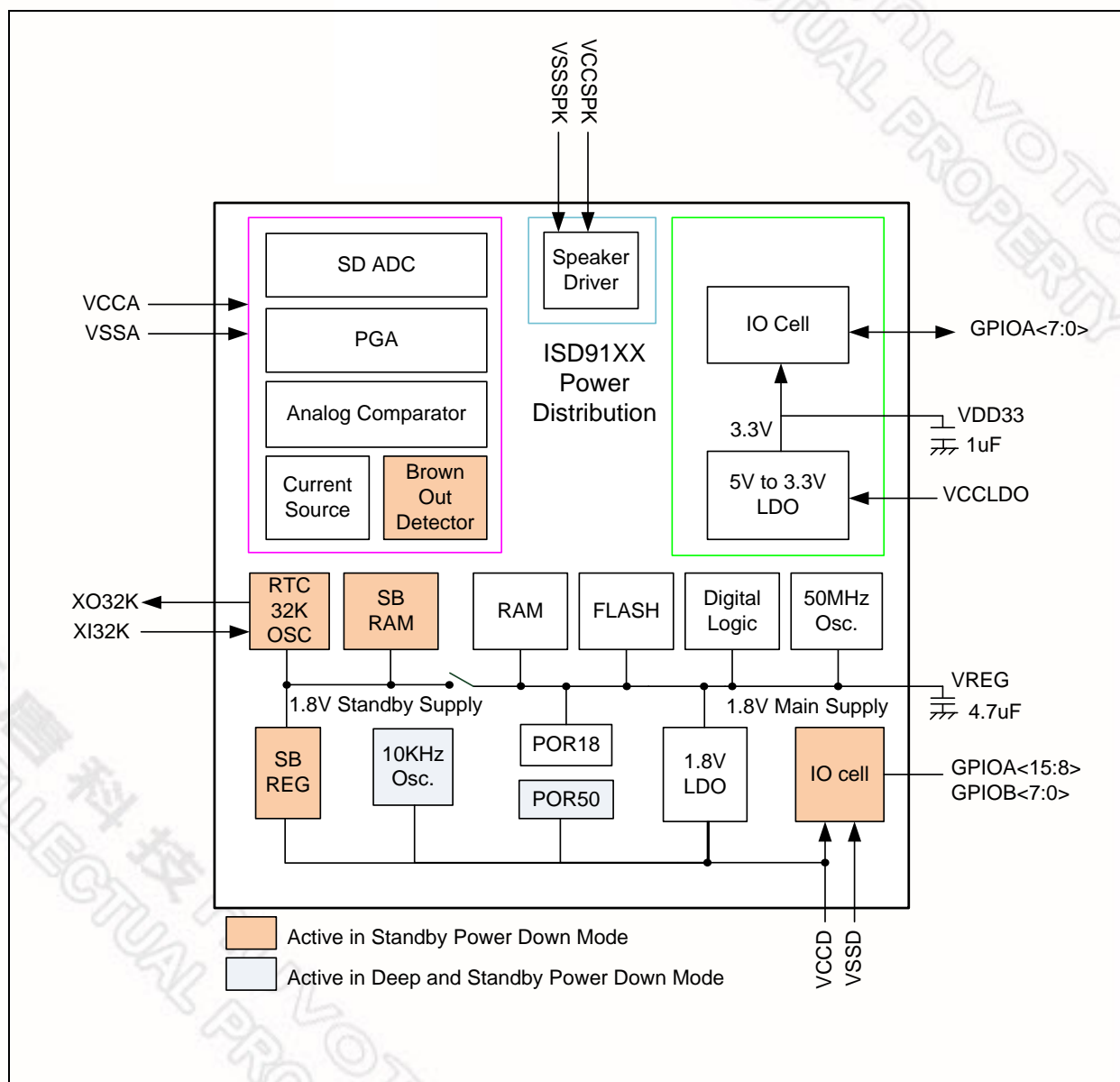
Figure 5-2ISD93xx Series Power Distribution Diagram

5.2.4   System Memory Map

The ISD93xx provides 4G-byte address space. The memory locations assigned to each on-chip module is shown in Table. The detailed register definition, memory space, and programming detailed will be described in the following sections for each on-chip module. The ISD93xx series supports little-endian data format.

Table 5-1Address Space Assignments for On-Chip Modules

Address Space	Token	Modules	Reference
Flash & SRAM Memory Space			
0x0000_0000 – 0x0002_33FF	FLASH_BA	FLASH Memory Space (141KB)	
0x0000_0000 – 0x0002_43FF	FLASH_BA	FLASH Memory Space (145KB)	
0x2000_0000 – 0x2000_2FFF	SRAM_BA	SRAM Memory Space (16KB)	5.3.4.5
AHB Modules Space (0x5000_0000 – 0x501F_FFFF)			
0x5000_0000 – 0x5000_01FF	GCR_BA	System Global Control Registers	5.2.5.1
0x5000_0200 – 0x5000_02FF	SYSCLK_BA	Clock Control Registers	5.3.4.5
0x5000_0300 – 0x5000_03FF	INT_BA	Interrupt Source Control Registers	5.2.7.5
0x5000_4000 – 0x5000_7FFF	GPIO_BA	GPIO Control Registers	5.4.3
0x5000_8000 – 0x5000_BFFF	PDMA_BA	SRAM_APB DMA Control Registers	5.14
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash Memory Control Registers	6.3
APB1 Modules Space (0x4000_0000 ~ 0x400F_FFFF)			
0x4000_4000 – 0x4000_7FFF	WDT_BA	Watch-Dog Timer Control Registers	5.11
0x4000_8000 – 0x4000_BFFF	RTC_BA	Real Time Clock (RTC) Control Register	5.8
0x4001_0000 – 0x4001_3FFF	TIMER0_BA	Timer0/Timer1 Control Registers	5.10
0x4002_0000 – 0x4002_3FFF	I2C0_BA	I2C0 Interface Control Registers	5.6
0x4003_0000 – 0x4003_3FFF	SPI0_BA	SPI0 Serial Interface Control Registers	5.9
0x4004_0000 – 0x4004_3FFF	PWMA_BA	PWM0/1 Control Registers	5.7
0x4005_0000 – 0x4005_3FFF	UART0_BA	UART0 Control Registers	5.12
0x4007_0000 – 0x4007_3FFF	DPWM_BA	Differential Audio PWM Speaker Driver	7.2
0x4008_0000 – 0x4008_3FFF	ANA_ BASE	Analog Block Control Registers	7.4

0x4008_4000 – 0x4008_7FFF	BOD_BA	Brown Out Detector Control Registers	5.5.1
0x4009_0000 – 0x4009_7FFF	CRC_BA	CRC Block Control Registers	CRC Control Register
0x400A_0000 - 0x400A_FFFF	I2S_BA	I2S Interface Control registers	5.13
0x400B_0000 - 0x400B_FFFF	BIQ_BA	Biquad Filter Control Registers	7.6
0x400D_0000 – 0x400D_3FFF	ACMP_BA	Analog Comparator Control Registers	7.3
0x400E_0000 – 0x400E_FFFF	ADC0_BA	Analog-Digital-Converter (ADC) Registers	7.1
0x400F_0000 – 0x400F_7FFF	SBRAM_BA	Standby RAM Block Address space	
System Control Space (0xE000_E000 ~ 0xE000_EFFF)			
0xE000_E010 – 0xE000_E0FF	SCS_BA	System Timer Control Registers	5.2.6
0xE000_E100 – 0xE000_ECFF	SCS_BA	External Interrupt Controller Control Registers	5.2.7
0xE000_ED00 – 0xE000_ED8F	SCS_BA	System Control Registers	5.2.8

5.2.5 GPIO, Reset, System Manager Registers

5.2.5.1 *System Manager Control Registers*

Register	Offset	R/W	Description	Reset Value	Reference
GCR_BA = 0x5000_0000					
RSTSRC	GCR_BA+0x04	R/W	System Reset Source Register	0x0000_00XX	Table 5-2
IPRSTC1	GCR_BA+0x08	R/W	IP Reset Control Resister1	0x0000_0000	Table 5-3
IPRSTC2	GCR_BA+0x0C	R/W	IP Reset Control Resister2	0x0000_0000	Table 5-4
GPA_INP	GCR_BA+0x30	R/W	GPIOA input type control register	0x0000_0000	Table 5-5
GPB_INP	GCR_BA+0x34	R/W	GPIOB input type control register	0x0000_0000	Table 5-6
GPA_ALT	GCR_BA+0x38	R/W	GPIOA multiple function control register	0x0000_0000	Table 5-7
GPB_ALT	GCR_BA+0x3C	R/W	GPIOB multiple function control register	0x0000_0000	Table 5-8
WAKECR	GCR_BA+0x54	R/W	WAKEUP pad control register	0x0000_0006	Table 5-9
REGLOCK	GCR_BA+0x100	R/W	Register Lock Key address	0x0000_0000	Table 5-10
OSCTRIM	GCR_BA+0x110	R/W	Internal oscillator trim register	0xXXXX_XXXX	Table 5-11
OSC16K	GCR_BA+0x114	R/W	16K Oscillator trim register	0xXXXX_XXXX	

System Reset Source Register (RSTSRC)

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

Register	Offset	R/W	Description	Reset Value
RSTSRC	GCR_BA+04	R/W	System Reset Source Register	0x0000_0xxx

15	14	13	12	11	10	9	8
Reserved	Reserved	Reserved	Reserved	Reserved	RSTS_POI	RSTS_TIM	RSTS_PIN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RSTS_CPU	RSTS_PMU	RSTS_SYS	Reserved	Reserved	RSTS_WDG	RSTS_PAD	RSTS_POR

Table 5-2 System Reset Source Register (RSTSRC, address 0x5000_0004) Bit Description.

Bits	Symbol	Descriptions
[31:11]	Reserved	Reserved
[10]	**RSTS_POI**	The **RSTS_POI** flag is set by hardware if device has powered up from a power on reset condition. 1= A power on Reset has occurred. This bit is cleared by writing 1 to itself. Writing 1 to this bit will clear bits RSTS_POI, RSTS_TIM, and RSTS_PIN
[9]	**RSTS_TIM**	The **RSTS_TIM** flag is set by hardware if device has powered up due to the DPD timer function. 1= A power on was triggered by DPD timer. This bit is cleared by writing 1 to itself. Writing 1 to this bit will clear bits RSTS_POI, RSTS_TIM, and RSTS_PIN
[8]	**RSTS_PIN**	The **RSTS_PIN** flag is set by hardware if device has powered up from deep power down (DPD) due to action of the WAKEUP pin. 1= A power on was triggered by WAKEUP pin. This bit is cleared by writing 1 to itself. Writing 1 to this bit will clear bits RSTS_POI, RSTS_TIM, and RSTS_PIN
[7]	**RSTS_CPU**	The **RSTS_CPU** flag is set by hardware if software writes **CPU_RST** (**IPRSTCR1[1]**) with a "1" to reset Cortex-M0 CPU kernel and Flash memory controller (FMC). 1= The Cortex-M0 CPU kernel and FMC has been reset by software setting **CPU_RST** to 1. This bit is cleared by writing 1 to itself.

[6]	**RSTS_PMU**	The **RSTS_PMU** flag is set if the PMU has called for a reset of logic. This indicates device has come from a power down condition. 1= PMU reset the system from a power down/standby event. This bit is cleared by writing 1 to itself.
[5]	**RSTS_SYS**	The **RSTS_SYS** flag is set if the previous reset source originates from the Cortex_M0 kernel. 1= The Cortex_M0 MCU issued a reset signal to reset the system by software writing 1 to bit SYSRESTREQ(AIRCR[2], Application Interrupt and Reset Control Register) in system control registers of Cortex_M0 kernel. This bit is cleared by writing 1 to itself.
[2]	**RSTS_WDG**	The **RSTS_WDG** flag is set if pervious reset source originates from the Watch-Dog module. 1= The Watch-Dog module issued the reset signal to reset the system. This bit is cleared by writing 1 to itself.
[1]	**RSTS_PAD**	The **RSTS_PAD** flag is if pervious reset source originates from the /RESET pin. 1 = Pin /RESET had issued the reset signal to reset the system. This bit is cleared by writing 1 to itself.
[0]	**RSTS_POR**	The **RSTS_POR** flag is set if the previous reset source was the Power-On Reset (POR) module or CHIP_RST (IPRSTC1[0]). 1 = Core was reset by hardware block. This bit is cleared by writing 1 to itself.

IP Reset Control Register1(IPRSTC1)

Register	Offset	R/W	Description	Reset Value
IPRSTC1	GCR_BA+08	R/W	IP Reset Control Resister 1	0x0000_0000

7	6	5	4	3	2	1	0
		Reserved			PDMA_RST	CPU_RST	CHIP_RST

Table 5-3 IP Reset Control Register 1 (IPRSTC1 address 0x5000_0008) Bit Description.

Bits	Descriptions	
[31:3]	Reserved	Reserved
[2]	**PDMA_RST**	**PDMA Controller Reset** Set "1" will generate a reset signal to the PDMA Block. User needs to set this bit to "0" to release from the reset state 0= Normal operation 1= PDMA IP reset
[1]	**CPU_RST**	**CPU kernel one shot reset.** Setting this bit will reset the CPU kernel and Flash Memory Controller(FMC), this bit will automatically return to "0" after the 2 clock cycles This bit is a protected bit, to program first issue the unlock sequence (see Protected) 0= Normal 1= Reset CPU
[0]	**CHIP_RST**	**CHIP one shot reset.** Set this bit will reset the whole chip, this bit will automatically return to "0" after the 2 clock cycles. CHIP_RST has same behavior as POR reset, all the chip modules are reset and the chip configuration settings from flash are reloaded. This bit is a protected bit, to program first issue the unlock sequence (see Protected Register Lock Key Register (REGLOCK)) 0= Normal 1= Reset CHIP

IP Reset Control Register2 (IPRSTC2)

Setting these bits "1" will generate an asynchronous reset signal to the corresponding peripheral block. The user needs to set bit to "0" to release block from the reset state.

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

31	30	29	28	27	26	25	24
Reserved	ANA_RST	I2S_RST	ADC_RST	Reserved	Reserved	Reserved	Reserved
23	22	21	20	19	18	17	16
Reserved	ACMP_RST	Reserved	PWM10_RST	CRC_RST	BIQ_RST	Reserved	UART0_RST
15	14	13	12	11	10	9	8
Reserved	Reserved	DPWM_RST	SPI0_RST	Reserved	Reserved	Reserved	I2C0_RST
7	6	5	4	3	2	1	0
TMR1_RST	TMR0_RST	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Table 5-4 IP Reset Control Register 2 (IPRSTC2 address 0x5000_000C) Bit Description.

Bits	Descriptions	
[30]	**ANA_RST**	**Analog block control Reset.** 1=Reset, 0=Normal Operation.
[29]	**I2S_RST**	**I2S Controller Reset.** 1=Reset, 0=Normal Operation.
[28]	**ADC_RST**	**ADC Controller Reset.** 1=Reset, 0=Normal Operation.
[22]	**ACMP_RST**	**Analog Comparator Reset.** 1=Reset, 0=Normal Operation.
[20]	**PWM10_RST**	**PWM10 controller Reset.** 1=Reset, 0=Normal Operation.
[19]	**CRC_RST**	**CRC Generation Block Reset.** 1=Reset, 0=Normal Operation.
[18]	**BIQ_RST**	**Biquad Filter Block Reset.** 1=Reset, 0=Normal Operation.
[16]	**UART0_RST**	**UART0 controller Reset.** 1=Reset, 0=Normal Operation.
[13]	**DPWM_ RST**	**DPWM Speaker Driver Reset.** 1=Reset, 0=Normal Operation.
[12]	**SPI0_ RST**	**SPI0 controller Reset.** 1=Reset, 0=Normal Operation.
[8]	**I2C0_RST**	**I2C0 controller Reset.** 1=Reset, 0=Normal Operation.
[7]	**TMR1_RST**	**Timer1 controller Reset.** 1=Reset, 0=Normal Operation.
[6]	**TMR0_RST**	**Timer0 controller Reset.** 1=Reset, 0=Normal Operation.

GPIOA Input Type Control Register (GPA_INP)

Register	Offset	R/W	Description	Reset Value
GPA_INP	GCR_BA+30	R/W	GPIOA input type control register	0x0000_0000

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

Table 5-5 GPIOA Input Type Control Register (GPA_INP address 0x5000_0030) Bit Description.

Bits	Descriptions	
[31:16]	**SCHMITT**	This register controls whether the GPIO input buffer Schmitt trigger is enabled. 1= GPIOA[15:0] I/O input Schmitt Trigger enabled 0= GPIOA[15:0] I/O input Schmitt Trigger disabled
[15:0]	**Reserved**	

GPIOB Input Type Control Register (GPB_INP)

Register	Offset	R/W	Description	Reset Value
GPB_INP	GCR_BA+34	R/W	GPIOB input type control register	0x0000_0000

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

Table 5-6 GPIOB Input Type Control Register (GPB_INP address 0x5000_0034) Bit Description.

Bits	Descriptions	
[23:16]	**SCHMITT**	This register controls whether the GPIO input buffer Schmitt trigger is enabled. 1= GPIOB[7:0] I/O input Schmitt Trigger enabled 0= GPIOB[7:0] I/O input Schmitt Trigger disabled

GPIO Alternative Function Control Register (GPA_ALT, GPB_ALT)

Each GPIO pin can take on multiple alternate functions depending on the setting of this register. Each pin has two bits of alternate function control. Set to 00 the pin is a standard GPIO pin whose attributes are defined by the GPIO control registers (See Section 5.4). Set to other values the pin is assigned to a peripheral as outlined in table below.

Register	Offset	R/W	Description	Reset Value
GPA_ALT	GCR_BA+38	R/W	Alternative Function Pin Control Register.	0x0000_0000
GPB_ALT	GCR_BA+3C	R/W	Alternative Function Pin Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
GPA15		GPA14		GPA13		GPA12	
23	22	21	20	19	18	17	16
GPA11		GPA10		GPA9		GPA8	
15	14	13	12	11	10	9	8
GPA7		GPA6		GPA5		GPA4	
7	6	5	4	3	2	1	0
GPA3		GPA2		GPA1		GPA0	

Table 5-7 GPIOA Alternate Function Register (GPA_ALT address 0x5000_0038)

GPIO	Power Domain	GPAn=01		GPAn =10		GPAn =11	
		Function	Type	Function	Type	Function	Type
GPIOA0	VDD33	SPI_MOSI0	O	MCLK	O		
GPIOA1	VDD33	SPI_SCLK	IO	I2C_SCL	IO		
GPIOA2	VDD33	SPI_SSB0	IO				
GPIOA3	VDD33	SPI_MISO0	I	I2C_SDA	IO		
GPIOA4	VDD33	I2S_FS	IO				
GPIOA5	VDD33	I2S_BCLK	IO			SPI_SSB1	O
GPIOA6	VDD33	I2S_SDI	I	UART_TX	O		
GPIOA7	VDD33	I2S_SDO	O				
GPIOA8	VCCD	UART_TX	O	CMP12	AIO	PWM2	O
GPIOA9	VCCD	UART_RX	I	CMP13	AIO	PWM3	O
GPIOA10	VCCD	I2C_SDA	IO	CMP14	AIO	UART_RTSn	O

GPIOA11	VCCD	I2C_SCL	IO	CMP15	AIO	UART_CTSn	I
GPIOA12	VCCD	PWM0	O	SPKP	O	I2S_FS	IO
GPIOA13	VCCD	PWM1	O	SPKM	O	I2S_BCLK	IO
GPIOA14	VCCD	TM0	I	SDCLK	O	PWM4	O
GPIOA15	VCCD	TM1	I	SDIN	I	PWM5	O

Table 5-8 GPIOB Alternate Function Register (GPB_ALT address 0x5000_003C)

GPIO	Power Domain	GPBn=01		GPBn =10		GPBn =11	
		Function	Type	Function	Type	Function	Type
GPIOB0	VCCD	SPI_SSB1	O	CMP0	AIO	SPI_SSB0	IO
GPIOB1	VCCD	MCLK	O	CMP1	AIO	SPI_SSB1	O
GPIOB2	VCCD	I2C_SCL	IO	CMP2	AIO	SPI_SCLK	IO
GPIOB3	VCCD	I2C_SDA	IO	CMP3	AIO	SPI_MISO0	I
GPIOB4	VCCD	PWM0B	O	CMP4	AIO	SPI_MOSI0	O
GPIOB5	VCCD	PWM1B	O	CMP5	AIO	SPI_MISO1	I
GPIOB6	VCCD	I2S_SDI	I	CMP6	AIO	SPI_MOSI1	O
GPIOB7	VCCD	I2S_SDO	O	CMP7	AIO		
GPIOB8	VCCD	I2S_FS(master)	O	CMP8	AIO		
GPIOB9	VCCD	I2S_BCLK (master)	O	CMP9	AIO		
GPIOB10	VCCD			CMP10	AIO		
GPIOB11	VCCD	I2S_SDO	O	CMP11	AIO		
GPIOB12	VDD33	SPI_MISO1	IO	PWM4	O	PWM5B	O
GPIOB13	VDD33	SPI_MOSI1	IO	PWM5	O	PWM4B	O
GPIOB14	VCCD	PWM2	O	UART_TX	O	PWM3B	O
GPIOB15	VCCD	PWM3	O			PWM2B	

Wakeup Pin Control Register (WAKECR)

The WAKEUP pin of the ISD93xx is a special purpose pin that can be used to wake the device from a deep power down condition when all other pins of the device are inactive. When the device is active, this register can be used to set the state of the WAKEUP pin. The default state of the pin is as a tri-state input.

Register	Offset	R/W	Description	Reset Value
WAKECR	GCR_BA+54	R/W	Wakeup Pin control register	0x0000_0006

7	6	5	4	3	2	1	0
Reserved				WAKE_DIN	WAKE_TRI	WAKE_OENB	WAKE_DOUT

Table 5-9 Wakeup Pin Control Register (WAKECR, address 0x5000_0054) Bit Description.

Bits	Descriptions	
[3]	WAKE_DOUT	Output state. Default=0
[2]	WAKE_OENB	Pin output enable bar. 1= tristate (default), 0=drive WAKE_DOUT to pin.
[1]	WAKE_TRI	Pin pull-up control. 1= tristate (default). 0=pull-up enable. This signal is latched in deep power down and preserved.
[0]	WAKE_DIN	State of WAKEUP pin. Read only.

Protected Register Lock Key Register (REGLOCK)

Certain critical system control registers are protected against inadvertent write operations which may disturb chip operation. These system control registers are locked after power on reset until the user specifically issues an unlock sequence to open the lock. The unlock sequence is to write to REGLOCK the data 0x59, 0x16, 0x88. Any different sequence, data or a write to any other address will abort the unlock sequence.

MDK provides the defined function UNLOCKREG(x); which will execute this sequence.

The status of the lock can be determined by reading REGLOCK bit0: "1" is unlocked, "0" is locked. Once unlocked, user can update protected register values. To lock registers again, write any data to REGLOCK.

This register is write accessible for writing key values and read accessible to determine RegUnLock status.

Register	Offset	R/W	Description	Reset Value
REGLOCK	GCR_BA+100	R/W	Register Lock Key Address register	0x0000_0000

7	6	5	4	3	2	1	0
							RegUnLock

Table 5-10 Protected Register Lock Key Register (REGLOCK address 0x5000_0100) Bit Description.

Bits	Descriptions	
[31:16]	Reserved	Reserved
[0]	**RegUnLock**	1 = Protected registers are unlocked 0 = Protected registers are locked. Any write to the target register is ignored.

Oscillator Trim Control Register (OSCTRIM)

The master oscillator of the ISD93xx has an adjustable frequency and is controlled by the OSCTRIM register. This register contains two oscillator frequency trim values, which one is active depends upon the setting of register CLKSEL0 OSCFSel bit. If this bit is 0, OSCTRIM[0] is active, if 1 then OSCTRIM[1] is active. Upon power on reset this register is loaded from flash memory with factory stored values to give oscillator frequencies of 49.152MHz for OSCTRIM[0] and 32.768MHz for OSCTRIM[1]. If users wish to change either of the default frequencies it is possible to do so by setting this register. An additional SUPERFINE trim register is also available to interpolate frequencies between the available OSCTRIM settings (see Table 7-36)

This register is a protected register, to write to register first issue the unlock sequence (see Protected Register Lock Key Register (REGLOCK))

Register	Offset	R/W	Description	Reset Value
OSCTRIM	GCR_BA+110	R/W	Oscillator Frequency Adjustment control register	0xXXXX_XXXX

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

Table 5-11 Oscillator Frequency Adjustment Control Register (OSCTRIM, address 0x5000_0110).

Bits	Descriptions	
[24]	OSCTRIM[1].RANGE	Range bit for oscillator. 1= low range, 0= high range.
[23:16]	OSCTRIM[1].TRIM	8 bit trim for oscillator. TRIM[7:5] are 8 coarse trim ranges which overlap in frequency. TRIM[4:0] are 32 fine trim steps of approximately 0.5% resolution.
[8]	OSCTRIM[0].RANGE	Range bit for oscillator. 1= low range, 0= high range.
[7:0]	OSCTRIM[0].TRIM	8 bit trim for oscillator. TRIM[7:5] are 8 coarse trim ranges which overlap in frequency. TRIM[4:0] are 32 fine trim steps of approximately 0.5% resolution.

5.2.6   System Timer (SysTick)

The Cortex-M0 includes an integrated system timer, SysTick. SysTick provides a simple, 24-bit, Clear-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. The counter can be used in several different ways, for example:

- An RTOS tick timer which fires at a programmable rate (for example 100Hz) and invokes a SysTick routine.
- A high speed alarm timer using Core clock.
- A variable rate alarm or signal timer – the duration range dependent on the reference clock used and the dynamic range of the counter.
- A simple counter. Software can use this to measure time to completion and time used.
- An internal clock source control based on missing/meeting durations. The COUNTFLAG bit-field in the control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop.

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

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

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

In DEEPSLEEP and power down modes, the SysTick timer is disabled so cannot be used to wake up the device.

For more detailed information, please refer to the documents "ARM®Cortex™-M0 Technical Reference Manual" and "ARM® v6-M Architecture Reference Manual".

5.2.6.1   *System Timer Control Register Map*

R: read only, **W**: write only, **R/W**: both read and write, **W&C**: Write 1 clear

Register	Offset	R/W	Description	Reset Value
SCS_BA = 0xE000_E000				
CTRL	SCS_BA + 010	R/W	SysTick Control and Status	0x0000_0004
LOAD	SCS_BA + 014	R/W	SysTick Reload value	0xXXXX_XXXX
VAL	SCS_BA + 018	R/W	SysTick Current value	0xXXXX_XXXX

5.2.6.2   *System Timer Control Register Description*

SysTick Control and Status（SysTick->CTRL）

Register	Offset	R/W	Description	Reset Value
CTRL	SCS_BA + 010	R/W	SysTick Control and Status	0x0000_0004

Table 5-12SysTick Control and Status Register (SysTick->CTRL, address 0xE000_E010)

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							COUNTFLAG

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved					CLKSRC	TICKINT	ENABLE

Bits	Descriptions	
[31:17]	Reserved	Reserved
[16]	**COUNTFLAG**	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**	1 : Core clock used for SysTick. 0: Clock selected from SYSCLK->CLKSEL0.STCLK_S is used as clock source.
[1]	**TICKINT**	Enables SysTick Exception request. 1 : Counting down to 0 will cause 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 : Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to zero has occurred.
[0]	**ENABLE**	1 : The counter will operate in a multi-shot manner. 0 : The counter is disabled

SysTick Reload Value Register（SysTick->LOAD）

Register	Offset	R/W	Description	Reset Value
LOAD	SCS_BA + 014	R/W	SysTick Reload Value Register	0xXXXX_XXXX

Table 5-13SysTick Reload Value Register (SysTick->LOAD, address 0xE000_E014)

31	30	29	28	27	26	25	24
Reserved							

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

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	**RELOAD**	Value to load into the Current Value register when the counter reaches 0. To generate a multi-shot timer with a period of N processor clock cycles, use a RELOAD value of N-1. For example, if the SysTick interrupt is required every 200 clock pulses, set RELOAD to 199.

SysTick Current Value Register（SysTick->VAL）

Register	Offset	R/W	Description	Reset Value
VAL	SCS _BA + 018	R/W	SysTick Current Value Register	0xXXXX_XXXX

Table 5-14 SysTick Current Value Register(SysTick->VAL, address 0xE000_E018)

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

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	**CURRENT**	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 and also clear the COUNTFLAG bit.

5.2.7　Nested Vectored Interrupt Controller (NVIC)

Cortex-M0 includes an interrupt controller the "Nested Vectored Interrupt Controller (NVIC)". It is closely coupled to the processor kernel and provides following features:

- Nested and Vectored interrupt support

- Automatic processor state saving and restoration

- Dynamic priority changing

- 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 any 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 corresponding 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 above mentioned registers from the stack and resume normal execution. This provides a high speed and deterministic time to process any interrupt request.

The NVIC supports "Tail Chaining" which handles back-to-back interrupts efficiently without the overhead of state saving and restoration and therefore reduces delay time in switching to a pending ISR at the end of the 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. This aids real-time, high priority, interrupt capability.

For more detailed information, please refer to the documents "ARM® Cortex™-M0 Technical Reference Manual" and ""ARM® v6-M Architecture Reference Manual".

5.2.7.1 *Exception Mode land System Interrupt Map*

The following table lists the exception model supported by ISD93xxseries. Software can set four levels of priority on certain 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 priority "0" is treated as the fourth priority on the system, after three system exceptions "Reset", "NMI" and "Hard Fault".

Table 5-15Exception Model

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

Table 5-16System Interrupt Map

Vector Number	Interrupt Number (Bit in Interrupt Registers)	Interrupt Name	Source IP	Interrupt description
0 ~ 15	-	-	-	System exceptions
16	0	BOD_IRQn	Brown-Out	Brownout low voltage detector interrupt
17	1	WDT_IRQn	WDT	Watch Dog Timer interrupt
18	2	EINT0_IRQn	GPIO	External signal interrupt from PB.0 pin
19	3	EINT1_IRQn	GPIO	External signal interrupt from PB.1 pin
20	4	GPAB_IRQn	GPIO	External signal interrupt from PA[15:0] / PB[7:2]
21	5	ALC_IRQn	ALC	Automatic Level Control Interrupt
22	6	PWMA_IRQn	PWM01	PWM0, PWM1 interrupt
23	7	Reserved		
24	8	TMR0_IRQn	TMR0	Timer 0 interrupt

25	9	**TMR1_IRQn**	TMR1	Timer 1 interrupt
26	10	Reserved		
27	11	Reserved		
28	12	**UART0_IRQn**	UART0	UART0interrupt
29	13	Reserved		
30	14	**SPI0_IRQn**	SPI0	SPI0 interrupt
31	15	Reserved		
32	16	Reserved		
33	17	Reserved		
34	18	**I2C0_IRQn**	I2C0	I2C0 interrupt
35	19	Reserved		
36	20	Reserved		
37	21	**TALARM_IRQn**	TALARM	Temperature Alarm Interrupt
38	22	Reserved		
39	23	Reserved		
40	24	Reserved		
41	25	**ACMP_IRQn**	ACMP	Analog Comparator-0 or Comaprator-1 interrupt
42	26	**PDMA_IRQn**	PDMA	PDMA interrupt
43	27	**I2S_IRQn**	I2S	I2S interrupt
44	28	**CAPS_IRQn**	ANA	CapSense Relaxation Oscillator Interrupt
45	29	**ADC_INT**	SDADC	Audio ADC interrupt
46	30	Reserved		
47	31	**RTC_INT**	RTC	Real time clock interrupt

5.2.7.2   *Vector Table*

When an interrupt is accepted, the processor will automatically fetch the starting address of the interrupt service routine (ISR) from the vector table in memory. For ARMv6-M, the vector table base address is fixed in flash 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.

Vector Table Word Offset	Description
0	SP_main - The Main stack pointer
Vector Number	Exception Entry Pointer using that Vector Number

Table 5-17 Vector Table Format

5.2.7.3   *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 activate. 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.

5.2.7.4   *NVIC Control Registers*

R: read only, **W**: write only, **R/W**: both read and write, **W&C**: Write 1 clear

Register	Offset	R/W	Description	Reset Value	Reference
SCS_BA = 0xE000_E000					
NVIC _ISER	SCS_BA + 100	R/W	IRQ0 ~ IRQ31 Set-Enable Control Register	0x0000_0000	Table 5-18
NVIC_ICER	SCS_BA + 180	R/W	IRQ0 ~ IRQ31 Clear-Enable Control Register	0x0000_0000	Table 5-19
NVIC_ISPR	SCS_BA + 200	R/W	IRQ0 ~ IRQ31 Set-Pending Control Register	0x0000_0000	Table 5-20
NVIC_ICPR	SCS_BA + 280	R/W	IRQ0 ~ IRQ31 Clear-Pending Control Register	0x0000_0000	Table 5-21
NVIC_IPR0	SCS_BA + 400	R/W	IRQ0 ~ IRQ3 Priority Control Register	0x0000_0000	Table 5-22
NVIC_IPR1	SCS_BA + 404	R/W	IRQ4 ~ IRQ7 Priority Control Register	0x0000_0000	Table 5-23
NVIC_IPR2	SCS_BA + 408	R/W	IRQ8 ~ IRQ11 Priority Control Register	0x0000_0000	Table 5-24
NVIC_IPR3	SCS_BA + 40C	R/W	IRQ12 ~ IRQ15 Priority Control Register	0x0000_0000	Table 5-25
NVIC_IPR4	SCS_BA + 410	R/W	IRQ16 ~ IRQ19 Priority Control Register	0x0000_0000	Table 5-26
NVIC_IPR5	SCS_BA + 414	R/W	IRQ20 ~ IRQ23 Priority Control Register	0x0000_0000	Table 5-27
NVIC_IPR6	SCS_BA + 418	R/W	IRQ24 ~ IRQ27 Priority Control Register	0x0000_0000	Table 5-28
NVIC_IPR7	SCS_BA + 41C	R/W	IRQ28 ~ IRQ31 Priority Control Register	0x0000_0000	Table 5-29

IRQ0 ~ IRQ31 Set-Enable Control Register（NVIC_ISER）

Register	Offset	R/W	Description	Reset Value
NVIC_ISER	SCS _BA + 100	R/W	IRQ0 ~ IRQ31 Set-Enable Control Register	0x0000_0000

If a pending interrupt is enabled, the NVIC activates the interrupt based on its priority. If an interrupt is not enabled, asserting its interrupt signal changes the interrupt state to pending, but the NVIC never activates the interrupt, regardless of its priority.

Table 5-18Interrupt Set-Enable Control Register (ISER, address 0xE000_E100) Bit Description

Bits	Descriptions	
[31:0]	**SETENA**	Enable one or more interrupts within a group of 32. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Writing 1 will enable the associated interrupt. Writing 0 has no effect. The register reads back the current enable state.

IRQ0 ~ IRQ31 Clear-Enable Control Register(NVIC_ICER）

Register	Offset	R/W	Description	Reset Value
NVIC_ICER	SCS _BA + 180	R/W	IRQ0 ~ IRQ31 Clear-Enable Control Register	0x0000_0000

Table 5-19 Interrupt Clear-Enable Control Register (ICER, address 0xE000_E180) Bit Description

Bits	Descriptions	
[31:0]	**CLRENA**	Disable one or more interrupts within a group of 32. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Writing 1 will disable the associated interrupt. Writing 0 has no effect. The register reads back with the current enable state.

IRQ0 ~ IRQ31 Set-Pending Control Register（NVIC_ISPR）

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR	SCS _BA + 200	R/W	IRQ0 ~ IRQ31 Set-Pending Control Register	0x0000_0000

Table 5-20Interrupt Set-Pending Control Register (ISPR, address 0xE000_E200)

Bits	Descriptions	
[31:0]	**SETPEND**	Writing 1 to a bit forces pending state of the associated interrupt under software control. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Writing 0 has no effect. The register reads back with the current pending state.

IRQ0 ~ IRQ31 Clear-Pending Control Register（NVIC_ICPR）

Register	Offset	R/W	Description	Reset Value
NVIC_ICPR	SCS _BA + 280	R/W	IRQ0 ~ IRQ31 Clear-Pending Control Register	0x0000_0000

Table 5-21 Interrupt Clear-Pending Control Register (ICPR, address 0xE000_E280)

Bits	Descriptions	
[31:0]	**CLRPEND**	Writing 1 to a bit to clear the pending state of associated interrupt under software control. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Writing 0 has no effect. The register reads back with the current pending state.

IRQ0 ~ IRQ3 Interrupt Priority Register（NVIC_IPR0）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR0	SCS _BA + 400	R/W	IRQ0 ~ IRQ3 Interrupt Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_3		**Reserved**					
23	22	21	20	19	18	17	16
PRI_2		**Reserved**					
15	14	13	12	11	10	9	8
PRI_1		**Reserved**					
7	6	5	4	3	2	1	0
PRI_0		**Reserved**					

Table 5-22Interrupt Priority Register (IPR0, address 0xE000_E400)

Bits	Descriptions	
[31:30]	**PRI_3**	Priority of IRQ3 "0" denotes the highest priority and "3" denotes lowest priority
[23:22]	**PRI_2**	Priority of IRQ2 "0" denotes the highest priority and "3" denotes lowest priority
[15:14]	**PRI_1**	Priority of IRQ1 "0" denotes the highest priority and "3" denotes lowest priority
[7:6]	**PRI_0**	Priority of IRQ0 "0" denotes the highest priority and "3" denotes lowest priority

IRQ4 ~ IRQ7 Interrupt Priority Register（NVIC_IPR1）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR1	SCS _BA + 404	R/W	IRQ4 ~ IRQ7 Interrupt Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_7		**Reserved**					
23	22	21	20	19	18	17	16
PRI_6		**Reserved**					
15	14	13	12	11	10	9	8
PRI_5		**Reserved**					
7	6	5	4	3	2	1	0
PRI_4		**Reserved**					

Table 5-23Interrupt Priority Register (IPR1, address 0xE000_E404)

Bits	Descriptions	
[31:30]	**PRI_7**	Priority of IRQ7 "0" denotes the highest priority and "3" denotes lowest priority
[23:22]	**PRI_6**	Priority of IRQ6 "0" denotes the highest priority and "3" denotes lowest priority
[15:14]	**PRI_5**	Priority of IRQ5 "0" denotes the highest priority and "3" denotes lowest priority
[7:6]	**PRI_4**	Priority of IRQ4 "0" denotes the highest priority and "3" denotes lowest priority

IRQ8 ~ IRQ11 Interrupt Priority Register（NVIC_IPR2）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR2	SCS _BA + 408	R/W	IRQ8 ~ IRQ11 Interrupt Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		Reserved					
23	22	21	20	19	18	17	16
PRI_10		Reserved					
15	14	13	12	11	10	9	8
PRI_9		Reserved					
7	6	5	4	3	2	1	0
PRI_8		Reserved					

Table 5-24Interrupt Priority Register (IPR2, address 0xE000_E408)

Bits	Descriptions	
[31:30]	**PRI_11**	**Priority of IRQ11** "0" denotes the highest priority and "3" denotes lowest priority
[23:22]	**PRI_10**	**Priority of IRQ10** "0" denotes the highest priority and "3" denotes lowest priority
[15:14]	**PRI_9**	**Priority of IRQ9** "0" denotes the highest priority and "3" denotes lowest priority
[7:6]	**PRI_8**	**Priority of IRQ8** "0" denotes the highest priority and "3" denotes lowest priority

IRQ12 ~ IRQ15 Interrupt Priority Register（NVIC_IPR3）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR3	SCS _BA + 40C	R/W	IRQ12 ~ IRQ15 Interrupt Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15		**Reserved**					
23	22	21	20	19	18	17	16
PRI_14		**Reserved**					
15	14	13	12	11	10	9	8
PRI_13		**Reserved**					
7	6	5	4	3	2	1	0
PRI_12		**Reserved**					

Table 5-25Interrupt Priority Register (IPR3, address 0xE000_E40C)

Bits	Descriptions	
[31:30]	**PRI_15**	**Priority of IRQ15** "0" denotes the highest priority and "3" denotes lowest priority
[23:22]	**PRI_14**	**Priority of IRQ14** "0" denotes the highest priority and "3" denotes lowest priority
[15:14]	**PRI_13**	**Priority of IRQ13** "0" denotes the highest priority and "3" denotes lowest priority
[7:6]	**PRI_12**	**Priority of IRQ12** "0" denotes the highest priority and "3" denotes lowest priority

IRQ16 ~ IRQ19 Interrupt Priority Register（NVIC_IPR4）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR4	SCS _BA + 410	R/W	IRQ16 ~ IRQ19 Interrupt Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_19		Reserved					

23	22	21	20	19	18	17	16
PRI_18		Reserved					

15	14	13	12	11	10	9	8
PRI_17		Reserved					

7	6	5	4	3	2	1	0
PRI_16		Reserved					

Table 5-26Interrupt Priority Register (IPR4, address 0xE000_E410)

Bits	Descriptions	
[31:30]	**PRI_19**	**Priority of IRQ19** "0" denotes the highest priority and "3" denotes lowest priority
[23:22]	**PRI_18**	**Priority of IRQ18** "0" denotes the highest priority and "3" denotes lowest priority
[15:14]	**PRI_17**	**Priority of IRQ17** "0" denotes the highest priority and "3" denotes lowest priority
[7:6]	**PRI_16**	**Priority of IRQ16** "0" denotes the highest priority and "3" denotes lowest priority

IRQ20 ~ IRQ23 Interrupt Priority Register（NVIC_IPR5）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR5	SCS _BA + 414	R/W	IRQ20 ~ IRQ23 Interrupt Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_23		**Reserved**					
23	22	21	20	19	18	17	16
PRI_22		**Reserved**					
15	14	13	12	11	10	9	8
PRI_21		**Reserved**					
7	6	5	4	3	2	1	0
PRI_20		**Reserved**					

Table 5-27Interrupt Priority Register (IPR5, address 0xE000_E414)

Bits	Descriptions	
[31:30]	**PRI_23**	**Priority of IRQ23** "0" denotes the highest priority and "3" denotes lowest priority
[23:22]	**PRI_22**	**Priority of IRQ22** "0" denotes the highest priority and "3" denotes lowest priority
[15:14]	**PRI_21**	**Priority of IRQ21** "0" denotes the highest priority and "3" denotes lowest priority
[7:6]	**PRI_20**	**Priority of IRQ20** "0" denotes the highest priority and "3" denotes lowest priority

IRQ24 ~ IRQ27 Interrupt Priority Register（NVIC_IPR6）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR6	SCS _BA + 418	R/W	IRQ24 ~ IRQ27 Interrupt Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_27		Reserved					
23	22	21	20	19	18	17	16
PRI_26		Reserved					
15	14	13	12	11	10	9	8
PRI_25		Reserved					
7	6	5	4	3	2	1	0
PRI_24		Reserved					

Table 5-28Interrupt Priority Register (IPR6, address 0xE000_E418)

Bits	Descriptions	
[31:30]	**PRI_27**	**Priority of IRQ27** "0" denotes the highest priority and "3" denotes lowest priority
[23:22]	**PRI_26**	**Priority of IRQ26** "0" denotes the highest priority and "3" denotes lowest priority
[15:14]	**PRI_25**	**Priority of IRQ25** "0" denotes the highest priority and "3" denotes lowest priority
[7:6]	**PRI_24**	**Priority of IRQ24** "0" denotes the highest priority and "3" denotes lowest priority

IRQ28 ~ IRQ31 Interrupt Priority Register（NVIC_IPR7）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR7	SCS _BA + 41C	R/W	IRQ28 ~ IRQ31 Interrupt Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_31		Reserved					
23	22	21	20	19	18	17	16
PRI_30		Reserved					
15	14	13	12	11	10	9	8
PRI_29		Reserved					
7	6	5	4	3	2	1	0
PRI_28		Reserved					

Table 5-29Interrupt Priority Register (IPR7, address 0xE000_E41C)

Bits	Descriptions	
[31:30]	**PRI_31**	**Priority of IRQ31** "0" denotes the highest priority and "3" denotes lowest priority
[23:22]	**PRI_30**	**Priority of IRQ30** "0" denotes the highest priority and "3" denotes lowest priority
[15:14]	**PRI_29**	**Priority of IRQ29** "0" denotes the highest priority and "3" denotes lowest priority
[7:6]	**PRI_28**	**Priority of IRQ28** "0" denotes the highest priority and "3" denotes lowest priority

ISD93xx Series Design Guide

5.2.7.5 *Interrupt Source Control Registers*

Along with the interrupt control registers associated with the NVIC, the ISD93xx also implements some specific control registers to facilitate the interrupt functions, including "interrupt source identify", "NMI source selection" and "interrupt test mode". They are described as below.

R: read only, **W**: write only, **R/W**: both read and write, **W&C**: Write 1 clear

Register	Offset	R/W	Description	Reset Value
INT_BA = 0x5000_0300				
IRQn_SRC	INT_BA+0x04*n	R	MCU IRQn interrupt source identify (n=0,..,31)	0xXXXX_XXXX
NMI_SEL	INT_BA+0x80	R/W	NMI source interrupt select control register	0x0000_0000
MCU_IRQ	INT_BA+0x84	R/W	MCU IRQ Number identity register	0x0000_0000

Interrupt Source Identify Register (IRQn_SRC)

Register	Offset	R/W	Description	Reset Value
IRQn_SRC	INT_BA+0x00 …….. INT_BA+0x7C	R	MCU IRQ0 interrupt source identify : MCU IRQ31 interrupt source identify	0xXXXX_XXXX

7	6	5	4	3	2	1	0
Reserved					INT_SRC[2:0]		

The IRQn_SRC register reports which of multiple sources generated an interrupt. For the ISD93xx series the only interrupts that have multiple sources are IRQ4 and IRQ6. The return of the IRQn_SRC register for these interrupts is shown below. All other IRQn_SRC registers will report the interrupt status in bit 0 of field.

Address	INT-Num	Bits	Descriptions
INT_BA+0x10	4	[2:0]	Bit2:1'b0 Bit1:GPB_INT Bit0:GPA_INT
INT_BA+0x18	6	[2:0]	Bit1:PWM1_INT Bit0:PWM0_INT

NMI Interrupt Source Select Control Register (NMI_SEL)

Register	Offset	R/W	Description	Reset Value
NMI_SEL	INT_BA+0x80	R/W	NMI source interrupt select control register	0x0000_0000

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

Bits	Descriptions	
[31:7]	Reserved	Reserved
[7]	**IRQ_TM**	IRQ Test mode. If set to 1 then peripheral IRQ signals (0-31) are replaced by the value in the MCU_IRQ register. This is a protected register to program first issue the unlock sequence (see Protected Register Lock Key Register (REGLOCK))
[4:0]	**NMI_SEL**	The NMI interrupt to Cortex-M0 can be selected from one of the interrupt[31:0] The NMI_SEL bit[4:0] used to select the NMI interrupt source

MCU Interrupt Request Source Test Mode Register (MCU_IRQ)

Register	Offset	R/W	Description	Reset Value
MCU_IRQ	INT_BA+0x84	R/W	MCU Interrupt Request Source Register	0x0000_0000

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

Bits	Descriptions	
[31:0]	**MCU_IRQ**	**MCU IRQ Source Test Mode Register** In Normal mode (NMI_SEL register bit [7] = 0) The device collects interrupts from each peripheral and synchronizes them to interrupt the Cortex-M0. In Test mode (NMI_SEL register bit [7] = 1), the interrupts from peripherals are blocked, and the interrupts are replaces by MCU_IRQ[31:0]. When MCU_IRQ[n] is "0" : Writing MCU_IRQ[n] "1"will generate an interrupt to Cortex_M0 NVIC[n]. When MCU_IRQ[n] is "1" (meaning an interrupt is asserted) writing MCU_bit[n] '1' will clear the interrupt Writing MCU_IRQ[n] "0" : has no effect.

5.2.8　System Control Registers

Key control and status features of Coterx-M0 are managed centrally in a System Control Block within the System Control Registers.

For more detailed information, please refer to the documents "ARM® Cortex™-M0 Technical Reference Manual" and ""ARM® v6-M Architecture Reference Manual".

R: read only, **W**: write only, **R/W**: both read and write, **W&C**: Write 1 clear

Register	Offset	R/W	Description	Reset Value
SCS_BA = 0xE000_E000				
CPUID	SCS_BA + D00	R	CPUID Base Register	0x0000_0000
ICSR	SCS_BA + D04	R/W	Interrupt Control State Register	0x0000_0000
AIRCR	SCS_BA + D0C	R.W	Application Interrupt / Reset Control Register	0x0000_0000
SCR	SCS_BA + D10	R/W	System Control Register	0x0000_0000
SHPR2	SCS_BA + D1C	R/W	System Handler Priority Register 2	0x0000_0000
SHPR3	SCS_BA + D20	R/W	System Handler Priority Register 3	0x0000_0000

CPUID Base Register (CPUID)

Register	Offset	R/W	Description	Reset Value
CPUID	SCS_BA + D00	R	CPUID Base Register	0x410CC200

31	30	29	28	27	26	25	24
IMPLEMENTER[7:0]

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

Bits	Descriptions	
[31:24]	**IMPLEMENTER**	Implementer code assigned by ARM. (ARM = 0x41)
[23:20]	Reserved	Reserved
[19:16]	**PART**	Reads as 0xC for ARMv6-M parts
[15:4]	**PARTNO**	Reads as 0xC20.
[3:0]	**REVISION**	Reads as 0x0

Interrupt Control State Register (ICSR)

Register	Offset	R/W	Description	Reset Value
ICSR	SCS_BA + D04	R/W	Interrupt Control State Register	0x00000000

31	30	29	28	27	26	25	24
NMIPENDSET	Reserved		PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	Reserved
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISRPREEMPT	ISRPENDING	Reserved	VECTPENDING[8:4]				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VECTPENDING[3:0]				Reserved			VECTACTIVE[8]
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VECTACTIVE[7:0]							

Bits	R/W	Descriptions	
[31]	R/W	**NMIPENDSET**	Setting this bit will activate an NMI. Since NMI is the highest priority exception, it will activate as soon as it is registered. Reads back with current state (1 if Pending, 0 if not).
[28]	R/W	**PENDSVSET**	Set a pending PendSV interrupt. This is normally used to request a context switch. Reads back with current state (1 if Pending, 0 if not).
[27]	W	**PENDSVCLR**	Write 1 to clear a pending PendSV interrupt.
[26]	R/W	**PENDSTSET**	Set a pending SysTick. Reads back with current state (1 if Pending, 0 if not).
[25]	W	**PENDSTCLR**	Write 1 to clear a pending SysTick.
[23]	R	**ISRPREEMPT**	If set, a pending exception will be serviced on exit from the debug halt state.
[22]	R	**ISRPENDING**	Indicates if an external configurable (NVIC generated) interrupt is pending.
[20:12]	R	**VECTPENDING**	Indicates the exception number for the highest priority pending exception. The pending state includes the effect of memory-mapped enable and mask registers. It does not include the PRIMASK special-purpose register qualifier. A value of zero indicates no pending exceptions.
[8:0]	R	**VECTACTIVE**	0: Thread mode Value> 1: the exception number for the current executing exception.

Application Interrupt and Reset Control Register(AIRCR)

Register	Offset	R/W	Description	Reset Value
AIRCR	SCS_BA + D0C	R/W	**Application Interrupt and Reset Control Register**	0x00000000

31	30	29	28	27	26	25	24
VECTKEY							

23	22	21	20	19	18	17	16
VECTKEY							

15	14	13	12	11	10	9	8
ENDIANESS	Reserved						

7	6	5	4	3	2	1	0
Reserved					SYSRESETREQ	VECTCLRACTIVE	Reserved

Bits	Descriptions	
[31:16]	**VECTKEY**	Vector Key. The value 0x05FA must be written to this register, otherwise a write to the register is UNPREDICTABLE.
[15]	**ENDIANESS**	Read Only. Reads 0 indicating little endian machine.
[2]	**SYSRESETREQ**	System Reset Request: **0** do not request a reset. **1** request reset. Writing 1 to this bit asserts a signal to request a reset by the external system.
[1]	**VECTCLRACTIVE**	Clears all active state information for fixed and configurable exceptions: **0** do not clear state information. **1** clear state information. The effect of writing a 1 to this bit if the processor is not halted in Debug, is UNPREDICTABLE.

System Control Register (SCR)

Register	Offset	R/W	Description	Reset Value
SCR	SCS_BA + D10	R/W	System Control Register	0x00000000

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved			SEVONPEND	Reserved	SLEEPDEEP	SLEEPONEXIT	Reserved

Bits	Descriptions	
[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 enabled, interrupt transitions from Inactive to Pending are included in the list of wakeup events for the WFE instruction. When an event or interrupt enters pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event is registered and affects the next WFE. The processor also wakes up on execution of an SEV instruction.
[2]	**SLEEPDEEP**	Controls whether the processor uses sleep or deep sleep as its low power mode: 0 = sleep 1 = deep sleep. The SLEEPDEEP flag is also used in conjunction with PWRCON register to enter deeper power-down states than purely core sleep states.
[1]	**SLEEPONEXIT**	When set to 1, the core can enter a sleep state on an exception return to Thread mode. This is the mode and exception level entered at reset, the base level of execution. Setting this bit to 1 enables an interrupt driven application to avoid returning to an empty main application.

System Handler Priority Register2 (SHPR2)

Register	Offset	R/W	Description	Reset Value
SHPR2	SCS_BA + D1C	R/W	System Handler Priority Register 2	0x00000000

31	30	29	28	27	26	25	24
	PRI_11			Reserved			

31	30	29	28	27	26	25	24
PRI_11		Reserved					

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved							

Bits	Descriptions	
[31:30]	**PRI_11**	**Priority of system handler 11 – SVCall** "0" denotes the highest priority and "3" denotes lowest priority

System Handler Priority Register3 (SHPR3)

Register	Offset	R/W	Description	Reset Value
SHPR3	SCS_BA + D20	R/W	System Handler Priority Register 3	0x00000000

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

Bits	Descriptions	
[31:30]	**PRI_15**	**Priority of system handler 15 – SysTick** "0" denotes the highest priority and "3" denotes lowest priority
[23:22]	**PRI_14**	**Priority of system handler 14 – PendSV** "0" denotes the highest priority and "3" denotes lowest priority

5.3 Clock Controller and Power Management Unit (PMU)

The clock controller generates the clock sources for the whole device, including all AMBA interface modules and all peripheral clocks. Clock gating is provided on all peripheral clocks to minimize power consumption. The Power Management Unit (PMU) implements power control functions which can place the device into various power saving modes. The device will enter these various modes by requesting a power mode then requesting the Cortex-M0 to execute the WFI or the WFE instruction.

5.3.1 Clock Generator

The clock generator consists of 3 sources listed below:

- An internal programmable high frequency oscillator factory trimmed to provide frequencies of 49.152MHz, 32.768MHz and 36.864MHz to 1% accuracy. On H and M speed grade devices a doubled clock frequency is available on the CLK2X output.

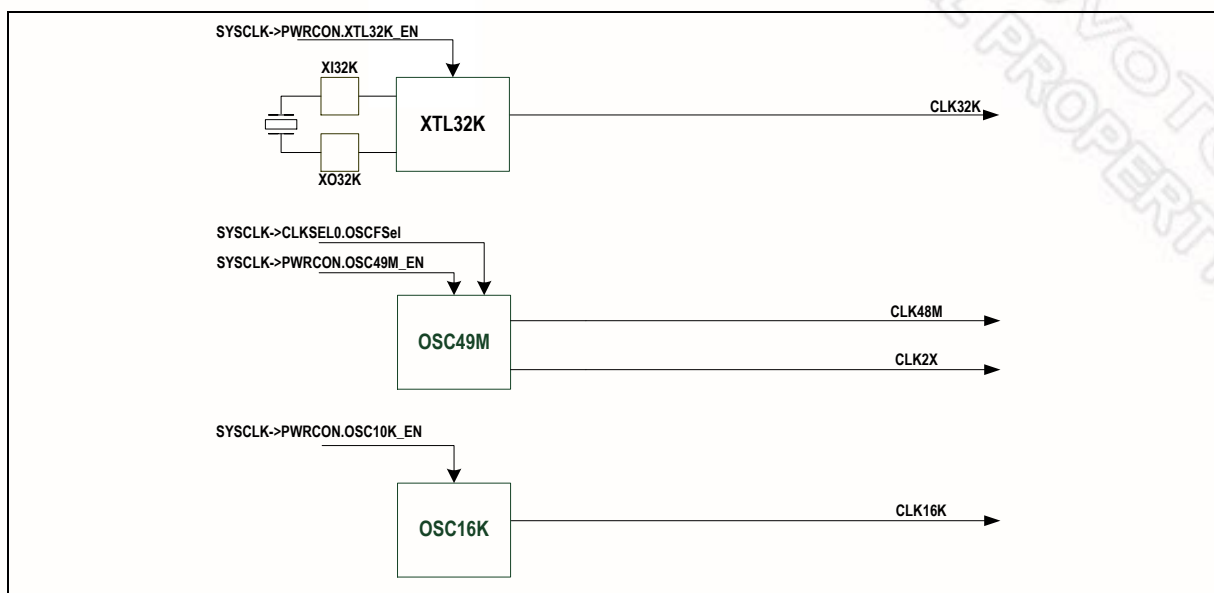- An external 32kHz crystal

- An internal low power16kHz oscillator.

Figure 5-3 Clock generator block diagram

5.3.2   System Clock & SysTick Clock

The system clock has 4clock sources from clock generator block. The clock source switch depends on the register HCLK_S(CLKSEL0[2:0]). The clock is then divided by HCLK_N+1 to produce the master clock for the device. Note that CLK2X source is only available on M and H speed grade devices of the series. Care must be taken to set correct FMC wait state control for frequencies greater than 50MHz for these devices.
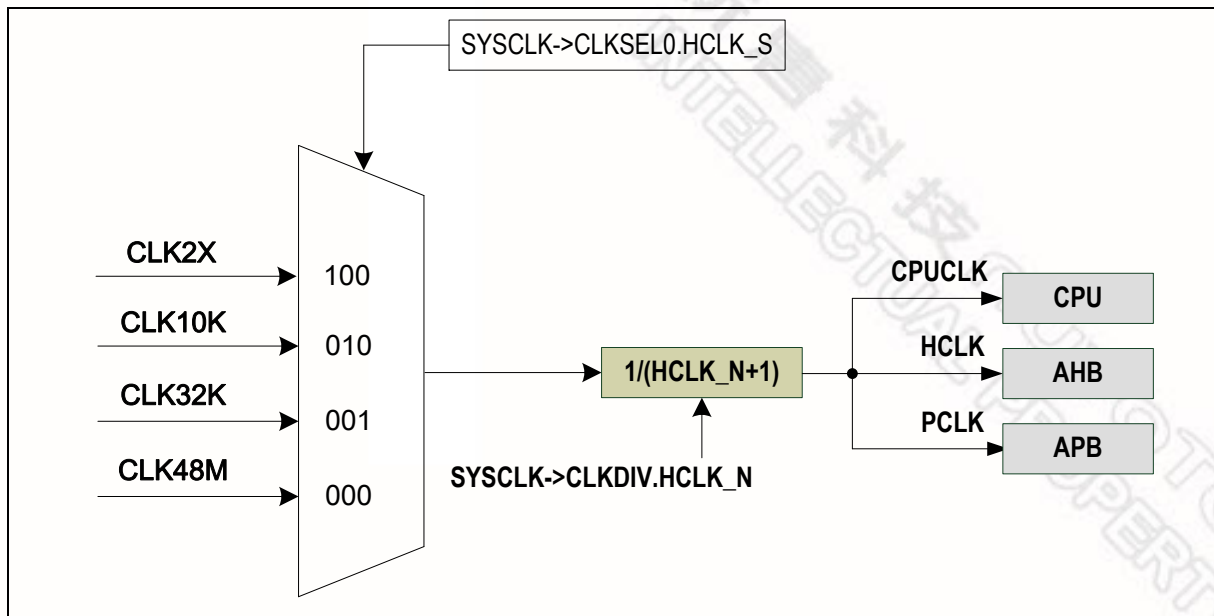

Figure 5-4 System Clock Block Diagram

The SysTick clock (STCLK) has five clock sources. The clock source switch depends on the setting of the register STCLK_S (CLKSEL0[5:3]).
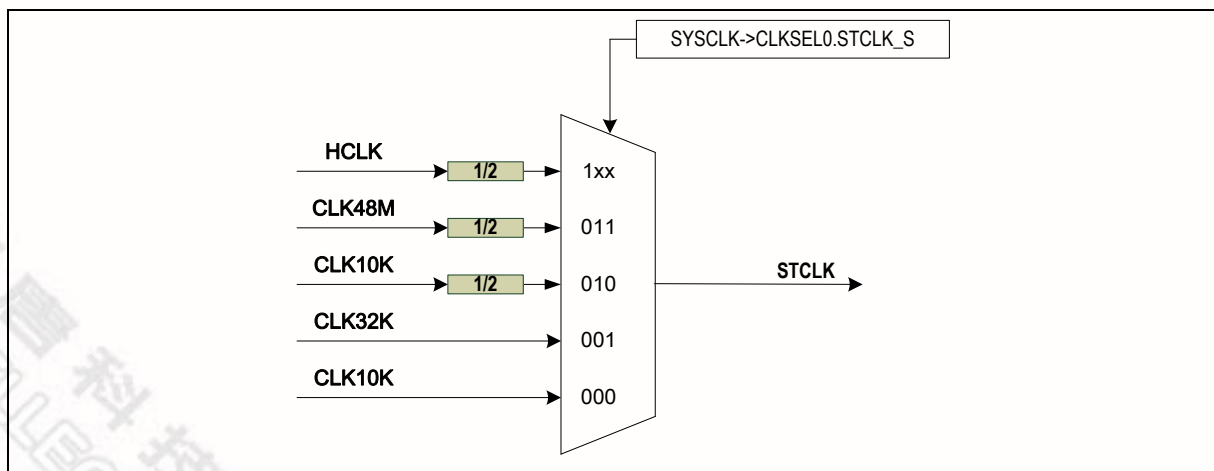

Figure 5-5 SysTick Clock Control Block Diagram

5.3.3 Peripheral Clocks

Each peripheral has a selectable clock gate. The register APBCLK determines whether the clock is active for each peripheral. In addition, the CLK_SLEEP register determines whether these clocks remain on during M0 sleep mode. Certain peripheral clocks have selectable sources these are controlled by the CLKSEL1 &CLKSEL2 register.

5.3.4 Power Management

The ISD93xx is equipped with a Power Management Unit (PMU) that implements a variety of power saving modes. There are four levels of power control with increasing functionality (and power consumption):

- Level0 : Deep Power Down (DPD)
- Level1 : Standby Power Down (SPD)
- Level2 : Deep Sleep
- Level3 : Sleep
- Level4 : Normal Operation

Within each of these levels there are further options to optimize power consumption.

5.3.4.1 Level0: Deep Power Down (DPD)

Deep Power Down (DPD) is the lowest power state the device can obtain. In this state there is no power provided to the logic domain and power consumption is only from the higher voltage chip supply domain. All logic state in the Cortex-M0 is lost as is contents of all RAM. All IO pins of the device are in a high impedance state. On a release from DPD the Cortex-M0 boots as if from a power-on reset. There are certain registers that can be interrogated to allow software to determine that previous state was a DPD state.

In DPD there are three ways to wake up the device:

1. A high to low transition on the WAKEUP pin.
2. A timed wakeup where the 16kHz oscillator is configured active and reaches a certain count.
3. A power cycle of main chip supply triggering a POR event.

To assist software in determining previous state of device before a DPD, a one-byte register is available PD_STATE[7:0] that can be loaded with a value to be preserved before issuing a DPD request.

To configure the device for DPD the user sets the following options:

- SYSCLK->PWRCON.PIN_ENB: If set to '1' then the WAKEUP pin is disabled and will not wake up the chip.
- SYSCLK->PWRCON.OSC16K_ENB: If set to '1' then the 16kHz oscillator will power down in DPD. No timed wakeup is possible.
- SYSCLK->PWRCON.TIMER_SEL: Each bit in this register will trigger a wakeup event after a certain number of OSC16K clock cycles.

When a WAKEUP event occurs the PMU will start the Cortex-M0 processor and execute the reset vector. The condition that generated the WAKEUP event can be interrogated by reading the registers SYSCLK->PWRCON.PIN_WAKE, SYSCLK->PWRCON.TIMER_WAKE and SYSCLK->PWRCON. POI_WAKE.

To enter the DPD state the user must set the register bit SYSCLK->PWRCON.DEEP_PD then execute a WFI or WFE instruction. Note that when debug interface is active, device will not enter DPD. Also once device enters DPD the debug interface will be inactive. It is possible that user could write code that makes it impossible to activate the debug interface and reprogram device, for instance if device re-enters DPD mode with insufficient time to allow an ICE tool to activate the SWD debug port.

Especially during development it is recommended that some checks are placed in the boot sequence to prevent device going to power down. A register bit, SYSCLK->DBGPD.DISABLE_PD is included for this purpose that will disable power down features. A check such as:

```
void Reset_Handler(void){
/*  check ICE_CLK and ICE_DAT to disable power down to the chip */
    if (SYSCLK->DBGPD.ICE_CLK == 0 && SYSCLK->DBGPD.ICE_DAT == 0)
        SYSCLK->DBGPD.DISABLE_PD = 1;
    __main();
}
```

Can check the SWD pin state on boot and prevent power down from occurring.

5.3.4.2   *Level1: Standby Power Down (SPD) mode.*

Standby Power Down mode is the lowest power state that some logic operation can be performed. In this mode power is removed from the majority of the core logic, including the Cortex-M0 and main RAM. A low power standby reference is enabled however that supplies power to a subset of logic including the IO ring, GPIO control, RTC module, 32kHz Crystal Oscillator, Brownout Detector and a 256Byte Standby RAM.

In Standby mode there are three ways to wake up the device:

1. An interrupt from the GPIO block, for instance a pin transition.

2. An interrupt from the RTC module, for instance an alarm or timer event.

3. A power cycle of main chip supply triggering a POR event.

When a wake up event occurs the PMU will start the Cortex-M0 processor and execute the reset vector. Software can determine whether the device woke up from SPD by interrogating the register bit SYSCLK->PFLAGCON.PD_FLAG.

To enter the SPD state the user must set the register bit SYSCLK->PWRCON.PD then execute a WFI or WFE instruction. Note that when debug interface is active, device will not enter SPD. Also once device enters SPD the debug interface will be inactive.

5.3.4.3   *Level2: Deep Sleep mode.*

The Deep Sleep mode is the lowest power state where the Cortex-M0 and all logic state are preserved. In Deep Sleep mode the CLK48M oscillator is shut down and a low speed oscillator is selected, if CLK32K is active this source is selected, if not then CLK16K is enabled and selected. All clocks to the Cortex-M0 core are gated eliminating dynamic power in the core. Clocks to peripheral are gated according to the CLK_SLEEP register, note however that HCLK is operating at a low frequency and CLK48M is not available. Deep Sleep mode is entered by setting System Control register bit 2: SCB->SCR |= (1UL << 2) and executing a WFI/WFE instruction. Software can determine whether the device woke up from Deep Sleep by interrogating the register bit SYSCLK->PFLAGCON.DS_FLAG.

5.3.4.4   *Level3: Sleep mode.*

The Sleep mode gates all clocks to the Cortex-M0 eliminating dynamic power in the core. In addition, clocks to peripherals are gated according to the CLK_SLEEP register. The mode is entered by executing a WFI/WFE instruction and is released when an event occurs. Peripheral functions, including PDMA can be continued while in Sleep mode. Using this mode power consumption can be minimized while waiting for events such as a PDMA operation collecting data from the ADC, once PDMA has finished the core can be woken up to process the data.

5.3.4.5   *Clock Control Register Map*

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

Register	Offset	R/W	Description	Reset Value	Reference
CLK_BA = 0x5000_0200					
PWRCON	CLK_BA + 00	R/W	System Power Control Register	0x0000_001x	Table 5-30
AHBCLK	CLK_BA + 04	R/W	AHB Device Clock Enable Control Register	0x0000_0001	Table 5-32
APBCLK	CLK_BA + 08	R/W	APB Device Clock Enable Control Register	0x0000_0000	Table 5-33
DPDSTATE	CLK_BA + 0C	R/W	DPD State Register	0x0000_xx00	Table 5-34
CLKSEL0	CLK_BA + 10	R/W	Clock Source Select Control Register 0	0x0000_0038	Table 5-35
CLKSEL1	CLK_BA + 14	R/W	Clock Source Select Control Register 1	0x3300_771F	Table 5-36
CLKDIV	CLK_BA + 18	R/W	Clock Divider Register	0x0000_0000	Table 5-37
CLKSEL2	CLK_BA + 1C	R/W	Clock Source Select Control Register 2	0xFFFF_FFFx	Table 5-38
CLKSLEEP	CLK_BA + 20	R/W	Sleep Clock Source Select Register	0xFFFF_FFFF	Table 5-39
PDFLAG	CLK_BA + 24	R/W	Power Down Flag Register	0x0000_0000	Table 5-40
DBGPD	CLK_BA + 28	R/W	Debug Port Power Down Disable Register	0x0000_00XX	Table 5-41

5.3.5 Clock Control Register Description

System Power Control Register （PWRCON）

This is a protected register, to write to register, first issue the unlock sequence (see Protected Register Lock Key Register (REGLOCK))

Register	Offset	R/W	Description	Reset Value
PWRCON	CLK_BA + 00	R/W	System Power Control Register	0xXX00_0006

Table 5-30 System Power Control Register （PWRCON, address 0x5000_0200）

31	30	29	28	27	26	25	24
TIMER_SEL_RD				reserved	POI_WAKE	TIMER_WAKE	PIN_WAKE
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TIMER_SEL				Reserved		OSC16K_ENB	PIN_ENB
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
	IO_STATE	RELEASE_IO	HOLD_IO	DEEP_PD	STANDBY_PD	STOP	Reserved
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				OSC16K_EN	OSC49M_EN	XTL32K_EN	Reserved

Table 5-31 System Power Control Register (PWRCON, address 0x5000_0200) Bit Description.

Bits	Descriptions	
[31:28]	**TIMER_SEL_RD**	Read-Only. Read back of the current WAKEUP timer setting. This value is updated with TIMER_SEL upon entering DPD mode.
[27]	**reserved**	
[26]	**POI_WAKE**	Read Only. This flag indicates that wakeup of device was requested with a power-on reset. Flag is cleared when DPD mode is entered or any of the DPD bits of RSTSRC register (RSTSRC[10:8]) are cleared.
[25]	**TIMER_WAKE**	Read Only. This flag indicates that wakeup of device was requested with TIMER count of the 16Khz oscillator. Flag is cleared when DPD mode is entered or any of the DPD bits of RSTSRC register (RSTSRC[10:8]) are cleared.
[24]	**PIN_WAKE**	Read Only. This flag indicates that wakeup of device was requested with a high to low transition of the WAKEUP pin. Flag is cleared when DPD mode is entered or any of the DPD bits of RSTSRC register (RSTSRC[10:8]) are cleared.
[23:20]	**TIMER_SEL**	Select WAKEUP Timer: TIMER_SEL[0]=1: WAKEUP after 128 OSC16K clocks (8 ms) TIMER_SEL[1]=1: WAKEUP after 256 OSC16K clocks (16 ms) TIMER_SEL[2]=1: WAKEUP after 512 OSC16K clocks (32 ms) TIMER_SEL[3]=1: WAKEUP after 1024 OSC16K clocks (64ms)

[19:18]	Reserved	
[17]	OSC16K_ENB	Determines whether OSC16K is enabled in DPD mode. 0=enabled, 1=disabled in DPD. If OSC16K is disabled, device cannot wake from DPD with TIMER_SEL delay.
[16]	PIN_ENB	Determines whether WAKEUP pin is enabled in DPD mode. 0=enabled, 1=disabled.
[15]	Reserved	
[14]	IO_STATE	'1': IO held from SPD. '0': IO released.
[13]	RELEASE_IO	Write '1' to this bit to release IO state after exiting SPD if hold request was made with the HOLD_IO bit.
[12]	HOLD_IO	When entering SPD mode, IO state is automatically held. If this bit is set to '1' then this sate upon resuming full power mode will be hold until the RELEASE_IO bit is written '1'
[11]	DEEP_PD	Deep Power Down (DPD) bit. Set to '1' and issue WFI/WFE instruction to enter DPD mode.
[10]	STANDBY_PD	Standby Power Down (SPD) bit. Set to '1' and issue WFI/WFE instruction to enter SPD mode.
[9]	STOP	STOP mode bit. Set to '1' and issue WFI/WFE instruction to enter STOP mode.
[8:4]	Reserved	
[3]	OSC16K_EN	OSC16K Oscillator enable bit. 1=enable (default), 0=disable
[2]	OSC49M_EN	OSC49M Oscillator enable bit. 1=enable (default), 0=disable
[1]	XTAL32K_EN	External 32.768 kHz Crystal Enable bit. 1=enable, 0=disable (default).
[0]	Reserved	

AHB Device Clock Enable Control Register （**AHBCLK**）

These register bits are used to enable/disable the clock source for AHB (Advanced High-Performance Bus) blocks. This is a protected register, to write to register, first issue the unlock sequence (see Protected Register Lock Key Register (REGLOCK))

Register	Offset	R/W	Description	Reset Value
AHBCLK	CLK_BA + 04	R/W	AHB Device Clock Enable Control Register	0x0000_0005

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

Table 5-32 AHB Device Clock Enable Register (AHBCLK, address 0x5000_0204) Bit Description.

Bits	Symbol	Description
[31:3]	Reserved	Reserved
[2]	ISP_EN	**Flash ISP Controller Clock Enable Control.** 1 = To enable the Flash ISP engine clock. 0 =To disable the Flash ISP engine clock.
[1]	PDMA_EN	**PDMA Controller Clock Enable Control.** 1 = To enable the PDMA engine clock. 0 =To disable the PDMA engine clock
[0]	CPU_EN	CPU Clock Enable (HCLK) must be left as '1' for normal operation.

APB Device Clock Enable Control Register （**APBCLK**）

These register bits are used to enable/disable clocks for APB (Advanced Peripheral Bus) peripherals. To enable the clocks write '1' to the appropriate bit. To reduce power consumption and disable the peripheral, write '0' to the appropriate bit.

Register	Offset	R/W	Description	Reset Value
APBCLK	CLK_BA + 08	R/W	APB Device Clock Enable Control Register	0x0000_000x

31	30	29	28	27	26	25	24
PWM54_EN	ANA_EN	I2S_EN	ADC_EN	Reserved	SBRAM_EN	Reserved	Reserved
23	22	21	20	19	18	17	16
Reserved	ACMP_EN	PWM32_EN	PWM01_EN	Reserved	BIQALC_EN	Reserved	UART0_EN
15	14	13	12	11	10	9	8
Reserved	Reserved	DPWM_EN	SPI0_EN	Reserved	Reserved	Reserved	I2C0_EN
7	6	5	4	3	2	1	0
TMR1_EN	TMR0_EN	RTC_EN	WDG_EN	Reserved	Reserved	Reserved	Reserved

Table 5-33 APB Device Clock Enable Control Register (APBCLK, address 0x5000_0208) Bit Description.

Bits	Symbol	Description
[30]	ANA_EN	Analog Block Clock Enable.
[29]	I2S_EN	I2S Clock Enable Control.
[28]	ADC_EN	Audio Analog-Digital-Converter (ADC) Clock Enable Control.
[26]	SBRAM_EN	Standby RAM Clock Enable Control.
[22]	ACMP_EN	Analog Comparator Clock Enable Control.
[21]	PWM32_EN	PWM Block Clock Enable Control.
[20]	PWM01_EN	PWM Block Clock Enable Control.
[18]	BIQALC_EN	Biquad filter and Automatic Level Control block clock enable.
[16]	UART0_EN	UART0 Clock Enable Control.
[13]	DPWM_EN	Differential PWM Speaker Driver Clock Enable Control.
[12]	SPI0_EN	SPI0 Clock Enable Control.
[8]	I2C0_EN	I2C0 Clock Enable Control.
[7]	TMR1_EN	Timer1 Clock Enable Control

[6]	TMR0_EN	Timer0 Clock Enable Control
[5]	RTC_EN	Real-Time-Clock APB Interface Clock Control.
[4]	WDG_EN	Watchdog Clock Enable Control

DPD State Register （DPDSTATE）

The Deep Power Down State register is a user settable register that is preserved during Deep Power Down (DPD). Software can use this register to store a single byte during a DPD event. The DPD_STATE_RD register reads back the current state of the DPDSTATE register. To write to this register, set desired value in the DPD_STATE_WR register, this value will be latched in to the DPDSTATE register on next DPD event.

Register	Offset	R/W	Description	Reset Value
DPDSTATE	CLK_BA + 0C	R/W	Deep Power Down State Register	0x0000_XX00

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
DPD_STATE_RD							
7	6	5	4	3	2	1	0
DPD_STATE_WR							

Table 5-34DPD State Register (DPDSTATE, address 0x5000_020C) Bit Description.

Bits	Symbol	Description
[15:8]	DPD_STATE_RD	Read back of DPDSTATE register. This register was preserved from last DPD event .
[7:0]	DPD_STATE_WR	To set the DPDSTATE register, write value to this register. Data is latched on next DPD event.

Clock Source Select Control Register 0 （**CLKSEL0**）

Register	Offset	R/W	Description	Reset Value
CLKSEL0	CLK_BA + 10	R/W	Clock Source Select Control Register 0	0x0000_0038

7	6	5	4	3	2	1	0
OSCFSel		STCLK_S			HCLK_S		

Table 5-35 Clock Source Select Register 0 (CLKSEL0, address 0x5000_0210) Bit Description.

Bits	Descriptions	
[7:6]	**OSCFSel**	**OSC48M Frequency Select.** Determines which trim setting to use for OSC48M internal oscillator. Oscillator is factory trimmed within 1% to: 0: 49.152MHz (Default) 1: 32.768MHz 2: 36.864MHz
[5:3]	**STCLK_S**	**MCU Cortex_M0 SysTick clock source select.** These bits are protected, to write to bits first perform the unlock sequence (see Protected Register Lock Key Register (REGLOCK)) 000 = clock source from 16kHz internal clock 001 = clock source from external 32kHz crystal clock 010 = clock source from 16kHz internaloscillator divided by 2 011 = clock source from OSC49M internal oscillator divided by 2 1xx = clock source from HCLK÷2 **(Default)** **Note that to use STCLK_S as source of SysTic timer the CLKSRC bit of SysTick->CTRL must be set to 0.**
[2:0]	**HCLK_S**	**HCLK clock source select.** **Note:** 1. Ensure that related clock sources (pre-select and new-select) are enabled before updating register. 2. These bits are protected, to write to bits first perform the unlock sequence (see Protected Register Lock Key Register (REGLOCK)) 000 = clock source from internal OSC48M oscillator. 001 = clock source from external 32kHz crystal clock 010 = clock source from internal16kHz oscillator clock 100= CLK2X a frequency doubled output of OSC48M, only available on M and H speed grade devices. Others = reserved

Clock Source Select Control Register 1（CLKSEL1）

Clock multiplexors are a glitch free design to ensure smooth transitions between asynchronous clock sources. As such, both the current clock source and the target clock source must be enabled for switching to occur. Beware when switching from a low speed clock to a high speed clock that low speed clock remains on for at least one period before disabling.

Register	Offset	R/W	Description	Reset Value
CLKSEL1	CLK_BA + 14	R/W	Clock Source Select Control Register 1	0x3300_771F

31	30	29	28	27	26	25	24
Reserved		PWM01_S		Reserved			
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved	TMR1_S			Reserved	TMR0_S		
7	6	5	4	3	2	1	0
Reserved			DPWM_S			WDG_S	

Table 5-36 Clock Source Select Register 1 (CLKSEL1, address 0x5000_0214) Bit Description.

Bits	Descriptions	
[29:28]	**PWM01_S**	**PWM0 and PWM1 clock source select.** PWM0 and PWM1 uses the same clock source, and prescaler 00 = clock source from internal 16kHz oscillator 01 = clock source from external 32kHz crystal clock 10 = clock source from HCLK 11 = clock source from internal OSC48M oscillator clock
[14:12]	**TMR1_S**	**TIMER1 clock source select.** 000 = clock source from internal 16kHz oscillator 001 = clock source from external 32kHz crystal clock 010 = clock source from HCLK 011 = clock source from external pin (GPIOA[15]) 1xx = clock source from internal OSC48M oscillator clock

[10:8]	TMR0_S	**TIMER0 clock source select.** 000 = clock source from internal 16kHz oscillator 001 = clock source from external 32kHz crystal clock 010 = clock source from HCLK 011 = clock source from external pin (GPIOA[14]) 1xx = clock source from internal OSC48M oscillator clock
[4]	DPWM_S	**Differential Speaker Driver PWM Clock Source Select.** 0 = OSC48M clock 1 = 2x OSC48M clock
[1:0]	WDG_S	**WDG CLK clock source select.** 00 = clock source from internal OSC48M oscillator clock 01 = clock source from external 32kHz crystal clock 10 = clock source from HCLK/2048 clock 11 = clock source from internal 16kHz oscillator clock

Clock Divider Register (CLKDIV)

Register	Offset	R/W	Description	Reset Value
CLKDIV	CLK_BA + 18	R/W	Clock Divider Number Register	0x0000_0000

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

Table 5-37 Clock Divider Register (CLKDIV, address 0x5000_0218) Bit Description.

Bits	Symbol	Description
[23:16]	**ADC_N**	**ADC clock divide number from ADC clock source** The ADC clock frequency = (ADC clock source frequency) / (ADC_N + 1)
[11:8]	**UART_N**	**UART clock divide number from UART clock source** The UART clock frequency = (UART clock source frequency) / (UART_N + 1)
[3:0]	**HCLK_N**	**HCLK clock divide number from HCLK clock source** The HCLK clock frequency = (HCLK clock source frequency) / (HCLK_N + 1)

Clock Source Select Control Register 2（CLKSEL2）

Before changing clock source, ensure that related clock sources (pre-select and new-select) are enabled.

Register	Offset	R/W	Description	Reset Value
CLKSEL2	CLK_BA + 1C	R/W	Clock Source Select Control Register 2	0xFFFF_FFFx

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						I2S_S	

Table 5-38 Clock Source Select Control Register 2 (CLKSEL2, address 0x5000_021C) Bit Description.

Bits	Descriptions	
[31:2]	Reserved	Reserved
[1:0]	**I2S_S**	**I2S Clock Source Select** 00 = clock source from internal 16kHz oscillator 01 = clock source from external 32kHz crystal clock 10 = clock source from HCLK 11 = clock source from internal OSC48M oscillator clock

Sleep Clock Enable Control Register （CLKSLEEP）

These register bits are used to enable/disable clocks during sleep mode. It works in conjunction with **AHBCLK** and **APBCLK** clock register to determine whether a clock source remains active during CPU Sleep mode. For a clock to be active in Sleep mode, the appropriate clock must be enabled in the **AHBCLK** or **APBCLK** register **and** the bit must also be enabled in the **CLKSLEEP** register. In other words, to disable a clock in Sleep mode, write '0' to the appropriate bit in **CLKSPEEP**.

Register	Offset	R/W	Description	Reset Value
CLKSLEEP	CLK_BA + 20	R/W	APB Device Clock Enable Control Register	0xFFFF_FFFF

Table 5-39 Sleep Clock Enable Control Register (CLKSLEEP, address 0x5000_0220). Bit Description.

31	30	29	28	27	26	25	24
Reserved	ANA_EN	I2S_EN	ADC_EN	Reserved	SBRAM_EN	Reserved	Reserved
23	22	21	20	19	18	17	16
Reserved	ACMP_EN	Reserved	PWM01_EN	CRC_EN	BIQALC_EN	Reserved	UART0_EN
15	14	13	12	11	10	9	8
Reserved	Reserved	DPWM_EN	SPI0_EN	Reserved	Reserved	Reserved	I2C0_EN
7	6	5	4	3	2	1	0
TMR1_EN	TMR0_EN	RTC_EN	WDG_EN	Reserved	ISP_EN	PDMA_EN	CPU_EN

Bits	Symbol	Description
[30]	ANA_EN	Analog Block Sleep Clock Enable.
[29]	I2S_EN	I2S Sleep Clock Enable Control.
[28]	ADC_EN	Audio Analog-Digital-Converter (ADC) Sleep Clock Enable Control.
[26]	SBRAM_EN	Standby RAM Sleep Clock Enable Control.
[22]	ACMP_EN	Analog Comparator Sleep Clock Enable Control.
[20]	PWM01_EN	PWM Block Sleep Clock Enable Control.
[19]	CRC_EN	Cyclic Redundancy Check Sleep Block clock enable.
[18]	BIQALC_EN	Bi-quad filter/ALC block Sleep Clock Enable Control.
[16]	UART0_EN	UART0 Sleep Clock Enable Control.
[13]	DPWM_EN	Differential PWM Speaker Driver Sleep Clock Enable Control.
[12]	SPI0_EN	SPI0 Sleep Clock Enable Control.
[8]	I2C0_EN	I2C0 Sleep Clock Enable Control.
[7]	TMR1_EN	Timer1 Sleep Clock Enable Control
[6]	TMR0_EN	Timer0 Sleep Clock Enable Control
[5]	RTC_EN	Real-Time- Sleep Clock APB Interface Clock Control.
[4]	WDG_EN	Watchdog Sleep Clock Enable Control
[2]	ISP_EN	Flash ISP Controller Sleep Clock Enable Control
[1]	PDMA_EN	PDMA Controller Sleep Clock Enable Control.
[0]	CPU_EN	CPU Clock Sleep Enable (HCLK) must be left as '1' for normal operation.

Power State Flag Register (PFLAGCON)

Register	Offset	R/W	Description	Reset Value
PFLAGCON	CLK_BA+ 24	R/W	Power State Flag Register	0x0000_0000

7	6	5	4	3	2	1	0
		Reserved			PD_FLAG	STOP_FLAG	DS_FLAG

Table 5-40 Power State Flag Register (PFLAGCON, address 0x5000_0224) Bit Description.

Bits	Descriptions	
[2]	**PD_FLAG**	This flag is set if core logic was powered down to Standby (SPD). Write '1' to clear flag.
[1]	**STOP_FLAG**	This flag is set if core logic was stopped but not powered down. Write '1' to clear flag.
[0]	**DS_FLAG**	This flag is set if core logic was placed in Deep Sleep mode. Write '1' to clear flag.

Debug Power Down Register (DBGPD)

Register	Offset	R/W	Description	Reset Value
DBGPD	CLK_BA+ 28	R/W	Debug Power Down Register	0x0000_000X

Table 5-41 Debug Power Down Register (DBGPD, address 0x5000_0228) Bit Description.

7	6	5	4	3	2	1	0
ICE_DAT	ICE_CLK		Reserved				DISABLE_PD

Bits	Descriptions	
[7]	**ICE_DAT**	Read Only. Current state of ICE_DAT pin.
[6]	**ICE_CLK**	Read Only. Current state of ICE_CLK pin.
[0]	**DISABLE_PD**	1 = Disable power down requests. 0 = enable power down requests.

5.4 General Purpose I/O

5.4.1 Overview and Features

Up to 32 General Purpose I/O pins are available on the ISD93xx series. These are shared peripheral special function pins under control of the alternate configuration registers. These 32 pins are arranged in 2 ports named with GPIOA, and GPIOB. Each one of the 32 pins is independent and has corresponding register bits to control the pin mode function and data.

The I/O type of each GPIO pin can be independently configured as an input, output, open-drain or in a quasi-bidirectional mode. Upon chip reset, all GPIO pins are configured in quasi-bidirectional mode and port data register resets high.

When device is in deep power down (DPD) mode, all GPIO pins become high impedance.

GPIO can generate interrupt signals to the core as either level sensitive or edge sensitive inputs. Edge sensitive inputs can also be de-bounced.

In quasi-bidirectional mode, each GPIO pin has a weak pull-up resistor which is approximately 110KΩ~300KΩ for V_{DD} from 5.0V to 2.4V.

Each pin can generate and interrupt exception to the Cortex M0 core. GPIOB[0] and GPIOB[1] can generate interrupts to system interrupt number IRQ2 and IRQ3 respectively (see Table 5-15). All other GPIO generate and exception to interrupt number IRQ4.

5.4.2 GPIO I/O Modes

The I/O mode of each GPIO pin is controlled by the register GPIO*x*->PMD. *(x=A or B).* Each pin has two bits of control giving four possible states:

5.4.2.1 *Input Mode*

For GPIO*x*->PMD.PMD*n* = 00b the GPIO*x* port [*n*] pin is in Input Mode. The GPIO pin is in a tri-state (high impedance) condition without output drive capability. The GPIO*x*->PIN value reflects the status of the corresponding port pins.

5.4.2.2 *Output Mode*

For GPIO*x*->PMD.PMD*n* = 01b the GPIO*x* port [*n*] pin is in Output Mode. The GPIO pin supports a digital output function with current source/sink capability. The bit value in the corresponding bit [n] of GPIO*x*->DOUT is driven to the pin.
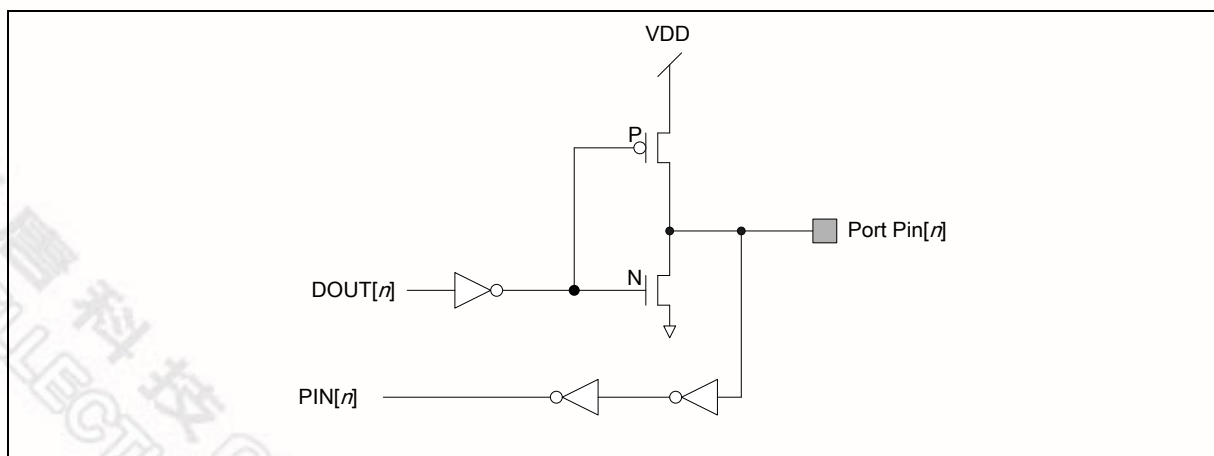

Figure 5-6 Output Mode: Push-Pull Output

5.4.2.3   *Open-Drain Mode*

For GPIO*x*->PMD.PMD*n* = 10b the GPIO*x* port [*n*] pin is in Open-Drain mode. The GPIO pin supports a digital output function but only with sink current capability, an additional pull-up resister is needed for defining a high state. If the bit value in the corresponding bit [n] of GPIOx_DOUT is "0", pin is driven low. If the bit value in the corresponding bit [n] of GPIOx_DOUT is "1", the pin state is defined by the external load on the pin.
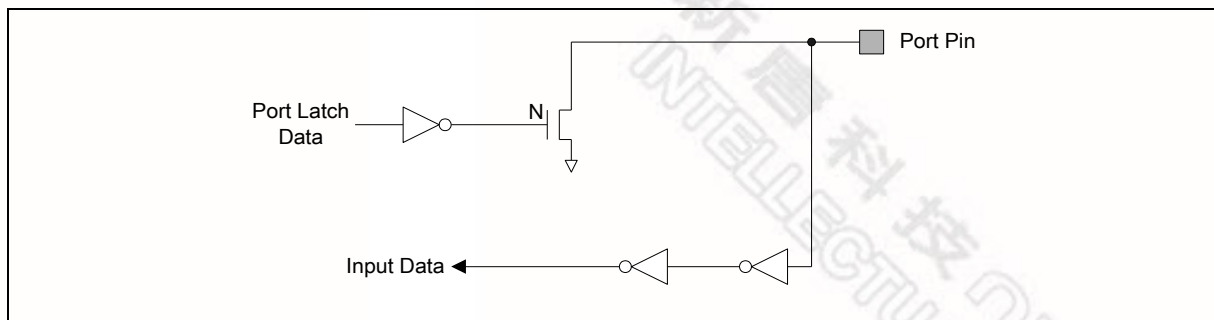

Figure 5-7Open-Drain Output

5.4.2.4   *Quasi-bidirectional Mode Explanation*

For GPIO*x*->PMD.PMD*n* = 11b the GPIO*x* port [*n*] pin is in Quasi-bidirectional mode and the I/O pin supports digital output and input function where the source current is only between 30-200uA.Before input function is performed the corresponding bit in GPIOx_DOUT must be set to 1. The quasi-bidirectional output is common on the 80C51 and most of its derivatives. If the bit value in the corresponding bit [n] of GPIOx_DOUT is "0", the pin will drive a "low" output to the pin. If the bit value in the corresponding bit [n] of GPIOx_DOUT is "1", the pin will check the pin value. If pin value is high, no action is taken. If pin state is low, then pin will drive a strong high for2 clock cycles. After this the pin has an internal pull-up resistor connected. Note that the source current capability in quasi-bidirectional mode is approximately200uA to 30uA for VDD form 5.0V to 2.4V.
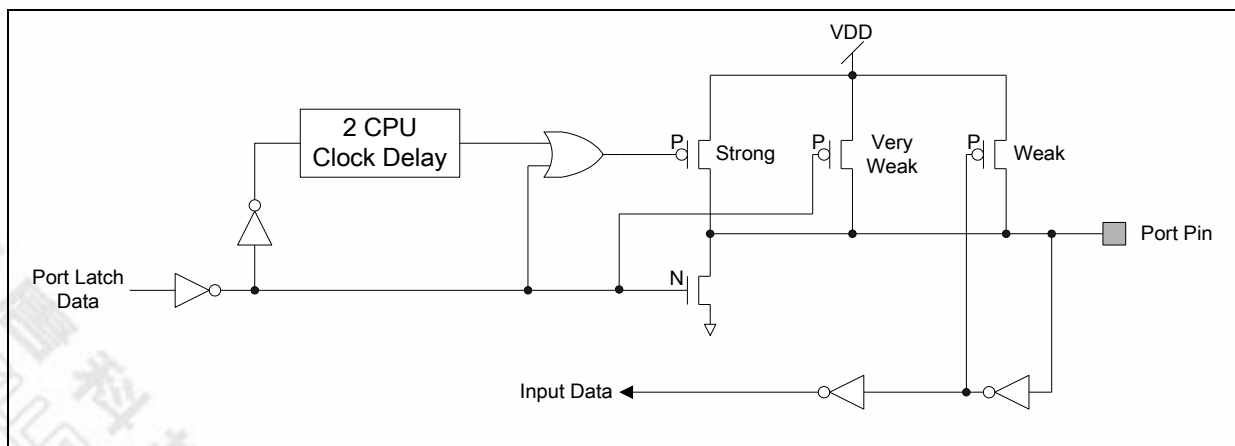

Figure 5-8Quasi-bidirectional GPIO Mode

5.4.3 GPIO Control Register Map

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

Register	Offset	R/W	Description	Reset Value	Reference
GP_BA = 0x5000_4000					
GPIOA_PMD	GP_BA+0x000	R/W	GPIO Port A Bit Mode Control	0xFFFF_FFFF	Table 5-42
GPIOA_INDIS	GP_BA+0x004	R/W	GPIO Port A Bit Input Disable	0x0000_0000	Table 5-43
GPIOA_DOUT	GP_BA+0x008	R/W	GPIO Port A Data Output Value	0x0000_FFFF	Table 5-44
GPIOA_DMASK	GP_BA+0x00C	R/W	GPIO Port A Data Output Write Mask	0x0000_0000	Table 5-45
GPIOA_PIN	GP_BA+0x010	R	GPIO Port A Pin Value	0x0000_XXXX	Table 5-46
GPIOA_DBEN	GP_BA+0x014	R/W	GPIO Port A De-bounce Enable	0x0000_0000	Table 5-47
GPIOA_IMD	GP_BA+0x018	R/W	GPIO Port A Interrupt Mode Control	0x0000_0000	Table 5-48
GPIOA_IEN	GP_BA+0x01C	R/W	GPIO Port A Interrupt Enable	0x0000_0000	Table 5-49
GPIOA_ISRC	GP_BA+0x020	R/WC	GPIO Port A Interrupt Source Flag	0xXXXX_XXXX	Table 5-50
GPIOB_PMD	GP_BA+0x040	R/W	GPIO Port B Bit Mode Enable	0xFFFF_FFFF	Table 5-42
GPIOB_INDIS	GP_BA+0x044	R/W	GPIO Port B Bit Input Disable	0x0000_0000	Table 5-43
GPIOB_DOUT	GP_BA+0x048	R/W	GPIO Port B Data Output Value	0x0000_FFFF	Table 5-44
GPIOB_DMASK	GP_BA+0x04C	R/W	GPIO Port B Data Output Write Mask	0x0000_0000	Table 5-45
GPIOB_PIN	GP_BA+0x050	R	GPIO Port B Pin Value	0x0000_XXXX	Table 5-46
GPIOB_DBEN	GP_BA+0x054	R/W	GPIO Port B De-bounce Enable	0x0000_0000	Table 5-47
GPIOB_IMD	GP_BA+0x058	R/W	GPIO Port B Interrupt Mode Control	0x0000_0000	Table 5-48
GPIOB_IEN	GP_BA+0x05C	R/W	GPIO Port B Interrupt Enable	0x0000_0000	Table 5-49
GPIOB_ISRC	GP_BA+0x060	R/WC	GPIO Port B Interrupt Source Flag	0xXXXX_XXXX	Table 5-50
DBNCECON	GP_BA+0x180	R/W	De-bounce Cycle Control	Re0x0000_0020	Table 5-51

5.4.4 GPIO Control Register Description

GPIO Port [A/B] I/O Mode Control (GPIOx_PMD)

Register	Offset	R/W	Description	Reset Value
GPIOA_PMD	GP_BA+0x000	R/W	GPIO Port A Pin I/O Mode Control	0xFFFF_FFFF
GPIOB_PMD	GP_BA+0x040	R/W	GPIO Port B Pin I/O Mode Control	0xXXXX_FFFF

Table 5-42 GPIO Mode Control Register

31	30	29	28	27	26	25	24
PMD15		PMD14		PMD13		PMD12	
23	22	21	20	19	18	17	16
PMD11		PMD10		PMD9		PMD8	
15	14	13	12	11	10	9	8
PMD7		PMD6		PMD5		PMD4	
7	6	5	4	3	2	1	0
PMD3		PMD2		PMD1		PMD0	

Bits	Descriptions	
[2n+1 :2n]	**PMD***n*	**GPIO*x* I/O Pin[*n*] Mode Control** Determine each I/O type of GPIO*x* pins 00 = GPIO port [n] pin is in INPUT mode. 01 = GPIO port [n] pin is in OUTPUT mode. 10 = GPIO port [n] pin is in Open-Drain mode. 11 = GPIO port [n] pin is in Quasi-bidirectional mode.

GPIO Port [A/B] Input Disable (GPIOx_INDIS)

Register	Offset	R/W	Description	Reset Value
GPIOA_INDIS	GP_BA+0x004	R/W	GPIO Port A Pin Input Disable	0x0000_0000
GPIOB_INDIS	GP_BA+0x044	R/W	GPIO Port B Pin Input Disable	0x0000_0000

Table 5-43 GPIO Input Disable Register

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

Bits	Descriptions	
[31:16]	**INDIS**	**INDIS: GPIOx Pin[n] OFF digital input path Enable** 1 = Disable IO digital input path (low leakage mode) 0 = Enable IO digital input path (Default)
[15:0]	Reserved	Reserved

GPIO Port [A/B] Data Output Value (GPIOx_DOUT)

Register	Offset	R/W	Description	Reset Value
GPIOA_DOUT	GP_BA+0x008	R/W	GPIO Port A Data Output Value	0x0000_FFFF
GPIOB_DOUT	GP_BA+0x048	R/W	GPIO Port B Data Output Value	0x0000_XXFF

Table 5-44 GPIO Data Output Register (GPIOx_DOUT)

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
DOUT[15:8]							
7	6	5	4	3	2	1	0
DOUT[7:0]							

Bits	Descriptions	
[*n*]	**DOUT[*n*]**	**GPIOx Pin[n] Output Value** Each of these bits controls the status of a GPIO pin when the GPIO pin is configured as output, open-drain or quasi-bidirectional mode. 1 = GPIO port [A/B] Pin[*n*] will drive High if the corresponding output mode bit is set. 0 = GPIO port [A/B] Pin[*n*] will drive Low if the corresponding output mode bit is set.

GPIO Port [A/B] Data Output Write Mask (GPIOx _DMASK)

Register	Offset	R/W	Description	Reset Value
GPIOA_DMASK	GP_BA+0x00C	R/W	GPIO Port A Data Output Write Mask	0xXXXX_0000
GPIOB_DMASK	GP_BA+0x04C	R/W	GPIO Port B Data Output Write Mask	0xXXXX_0000

Table 5-45 GPIO Data Output Write Mask Register (GPIOx_MASK)

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
DMASK[15:8]							
7	6	5	4	3	2	1	0
DMASK[7:0]							

Bits	Descriptions	
[*n*]	**DMASK[*n*]**	**Port [A/B] Data Output Write Mask** These bits are used to protect the corresponding register of GPIOx_DOUT bit[*n*] . When set the DMASK bit[*n*] to "1", the corresponding DOUT*n* bit is write protected. 0 = The corresponding GPIO_DOUT[*n*] bit can be updated 1 = The corresponding GPIO_DOUT[*n*] bit is read only

GPIO Port [A/B] Pin Value (GPIOx _PIN)

Register	Offset	R/W	Description	Reset Value
GPIOA_PIN	GP_BA+0x010	R	GPIO Port A Pin Value	0x0000_XXXX
GPIOB_PIN	GP_BA+0x050	R	GPIO Port B Pin Value	0x0000_XXXX

Table 5-46 GPIO PIN Value Register (GPIOx_PIN)

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
PIN[15:8]							
7	6	5	4	3	2	1	0
PIN[7:0]							

Bits	Descriptions	
[n]	**PIN[n]**	**Port [A/B] Pin Values** The value read from each of these bit reflects the actual status of the respective GPIO pin

GPIO Port [A/B] De-bounce Enable (GPIOx _DBEN)

Register	Offset	R/W	Description	Reset Value
GPIOA_DBEN	GP_BA+0x014	R/W	GPIO Port A De-bounce Enable	0xXXXX_0000
GPIOB_DBEN	GP_BA+0x054	R/W	GPIO Port B De-bounce Enable	0xXXXX_0000

Table 5-47 GPIO Debounce Enable Register (GPIOx_DBEN)

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
DBEN[15:8]							
7	6	5	4	3	2	1	0
DBEN[7:0]							

Bits	Descriptions	
[*n*]	**DBEN[*n*]**	**Port [A/B] Input Signal De-bounce Enable** DBEN[n]used to enable the de-bounce function for each corresponding bit. For an edge triggered interrupt to be generated, input signal must be valid for two consecutive de-bounce periods. The de-bounce time is controlled by the **DBNCECON** register. The DBEN[*n*] is used for "edge-trigger" interrupt only; it is ignored for "level trigger" interrupt 0 = The bit[*n*] de-bounce function is disabled 1 = The bit[*n*] de-bounce function is enabled

GPIO Port [A/B] Interrupt Mode Control (GPIOx _IMD)

Register	Offset	R/W	Description	Reset Value
GPIOA_IMD	GP_BA+0x018	R/W	GPIO Port A Interrupt Mode Control	0xXXXX_0000
GPIOB_IMD	GP_BA+0x058	R/W	GPIO Port B Interrupt Mode Control	0xXXXX_0000

Table 5-48 GPIO Interrupt Mode Control (GPIOx_IMD)

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
IMD[15:8]							
7	6	5	4	3	2	1	0
IMD[7:0]							

Bits	Descriptions	
[*n*]	**IMD[*n*]**	**Port [A/B] Edge or Level Detection Interrupt Control** IMD[*n*] used to control whether the interrupt mode is level triggered or edge triggered. If the interrupt mode is edge triggered, edge de-bounce is controlled by the DBEN register. If the interrupt mode is level triggered, the input source is sampled each clock to generate an interrupt 0 = Edge triggered interrupt 1 = Level triggered interrupt If level triggered interrupt is selected, then only one level can be selected in the GPIOX_IEN register. If both levels are set no interrupt will occur

GPIO Port [A/B] Interrupt Enable Control (GPIOx _IEN)

Register	Offset	R/W	Description	Reset Value
GPIOA_IEN	GP_BA+0x01C	R/W	GPIO Port A Interrupt Enable	0x0000_0000
GPIOB_IEN	GP_BA+0x05C	R/W	GPIO Port B Interrupt Enable	0x0000_0000

Table 5-49 GPIO Interrupt Enable Control Register (GPIOx_IEN)

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

Bits	Descriptions	
[n+16]	**IR_EN[n]**	**Port [A/B] Interrupt Enable by Input Rising Edge or Input Level High** IR_EN[n] is used to enable the rising/high interrupt for each of the corresponding GPIO pins. It also enables the pin wakeup function. If the interrupt is configured in level trigger mode, a level "high" will generate an interrupt. If the interrupt is configured in edge trigger mode, a state change from "low-to-high" will generate an interrupt. GPB.0 and GPB.1 trigger individual IRQ vectors (IRQ2/IRQ3) while remaining GPIO trigger a single interrupt vector IRQ4. 1 = Enable GPIOx[n] for level-high or low-to-high interrupt 0 = Disable GPIOx[n] for level-high or low-to-high interrupt.
[n]	**IF_EN[n]**	**Port [A/B] Interrupt Enable by Input Falling Edge or Input Level Low** IF_EN[n] is used to enable the falling/low interrupt for each of the corresponding GPIO pins. It also enables the pin wakeup function. If the interrupt is configured in level trigger mode, a level "low" will generate an interrupt. If the interrupt is configured in edge trigger mode, a state change from "high-to-low" will generate an interrupt. GPB.0 and GPB.1 trigger individual IRQ vectors (IRQ2/IRQ3) while remaining GPIO trigger a single interrupt vector IRQ4. 1 = Enable GPIOx[n] for low-level or high-to-low interrupt 0 = Disable GPIOx[n] for low-level or high-to-low interrupt

GPIO Port [A/B] Interrupt Trigger Source (GPIOx _ISRC)

Register	Offset	R/W	Description	Reset Value
GPIOA_ISRC	GP_BA+0x020	R/WC	GPIO Port A Interrupt Trigger Source Indicator	0x0000_0000
GPIOB_ISRC	GP_BA+0x060	R/WC	GPIO Port B Interrupt Trigger Source Indicator	0x0000_0000

Table 5-50 GPIO Interrupt Trigger Source Register (GPIOx_ISRC)

15	14	13	12	11	10	9	8
IF_ ISRC[15:8]							
7	6	5	4	3	2	1	0
IF_ ISRC[7:0]							

Bits	Descriptions	
[n]	**ISRC[n]**	**Port [A/B] Interrupt Trigger Source Indicator** Read : 1 = Indicates GPIOx[n] generated an interrupt 0 = No interrupt from GPIOx[n] Write : 1= Clear the corresponding pending interrupt. 0= No action

Interrupt De-bounce Control (DBNCECON)

Register	Offset	R/W	Description	Reset Value
DBNCECON	GP_BA+0x180	R/W	External Interrupt De-bounce Control	0x0000_0020

Table 5-51 GPIO Interrupt De-bounce Control Register (DBNCECON)

7	6	5	4	3	2	1	0
Reserved		ICLK_ON	DBCLKSRC	DBCLKSEL			

Bits	Descriptions	
[5]	**ICLK_ON**	**Interrupt clock On mode** Set this bit "0" will gate the clock to the interrupt generation circuit if the GPIOx[n] interrupt is disabled. 0 = disable the clock if the GPIOx[n] interrupt is disabled 1 = Interrupt generation clock always active.
[4]	**DBCLKSRC**	**De-bounce counter clock source select** 1 = De-bounce counter clock source is the internal 16kHz clock 0 = De-bounce counter clock source is HCLK
[3:0]	**DBCLKSEL**	**De-bounce sampling cycle selection.** For edge level interrupt GPIO state is sampled every 2^(DBCLKSEL) de-bounce clocks. For example if **DBCLKSRC**= 6, then interrupt is sampled every 2^6=64 de-bounce clocks. If **DBCLKSRC** is 16kHz oscillator this would be a 64ms debounce.

5.5 Brownout Detection and Temperature Alarm

The ISD93xx is equipped with a Brown-Out voltage detector and Over Temperature Alarm. The Brown-Out detector features a configurable trigger level and can be configured by flash to be active upon reset. The Brown-Out detector also has a power saving mode where detection can be set up to be active for a configurable on and off time.

TALARM and BOD operation require that the OSC16K low power oscillator is enabled (SYSCLK->PWRCON.OSC16K_ENB = 0).

The over temperature alarm is designed to protect the chip from dangerously high internal temperatures, generally associated with excessive load (or short circuit) on the speaker driver. The temperature alarm can generate an interrupt to which the CPU can respond and shut down the speaker driver. It is recommended that users implement this function due to the drive strength of the speaker driver has the capability of damaging the chip.

5.5.1 Brownout and Temperature Alarm Register Map

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

Register	Offset	R/W	Description	Reset Value	Reference
BOD_BA = 0x4008_4000					
BOD_SEL	BOD_BA+0x00	R/W	Brownout Detector Select Register	0x0000_0000	Table 5-52
BOD_EN	BOD_BA+0x04	R/W	Brownout Detector Enable Register	0x0000_00XX	Table 5-53
TALARM_SEL	BOD_BA+0x08	R/W	Temperature Alarm Select Register	0x0000_0000	Table 5-55
TALARM_EN	BOD_BA+0x0C	R/W	Temperature Alarm Enable Register	0x0000_0000	Table 5-56
DET_TIMER	BOD_BA+0x10	R/W	Brownout Detection Timer Register	0x0000_0000	Table 5-54

Brown-Out Detector Select Register (BOD_SEL)

Register	Offset	R/W	Description	Reset Value
BOD_SEL	BOD_BA+0	R/W	Brown Out Detector Select Register	0x0000_0000

Table 5-52 Brownout Detector Select Register (BOD_SEL, address 0x4008_4000)

7	6	5	4	3	2	1	0
Reserved			RANGE	BOD_HYS	BOD_LVL		

Bits	Descriptions	
[31:5]	Reserved	Reserved
[4]	**RANGE**	Range setting for BOD_LVL
[3]	**BOD_HYS**	**BOD Hysteresis** 1= Enable Hysteresis of BOD detection. 0= Hysteresis Disabled.
[2:0]	**BOD_LVL**	**BOD Voltage Level** RANGE=0 / RANGE=1 111b = 4.6V / 111b = 4.2V 110b = 3.0V / 110b = 3.9V 101b = 2.8V / 101b = 3.8V 100b = 2.65V / 100b = 3.7V 011b = 2.5V / 011b = 3.6V 010b = 2.4V / 010b = 3.4V 001b = 2.2V / 001b = 3.2V 000b = 2.1V / 000b = 3.1V

Brown-Out Detector Enable Register (BOD_EN)

This register is initialized by user flash configuration bit config0[23] (see Section 6.7). If config0[23]=0, then reset value of BODEN is 0x5. The effect of this is to generate a NMI interrupt (default NMI interrupt is BOD interrupt) if BOD circuit detects a voltage below 2.1V. The NMI ISR can be defined by the user to respond to this low voltage level.

Register	Offset	R/W	Description	Reset Value
BOD_EN	BOD_BA+4	R/W	Brown Out Detector Enable Register	0x0000_00XX

Table 5-53Detector Enable Register (BOD_EN, address 0x4008_4004)

7	6	5	4	3	2	1	0
Reserved			BOD_OUT	INT	IE	EN	

Bits	Descriptions	
[31:5]	Reserved	Reserved
[4]	**BOD_OUT**	Output of BOD detection block. This signal can be monitored to determine the current state of the BOD comparator. BOD_OUT=1 implies that VCC is less than BOD_LVL.
[3]	**INT**	**Current status of interrupt.** Latched whenever a BOD event occurs and IE=1. Write '1' to clear.
[2]	**IE**	**BOD Interrupt Enable** 1= Enable BOD Interrupt. 0= Disable BOD Interrupt.
[1:0]	**EN**	**BOD Enable.** 1xb = Enable continuous BOD detection. 01b = Enable time multiplexed BOD detection. See DET_TIMER register. 00b = Disable BOD Detection.

Brown-Out Detection Timer Register (DET_TIMER)

The BOD detector can be set up to take periodic samples of the supply voltage to minimize power consumption. The circuit can be configured and used in Standby Power Down (SPD) mode and can wake up the device if a BOD is event detected. The detection timer uses the OSC16K oscillator as time base so this oscillator must be active for timer operation. When active the BOD circuit requires ~165uA. With default timer settings, average current reduces to 500nA 165uA*ON_DUR/(ON_DUR+OFF_DUR).

Register	Offset	R/W	Description	Reset Value
DET_TIMER	BOD_BA+10	R/W	Detection Time Multiplex Register	0x0003_03E3

Table 5-54 Detection Time Multiplex Register (DET_TIMER, address 0x4008_4010)

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

Bits	Descriptions	
[31:20]	Reserved	Reserved
[19:16]	**ON_DUR**	Time BOD detector is active: (ON_DUR+1) * 100us. Minimum value is 1. (default = 400us)
[15:0]	**OFF_DUR**	Time BOD detector is off: (OFF_DUR+1)*100us . Minimum value is 7. (default = 99.6ms)

Temperature Alarm Select Register (TALARM_SEL)

Register	Offset	R/W	Description	Reset Value
TALARM_SEL	BOD_BA+8	R/W	Temperature Alarm Select Register	0x0000_0000

Table 5-55 Temperature Alarm Select Register (TALARM_SEL, address 0x4008_4008)

7	6	5	4	3	2	1	0
Reserved				LVL			

Bits	Descriptions	
[31:4]	Reserved	Reserved
[3:0]	**LVL**	**Temperature Alarm Sense Level** 1000b = 145C0100b = 135C 0010b = 125C0001b = 115C 0000b = 105C

Temperature Alarm Enable Register (TALARM EN)

Register	Offset	R/W	Description	Reset Value
TALARM _EN	BOD_BA+C	R/W	Temperature Alarm Enable Register	0x0000_00XX

Table 5-56Temperature Alarm Enable Register (TALARM _EN, address 0x4008_400C)

7	6	5	4	3	2	1	0
Reserved				INT	IE	TALARM	EN

Bits	Descriptions	
[31:4]	Reserved	Reserved
[3]	**INT**	**Current status of interrupt.** Latched whenever a Temperature Sense event occurs and IE=1. Write '1' to clear.
[2]	**IE**	**TALARM Interrupt Enable** 1= EnableTALARM Interrupt., 0= Disable TALARM Interrupt.
[1]	**TALARM**	**Output of TALARM block.** Can be polled to determine whether TALARM active (=1).
[0]	**EN**	**TALARM Enable.** 1= Enable TALARM detection, 0 = Disable TALARM Detection.

5.6 I2C Serial Interface Controller (Master/Slave)

5.6.1 Introduction

I2C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. The I2C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously. Serial, 8-bit oriented, bi-directional data transfers can be made up 1.0 Mbps.

Data is transferred between a Master and a Slave synchronously to SCL on the SDA line on a byte-by-byte basis. Each data byte is 8 bits long. There is one SCL clock pulse for each data bit with the MSB being transmitted first. An acknowledge bit follows each transferred byte. Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL. A transition on the SDA line while SCL is high is interpreted as a command (START or STOP). Please refer to Figure 5-9 for more detail I2C BUS Timing.
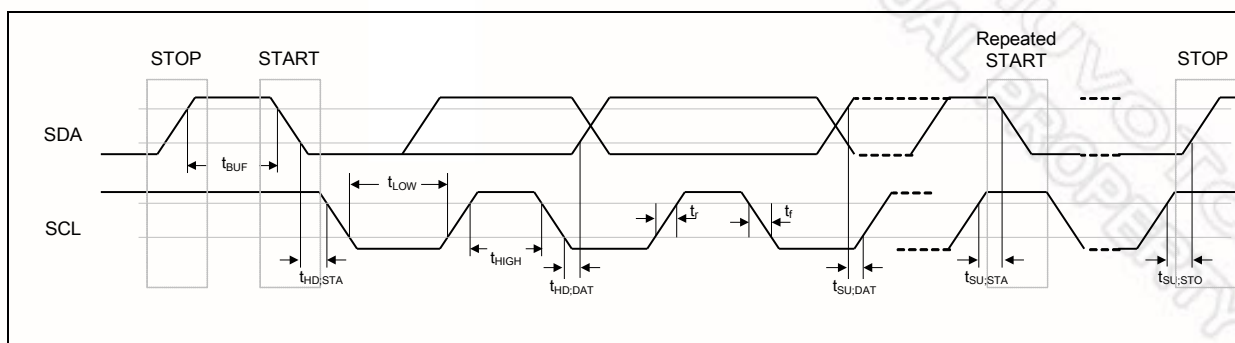
Figure 5-9I2C Bus Timing

The device's on-chip I2C logic provides the serial interface that meets the I2C bus standard mode specification. The I2C port handles byte transfers autonomously. To enable this port, the bit ENS1 in I2C->CON should be set to '1'. The I2C H/W interfaces to the I2C bus via two pins: I2C_SDA (which can be configured as GPIOA[10], GPIOB[3] or GPIOA[3], serial data line) and I2C_SCL (which can be configured as GPIOA[11], GPIOB[2] or GPIOA[1], serial clock line). See Table 5-7 and Table 5-8 for alternate GPIO pin functions. Pull up resistor is needed for these pins for I2C operation as these are open drain pins.

The I2C bus uses two wires (SDA and SCL) to transfer information between devices connected to the bus. The main features of the bus are:

- Master/Slave up to 1Mbit/s

- Bidirectional data transfer between masters and slaves

- Multi-master bus (no central master)

- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus

- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus

- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer

- Built-in a 14-bit time-out counter will request the I2C interrupt if the I2C bus hangs up and timer-out counter overflows.

- External pull-up are needed for high output

- Programmable clocks allow versatile rate control

- Supports 7-bit addressing mode

- I2C-bus controllers support multiple address recognition (Four slave address with mask option)

5.6.1.1 *I²C Protocol*

Normally, a standard communication consists of four parts:

1) START or Repeated START signal generation

2) Slave address transfer
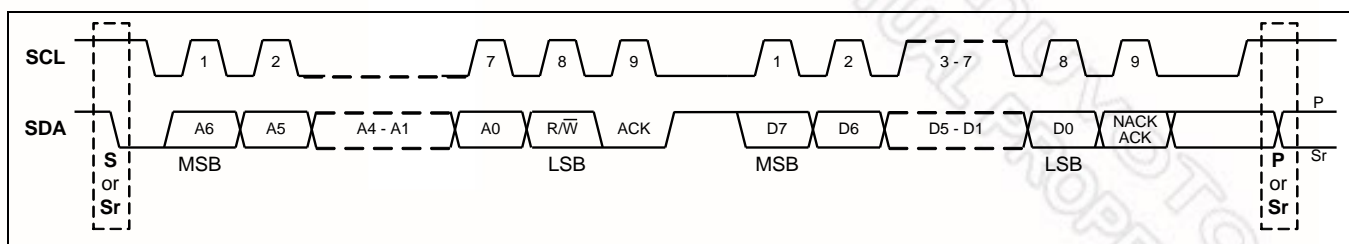
3) Data transfer

4) STOP signal generation

Figure 5-10I2C Protocol

5.6.1.2 *Data transfer on the I2C-bus*

A master-transmitter always begins by addressing a slave receiver with a 7-bit address. For a transaction where the master-transmitter is sending data to the slave, the transfer direction is not changed, master is always transmitting and slave acknowledges the data, see Figure 5-11.
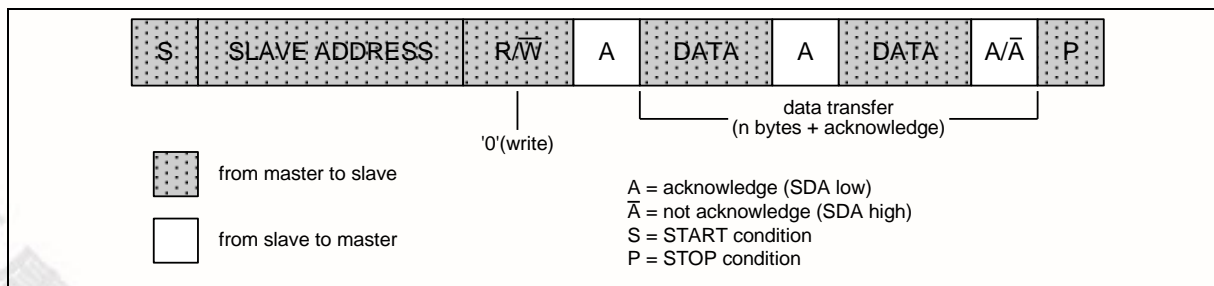

Figure 5-11Master Transmits Data to Slave

For a master to read data from a slave, master addresses slave with the R/W bit set to '1', immediately after the first byte (address) is acknowledged by the slave the transfer direction is changed and slave sends data to the master and master acknowledges the data transfer.
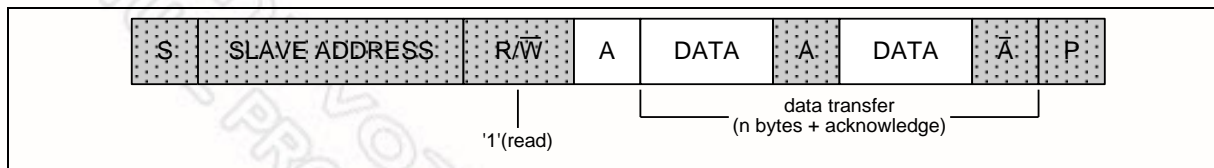

Figure 5-12Master Reads Data from Slave

5.6.1.3 *START or Repeated START signal*

When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the S-bit, is defined as a HIGH to LOW transition on the SDA line while SCL is HIGH. The START signal denotes the beginning of a new data transfer.

A Repeated START (Sr) is a START signal without first generating a STOP signal. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

STOP signal

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the P-bit, is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH.
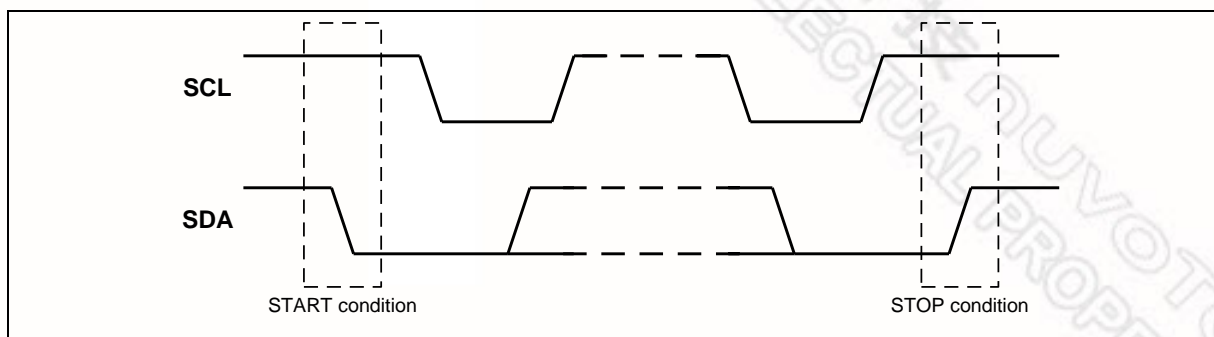

Figure 5-13START and STOP condition

5.6.1.4 *Slave Address Transfer*

The first byte of data transferred by the master immediately after the START signal is the slave address. This is a 7-bits calling address followed by a RW bit. The RW bit signals the slave the data transfer direction. No two slaves in the system can have the same address. Only the slave with an address that matches the one transmitted by the master will respond by returning an acknowledge bit by pulling the SDA low at the 9th SCL clock cycle.

5.6.1.5 *Data Transfer*

Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a Not Acknowledge (NACK), the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

If the master, as the receiving device, does Not Acknowledge (NACK) the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.
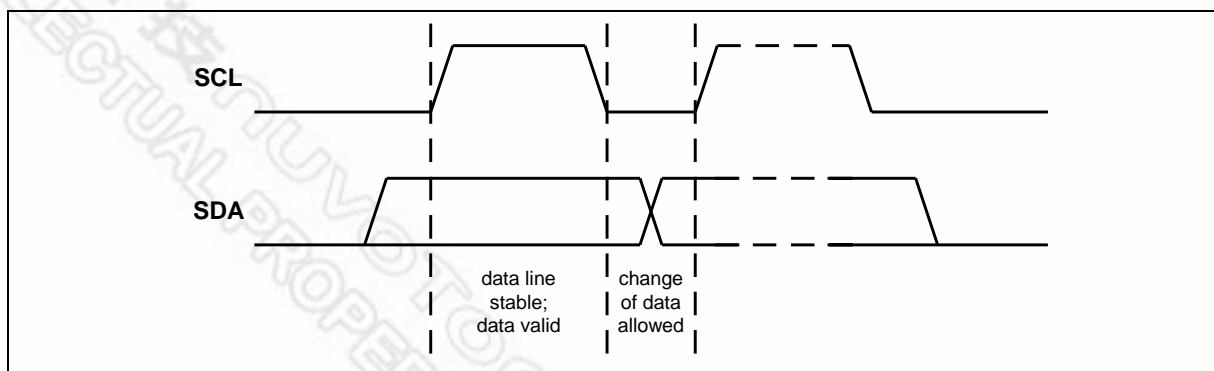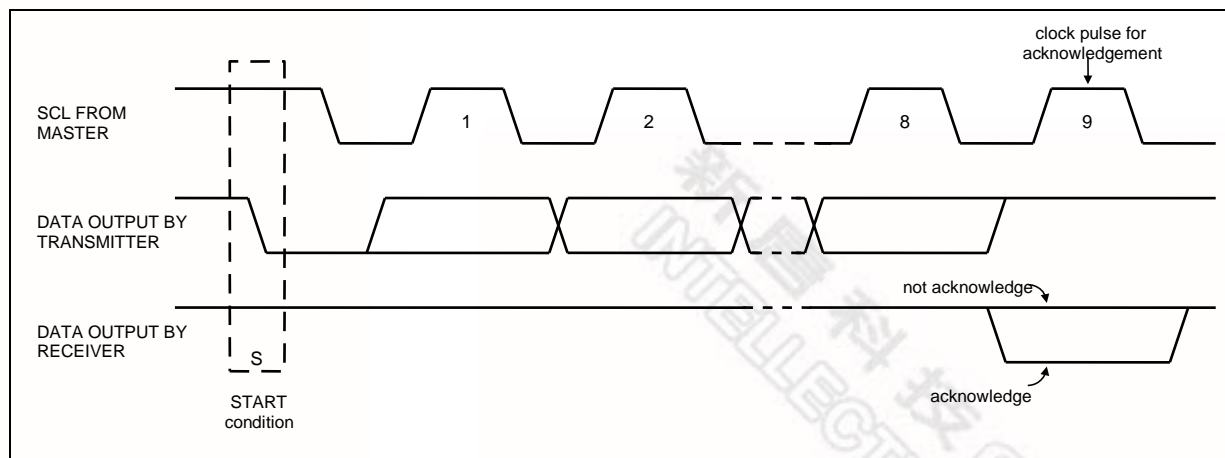
Figure 5-14Bit Transfer on the I2C bus

Figure 5-15Acknowledge on the I2C bus

5.6.2   I2C Protocol Registers

The CPU interfaces to the SIO port through the following thirteen special function registers: I2C->CON (control register), I2C->STATUS (status register), I2C->DATA (data register), I2C->ADDRn (address registers, n=0~3), I2C->ADRMn (address mask registers, n=0~3), I2C->CLK (clock rate register) and I2c->TOC (Time-out counter register). Bits 31~ bit 8 of these I2C special function registers are reserved. These bits do not have any functions and are all zero if read back.

When I2C port is enabled by setting ENSI (I2C->CON[6]) to high, the internal states will be controlled by I2C->CON and I2C logic hardware. Once a new status code is generated and stored in I2C->STATUS, the I2C Interrupt Flag bit SI (I2C->CON[3]) will be set automatically. If the Enable Interrupt bit EI (I2C->CON[7]) is set high at this time, the I2C interrupt will be generated. The bit field I2C->STATUS[7:3] stores the internal state code, the lowest 3 bits of I2C->STATUS are always zero and the contents are stable until SI is cleared by software. The base address of the I2C peripheral on theISD93xx is 0x4002_0000.

5.6.2.1   *Address Registers (I2C->ADDR)*

I2C port is equipped with four slave address registers I2C->ADDRn (n=0~3). The contents of the register are irrelevant when I2C is in master mode. In the slave mode, the bit field I2C->ADDRn[7:1] must be loaded with the MCU's own slave address. The I2C hardware will react if the contents of I2C->ADDR are matched with the received slave address.

The I2C ports support the "General Call" function. If the GC bit (I2C->ADDRn[0]) is set the I2C port hardware will respond to General Call address (00H). Clear GC bit to disable general call function.

When GC bit is set, the I2C is in Slave mode, it can be received the general call address by 00H after Master send general call address to I2C bus, then it will follow status of GC mode. If it is in master mode, the AA bit (I2C->CON[2], Assert Acknowledge control bit) must be cleared when it will send general call address of 00H to I2C bus.

I2C-bus controllers support multiple address recognition with four address mask registers I2ADRMn (n=0~3). When the bit in the address mask register is set to one, it means the received corresponding address bit is don't-care. If the bit is set to zero, that means the received corresponding register bit should be exact the same as address register.

5.6.2.2   *Data Register (I2C->DATA)*

This register contains a byte of serial data to be transmitted or a byte which has just been received. The CPU can read from or write to this 8-bit (I2C->DATA[7:0]) directly addressable SFR while it is not in the process of shifting a byte. This occurs when SIO is in a defined state and the serial interrupt flag (SI) is set. Data in I2C->DATA[7:0] remains stable as long as SI bit is set. While data is being shifted out, data on the bus is simultaneously being shifted in; I2C->DATA[7:0] always contains the last data byte present on the bus. Thus, in the event of arbitration lost, the transition from master transmitter to slave receiver is made with the correct data in I2C->DATA[7:0].

I2C->DATA[7:0] and the acknowledge bit form a 9-bit shift register, the acknowledge bit is controlled by the SIO hardware and cannot be accessed by the CPU. Serial data is shifted through the acknowledge bit into I2C->DATA[7:0] on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into I2C->DATA[7:0], the serial data is available in I2C->DATA[7:0], and the acknowledge bit (ACK or NACK) is returned by the control logic during the ninth clock pulse. Serial data is shifted out from I2C->DATA[7:0] on the falling edges of SCL clock pulses, and is shifted into I2C->DATA[7:0] on the rising edges of SCL clock pulses.
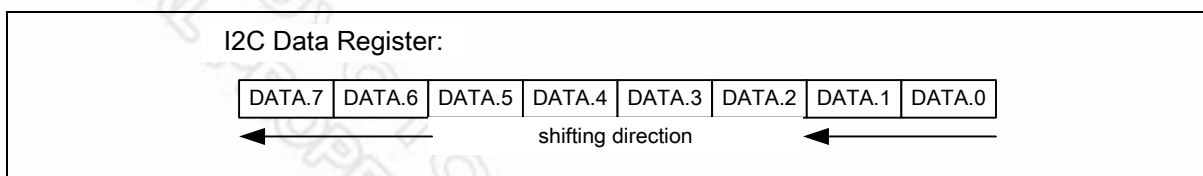
I2C Data Register:

DATA.7	DATA.6	DATA.5	DATA.4	DATA.3	DATA.2	DATA.1	DATA.0

shifting direction

Figure 5-16 I2C Data Shift Direction

5.6.2.3   *Control Register (I2C->CON)*

The CPU can read from and write to this 8-bit field of I2C->CON[7:0]. Two bits are affected by hardware: the SI bit is set when the I2C hardware requests a serial interrupt, and the STO bit is cleared when a STOP condition is present on the bus. The STO bit is also cleared when ENS1 = "0".

EI   Enable Interrupt.

ENSI   Set to enable I2C serial function block. When ENS=1 the I2C serial function is enabled.

STA   I2C START Control Bit. Setting STA to logic 1 enters master mode, the I2C hardware sends a START or repeat START condition to bus when the bus is free.

STO   I2C STOP Control Bit. In master mode, setting STO transmits a STOP condition to the bus. The I2C hardware will check the bus condition and if a STOP condition is detected this flag will be cleared by hardware. In a slave mode, setting STO resets I2C hardware to the defined "not addressed" slave mode. This means it is NO LONGER in the slave receiver mode to receive data from the master transmit device.

SI   I2C Interrupt Flag. When a new SIO state is present in the I2C->STATUS register, the SI flag is set by hardware, and if bit EI (I2C->CON[7]) is set, the I2C interrupt is requested. SI must be cleared by software. Clear SI is by writing one to this bit.

AA   Assert Acknowledge Control Bit. When AA=1 prior to address or data received, an acknowledged (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when:

      1.) A slave is acknowledging the address sent from master,
      2.) A receiver device is acknowledging thedata sent by a transmitter.

When AA=0 prior to address or data received, a Not acknowledged (high level to SDA) will be returned during the acknowledge clock pulse on the SCL line.

5.6.2.4   *Status Register (I2C->STATUS)*

I2C->STATUS[7:0] is an 8-bit read-only register. The three least significant bits are always 0. The bit field I2C->STATUS[7:3] contains the status code. There are 26 possible status codes. When I2C->STATUS[7:0] contains F8H, no serial interrupt is requested. All other I2C->STATUS[7:3] values correspond to defined SIO states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2C->STATUS[7:3] one machine cycle after SI is set by hardware and is still present one machine cycle after SI has been reset by software.

In addition, state 00H stands for a Bus Error. A Bus Error occurs when a START or STOP condition is present at an illegal position in the format frame. Examples of illegal positions are during the serial transfer of an address byte, a data byte or an acknowledge bit. To recover I2C from bus error, STO should be set and SI should be clear to enter not addressed slave mode. Then clear STO to release bus and to wait new communication. I2C bus cannot recognize stop condition during this action when bus error occurs.

5.6.2.5   *I2C Clock Baud Rate Bits (I2C->CLK)*

The data baud rate of I2C is determined by I2C->CLK[7:0] register when SIO is in a master mode. It is not important when SIO is in a slave mode. In the slave modes, SIO will automatically synchronize with any clock frequency up to 400 kHz from master I2C device.

Data Baud Rate of I2C = PCLK /(4x(I2C->CLK[7:0]+1)). If PCLK=16MHz, the I2C->CLK[7:0] = 40 (28H), data baud rate of I2C = 16MHz/(4x(40 +1)) = 97.5Kbits/sec.

5.6.2.6   *The I2C Time-out Counter Register (I2C->TOC)*

There is a 14-bit time-out counter which can be configured to deal with an I2C bus hang-up. If the time-out counter is enabled, the counter starts up-counting until it overflows (TIF=1) and generates I2C interrupt to CPU or stops counting by clearing ENTI to 0. When time-out counter is enabled, setting flag SI to high will reset counter. Counter will re-start after SI is cleared. If the I2C bus hangs up, counter will overflow and generate a CPU interrupt. Refer to Figure 5-17 for the 14-bit time-out counter. User can clear TIF by writing one to this bit.
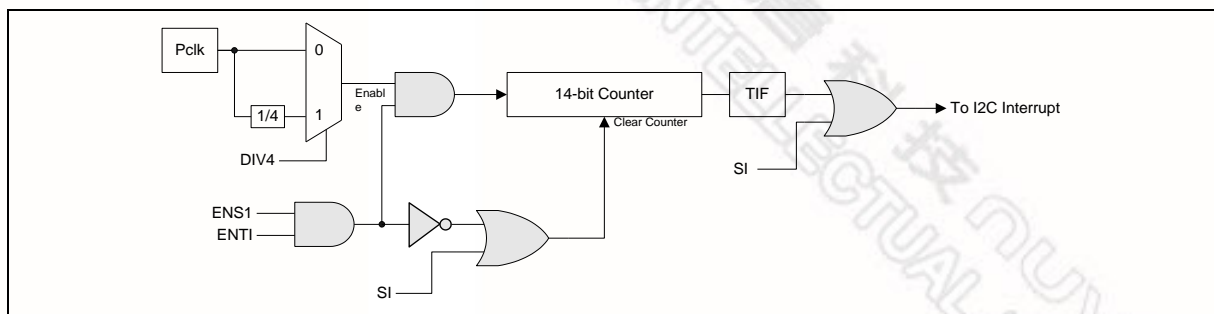
Figure 5-17: I2C Time-out Count Block Diagram

5.6.3　Register Mapping

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

Register	Offset	R/W	Description	Reset Value
I2C->CON	I2C_BA+0x00	R/W	I2C Control Register	0x0000_0000
I2C->ADRR0	I2C_BA+0x04	R/W	I2C Slave address Register0	0x0000_0000
I2C->DATA	I2C_BA+0x08	R/W	I2C DATA Register	0x0000_0000
I2C->STATUS	I2C_BA+0x0C	R	I2C Status Register	0x0000_00F8
I2C->CLK	I2C_BA+0x10	R/W	I2C clock divided Register	0x0000_0000
I2C->TOC	I2C_BA+0x14	R/W	I2C Time out control Register	0x0000_0000
I2C->ADDR1	I2C_BA+0x18	R/W	Slave address Register1	0x0000_0000
I2C->ADDR2	I2C_BA+0x1C	R/W	Slave address Register2	0x0000_0000
I2C->ADDR3	I2C_BA+0x20	R/W	Slave address Register3	0x0000_0000
I2C->ADM0	I2C_BA+0x24	R/W	Slave address Mask Register0	0x0000_0000
I2C->ADM1	I2C_BA+0x28	R/W	Slave address Mask Register1	0x0000_0000
I2C->ADM2	I2C_BA+0x2C	R/W	Slave address Mask Register2	0x0000_0000
I2C->ADM3	I2C_BA+0x30	R/W	Slave address Mask Register3	0x0000_0000

5.6.4 Register Description

I2C CONTROL REGISTER (I2C->CON)

Register	Offset	R/W	Description	Reset Value
I2C->CON	I2C_BA+0x00H	R/W	I2C Control Register	0x0000_0000

7	6	5	4	3	2	1	0
EI	ENSI	STA	STO	SI	AA	Reserved	Reserved

Bits	Descriptions	
[7]	EI	**Enable Interrupt.** 1 = Enable interrupt CPU. 0 = Disable interrupt.
[6]	ENSI	**I2C Controller Enable Bit.** 1 = Enable 0 = Disable Set to enable I2C serial function block.
[5]	STA	**I2C START Control Bit.** Setting STA to logic 1 will enter master mode, the I2C hardware sends a START or repeat START condition to bus when the bus is free.
[4]	STO	**I2C STOP Control Bit.** In master mode, set STO to transmit a STOP condition to bus. I2C hardware will check the bus condition, when a STOP condition is detected this bit will be cleared by hardware automatically. In slave mode, setting STO resets I2C hardware to the defined "not addressed" slave mode. This means it is NO LONGER in the slave receiver mode able receive data from the master transmit device.
[3]	SI	**I2C Interrupt Flag.** When a new SIO state is present in the I2C->STATUS register, the SI flag is set by hardware, and if bit EI (I2C->CON[7]) is set, the I2C interrupt is requested. SI must be cleared by software. Clear SI is by writing one to this bit.
[2]	AA	**Assert Acknowledge Control Bit**. When AA=1 prior to address or data received, an acknowledge (ACK - low level to SDA) will be returned during the acknowledge clock pulse on the SCL line when: 1. A slave is acknowledging the address sent from master, 2. The receiver devices are acknowledging the data sent by transmitter. When AA=0 prior to address or data received, a Not acknowledged (high level to SDA) will be returned during the acknowledge clock pulse on the SCL line.

I2C DATA REGISTER (I2C->DATA)

Register	Offset	R/W	Description	Reset Value
I2C->DATA	I2C_BA+0x08	R/W	I2C DATA Register	0x0000_0000

7	6	5	4	3	2	1	0
			I2C->DATA[7:0]				

Bits	Descriptions
[7:0]	**I2C->DATA** **I2C Data Register.** During master or slave transmit mode, data to be transmitted is written to this register. During master or slave receive mode, data that has been received may be read from this register.

I2C STATUS REGISTER (I2C->STATUS)

Register	Offset	R/W	Description	Reset Value
I2C->STATUS	I2C_BA+0x0C	R/W	I2C STATUS Register	0x0000_0000

7	6	5	4	3	2	1	0
			I2C->STATUS[7:0]				

Bits	Descriptions
[7:0]	**I2C->STATUS**

I2C Status Register

The status register of I2C:

The three least significant bits are always 0. The five most significant bits contain the status code. There are 26 possible status codes. When I2C->STATUS contains F8H, no serial interrupt is requested. All other I2C->STATUS values correspond to defined I2C states. When each of these states is entered, a status interrupt is requested (SI = 1). A valid status code is present in I2C->STATUS one PCLK cycle after SI is set by hardware and is still present one PCLK cycle after SI has been reset by software. In addition, states 00H stands for a Bus Error. A Bus Error occurs when a START or STOP condition is present at an illegal position in the frame. Example of illegal position are during the serial transfer of an address byte, a data byte or an acknowledge bit.

I2C BAUD RATE CONTROL REGISTER (I2C->CLK)

Register	Offset	R/W	Description	Reset Value
I2C->CLK	I2C_BA+0x10	R/W	I2C clock divided Register	0x0000_0000

7	6	5	4	3	2	1	0
			I2C->CLK[7:0]				

Bits	Descriptions	
[7:0]	**I2C->CLK**	**I2C clock divided Register** The I2C clock rate bits: Data Baud Rate of I2C = PCLK /(4x(I2C->CLK+1)).

I2C TIME-OUT COUNTER REGISTER (I2C->TOC)

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

7	6	5	4	3	2	1	0
Reserved					ENTI	DIV4	TIF

Bits	Descriptions	
[2]	**ENTI**	**Time-out counter is enabled/disable** 1 = Enable 0 = Disable When enabled, the 14 bit time-out counter will start counting when SI is clear. Setting flag SI to high will reset counter and re-start up counting after SI is cleared.
[1]	**DIV4**	**Time-Out counter input clock divide by 4** 1 = Enable 0 = Disable When enabled, the time-out clock is PCLK/4
[0]	**TIF**	**Time-Out Flag** 1 = Time-out flag is set by H/W. It can interrupt CPU. Write 1 to clear. 0 = No time-out.

I2C SLAVE ADDRESS REGISTER (I2CADDRx)

Register	Offset	R/W	Description	Reset Value
I2C->ADDR0	I2C_BA+0x04	R/W	I2C slave Address Register0	0x0000_0000
I2C->ADDR1	I2C_BA+0x18	R/W	I2C slave Address Register1	0x0000_0000
I2C->ADDR2	I2C_BA+0x1C	R/W	I2C slave Address Register2	0x0000_0000
I2C->ADDR3	I2C_BA+0x20	R/W	I2C slave Address Register3	0x0000_0000

7	6	5	4	3	2	1	0
ADDR[7:1]							GC

Bits	Descriptions
[7:1]	**ADDR** **I2C Address Register** The content of this register is irrelevant when I2C is in master mode. In the slave mode, the seven most significant bits must be loaded with the MCU's own address. The I2C hardware will react if any of the addresses are matched.
[0]	**GC** **General Call Function** 0 = Disable General Call Function. 1 = Enable General Call Function.

I2C SLAVE ADDRESS MASK REGISTER (I2CADMx)

Register	Offset	R/W	Description	Reset Value
I2C->ADM0	I2C_BA+0x24	R/W	I2C slave Address Mask Register0	0x0000_0000
I2C->ADM1	I2C_BA+0x28	R/W	I2C slave Address Mask Register1	0x0000_0000
I2C->ADM2	I2C_BA+0x2C	R/W	I2C slave Address Mask Register2	0x0000_0000
I2C->ADM3	I2C_BA+0x30	R/W	I2C slave Address Mask Register3	0x0000_0000

7	6	5	4	3	2	1	0
ADMx[7:1]							Reserved

Bits	Descriptions
[7:1]	ADMx	**I2C Address Mask register** 1 = Mask enable (the received corresponding address bit is don't care.) 0 = Mask disable. I2Cbus controllers support multiple address recognition with four address mask registers. Bits in this field mask the ADDRx registers masking bits from the address comparison.

5.6.5   Modes of Operation

The on-chip I2C ports support five operation modes, Master transmitter, Master receiver, Slave transmitter, Slave receiver, and GC call.

In a given application, I2C port may operate as a master or as a slave. In the slave mode, the I2C port hardware looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to master(by setting the AA bit), an acknowledge pulse will be transmitted out on the 9th clock. An interrupt is requested on both master and slave devices if interrupt is enabled. When the microcontroller wishes to become the bus master, the hardware waits until the bus is free before the master mode is entered so that a possible slave action is not interrupted. If bus arbitration is lost in the master mode, I2C port switches to the slave mode immediately and can detect its own slave address in the same serial transfer.

5.6.5.1   *Master Transmitter Mode*

Serial data output through SDA while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the data direction bit. In this case the data direction bit (R/W) will be logic 0, and it is represented by "W" in the flow diagrams. Thus the first byte transmitted is SLA+W. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

5.6.5.2   *Master Receiver Mode*

In this case the data direction bit (R/W) will be logic 1, and it is represented by "R" in the flow diagrams. Thus the first byte transmitted is SLA+R. Serial data is received via SDA while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are output to indicate the beginning and end of a serial transfer.

5.6.5.3   *Slave Receiver Mode*

Serial data and the serial clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit.

5.6.5.4   *Slave Transmitter Mode*

The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted via SDA while the serial clock is input through SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer.

5.6.6 Data Transfer Flow in Five Operating Modes

The five operating modes are: Master/Transmitter, Master/Receiver, Slave/Transmitter, Slave/Receiver and GC Call. Bits STA, STO and AA in I2C->CON register will determine the next state of the SIO hardware after SI flag is cleared. Upon completion of the new action, a new status code will be updated and the SI flag will be set. If the I2C interrupt control bit EI (I2C->CON[7]) is set, appropriate action or software branch of the new status code can be performed in the Interrupt service routine.

Data transfers in each mode are shown in the following figures.

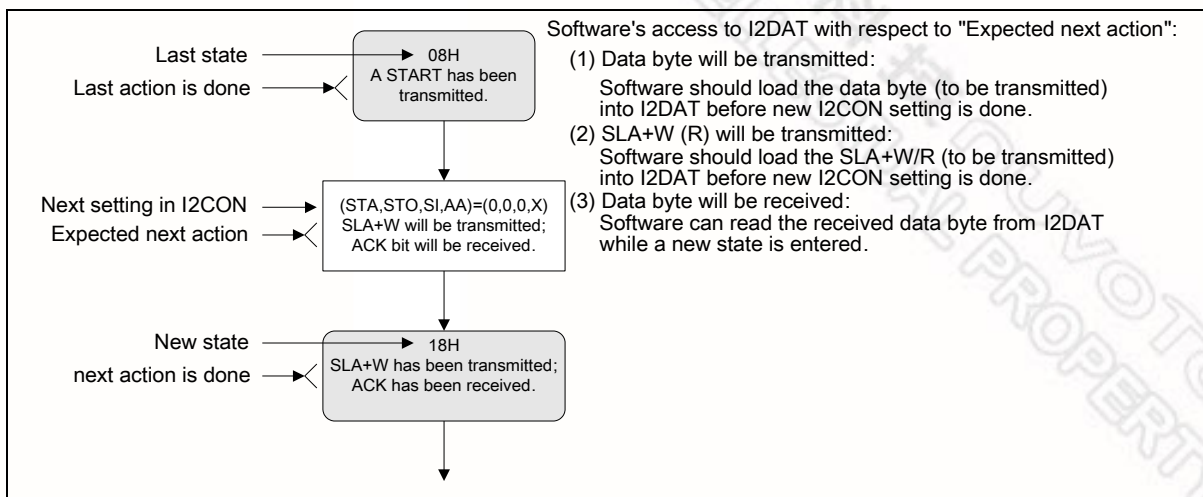*** Legend for the following five figures:
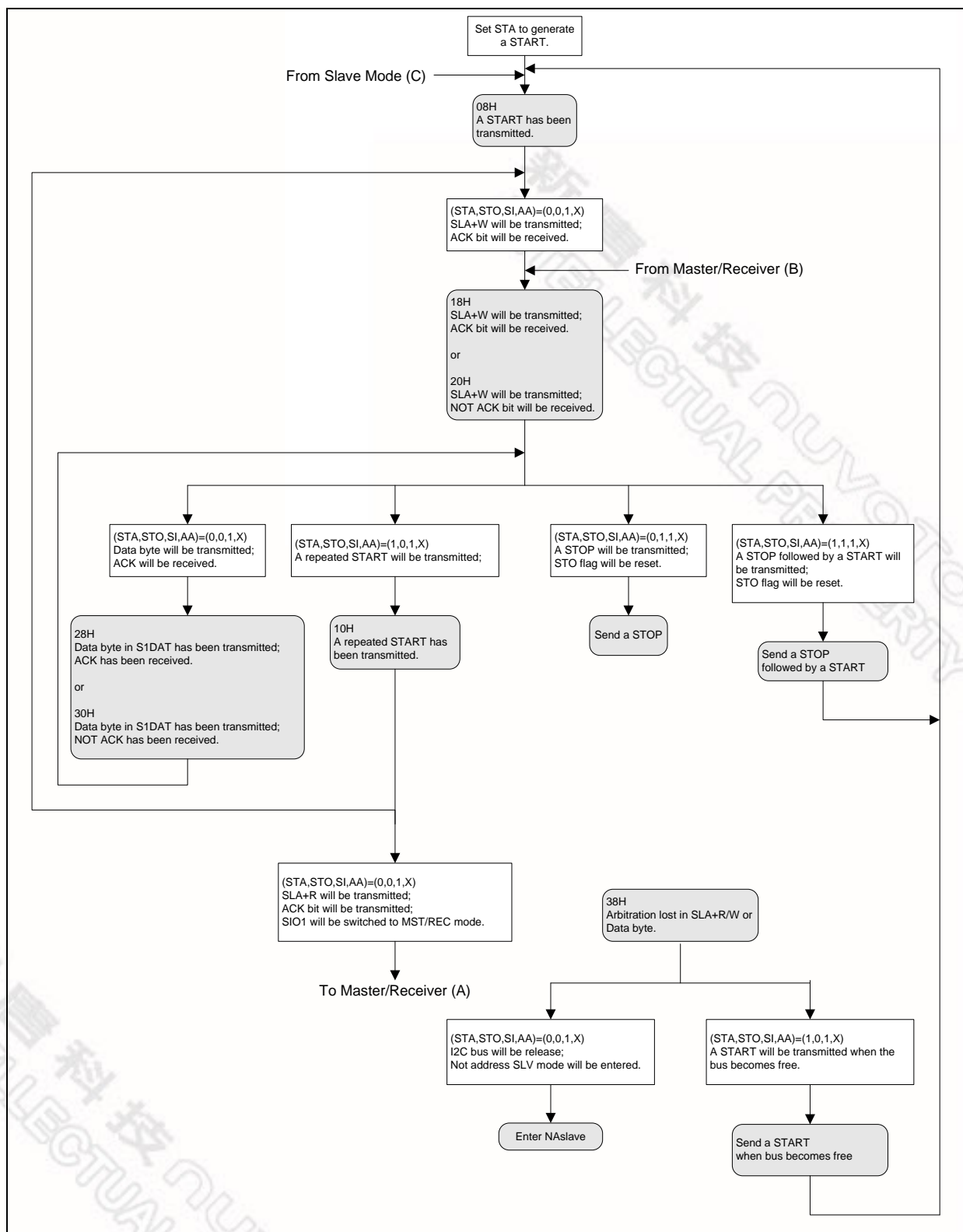

Figure 5-18 Legend for the following four figures

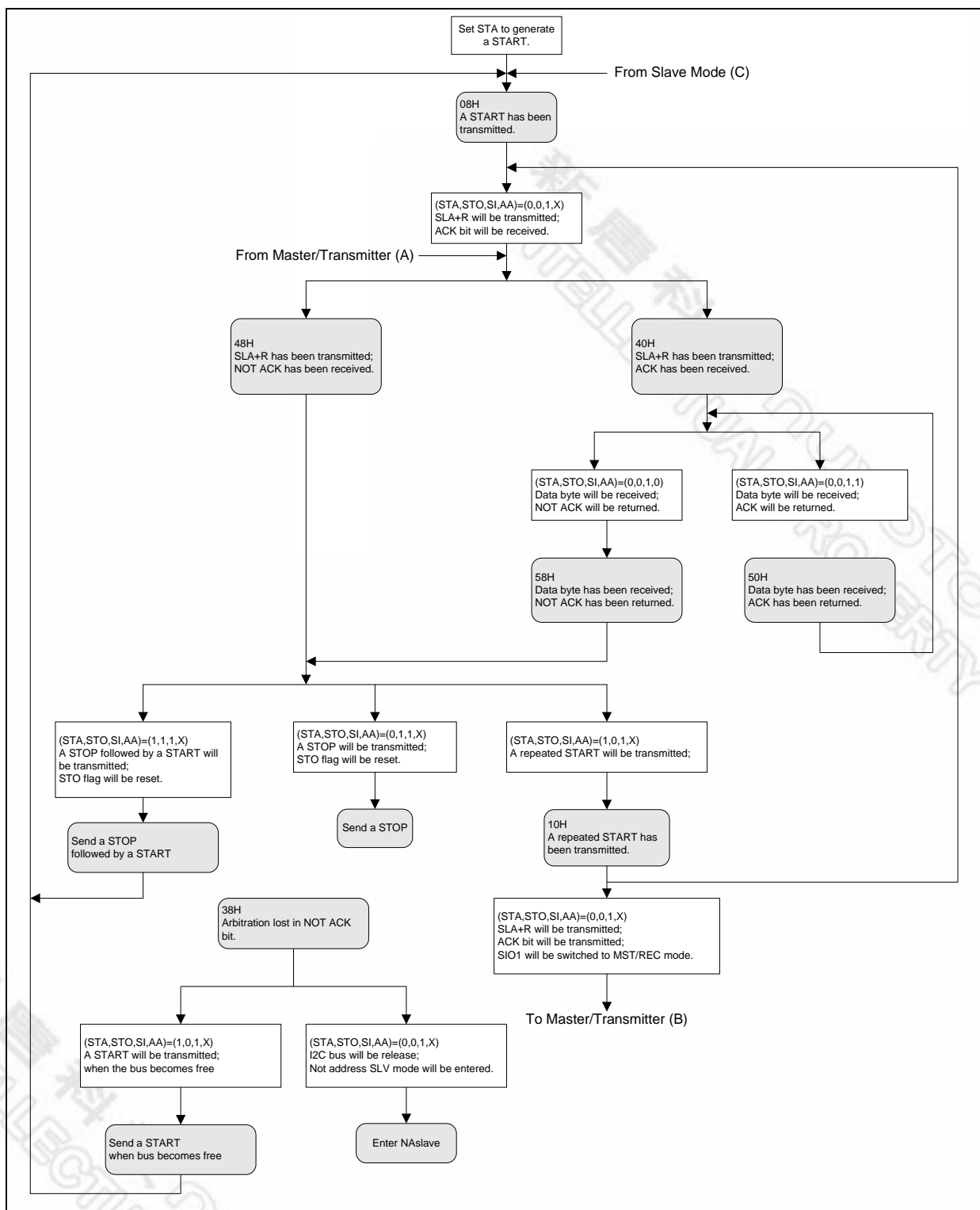Figure 5-19 Master Transmitter Mode

Figure 5-20 Master Receiver Mode

ISD93xx Series Design Guide

Set AA

A8H
Own SLA+R has been received;
ACK has been return.

or

B0H
Arbitration lost SLA+R/W as master;
Own SLA+R has been received;
ACK has been return.

(STA,STO,SI,AA)=(0,0,1,0)
Last data byte will be transmitted;
ACK will be received.

(STA,STO,SI,AA)=(0,0,1,1)
Data byte will be transmitted;
ACK will be received.

C8H
Last data byte in S1DAT has been transmitted;
ACK has been received.

C0H
Data byte or Last data byte in S1DAT has been transmitted;
NOT ACK has been received.

B8H
Data byte in S1DAT has been transmitted;
ACK has been received.

(STA,STO,SI,AA)=(0,0,1,0)
Last data will be transmitted;
ACK will be received.

(STA,STO,SI,AA)=(0,0,1,1)
Data byte will be transmitted;
ACK will be received.

A0H
A STOP or repeated START has been received while still addressed as SLV/TRX.

(STA,STO,SI,AA)=(1,0,1,1)
Switch to not address SLV mode;
Own SLA will be recognized;
A START will be transmitted when the bus becomes free.

(STA,STO,SI,AA)=(1,0,1,0)
Switch to not addressed SLV mode;
No recognition of own SLA;
A START will be transmitted when the becomes free.

(STA,STO,SI,AA)=(0,0,1,1)
Switch to not addressed SLV mode;
Own SLA will be recognized.

(STA,STO,SI,AA)=(0,0,1,0)
Switch to not addressed SLV mode;
No recognition of own SLA.

Send a START
when bus becomes free

Enter NAslave

To Master Mode (C)

Figure 5-21 Slave Transmitter Mode

Figure 5-22Slave Receiver Mode

Set AA

70H
Reception of the general call address
and one or more data bytes;
ACK has been return.

or

78H
Arbitration lost SLA+R/W as master;
and address as SLA by general call;
ACK has been return.

(STA,STO,SI,AA)=(X,0,1,0)
Data byte will be received;
NOT ACK will be returned.

(STA,STO,SI,AA)=(X,0,1,1)
Data byte will be received;
ACK will be returned.

98H
Previously addressed with General Call;
Data byte has been received;
NOT ACK has been returned.

90H
Previously addressed with General Call;
Data has been received;
ACK has been returned.

(STA,STO,SI,AA)=(X,0,1,0)
Data will be received;
NOT ACK will be returned.

(STA,STO,SI,AA)=(X,0,1,1)
Data will be received;
ACK will be returned.

A0H
A STOP or repeated START has been
received while still addressed as
SLV/REC.

(STA,STO,SI,AA)=(1,0,1,1)
Switch to not address SLV mode;
Own SLA will be recognized;
A START will be transmitted when
the bus becomes free.

(STA,STO,SI,AA)=(1,0,1,0)
Switch to not addressed SLV mode;
No recognition of own SLA;
A START will be transmitted when the
becomes free.

(STA,STO,SI,AA)=(0,0,1,1)
Switch to not addressed SLV mode;
Own SLA will be recognized.

(STA,STO,SI,AA)=(0,0,1,0)
Switch to not addressed SLV mode;
No recognition of own SLA.

Send a START
when bus becomes free

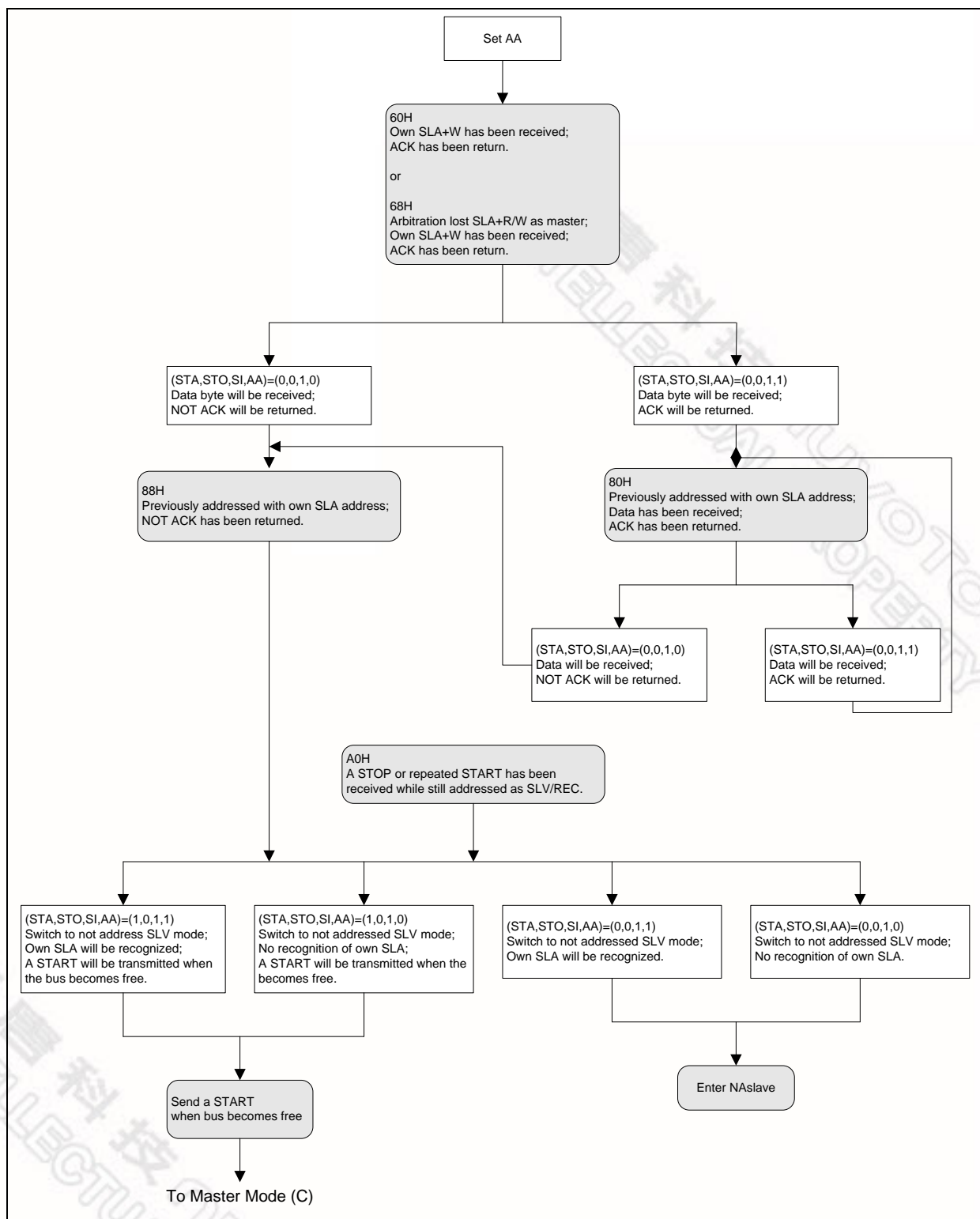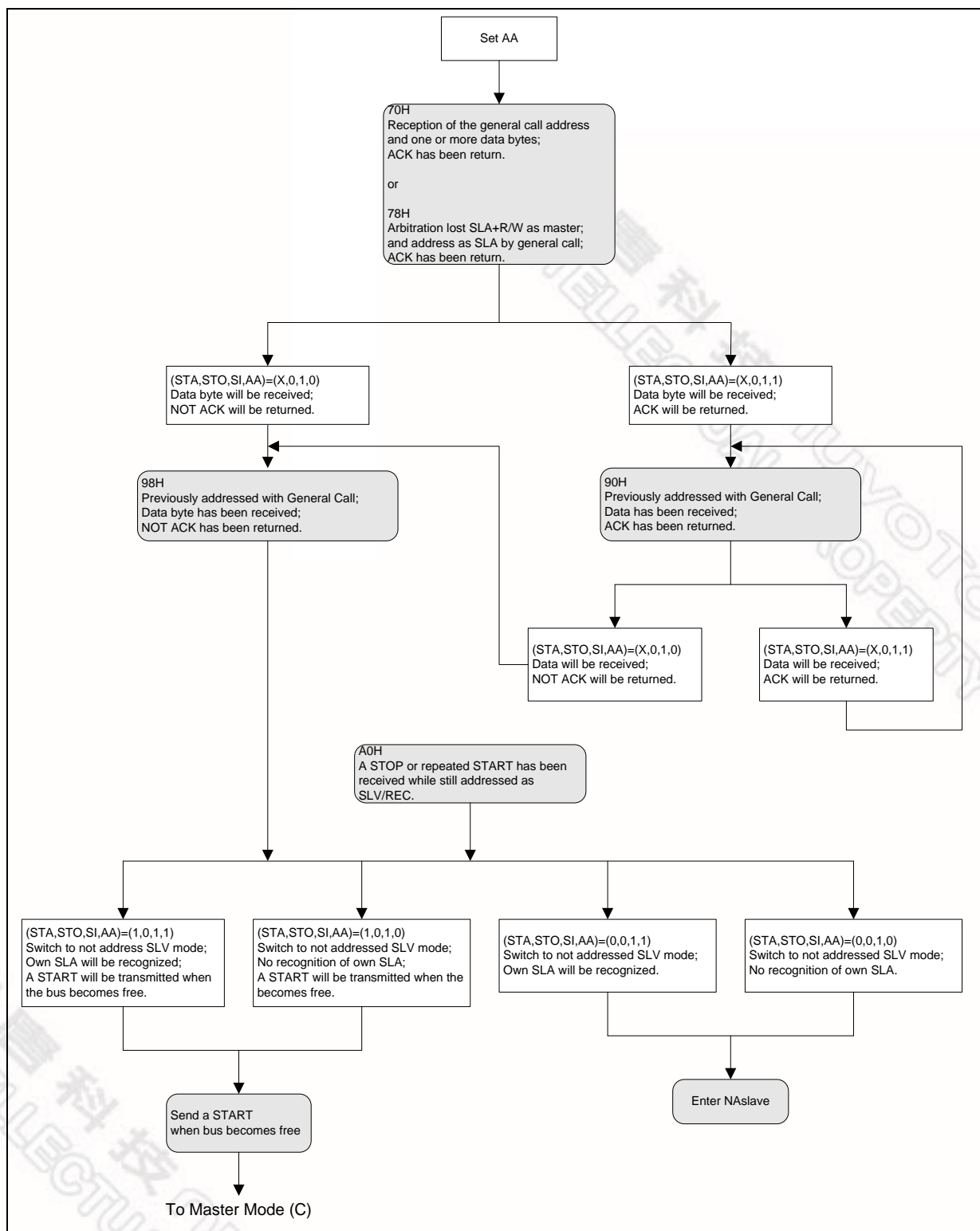Enter NAslave

To Master Mode (C)

Figure 5-23GC Mode

5.7 PWM Generator and Capture Timer

5.7.1 Introduction

The ISD93xx has three PWM generators which can be configured as6 independent PWM outputs, PWM0~PWM5, or as a complementary PWM pairs with programmable dead-zone generator. Each PWM Generator has an 8-bit prescaler, a clock divider providing 5 divided frequencies (1, 1/2, 1/4, 1/8, 1/16), two PWM Timers including two clock selectors, two 16-bit PWM down-counters for PWM period control, two 16-bit comparators for PWM duty control and one dead-zone generator. The PWM Generator provides PWM interrupt flags which are set by hardware when the corresponding PWM period down counter reaches zero. Each PWM interrupt source, with its corresponding enable bit, cangenerate a PWM interrupt request to the CPU. The PWM generator can be configured in one-shot mode to produce only one PWM cycle signal or continuous mode to output a periodic PWM waveform.

When PCR.DZEN01 is set, PWM0 and PWM1 perform complementary paired PWM function; the paired PWM timing, period, duty and dead-time are determined by PWM0 timer and Dead-zone generator0. Refer to Figure 5-25for the architecture of PWM Timers.

To prevent PWM driving glitches to an output pin, the 16-bit period down-counter and 16-bit comparator are implemented with a double buffer. When user writes data to the counter/comparator registers, the updated value will not be load into the 16-bit down-counter/comparator until the down-counter reaches zero.

When the 16-bit period down-counter reaches zero, the interrupt request is generated. If PWMtimer is configuredincontinuous mode, when the down counter reaches zero, it is reloaded with PWM Counter Register (CNRx) automatically and begins decrementing again. Ifthe PWM timer is configured in one-shot mode, the down counter will stop and generate a single interrupt request when it reaches zero.

The value of PWM counter comparator is used for pulse width modulation. The counter control logic inverts the output level when down-counter value matches the value of compare register.

The alternate functionof thePWM-timer is as a digital input capture timer. If Capture function is enabled the PWM output pin is switched as a capture input pin. The Capture0 and PWM0 share one timer whichis included in PWM0; and the Capture1 and PWM1 share PWM1 timer. User must setup the PWM-timer before enabling the Capture feature. After the capture feature is enabled, the count is latched to the Capture Rising Latch Register (CRLR) when input channel has a rising transition and latched to Capture Falling Latch Register(CFLR) when input channel has a falling transition. Capture channel 0 interrupt is programmable by setting CCR0.CRL_IE0[1] (Rising latch Interrupt enable) and CCR0.CFL_IE0[2]] (Falling latch Interrupt enable) to determine the condition of interrupt occurrence. Capture channel 1 has the same feature by setting CCR0.CRL_IE1[17] and CCR0.CFL_IE1[18]. Whenever Capture issues interrupt, the PWM counter will also be reloaded.

5.7.2   Features

5.7.2.1   *PWM function features:*

•   PWM Generator, incorporating an8-bit pre-scaler, clock divider, two PWM-timers (down counters), a dead-zone generator and two PWM outputs.

•   Up to 6PWM channels or three paired PWM channel.

•   16 bits resolution.

•   PWM Interrupt request synchronous with PWM period.

•   Single-shot or Continuous mode PWM.

•   Dead-Zone generator.

5.7.2.2   *Capture Function Features:*

•   Timing control logic shared with PWM Generators.

•   6Capture input channels shared with 6 PWM output channels.

•   Each channel supports a rising latch register (CRLR), a falling latch register (CFLR) and Capture interrupt flag (CAPIFx)

5.7.3   PWM Generator Architecture

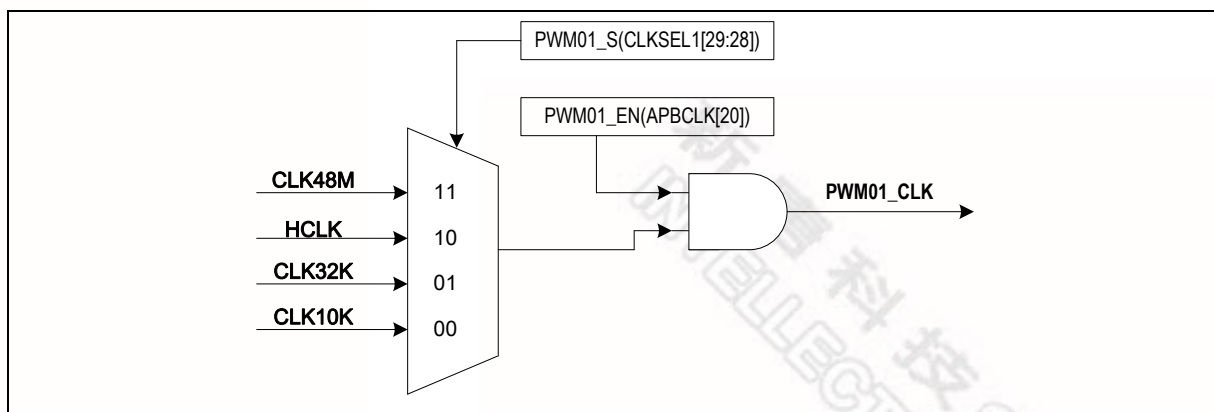The following figures illustrate the architecture of the PWM.

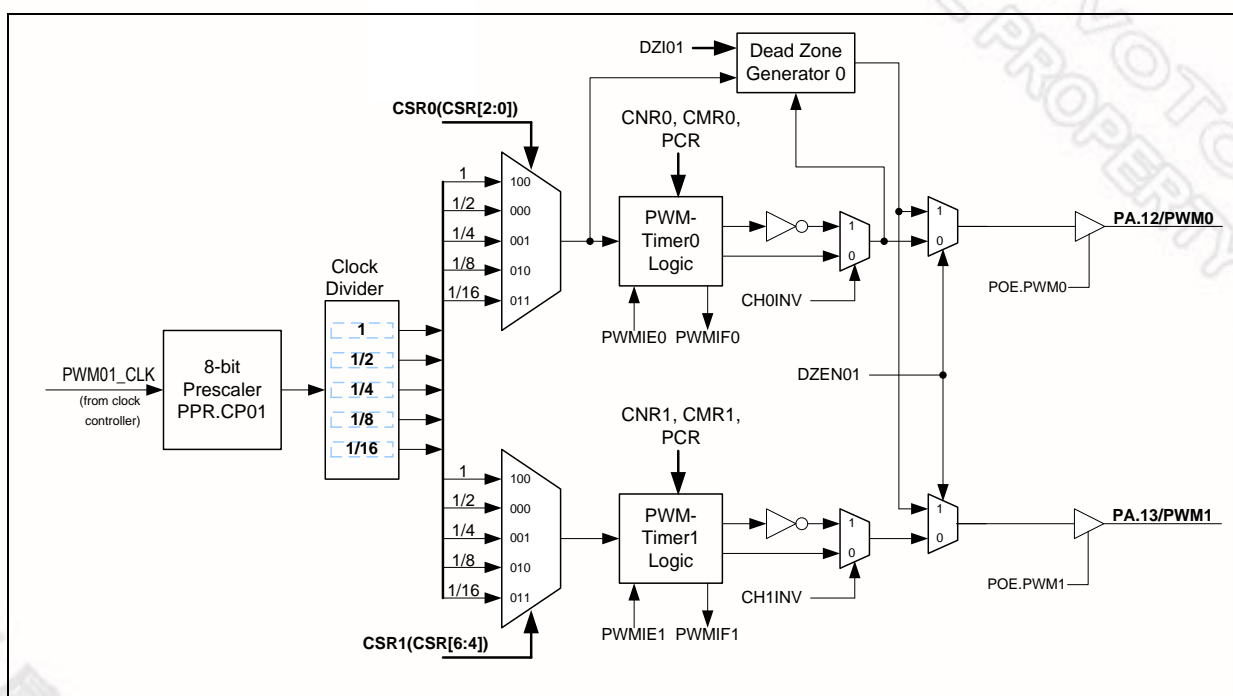Figure 5-24 PWM Generator Clock Source Control

Figure 5-25 PWM Generator Architecture Diagram

5.7.4   PWM-Timer Operation

The PWM period and duty control are configured by the PWM down-counter register (CNR) and PWM comparator register (CMR). Formulas for calculating the pulse width modulation are shown below and demonstrated in Figure 5-26. Note that the corresponding GPIO pins must be configured as the alternate function before PWM function is enabled.

•   PWM frequency = PWM01_CLK/(prescale+1)*(clock divider)/(CNR+1);

•   Duty cycle = (CMR+1)/(CNR+1).

•   CMR >= CNR: PWM output is always high.

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

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

Note: 1. Unit = one PWM clock cycle.

Note: x= 0~1.
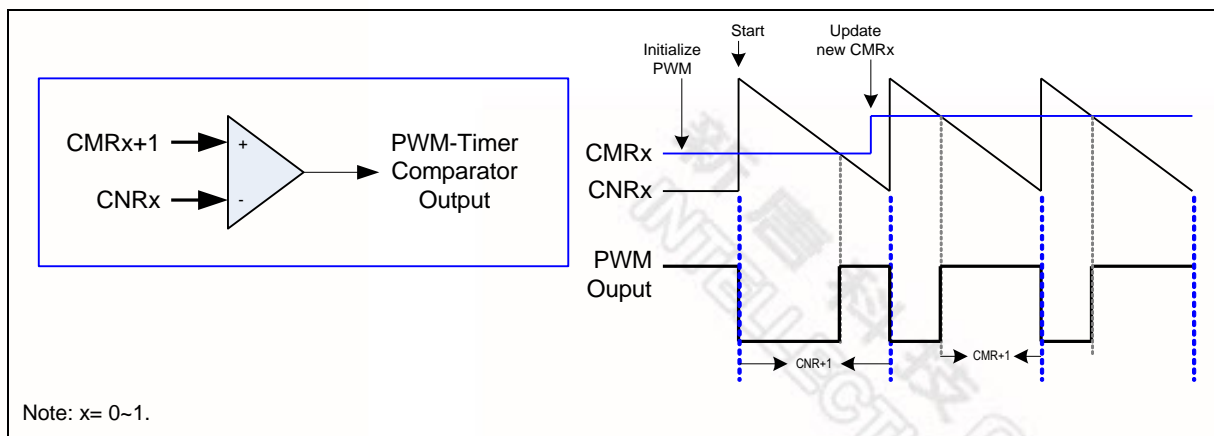
Figure 5-26PWM Generation Timing

The procedure to operate the PWM generator is shown in Figure 5-27. First initialize the PWM settings. At the same time ensure that GPIO are configured to PWM function. Next step is to enable PWM channel. After this, if CNR or CMR register is written by software, it is double buffered until the next counter reload, at which time the registers are updated to new values.
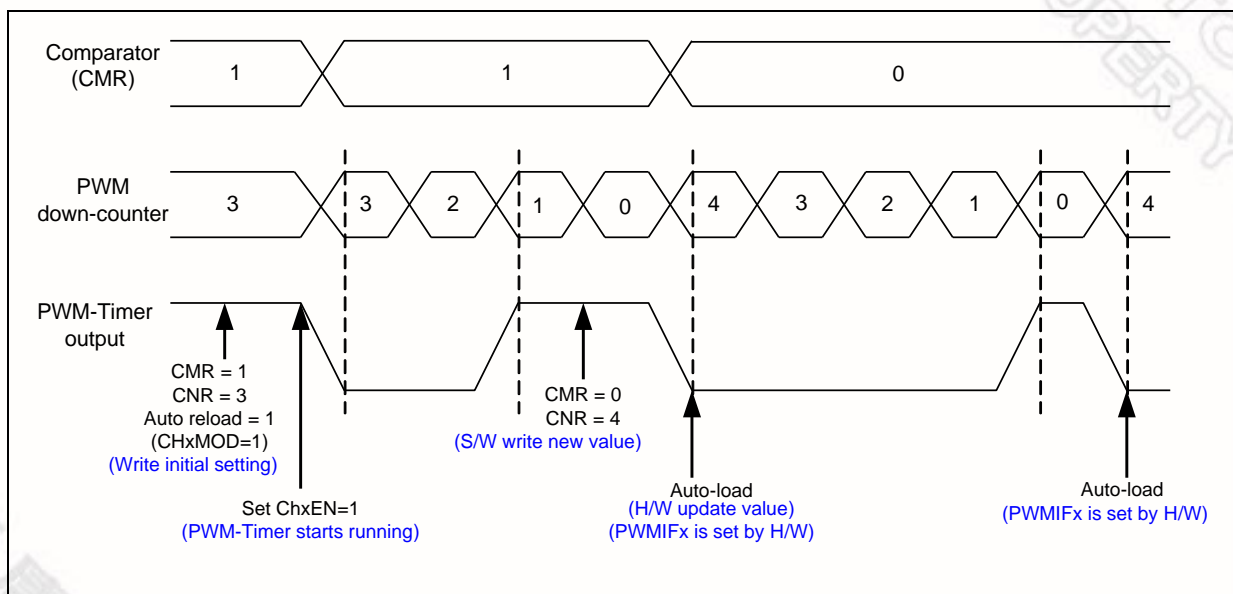

Figure 5-27PWM-Timer Operation Timing

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

The ISD93xx series PWM Timers are double buffered, the reload value is updated at the start of next period without affecting current timer operation. The PWM counter reset value can be written into CNR0~1 and current PWM counter value can be read from PDR0~1.

The bit CHxMOD in PWM Control Register (PCR) determines whether PWMx operates in auto-reload or one-shot mode. If CHxMOD is set to one, the auto-reload operation loads CNRx to PWM counter when PWM counter reaches zero. If CNRx is set to zero, PWM counter will halt when PWM counter counts to zero. If CHxMODis set as zero, counter will stop immediately.
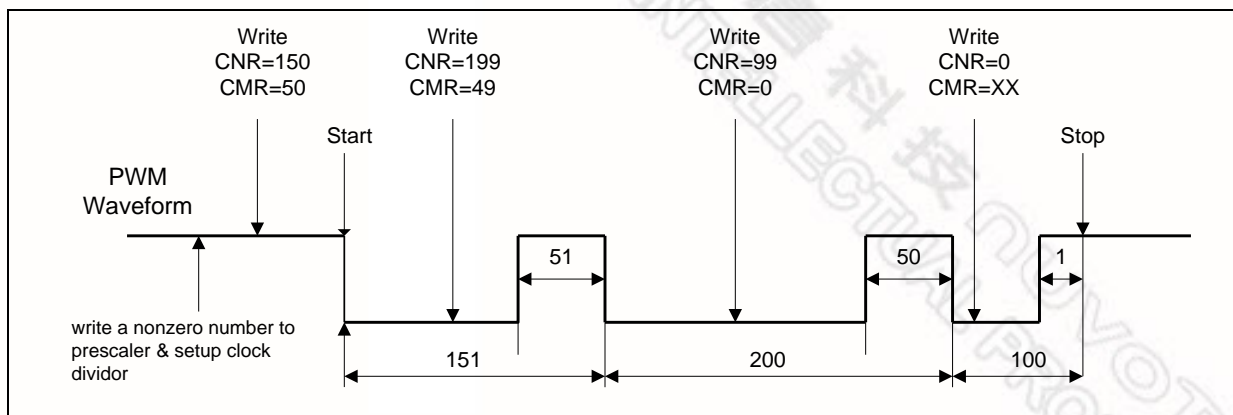
Figure 5-28PWM Double Buffering.

5.7.6 Modulate Duty Cycle

The double buffering allows CMR to be written at any point in current cycle. The loaded value will take effect from next cycle. This is demonstrated in Figure 5-29.
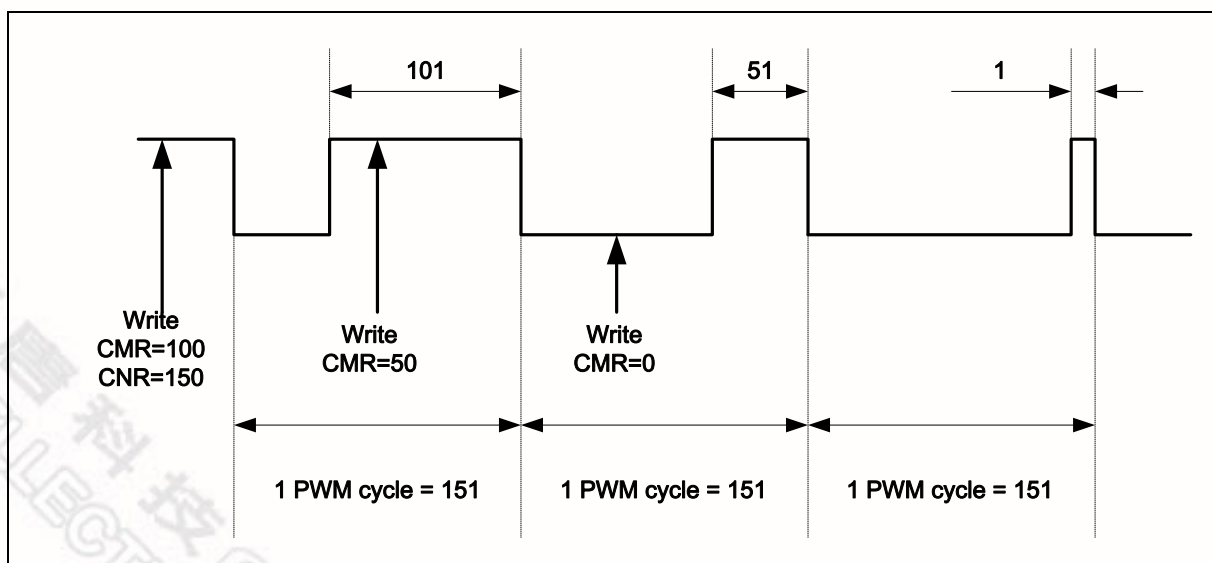
Figure 5-29PWM Controller Duty Cycle Modulation (CNR = 150).

5.7.7 Dead-Zone Generator

The ISD93xx PWM generator includes a Dead Zone generator. This is used to ensure neither PWM output is active simultaneously for power device protection. The function generates a programmable time gap between rising PWM outputs. The user can program PPRx.DZI to determine the Dead Zone interval. The Dead Zone generator behavior is demonstrated in Figure 5-30.
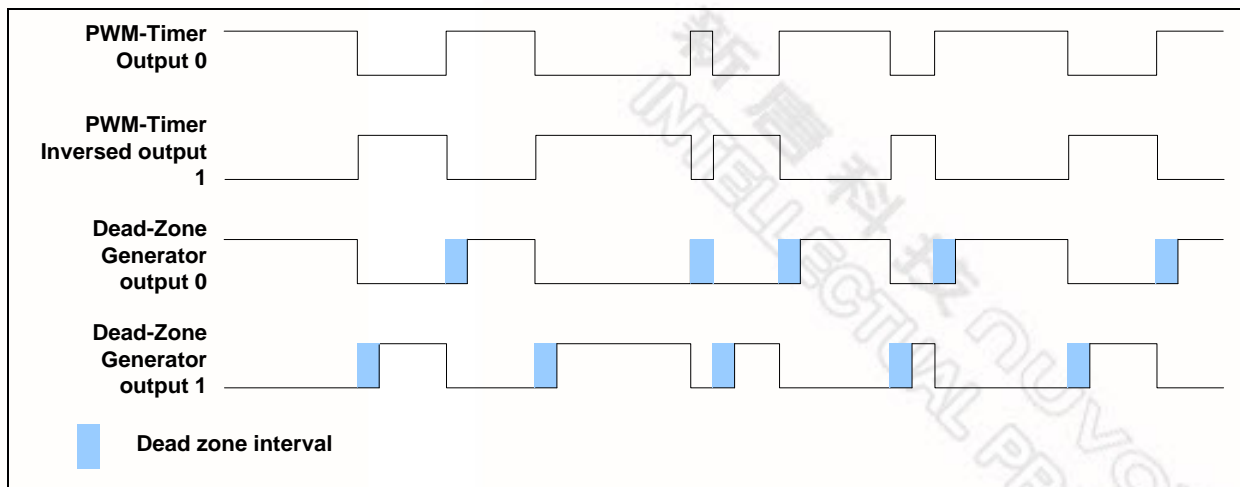
Figure 5-30Paired-PWM Output with Dead Zone Generation Operation

5.7.8 Capture Timer Operation

Instead of using the PWM generator to output a modulated signal, it can be configured as a capture timer to measure a modulated input. Capture channel 0 and PWM0 share one timer and Capture channel 1 and PWM1 share another timer. The capture timer latches PWM-counter to CRLR when input channel has a rising transition and latches PWM-counter to CFLR when input channel has a falling transition. Capture channel 0 interrupt is programmable by setting CCR0[1] (Rising latch Interrupt enable) and CCR0[2] (Falling latch Interrupt enable) to decide the condition of interrupt occurrence. Capture channel 1 has the same feature by setting CCR0[17] and CCR0[18]. Whenever the Capture module issues a capture interrupt, the corresponding PWM counter will be reloaded with CNRx at this moment. Note that the corresponding GPIO pins must be configured as their alternate function before Capture function is enabled.
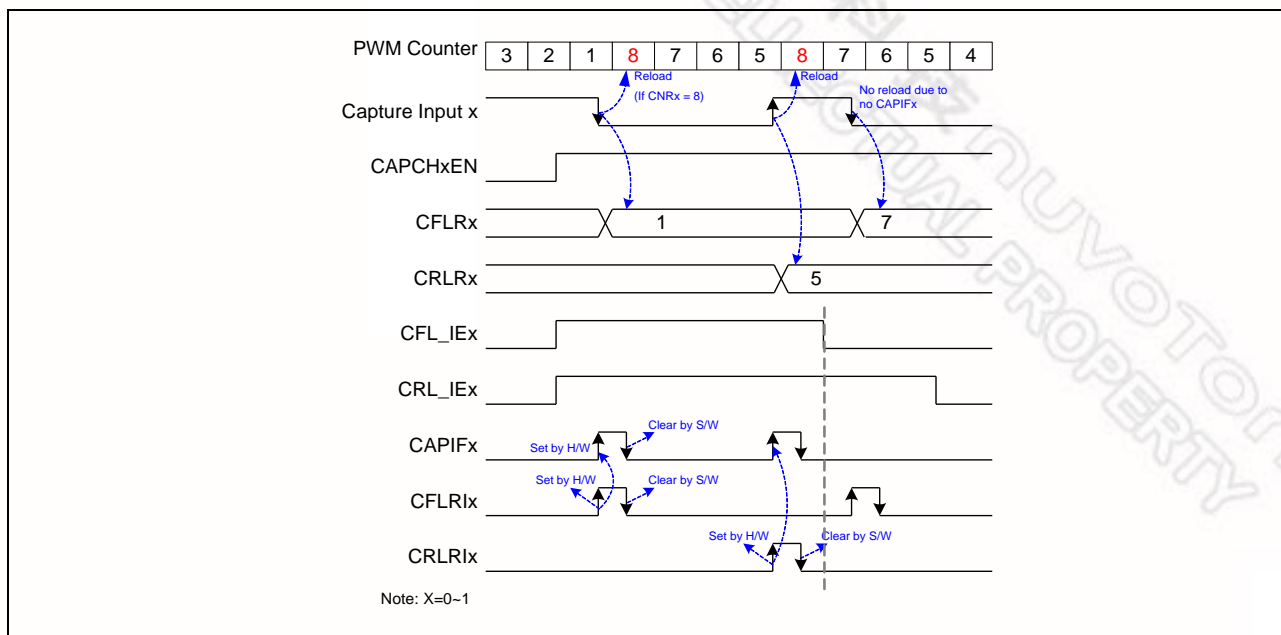
Figure 5-31 Capture Operation Timing

Figure 5-31 demonstrates the case where CNR = 8:

1. The PWM counter will be reloaded with CNRx=8 when a capture interrupt flag (CAPIFx) is set by a transition on the capture input.

2. The channel low pulse width is given by (CNR- CRLR).

3. The channel high pulse width is given by (CNR - CFLR).

5.7.9 PWM-Timer Interrupt Architecture

There are two PWM interrupts, PWM0_INT, PWM1_INT, which are multiplexed into PWMA_IRQ. PWM 0 and Capture 0 share one interrupt, PWM1 and Capture 1 share the same interrupt. Figure 5-32 demonstrates the architecture of PWM-Timer interrupts.
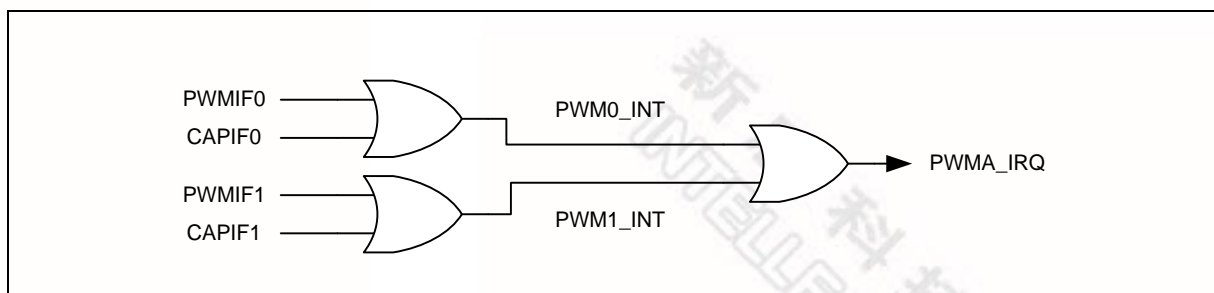
Figure 5-32 PWM-Timer Interrupt Architecture Diagram

5.7.10 PWM-Timer Initialization Procedure

The following procedure is recommended for starting a PWM generator.

1. Setup clock selector (CSR)

2. Setup prescaler (PPR)

3. Setup inverter on/off, dead zone generator on/off, auto-reload/one-shot mode and Stop PWM-timer (PCR)

4. Setup comparator register (CMR)to set PWM duty cycle.

5. Setup PWM down-counter register (CNR) to set PWM period.

6. Setup interrupt enable register (PIER)

7. Setup PWM output enable (POE)

8. Setup the corresponding GPIO pins to PWM function (GPA_ALT)

9. Enable PWM timer start (Set CHxEN = 1 in PCR)

5.7.11 PWM-Timer Stop Procedure

Method 1:
Set 16-bit down counter (CNR) as 0, and monitor PDR (current value of 16-bit down-counter). When PDR reaches to 0, disable PWM-Timer (CHxEN in PCR). *(Recommended)*

Method 2:
Set 16-bit down counter (CNR) as 0. When interrupt request occurs, disable PWM-Timer (CHxEN in PCR). *(Recommended)*

Method 3:
Disable PWM-Timer directly (CHxEN in PCR). *(Not recommended)*

5.7.12 Capture Start Procedure

1. Setup clock selector (CSR)

2. Setup prescaler (PPR)

3. Setup channel enable, rising/falling interrupt enable and input signal inverter on/off (CCR0, CCR1)

4. Setup PWM down-counter (CNR)

5. Set Capture Input Enable Register (CAPENR)

6. Setup the corresponding GPIO pins to PWM function (GPA_ALT)

7. Enable PWM timer start running (Set CHxEN = 1 in PCR)

5.7.13 Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Offset	R/W	Description	Reset Value	Reference
PWMA_BA = 0x4004_0000 (4 PWM Channels 0-3)					
PWMB_BA= 0x4004_0080 (2 PWM Channels only 0-1 active)					
PPR	PWMA_BA+0x000	R/W	PWM Prescaler Register	0x0000_0000	Table 5-57
CSR	PWMA_BA+0x004	R/W	PWM Clock Select Register	0x0000_0000	Table 5-58
PCR	PWMA_BA+0x008	R/W	PWM Control Register	0x0000_0000	Table 5-59
CNR0	PWMA_BA+0x00C	R/W	PWM Counter Register 0	0x0000_0000	Table 5-60
CMR0	PWMA_BA+0x010	R/W	PWM Comparator Register 0	0x0000_0000	Table 5-61
PDR0	PWMA_BA+0x014	R	PWM Data Register 0	0x0000_0000	Table 5-62
CNR1	PWMA_BA+0x018	R/W	PWM Counter Register 1	0x0000_0000	Table 5-60
CMR1	PWMA_BA+0x01C	R/W	PWM Comparator Register 1	0x0000_0000	Table 5-61
PDR1	PWMA_BA+0x020	R	PWM Data Register 1	0x0000_0000	Table 5-62
CNR2	PWMA_BA+0x024	R/W	PWM Counter Register 2	0x0000_0000	Table 5-60
CMR2	PWMA_BA+0x028	R/W	PWM Comparator Register 2	0x0000_0000	Table 5-61
PDR2	PWMA_BA+0x02C	R	PWM Data Register 2	0x0000_0000	Table 5-62
CNR3	PWMA_BA+0x030	R/W	PWM Counter Register 3	0x0000_0000	Table 5-60
CMR3	PWMA_BA+0x034	R/W	PWM Comparator Register 3	0x0000_0000	Table 5-61
PDR3	PWMA_BA+0x038	R	PWM Data Register 3	0x0000_0000	Table 5-62
PIER	PWMA_BA+0x040	R/W	PWM Interrupt Enable Register	0x0000_0000	Table 5-63
PIFR	PWMA_BA+0x044	R/C	PWM Interrupt Flag Register	0x0000_0000	Table 5-64
CCR0	PWMA_BA+0x050	R/W	Capture Control Register 0	0x0000_0000	Table 5-65
CCR1	PWMA_BA+0x054	R/W	Capture Control Register 1	0x0000_0000	Table 5-65
CRLR0	PWMA_BA+0x058	R/W	Capture Rising Latch Register (Channel 0)	0x0000_0000	Table 5-66
CFLR0	PWMA_BA+0x05C	R/W	Capture Falling Latch Register (Channel 0)	0x0000_0000	Table 5-67
CRLR1	PWMA_BA+0x060	R/W	Capture Rising Latch Register (Channel 1)	0x0000_0000	Table 5-66
CFLR1	PWMA_BA+0x064	R/W	Capture Falling Latch Register (Channel 1)	0x0000_0000	Table 5-67
CRLR2	PWMA_BA+0x068	R/W	Capture Rising Latch Register (Channel 2)	0x0000_0000	Table 5-66
CFLR2	PWMA_BA+0x06C	R/W	Capture Falling Latch Register (Channel 2)	0x0000_0000	Table 5-67
CRLR3	PWMA_BA+0x070	R/W	Capture Rising Latch Register (Channel 3)	0x0000_0000	Table 5-66
CFLR3	PWMA_BA+0x074	R/W	Capture Falling Latch Register (Channel 3)	0x0000_0000	Table 5-67

CAPENR	PWMA_BA+0x078	R/W	Capture Input 0~1 Enable Register	0x0000_0000	Table 5-68
POE	PWMA_BA+0x07C	R/W	PWM Output Enable for PWM0~PWM3	0x0000_0000	Table 5-69

5.7.14 Register Description

PWM Pre-Scale Register (PPR)

Register	Offset	R/W	Description	Reset Value
PPR	PWMA_BA+0x000	R/W	PWM Pre-scale Register	0x0000_0000

31	30	29	28	27	26	25	24
DZI23							
23	22	21	20	19	18	17	16
DZI01							
15	14	13	12	11	10	9	8
CP23							
7	6	5	4	3	2	1	0
CP01							

Table 5-57 PWM Pre-Scaler Register (PPR, address 0x4004_0000).

Bits	Descriptions	
[23:16]	DZI01	**Dead zone interval register for pair of PWM0 and PWM1** These 8 bits determine dead zone length. The unit time of dead zone length is that from clock selector 0.
[7:0]	CP01	**Clock pre-scaler** Clock input is divided by (CP01 + 1) **If CP01=0, then the pre-scaler output clock will be stopped.** This implies PWM counter 0 and 1 will also be stopped.

PWM Clock Select Register (CSR)

Register	Offset	R/W	Description	Reset Value
CSR	PWMA_BA+0x004	R/W	PWM Clock Select Register	0x0000_0000

15	14	13	12	11	10	9	8
Reserved	CSR3			Reserved	CSR2		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	CSR1			Reserved	CSR0		

Table 5-58 PWM Clock Select Register (CSR, address 0x4004_0004).

Bits	Descriptions	
[31:7]	-	**Reserved**
[6:4]	**CSR1**	**Timer 1 Clock Source Selection** <table><tr><td>CSR1</td><td>Input clock divided by</td></tr><tr><td>100</td><td>1</td></tr><tr><td>011</td><td>16</td></tr><tr><td>010</td><td>8</td></tr><tr><td>001</td><td>4</td></tr><tr><td>000</td><td>2</td></tr></table>
[2:0]	**CSR0**	**Timer 0 Clock Source Selection** (Table is as CSR1)

PWM Control Register (PCR)

Register	Offset	R/W	Description	Reset Value
PCR	PWMA_BA+0x008	R/W	PWM Control Register (PCR)	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				CH3MOD	CH3INV	Reserved	CH3EN
23	22	21	20	19	18	17	16
Reserved				CH2MOD	CH2INV	Reserved	CH2EN
15	14	13	12	11	10	9	8
Reserved				CH1MOD	CH1INV	Reserved	CH1EN
7	6	5	4	3	2	1	0
Reserved		DZEN23	DZEN01	CH0MOD	CH0INV	Reserved	CH0EN

Table 5-59 PWM Control Register (PCR, address 0x4004_008).

Bits	Descriptions	
[11]	CH1MOD	**PWM-Timer 1 Auto-reload/One-Shot Mode** 1 = Auto-load Mode 0 = One-Shot Mode **Note**: Arising transition of this bit will cause CNR1 and CMR1 to be cleared.
[10]	CH1INV	**PWM-Timer 1 Output Inverter ON/OFF** 1 = Inverter ON, 0 = Inverter OFF
[8]	CH1EN	**PWM-Timer 1 Enable/Disable Start Run** 1 = Enable PWM-Timer 1 Start/Run, 0 = Stop PWM-Timer 1
[4]	DZEN01	**Dead-Zone 0 Generator Enable/Disable** 1 = Enable, 0 = Disable **Note**: When Dead-Zone Generator is enabled, the pair of PWM0 and PWM1 become a complementary pair.
[3]	CH0MOD	**PWM-Timer 0 Auto-reload/One-Shot Mode** 1 = Auto-reload Mode 0 = One-Shot Mode **Note**: A rising transition of this bit will cause CNR0 and CMR0 to be cleared.
[2]	CH0INV	**PWM-Timer 0 Output Inverter ON/OFF** 1 = Inverter ON 0 = Inverter OFF

[0]	CH0EN	**PWM-Timer 0 Enable/Disable Start Run**
		1 = Enable PWM-Timer 0 Start/Run, 0 = Stop PWM-Timer 0 Running

PWM Counter Register 1-0 (CNR1-0)

Register	Offset	R/W	Description	Reset Value
CNR0	PWMA_BA+0x00C	R/W	PWM Counter Register 0	0x0000_0000
CNR1	PWMA_BA+0x018	R/W	PWM Counter Register 1	0x0000_0000

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

Table 5-60 PWM Counter Register (CNRx, address 0x4004_00C+C*x).

Bits	Descriptions	
[15:0]	**CNR**	**PWM Counter/Timer Reload Value** CNR determines the PWM period. • PWM frequency = PWM01_CLK/(prescale+1)*(clock divider)/(CNR+1); • Duty ratio = (CMR+1)/(CNR+1). • CMR >= CNR: PWM output is always high. • CMR < CNR: PWM low width = (CNR-CMR) unit; PWM high width = (CMR+1) unit. • CMR = 0: PWM low width = (CNR) unit; PWM high width = 1 unit (Unit = one PWM clock cycle) **Note:** Any write to CNR will take effect in next PWM cycle.

PWM Comparator Register 1-0 (CMR1-0)

Register	Offset	R/W	Description	Reset Value
CMR0	PWMA_BA+0x010	R/W	PWM Comparator Register 0	0x0000_0000
CMR1	PWMA_BA+0x01C	R/W	PWM Comparator Register 1	0x0000_0000

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

Table 5-61 PWM Comparator Register (CMRx, address 0x4004_0010 + C*x).

Bits	Descriptions	
[15:0]	**CMR**	**PWM Comparator Register** CMR determines the PWM duty cycle. • PWM frequency = PWM01_CLK/(prescale+1)*(clock divider)/(CNR+1); • Duty Cycle = (CMR+1)/(CNR+1). • CMR >= CNR: PWM output is always high. • CMR < CNR: PWM low width = (CNR-CMR) unit; PWM high width = (CMR+1) unit. • CMR = 0: PWM low width = (CNR) unit; PWM high width = 1 unit (Unit = one PWM clock cycle) **Note:** Any write to CMR will take effect in next PWM cycle.

PWM Data Register 1-0 (PDR1/PDR0)

Register	Offset	R/W	Description	Reset Value
PDR0	PWMA_BA+0x014	R	PWM Data Register 0	0x0000_0000
PDR1	PWMA_BA+0x020	R	PWM Data Register 1	0x0000_0000

15	14	13	12	11	10	9	8
PDR[15:8]							
7	6	5	4	3	2	1	0
PDR[7:0]							

Table 5-62 PWM Data Register (PDRx, address 0x4004_0014 + C*x).

Bits	Descriptions	
[15:0]	**PDR**	**PWM Data Register** Reports the current value of the 16-bit down counter.

PWM Interrupt Enable Register (PIER)

Register	Offset	R/W	Description	Reset Value
PIER	PWMA_BA+0x040	R/W	PWM Interrupt Enable Register	0x0000_0000

7	6	5	4	3	2	1	0
Reserved				PWMIE3	PWMIE2	PWMIE1	PWMIE0

Table 5-63 PWM Interrupt Enable Register (PIER, address 0x4004_0040).

Bits	Descriptions	
[n]	**PWMIEn**	**PWM Timer n Interrupt Enable** 1 = Enable, 0 = Disable
[0]	**PWMIE0**	**PWM Timer 0 Interrupt Enable** 1 = Enable, 0 = Disable

PWM Interrupt Flag Register (PIFR)

Register	Offset	R/W	Description	Reset Value
PIFR	PWMA_BA+0x044	R/W	PWM Interrupt Flag Register	0x0000_0000

7	6	5	4	3	2	1	0
Reserved				PWMIF3	PWMIF2	PWMIF1	PWMIF0

Table 5-64 PWM Interrupt Flag Register (PIFR, address 0x4004_0044).

Bits	Descriptions	
[n]	**PWMIFn**	**PWM Timer n Interrupt Flag** Flag is set by hardware when PWMn down counter reaches zero, software can clear this bit by writing '1' to it.
[0]	**PWMIF0**	**PWM Timer 0 Interrupt Flag** Flag is set by hardware when PWM0 down counter reaches zero, software can clear this bit by writing '1' to it.

Note: User can clear each interrupt flag by writing a one to corresponding bit in PIFR.

Capture Control Register (CCR0)

Register	Offset	R/W	Description	Reset Value
CCR0	PWMA_BA+0x050	R/W	Capture Control Register	0x0000_0000

31	30	29	28	27	26	25	24
				Reserved			
23	22	21	20	19	18	17	16
CFLRI1	CRLRI1	Reserved	CAPIF1	CAPCH1EN	CFL_IE1	CRL_IE1	INV1
15	14	13	12	11	10	9	8
				Reserved			
7	6	5	4	3	2	1	0
CFLRI0	CRLRI0	Reserved	CAPIF0	CAPCH0EN	CFL_IE0	CRL_IE0	INV0

Table 5-65 Capture Control Register (CCR0, address 0x4004_0050).

Bits	Descriptions	
[23]	CFLRI1	**CFLR1 Latched Indicator Bit** When input channel 1 has a **falling** transition, CFLR1 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing a *zero* to it.
[22]	CRLRI1	**CRLR1 Latched Indicator Bit** When input channel 1 has a **rising** transition, CRLR1 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing a *zero* to it.
[20]	CAPIF1	**Capture1 Interrupt Indication Flag** If channel 1 rising latch interrupt is enabled (CRL_IE1=1), a rising transition at input channel 1 will result in CAPIF1 to high; Similarly, a falling transition will cause CAPIF1 to be set high if channel 1 falling latch interrupt is enabled (CFL_IE1=1). This flag is cleared by software writing a '1' to it.
[19]	CAPCH1EN	**Capture Channel 1 transition Enable/Disable** 1 = Enable capture function on channel 1.0 = Disable capture function on channel 1 When enabled, Capture function latches the PMW-counter to CRLR (Rising latch) and CFLR (Falling latch) registers on input edge transition. When disabled, Capture function is inactive as is interrupt.
[18]	CFL_IE1	**Channel 1 Falling Latch Interrupt Enable** 1 = Enable falling edge latch interrupt. 0 = Disable falling edge latch interrupt When enabled, capture block generates an interrupt on falling edge of input.

[17]	CRL_IE1	**Channel 1 Rising Latch Interrupt Enable** 1 = Enable rising edge latch interrupt. 0 = Disable rising edge latch interrupt When enabled, capture block generates an interrupt on rising edge of input.
[16]	INV1	**Channel 1 Inverter ON/OFF** 1 = Inverter ON. Reverse the input signal from GPIO before Capture timer 0 = Inverter OFF
[7]	CFLRI0	**CFLR0 Latched Indicator Bit** When input channel 0 has a **falling** transition, CFLR0 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing a *zero* to it.
[6]	CRLRI0	**CRLR0 Latched Indicator Bit** When input channel 0 has a **rising** transition, CRLR0 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing a *zero* to it.
[4]	CAPIF0	**Capture0 Interrupt Indication Flag** If channel 0 rising latch interrupt is enabled (CRL_IE0=1), a rising transition at input channel 0 will result in CAPIF0 to high; Similarly, a falling transition will cause CAPIF0 to be set high if channel 0 falling latch interrupt is enabled (CFL_IE0=1). This flag is cleared by software writing a '1' to it.
[3]	CAPCH0EN	**Capture Channel 0 transition Enable/Disable** 1 = Enable capture function on channel0.0 = Disable capture function on channel0 When enabled, Capture function latches the PMW-counter to CRLR (Rising latch) and CFLR (Falling latch) registers on input edge transition. When disabled, Capture function is inactive as is interrupt.
[2]	CFL_IE0	**Channel 0 Falling Latch Interrupt Enable ON/OFF** 1 = Enable falling latch interrupt. 0 = Disable falling latch interrupt When enabled, capture block generates an interrupt on falling edge of input.
[1]	CRL_IE0	**Channel 0 Rising Latch Interrupt Enable ON/OFF** 1 = Enable rising latch interrupt. 0 = Disable rising latch interrupt When enabled, capture block generates an interrupt on rising edge of input.
[0]	INV0	**Channel 0 Inverter ON/OFF** 1 = Inverter ON. Reverse the input signal from GPIO before Capture timer 0 = Inverter OFF

Capture Rising Latch Register1-0 (CRLR0 CRLR1)

Register	Offset	R/W	Description	Reset Value
CRLR0	PWMA_BA+0x058	R	Capture Rising Latch Register (channel 0)	0x0000_0000
CRLR1	PWMA_BA+0x060	R	Capture Rising Latch Register (channel 1)	0x0000_0000

15	14	13	12	11	10	9	8
CRLR[15:8]							
7	6	5	4	3	2	1	0
CRLR[7:0]							

Table 5-66 Capture Rising Latch Register (CRLRx, address 0x4004_0058 +C*x).

Bits	Descriptions	
[15:0]	CRLR	In Capture mode, this register is latched with the value of the PWM counter on a rising edge of the input signal.

Capture Falling Latch Register1-0 (CFLR1 CFLR0)

Register	Offset	R/W	Description	Reset Value
CFLR0	PWMA_BA+0x05C	R	Capture Falling Latch Register (channel 0)	0x0000_0000
CFLR1	PWMA_BA+0x064	R	Capture Falling Latch Register (channel 1)	0x0000_0000

15	14	13	12	11	10	9	8
CFLR[15:8]							
7	6	5	4	3	2	1	0
CFLR[7:0]							

Table 5-67 Capture Falling Latch Register (CFLRx, address 0x4004_005C + C*x).

Bits	Descriptions	
[15:0]	CFLR	In Capture mode, this register is latched with the value of the PWM counter on a falling edge of the input signal.

Capture Input Enable Register (CAPENR)

Register	Offset	R/W	Description	Reset Value
CAPENR	PWMA_BA+0x078	R/W	Capture Input Enable Register	0x0000_0000

7	6	5	4	3	2	1	0
Reserved						CAPENR[1:0]	

Table 5-68 Capture Input Enable Register (CAPENR, address 0x4004_0078).

Bits	Descriptions	
[3:0]	CAPENR	Capture Input Enable Register 0= OFF (GPA[13:12] pin input disconnected from Capture block) 1= ON (GPA[13:12] pin, if in PWM alternative function, will be configured as an input and fed to capture function) CAPENR[1:0] <u>Bit 10</u> Bit x1➜ Capture channel 0 is from GPA [12] Bit 1x➜ Capture channel 1 is from GPA[13]

PWM Output Enable Register (POE) for PWM0~PWM1

Register	Offset	R/W	Description	Reset Value
POE	PWMA_BA+0x07C	R/W	PWM Output Enable Register for PWM0~PWM1	0x0000_0000

7	6	5	4	3	2	1	0
Reserved				PWM3	PWM2	PWM1	PWM0

Table 5-69 PWM Output Enable (POE, address 0x4004_007C).

Bits	Descriptions	
[n]	PWMn	**PWMn Output Enable Register** 1 = Enable PWMn output to pin. 0 = Disable PWMn output to pin. **Note**: The corresponding GPIO pin also must be switched to PWM function (refer to GPA_ALT Table 5-7)
[0]	PWM0	**PWM0 Output Enable Register** 1 = Enable PWM0 output to pin.0 = Disable PWM0 output to pin. **Note**: The corresponding GPIO pin also must be switched to PWM function (refer to GPA_ALT Table 5-7)

5.8   Real Time Clock (RTC)

5.8.1   Overview

Real Time Clock (RTC) unit provides real time clock, calendar and alarm functions. The clock source of the RTC is an external 32.768 kHz crystal connected at pins XI32K and XO32K or from an external 32.768 kHz oscillator output fed to pin XI32K. The RTC unit provides the time (second, minute, hour) in Time Load Register (TLR) as well as calendar (day, month, year) in Calendar Load Register (CLR). The data is expressed in BCD (Binary Coded Decimal) format. The unit offers an alarm function whereby the user can preset the alarm time in the Time Alarm Register (TAR) and alarm calendar in Calendar Alarm Register (CAR).

The RTC unit supports periodic Time-Tick and Alarm-Match interrupts. The periodic interrupt has 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second which are selected by TTR (TTR[2:0]). When RTC counter in TLR and CLR is equal to alarm setting registers TAR and CAR, the alarm interrupt flag (RIIR.AIF) is set and the alarm interrupt is requested if the alarm interrupt is enabled (RIER.AIER=1). The RTC Time Tick and Alarm Match can wake the CPU from sleep mode or Standby Power-Down (SPD) mode if the Wakeup CPU function is enabled (TWKE (TTR[3])=1).

5.8.2   RTC Features

- Consists of a time counter (second, minute, hour) and calendar counter (day, month, year).

- Alarm register (second, minute, hour, day, month, year).

- 12-hour or 24-hour mode is selectable.

- Automatic leap year compensation.

- Day of week counter.

- Frequency compensate register (FCR).

- All time and calendar registers are expressed in BCD code.

- Support periodic time tick interrupt with 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second.

- Support RTC Time-Tick and Alarm-Match interrupt

- Support CPU wakeup from sleep or standby power-down mode.
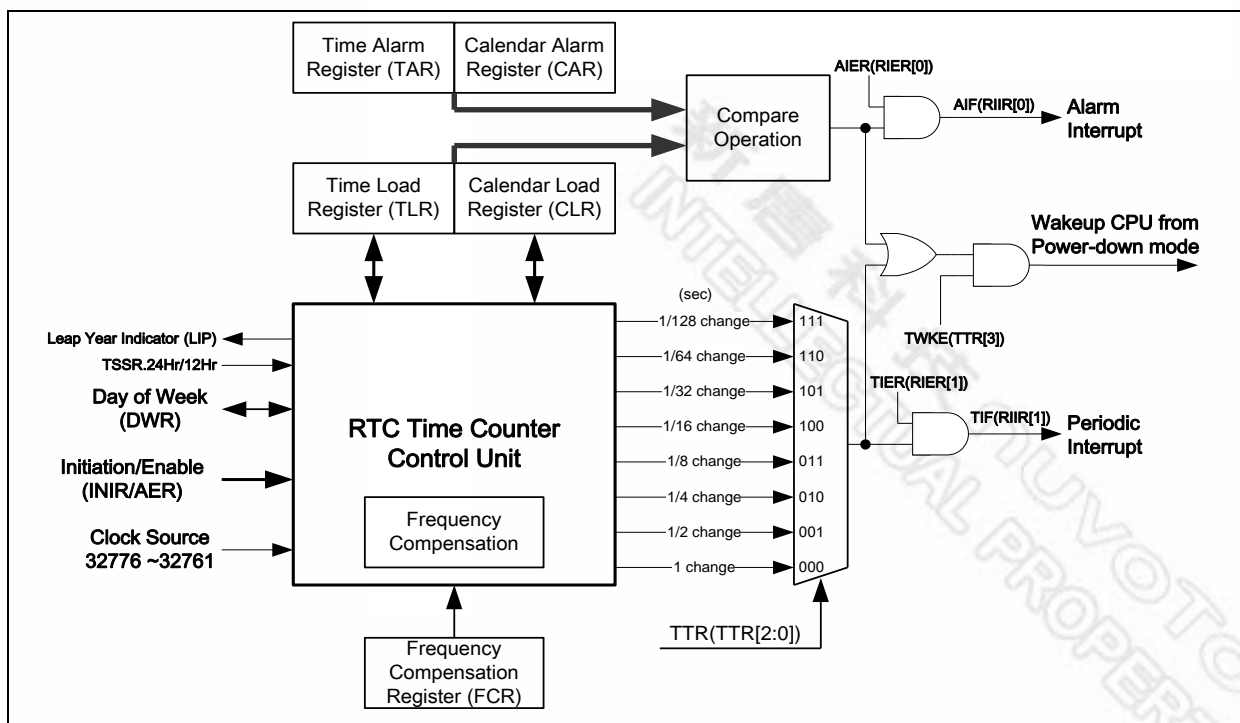
5.8.3　RTC Block Diagram

Figure 5-33 RTC Block Diagram

5.8.4 RTC Function Description

5.8.4.1 *Access to RTC register*

Due to clock frequency difference between RTC clock and system clock, when the user writes new data to any one of the RTC registers, the register will not be updated until 2 RTC clock periods later (60us). The programmer should take this into consideration for determining access sequence between TSSR, TAR and TLR.

In addition, the RTC block does not check whether written data is out of bounds for a valid BCD time or calendar load. RTC does not check validity of DWR and CLR write either.

5.8.4.2 *RTC Initiation*

When RTC block is powered on, programmer must write 0xA5EB1357 to INIR register to reset all logic. INIR acts as a hardware reset circuit. Once INIR has been set to0xA5EB1357, internal reset operation begins. When reset operation is finished, INIR[0] is set by hardware and RTC is ready for operation.

5.8.4.3 *RTC Read/Write Enable*

Register AER[15:0] serves as the RTC read/write password to protect RTC registers. AER[15:0] have to be set to 0xA965 to enable access. Once set, it will take effect 512 RTC clocks later (about 15ms). Programmer can read RTC enabled status flag in AER.ENF to check whether RTC is access enabled. Access is automatically cleared after 200ms.

5.8.4.4 *Frequency Compensation*

The RTC Frequency Compensation Register (FCR)allows software to configure digital compensation to the 32768Hz clock input. The FCR allows compensation of a clock input in the range from 32761Hz to 32776Hz. If desired, RTC clock can be measured during manufacture from a GPIO pin and compensation value calculated and stored in flash memory for retrieval when the product is first powered on. Following are compensation examples for a higher or lower measured frequency clock input.

Example 1:

> Frequency counter measurement : 32773.65Hz (> 32768 Hz)
>
> Integer part: 32773 = 0x8005
>
> FCR. INTEGER= (32773 – 32761) = 12 = 0x0C
>
> Fractional part: 0.65 x 60 = 39 = 0x27
>
> FCR. FRACTION= 0x27

Example 2

> Frequency counter measurement : 32765.27Hz (< 32768 Hz)
>
> Integer part: 32765 = 0x7ffd
>
> FCR.INTEGER = (32765 – 32761) = 4 = 0x04
>
> Fractional part: 0.27 x 60 = 16.2 = 0x10
>
> FCR.FRACTION = 0x10

5.8.4.5 *Time and Calendar counter*

TLR and CLR are used to load the time and calendar. TAR and CAR are used to set the alarm. They are all represented by a BCD format, see register descriptions for digit assignments.

5.8.4.6 *12/24 hour Time Scale Selection*

RTC can be selected to report time in either a 12 or 24hour time scale. If 12 hour mode is selected then AM/PM indication is provided by the hour digit being >=2, see register description Table 5-76 for details. The 12/24 hour time scale selection depends on TSSR bit 0.

5.8.4.7   *Day of the week counter*

The RTC unit provides day of week in Day of the Week Register (DWR). The value is defined from 0 to 6 to represent Sunday to Saturday respectively.

5.8.4.8   *Periodic Time Tick Interrupt*

The periodic interrupt has 8 period option 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second which are selected by TTR.TTR[2:0]. When periodic time tick interrupt is enabled by setting RIER.TIER to 1, the Periodic Time Tick Interrupt is requested as selected by TTR register.

5.8.4.9   *Alarm Time Interrupt*

When RTC counter in TLR and CLR is equal to alarm setting in TAR and CAR the alarm interrupt flag (RIIR.AIF) is set. If alarm interrupt is enabled (RIER.AIER=1) the alarm interrupt is also requested.

5.8.4.10   *Additional Notes*

1. TAR, CAR, TLR and CLR registers are all BCD counter.

2. Programmer has to make sure that values loaded are reasonable. For example, some invalid CLR values would be 201a (year), 13 (month), 00 (day).

3. Reset state :

Register	Reset State
AER	0
CLR	05/1/1 (year/month/day)
TLR	00:00:00 (hour : minute : second)
CAR	00/00/00 (year/month/day)
TAR	00:00:00 (hour : minute : second)
TSSR	1 (24 hr. mode)
DWR	6 (Saturday)
RIER	0
RIIR	0
LIR	0
TTR	0
PWRTOUT	5555

4. In TLR and TAR, only 2 BCD digits are used to express "year". It is assumed that 2 BCD digits of xY denote 20xY, but not 19xY or 21xY.

5.8.5 Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Offset	R/W	Description	Reset Value	Reference
RTC_BA = 0x4000_8000					
INIR	RTC_BA+0x000	R/W	RTC Initialization Register	0x0000_0000	Table 5-70
AER	RTC_BA+0x004	R/W	RTC Access Enable Register	0x0000_0000	Table 5-71
FCR	RTC_BA+0x008	R/W	RTC Frequency Compensation Register	0x0000_0700	Table 5-73
TLR	RTC_BA+0x00C	R/W	Time Load Register	0x0000_0000	Table 5-74
CLR	RTC_BA+0x010	R/W	Calendar Load Register	0x0005_0101	Table 5-75
TSSR	RTC_BA+0x014	R/W	Time Scale Selection Register	0x0000_0001	Table 5-76
DWR	RTC_BA+0x018	R/W	Day of the Week Register	0x0000_0006	Table 5-77
TAR	RTC_BA+0x01C	R/W	Time Alarm Register	0x0000_0000	Table 5-78
CAR	RTC_BA+0x020	R/W	Calendar Alarm Register	0x0000_0000	Table 5-79
LIR	RTC_BA+0x024	R	Leap year Indicator Register	0x0000_0000	Table 5-80
RIER	RTC_BA+0x028	R/W	RTC Interrupt Enable Register	0x0000_0000	Table 5-81
RIIR	RTC_BA+0x02C	R/C	RTC Interrupt Indicator Register	0x0000_0000	Table 5-82
TTR	RTC_BA+0x030	R/W	RTC Time Tick Register	0x0000_0000	Table 5-83

5.8.6 Register Description

RTC Initiation Register (INIR)

Register	Offset	R/W	Description	Reset Value
INIR	RTC_BA+0x000	R/W	RTC Initiation Register	0x0000_0000

31	30	29	28	27	26	25	24
INIR							

23	22	21	20	19	18	17	16
INIR							

15	14	13	12	11	10	9	8
INIR							

7	6	5	4	3	2	1	0
INIR							INIR/Active

Table 5-70 RTC Initialization Register (INIR, address 0x4000_8000).

Bits	Descriptions	
[31:0]	**INIR**	**RTC Initialization** After a power-on reset (POR) RTC block should be initialized by writing 0xA5EB1357 to INIR. This will force a hardware reset then release all logic and counters.
[0]	**Active**	**RTC Active Status** (Read only), 0: RTC is in reset state 1: RTC is in normal active state.

RTC Access Enable Register (AER)

Register	Offset	R/W	Description	Reset Value
AER	RTC_BA+0x004	R/W	RTC Access Enable Register	0x0000_0000

23	22	21	20	19	18	17	16
Reserved							ENF
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
AER							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AER							

Table 5-71 RTC Access Enable Register (AER, address 0x4000_8004).

Bits	Descriptions	
[16]	**ENF**	**RTC Register Access Enable Flag** (Read only) 1 = RTC register read/write enable. 0 = RTC register read/write disable This bit will be set after AER[15:0] register is set to 0xA965, it will clear automatically in 512 RTC clock cycles or AER[15:0] != 0xA965. The effect of AER.ENF on access to each register is given Table 5-72.	Table 5-72 AER.ENF Register Access Effect. See table below
[15:0]	**AER**	**RTC Register Access Enable Password** (Write only) 0xA965 = Enable RTC access Others = Disable RTC access	

Table 5-72 AER.ENF Register Access Effect.

Register \ AER.ENF	1	0
INIR	R/W	R/W
FCR	R/W	-
TLR	R/W	R
CLR	R/W	R
TSSR	R/W	R/W
DWR	R/W	R
TAR	R/W	-
CAR	R/W	-
LIR	R	R
RIER	R/W	R/W
RIIR	R/C	R/C
TTR	R/W	-

RTC Frequency Compensation Register (FCR)

Register	Offset	R/W	Description	Reset Value

FCR	RTC_BA+0x008	R/W	Frequency Compensation Register	0x0000_0700

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				INTEGER			
7	6	5	4	3	2	1	0
Reserved			FRACTION				

Table 5-73 RTC Frequency Compensation Register (FCR, address 0x4000_8008).

Bits	Descriptions	
[11:8]	INTEGER	**Integer Part.** Register should contain the value (INT(F_{actual}) – 32761) <table><tr><th>Integer part of detected value</th><th>FCR[11:8]</th><th>Integer part of detected value</th><th>FCR[11:8]</th></tr><tr><td>32776</td><td>1111</td><td>32768</td><td>0111</td></tr><tr><td>32775</td><td>1110</td><td>32767</td><td>0110</td></tr><tr><td>32774</td><td>1101</td><td>32766</td><td>0101</td></tr><tr><td>32773</td><td>1100</td><td>32765</td><td>0100</td></tr><tr><td>32772</td><td>1011</td><td>32764</td><td>0011</td></tr><tr><td>32771</td><td>1010</td><td>32763</td><td>0010</td></tr><tr><td>32770</td><td>1001</td><td>32762</td><td>0001</td></tr><tr><td>32769</td><td>1000</td><td>32761</td><td>0000</td></tr></table>
[5:0]	FRACTION	**Fractional Part** Formula = (fraction part of detected value) x 60 Refer to 5.8.4.4 for the examples.

Note: This register can be read back after the RTC enable is active.

RTC Time Load Register (TLR)

This register is Read Only until access enable password is written to **AER** register. The register returns the current time.Timer0 Control and Status Register Timer0 Control and Status Register Timer0 Control and Status Register Timer0 Control and Status Register Timer0 Control and Status Register Timer0 Control and Status Register

Register	Offset	R/W	Description	Reset Value
TLR	RTC_BA+0x00C	R/W	Time Load Register	0x0000_0000

23	22	21	20	19	18	17	16
Reserved		10HR		1HR			
15	14	13	12	11	10	9	8
Reserved	10MIN			1MIN			
7	6	5	4	3	2	1	0
Reserved	10SEC			1SEC			

Table 5-74 RTC Time Load Register (TLR, address 0x4000_800C).

Bits	Descriptions	
[21:20]	**10HR**	10 Hour Time Digit (0~3)[2]
[19:16]	**1HR**	1 Hour Time Digit (0~9)
[14:12]	**10MIN**	10 Min Time Digit (0~5)
[11:8]	**1MIN**	1 Min Time Digit (0~9)
[6:4]	**10SEC**	10 Sec Time Digit (0~5)
[3:0]	**1SEC**	1 Sec Time Digit (0~9)

Note:

1. TLR is a BCD counter and RTC will not check loaded data for validity.

2. Valid range is listed in the parenthesis.

RTC Calendar Load Register (CLR)

This register is Read Only until access enable password is written to **AER** register. The register returns the current date.

Register	Offset	R/W	Description	Reset Value
CLR	RTC_BA+0x010	R/W	Calendar Load Register	0x0005_0101

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

Table 5-75 RTC Calendar Load Register (CLR, address 0x4000_80010).

Bits	Descriptions	
[23:20]	**10YEAR**	10-Year Calendar Digit (0~9)[1]
[19:16]	**1YEAR**	1-Year Calendar Digit (0~9)
[12]	**10MON**	10-Month Calendar Digit (0~1)
[11:8]	**1MON**	1-Month Calendar Digit (0~9)
[5:4]	**10DAY**	10-Day Calendar Digit (0~3)
[3:0]	**1DAY**	1-Day Calendar Digit (0~9)

---

[1]CLR is a BCD counter and RTC will not check loaded data for validity. Valid range is listed in parenthesis.

RTC Time Scale Selection Register (TSSR)

Register	Offset	R/W	Description	Reset Value
TSSR	RTC_BA+0x014	R/W	Time Scale Selection Register	0x0000_0001

Table 5-76 RTC Time Scale Selection Register (TSSR, address 0x4000_8014).

7	6	5	4	3	2	1	0
Reserved							HR24

Bits	Descriptions	
[0]	**HR24**	**24-Hour / 12-Hour Mode Selection** Determines whether TLR and TAR are in 24-hour mode or 12-hour mode 1 = select 24-hour time scale 0 = select 12-hour time scale with AM and PM indication

24-hour time scale	12-hour time scale	24-hour time scale	12-hour time scale (PM time + 20)
00	12(AM12)	12	32(PM12)
01	01 (AM01)	13	21 (PM01)
02	02(AM02)	14	22(PM02)
03	03(AM03)	15	23(PM03)
04	04 (AM04)	16	24 (PM04)
05	05(AM05)	17	25(PM05)
06	06(AM06)	18	26(PM06)
07	07(AM07)	19	27(PM07)
08	08(AM08)	20	28(PM08)
09	09(AM09)	21	29(PM09)
10	10 (AM10)	22	30 (PM10)
11	11 (AM11)	23	31 (PM11)

RTC Day of the Week Register (DWR)

Register	Offset	R/W	Description	Reset Value
DWR	RTC_BA+0x018	R/W	Day of the Week Register	0x0000_0006

7	6	5	4	3	2	1	0
Reserved					DWR		

Table 5-77 RTC Day of Week Register (DWR, address 0x4000_8018).

Bits	Descriptions	
[2:0]	**DWR**	**Day of the Week Register** <table><tr><td>Value</td><td>Day of the Week</td></tr><tr><td>0</td><td>Sunday</td></tr><tr><td>1</td><td>Monday</td></tr><tr><td>2</td><td>Tuesday</td></tr><tr><td>3</td><td>Wednesday</td></tr><tr><td>4</td><td>Thursday</td></tr><tr><td>5</td><td>Friday</td></tr><tr><td>6</td><td>Saturday</td></tr></table>

RTC Time Alarm Register (TAR)

Register	Offset	R/W	Description	Reset Value
TAR	RTC_BA+0x01C	R/W	Time Alarm Register	0x0000_0000

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

Table 5-78 RTC Time Alarm Register (TAR, address 0x4000_801C).

Bits	Descriptions	
[21:20]	10HR	10 Hour Time Digit of Alarm Setting (0~3)[2]
[19:16]	1HR	1 Hour Time Digit of Alarm Setting (0~9)
[14:12]	10MIN	10 Min Time Digit of Alarm Setting (0~5)
[11:8]	1MIN	1 Min Time Digit of Alarm Setting (0~9)
[6:4]	10SEC	10 Sec Time Digit of Alarm Setting (0~5)
[3:0]	1SEC	1 Sec Time Digit of Alarm Setting (0~9)

Note:
1. TAR is a BCD digit counter and RTC will not check validity of loaded data. Valid range is listed in the parenthesis.
2. This register can be read back after the RTC unit is active.

RTC Calendar Alarm Register (CAR)

Register	Offset	R/W	Description	Reset Value
CAR	RTC_BA+0x020	R/W	Calendar Alarm Register	0x0000_0000

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

Table 5-79 RTC Calendar Alarm Register (CAR, address 0x4000_8020).

Bits	Descriptions	
[23:20]	10YEAR	10-Year Calendar Digit of Alarm Setting (0~9)
[19:16]	1YEAR	1-Year Calendar Digit of Alarm Setting (0~9)
[12]	10MON	10-Month Calendar Digit of Alarm Setting (0~1)
[11:8]	1MON	1-Month Calendar Digit of Alarm Setting (0~9)
[5:4]	10DAY	10-Day Calendar Digit of Alarm Setting (0~3)
[3:0]	1DAY	1-Day Calendar Digit of Alarm Setting (0~9)

Note:
1. TLR is a BCD digit counter and RTC will not check validity loaded data, valid range is listed in the parenthesis.
2. This register can be read back after the RTC unit is active.

RTC Leap year Indication Register (LIR)

Register	Offset	R/W	Description	Reset Value
LIR	RTC_BA+0x024	R	RTC Leap year Indication Register	0x0000_0000

7	6	5	4	3	2	1	0
Reserved							**LIR**

Table 5-80 RTC Leap Year Indicator Register (LIR, address 0x4000_8024).

Bits	Descriptions
[0]	**LIR**	**Leap Year Indication Register** (read only). 1 = Current year is leap year 0 = Current year is not a leap year

RTC Interrupt Enable Register (RIER)

Register	Offset	R/W	Description	Reset Value
RIER	RTC_BA+0x028	R/W	RTC Interrupt Enable Register	0x0000_0000

7	6	5	4	3	2	1	0
Reserved						TIER	AIER

Table 5-81 RTC Interrupt Enable Register (RIER, address 0x4000_8028).

Bits	Descriptions
[1]	**TIER** **Time-Tick Interrupt and Wakeup-by-Tick Enable** 1 = RTC Time-Tick Interrupt is enabled 0 = RTC Time-Tick Interrupt is disabled.
[0]	**AIER** **Alarm Interrupt Enable** 1 = RTC Alarm Interrupt is enabled 0 = RTC Alarm Interrupt is disabled

RTC Interrupt Indication Register (RIIR)

Register	Offset	R/W	Description	Reset Value
RIIR	RTC_BA+0x02C	R/C	RTC Interrupt Indication Register	0x0000_0000

7	6	5	4	3	2	1	0
Reserved						TIF	AIF

Table 5-82 RTC Interrupt Indication Register (RIIR, address 0x4000_802C).

Bits	Descriptions	
[1]	**TI**	**RTC Time-Tick Interrupt Flag** When RTC Time-Tick Interrupt is enabled (RIER.TIER=1), RTC unit will set TIF high at the rate selected by TTR[2:0]. This bit cleared/acknowledged by writing 1 to it. 1= Indicates RTC Time-Tick Interrupt generated. 0= Indicates no Time-Tick Interrupt condition
[0]	**AI**	**RTC Alarm Interrupt Flag** When RTC Alarm Interrupt is enabled (RIER.AIER=1), RTC unit will set AIF to high once the RTC real time counters TLR and CLR reach the alarm setting time registers TAR and CAR. This bit cleared/acknowledged by writing 1 to it. 1= Indicates RTC Alarm Interrupt generated. 0= Indicates no Alarm Interrupt condition.

RTC Time-Tick Register (TTR)

Register	Offset	R/W	Description	Reset Value
TTR	RTC_BA+0x030	R/C	RTC Time-Tick Register	0x0000_0000

7	6	5	4	3	2	1	0
Reserved				TWKE	TTR[2:0]		

Table 5-83 RTC Time-Tick Register (TTR, address 0x4000_8030).

Bits	Descriptions	
[3]	TWKE	**RTC Timer Wakeup CPU Function Enable Bit** If TWKE is set before CPU is in power-down mode, when a RTC Time-Tick or Alarm Match occurs, CPU will wake up. 1= Enable the Wakeup function. 0= Disable Wakeup CPU function.
[2:0]	TTR	**Time Tick Register** The RTC time tick period for Periodic Time-Tick Interrupt request. | TTR[2:0]	Time tick (second)
---	---
0	1
1	1/2
2	1/4
3	1/8
4	1/16
5	1/32
6	1/64
7	1/128	 Note: This register can be read back after the RTC is active.

5.9   Serial Peripheral Interface (SPI) Controller

5.9.1   Overview

The Serial Peripheral Interface (SPI) is a synchronous serial data communication protocol which operates in full duplex mode. Devices communicate in master/slave mode with 4-wire bi-directional interface. The ISD93XX series contains an SPI controller performing a serial-to-parallel conversion of data received from an external device, and a parallel-to-serial conversion of data transmitted to an external device. The SPI controller can be set as a master with up to 2 slave select (SSB) address lines to access two slave devices; it also can be set as a slave controlled by an off-chip master device.

In addition the SPI interface supports Dual and Quad IO as is common on serial flash memories, where data is transferred 2 or 4 bits per clock period.

5.9.2   Features

•   Supports master or slave mode operation.

•   Supports one or two channels of serial data.

•   8 word FIFO on transmit and receive.

•   MSB or LSB first transfer.

•   2 device/slave select lines in master mode, single device/slave select line in slave mode.

•   Byte or word Sleep Suspend Mode.

•   PDMA access support.

5.9.3 SPI Block Diagram
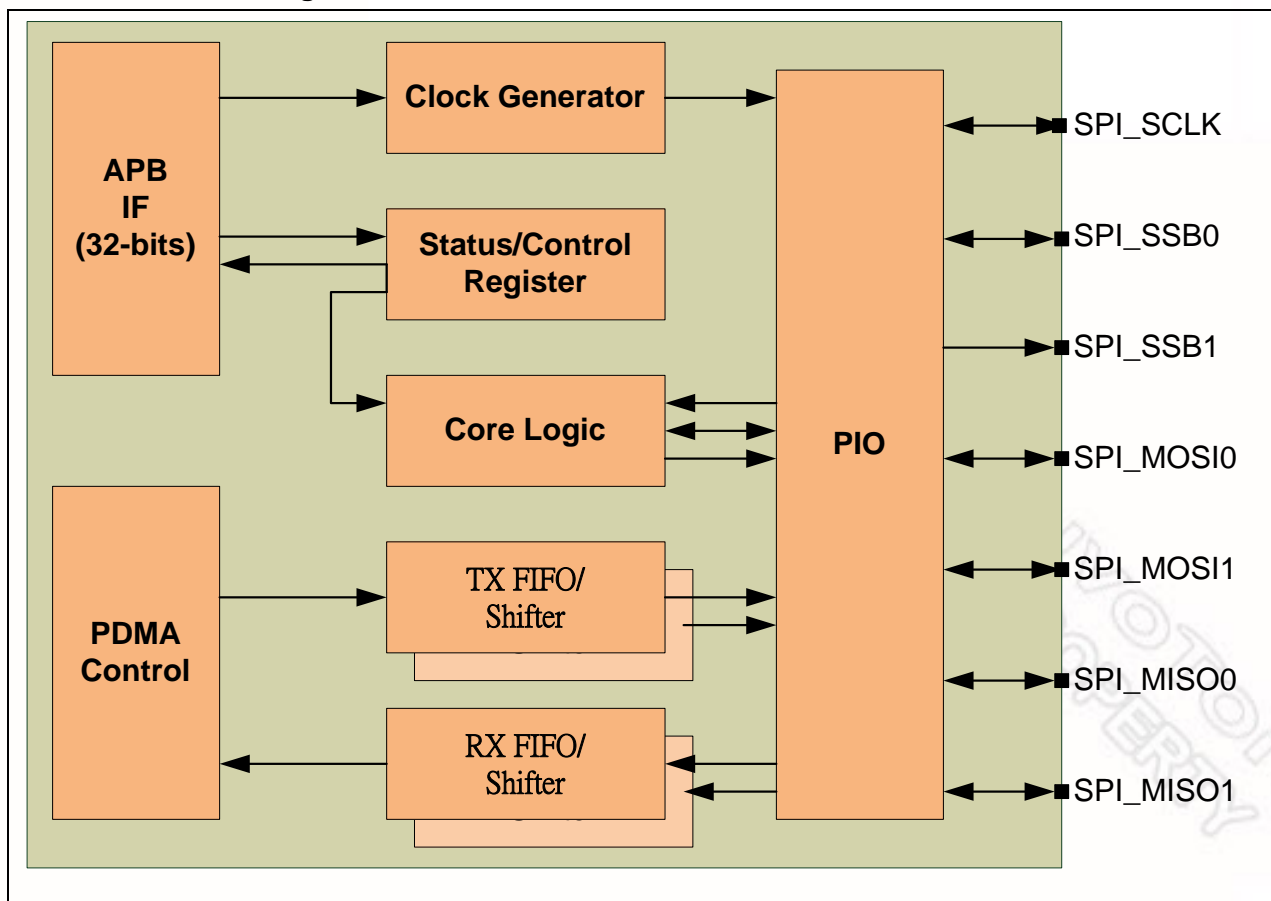
Figure 5-34SPI Block Diagram

5.9.4 SPI Function Descriptions

SPI Engine Clock and SPI Serial Clock

The SPI controller derives its clock source from the system HCLK as determined by the CLK_SEL1 register. The frequency of the SPI master clock is determined by the divisor ratio SPI_CLKDIV.

In Master mode, the output frequency of the SPI serial clock output pin is equal to the SPI engine clock rate. In general, the SPI serial clock is denoted as SPI clock. In Slave mode, the SPI serial clock is provided by an off-chip master device. The SPI engine clock rate of slave device must be faster than the SPI serial clock rate of the master device. The frequency of SPI engine clock cannot be faster than the APB clock rate regardless of Master or Slave mode.

Master/Slave Mode

This SPI controller can be configured as in master or slave mode by setting the SLAVE bit (SPI->CNTRL.SLAVE). In master mode the ISD93XX generates SCLK and SSB signals to access one or more slave devices. In slave mode the ISD93XX monitors SCLK and SSB signals to respond to data transactions from an off-chip master. The signal directions are summarized in the application block diagrams of and .
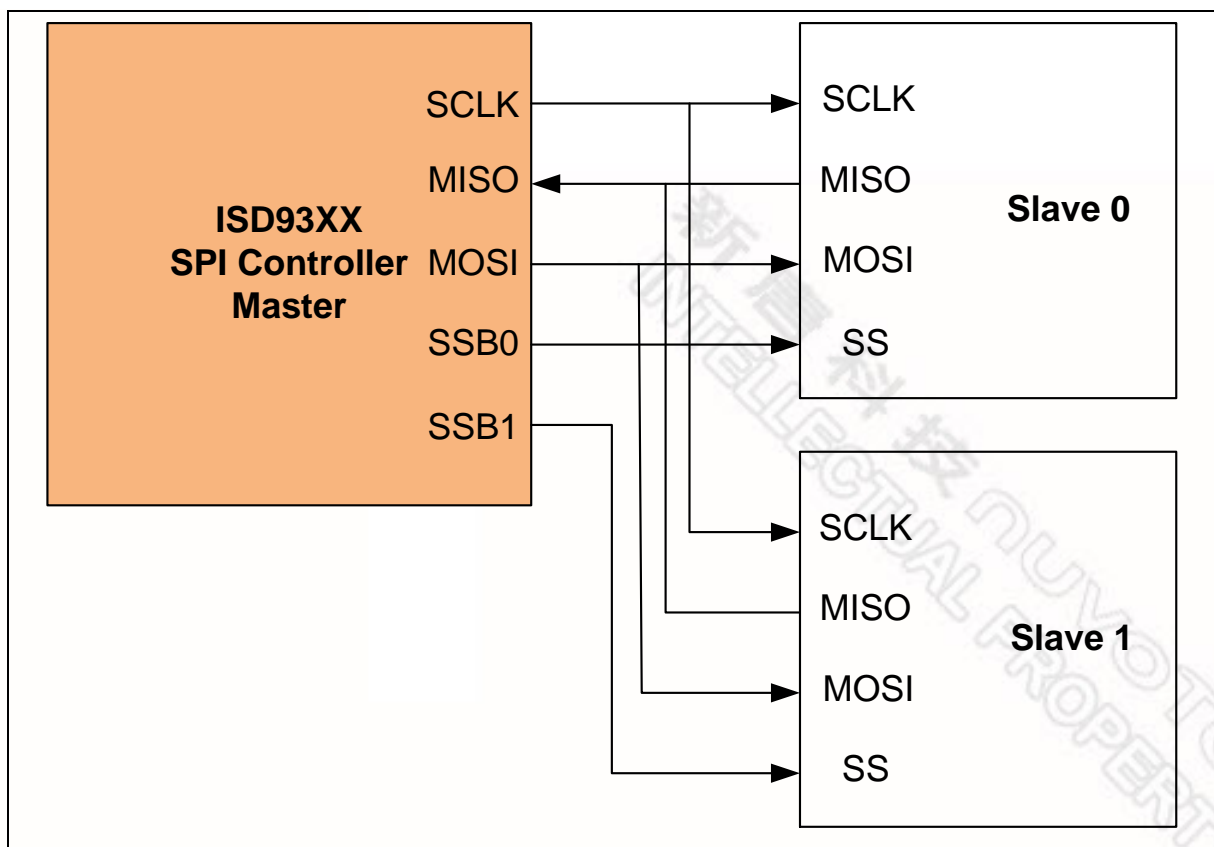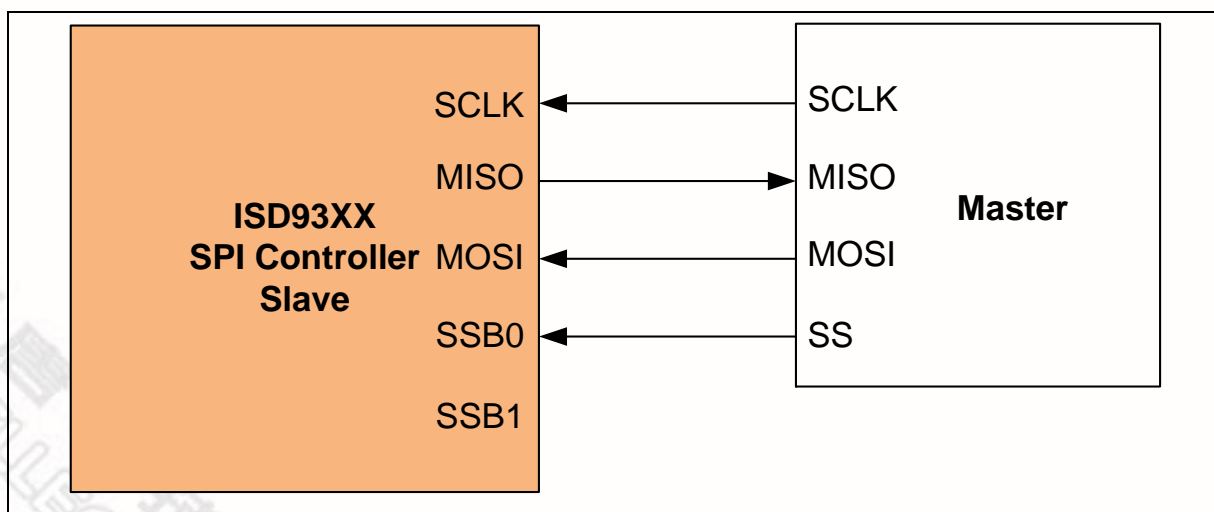
Figure 5-35SPI Master Mode Application Block Diagram

Figure 5-36SPI Slave Mode Application Block Diagram

Slave Select

In master mode, the SPI controller can address up to two off-chip slave devices through the slave select output pins SPI_SSB0 and SPI_SSB1. Only one slave can be addressed at any one time. If more slave address lines are required, GPIO pins can be manually configured to provide additional SSB lines. In slave mode, the off-chip master device drives the slave select signal SPI_SSB0 to address the SPI controller. The slave select signal can be programmed to be active low or active high via the SPI->SSR.SS_LVL bit. In addition the SPI->SSR.SS_LTRIG bit defines whether the slave select signals are level triggered or edge triggered. The selection of trigger condition depends on what type of peripheral slave/master device is connected.

Automatic Slave Select

In master mode, if the bit SPI->SSR.ASS is set, the slave select signals will be generated automatically and output to SPI_SSB0 and SPI_SSB1 pins according to registers SPI->SSR.SSR[0] and SPI->SSR.SSR[1]. In this mode, SPI controller will assert SSB when transaction is triggered and de-assert when data transfer is finished. If the SPI->SSR.ASS bit is cleared, the slave select output signals are asserted and de-asserted by manual setting and clearing the related bits in the SPI->SSR.SSR[1:0] register. The active level of the slave select output signals is specified by the SPI->SSR. SS_LVL bit.

In Master mode, if the value of SP_CYCLE[3:0] is less than 3 and AUTOSS is enabled, the slave select signal will be kept in active state between two successive transactions.

In Slave mode, to recognize the inactive state of the slave select signal, the inactive period of the slave select signal must be larger than or equal to 3 engine clock periods between two successive transactions.

Serial Clock

In master mode, writing a divisor into the SPI->DIVIDER.DIVIDER register will program the output frequency of serial clock to the SPI_SCLK output port. In slave mode, the off-chip master device drives the serial clock through the SPI_SCLK.

Clock Polarity

The SPI->CNTRL.CLKP bit defines the serial clock idle state in master mode. If CLKP = 1, the output SPI_SCLK is high in idle state. If CLKP=0,it is low in idle state.

Transmit/Receive Bit Length

The bit length of a transfer word is defined in SPI->CNTRL.DWIDTH bit field. It is set to define the length of a transfer word and can be up to 32 bits in length. DWIDTH=0x0 enables 32bit word length.

LSB First

The SPI->CNTRL.LSB bit defines the **bit** order of data transmission. If LSB=0 then MSB of transfer word is sent first in time. If LSB=1 then LSB of transfer word is sent first in time.

Transmit Edge

The SPI->CNTRL.TX_NEG bit determines whether transmit data is changed on the positive or negative edge of the SPI_SCLK serial clock. If TX_NEG=0 then transmitted data will change state on the rising edge of SPI_SCLK. If TX_NEG=1 then transmitted data will change state on the falling edge of SPI_SCLK.

Receive Edge

The SPI->CNTRL.RX_NEG bit determines whether data is received at either the negative edge or positive edge of serial clock SPI_SCLK. If RX_NEG=1 then data is clocked in on the falling edge of SPI_SCLK. If RX_NEG=0 data is clocked in on the rising edge of SPI_SCLK. Note that RX_NEG should be the inverse of TX_NEG for standard SPI operation.

Word Suspend

The four bit field SP_CYCLE (SPI_CTL[7:4]) provides a configurable suspend interval of 0.5 ~ 15.5SPI clock periods, between two successive transaction words in Master mode. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value of SP_CYCLE is 0x3 (3.5SPI clock cycles).
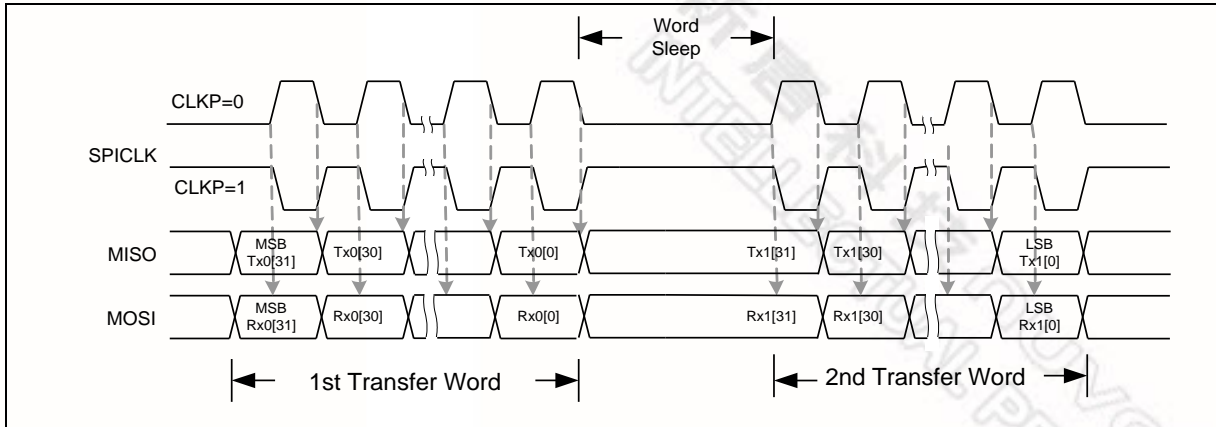
.

Figure 5-37 Word Sleep Suspend Mode

Byte Reorder

APB access to the SPI controller is via the 32bit wide TX and RX registers. When the transfer is set as MSB first (SPI->CNTRL.LSB = 0) and the SPI->CNTRL.REORDER bit is set, the data stored in the TX buffer and RX buffer will be rearranged such that the least significant **physical byte** is processed first. For DWIDTH =0 (32 bits transfer), the sequence of transmitted/received data will be BYTE0, BYTE1, BYTE2, and then BYTE3. If DWIDTH is set to 24-bits, the sequence will be BYTE0, BYTE1, and BYTE2.For Quad and Dual SPI transactions, REORDER is only valid for receive operation. For transmit in Dual/Quad modes, REORDER must be set to 0.
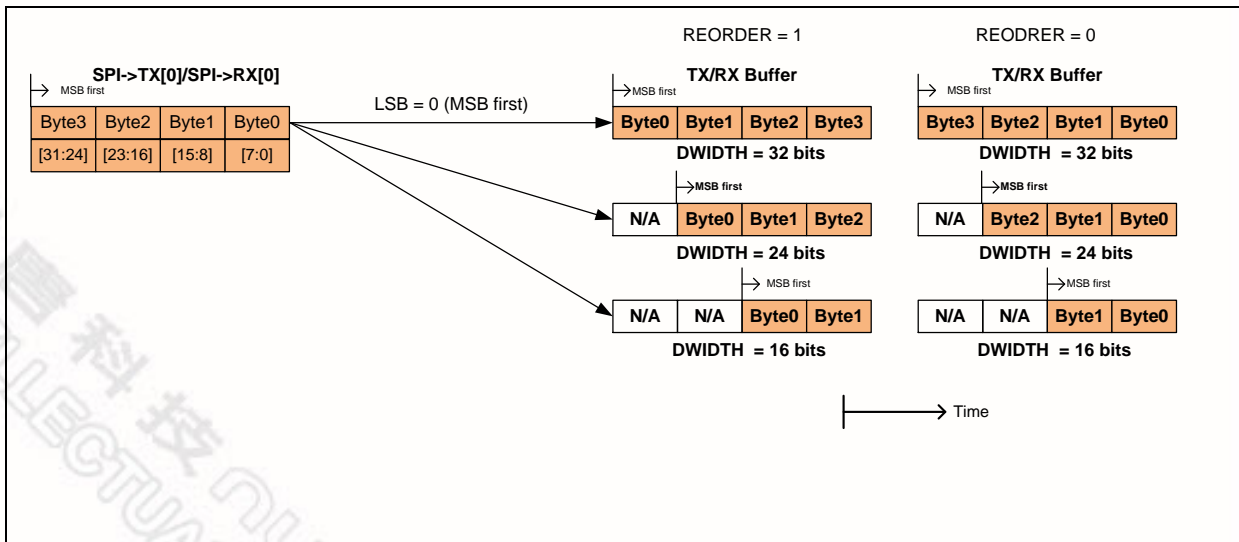

Figure 5-38 Byte Re-Ordering Transfer

Byte ordering can be a confusing issue when converting from arrays of data processed by the CPU for transmission out the SPI port. The CortexM0 stores data in a little endian format; that is the LSB of a multi-byte word or half-word are stored first in memory. Consider how the CortexM0 stores the following arrays in memory:

1. unsigned char ucSPI_DATA[]={0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08};

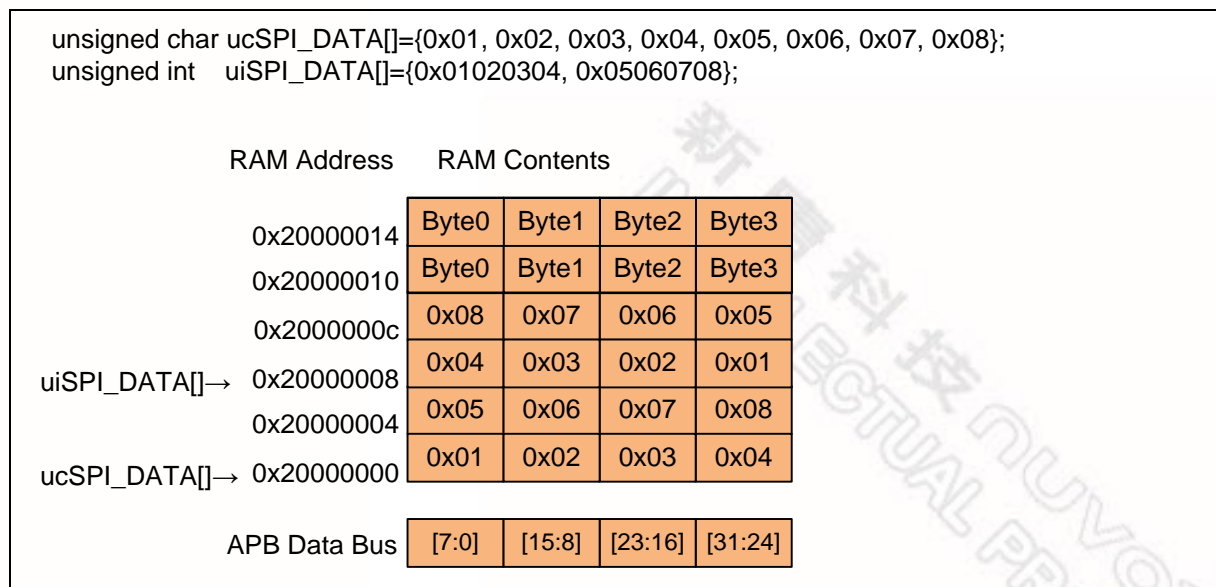2. unsigned int   uiSPI_DATA[]={0x01020304, 0x05060708};

Figure 5-39 Byte Order in Memory

It can be seen from that byte order for an array of bytes is different than that of an array of words. Now consider if this data were to be sent to the SPI port; the user could:

1. Set DWIDTH=8 and send data byte-by-byte SPI->TX = ucSPI_DATA[i++]
2. Set DWIDTH=32 and send word-by-word SPI->TX = uiSPI_DATA[i++]

Both of these would result in the byte stream {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08} being sent.

It would be common that a byte array of data is constructed but user, for efficiency, wishes to transfer data to SPI via word transfers. Consider the situation of where an int pointer points to the byte data array.
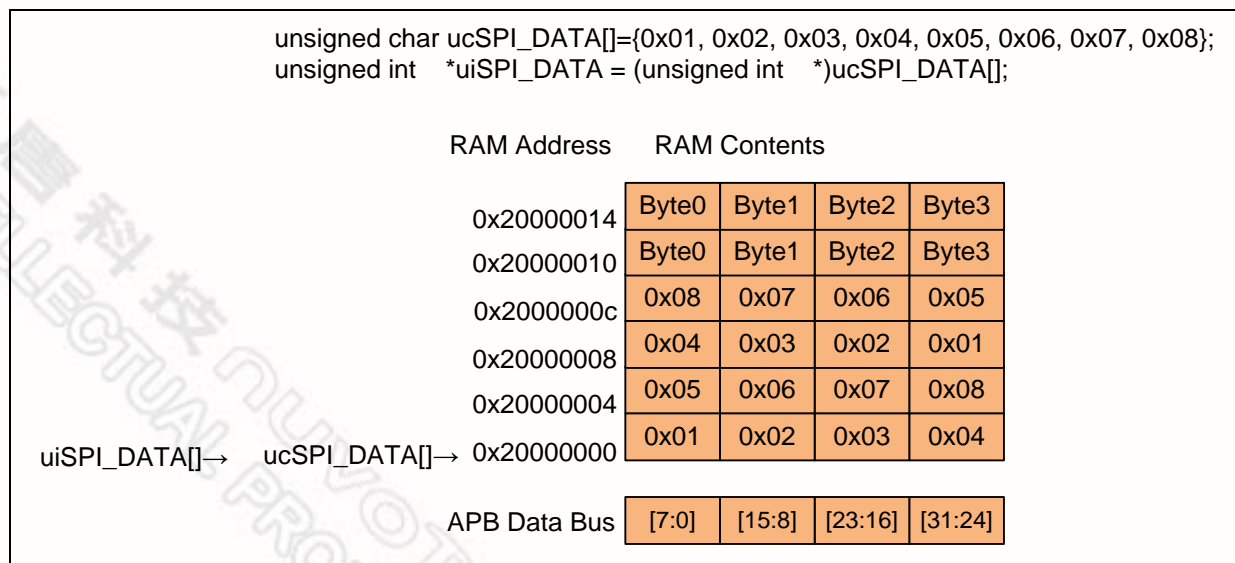

Figure 5-40 Byte Order in Memory

Now if we set DWIDTH=32 and sent word-by-word SPI->TX[0] = uiSPI_DATA[i++], the order

transmitted would be {0x04, 0x03, 0x02, 0x01, 0x08, 0x07, 0x06, 0x05}. However if we set REORDER=1, we would reverse this order to the desired stream: {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08}.

Interrupt

5.9.4.1.1.1   SPI unit transfer interrupt

As the SPI controller finishes a unit transfer, the unit transfer interrupt flag IF (SPI_STATUS[1]) will be set to 1. The unit transfer interrupt event will generate an interrupt to CPU if the unit transfer interrupt enable bit IE (SPI_CTL[17]) is set. The unit transfer interrupt flag is cleared by writing 1 to it.

5.9.4.1.1.2   SPI slave select interrupt

In slave mode, there are slave select active and in-active interrupt flag, SSACT_INTSTS and SSINA_INTSTS, will be set to 1 when the SPIEN and SLAVE bits were set to 1 and slave senses the slave select signal active or inactive. The SPI controller will issue an interrupt if the SSINA_INTEN or SSACT_INTEN, SPI_SSCR[13:12], are set to 1.

5.9.4.1.1.3   Slave Time-out interrupt

In Slave mode, there is slave time-out function for user to know that there is serial clock input but one transaction doesn't finish over the period of SLVTOPRD basing on engine clock.

When the Slave select is active and the value of SLVTOPRD is not 0, the Slave time-out counter in the SPI controller logic will start after the serial clock input. This counter will be clear after one transaction done or the SLVTOPRD is set to 0. If the value of the time-out counter greater or equal than the value of SLVTOPRD before one transaction done, the slave time-out event occurs and the SLVTO_INTSTS, SPI_STATUS[5], will be set to 1. The SPI controller will issue an interrupt if the SLVTO_INTEN, SPI_SSCR[5], is set to 1.

5.9.4.1.1.4   Slave Error 0 interrupt

In Slave mode,if the transmit/ receive bit count mismatch with the DWIDTH when the slave select line goes to inactive state, the Slave mode error 0, SLVER0_INTSTS, SPI_STATUS[6], will be set to 1. The SPI controller will issue an interrupt if the SLVER0_INTEN, SPI_SSCR[8], is set to 1.

Note: 1. In Slave transmit mode, if there is bit length transmit error (bit count mismatch), the user shall set the TX_CLR bit and write the transmit datum again to restart the next transaction.

2. If the slave select active but there is no any serial clock input, the SLVER0_INTSTS also active when the slave select goes to inactive state.

5.9.4.1.1.5   Slave Under-run and Slave Error 1 interrupts

In Slave mode, if there is no any data is written to the SPI_TX register, the under-run event, TXUDR_INTSTS (SPI_STATUS[19]) will active when the slave select active and the serial clock input this controller. The SPI controller will issue an interrupt if the SLVUDR_INTEN is set to 1.

Under the previous condition, the Slave mode error 1, SLVER1_INTSTS, SPI_STATUS[7], will be set to 1 when SS goes to inactive state and transmit under-run occurs. The SPI controller will issue an interrupt if the SLVER1_INTEN, SPI_SSCR[9], is set to 1.

Note: In SLV3WIRE mode, the slave select bus active all the time so that the user shall polling the TXUDR_INTSTS bit to know if there is transmit under-run event or not.

5.9.4.1.1.6   Receive Over-run interrupt

In Slave mode, if the receive FIFO buffer contains 8 unread data, the RX_FULL flag will be set to 1 and the RXOVR_INTSTS will be set 1 if there is more serial data is received from SPIMOSI and the RXOVR_INTSTS will be set to 1 and follow-up data will be dropped. The SPI controller will issue an

interrupt if the SLVOVR_INTEN, SPI_FIFOCTL[5], is set to 1.

5.9.4.1.1.7 Receive FIFO time-out interrupt

In FIFO mode, there is a time-out function to inform user. If there is a received data in the FIFO and it is not read by software over 64 SPI engine clock periods in Master mode or over 576 SPI engine clock periods in Slave mode, it will send a time-out interrupt to the system if the time-out interrupt enable bit, RXTO_INTEN, SPI_FIFOCTL[4], is set to 1.

5.9.4.1.1.8 Transmit FIFO interrupt

In FIFO mode, if the valid data count of the transmit FIFO buffer is less than or equal to the setting value of TX_THRESHOLD, the transmit FIFO interrupt flag will be set to 1. The SPI controller will generate a transmit FIFO interrupt to the system if the transmit FIFO interrupt enable bit, SPI_FIFOCTL[3], is set to 1.

5.9.4.1.1.9 Receive FIFO interrupt

In FIFO mode, if the valid data count of the receive FIFO buffer is larger than the setting value of RX_THRESHOLD, the receive FIFO interrupt flag will be set to 1. The SPI controller will generate a receive FIFO interrupt to the system if the receive FIFO interrupt enable bit, SPI_FIFOCTL[2], is set to 1.

3-Wire Mode

When the SLV3WIRE bit is set by software to enable the Slave 3-wire mode, the SPI controller can work with no slave select signal in Slave mode. The SLV3WIRE bit only takes effect in Slave mode. Only three pins, SPICLK, SPI_MISO, and SPI_MOSI, are required to communicate with a SPI master. The SPISS pin can be configured as a GPIO. When the SLV3WIRE bit is set to 1, the SPI slave will be ready to transmit/receive data after the SPIEN bit is set to 1.

2-Bit Mode

The SPI controller supports 2-bit Transfer mode when setting the TWOB bit (SPI_CTL[16]) to 1. In 2-bit mode, the SPI controller performs full duplex data transfer. In other words, the2-bit serial data can be transmitted and received simultaneously.

For example, in Master mode, the first data written to the TX FIFO will be transmitted through SPI_MOSI0and the second data written to the TX FIFO will be transmitted through the SPI_MOSI1 pin. After transmission, the first read of RX FIFO will result in the data received from SPI_MISO0 pin and the second read the data received from SPI_MISO1 pin.

In Slave mode, the first two data stored in the TX FIFO will be transmitted through the SPI_MISO0 and SPI_MISO1 pin respectively. Concurrently, the RX FIFO will store the data received from the SPI_MOSI0 and SPI_MOSI1 pin, same as Master mode.
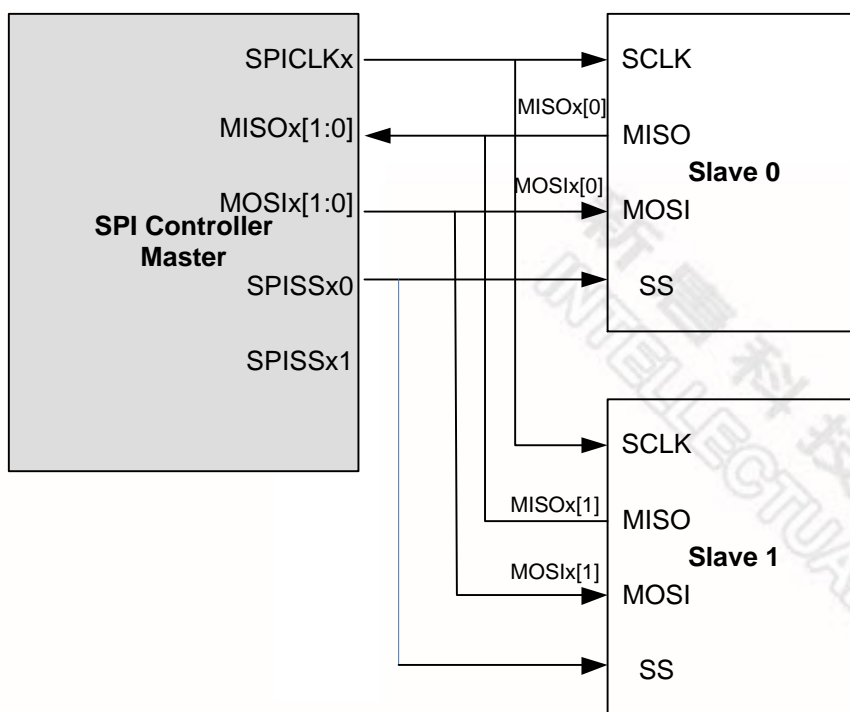
Figure 5-412-Bit Mode System Architecture
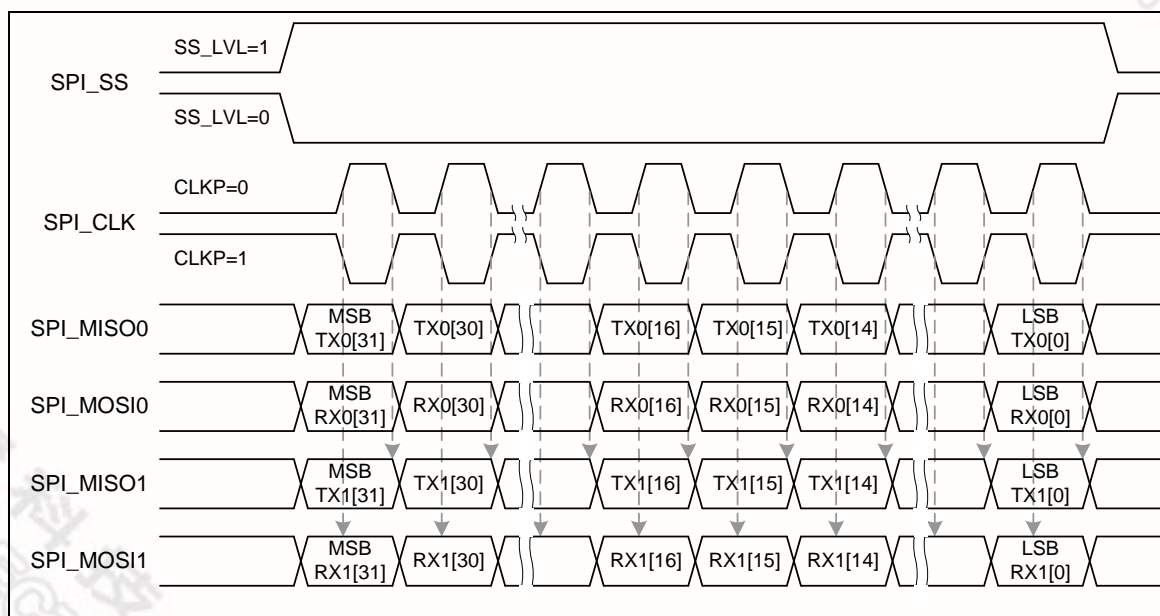
Figure 5-422-Bit Mode (Slave Mode)

Dual/Quad I/O Mode

The SPI controller supports dual and quad I/O transfer when setting the DUAL_IO_EN bit or the QUAD_IO_EN bit (SPI_CTL[21], SPI_CTL[22]) to 1. Many SPI Serial Flash devices support Dual/ Quad I/O transfer. The QD_IO_DIR bit (SPI_CTL[20]) is used to define the direction of the transfer data. When the QD_IO_DIR bit is set to 1, the controller will send the data to external device. When the QD_IO_DIR bit is set to 0, the controller will read the data from the external device. This function supports transfers of 8, 16, 24, and 32-bits.

The Dual/Quad I/O mode is not supported in the Slave 3-wire mode. The byte REORDER function is only available in receive mode for Dual/Quad transactions.

For Dual I/O mode, if both the DUAL_IO_EN and QD_IO_DIR bits are set as 1, the SPI_MOSI0 is the even bit data output and the SPI_MISO0 will be set as the odd bit data output. If the DUAL_IO_EN is set as 1 and QD_IO_DIR is set as 0, both the SPI_MISO0 and SPI_MOSI0 will be set as data input ports.
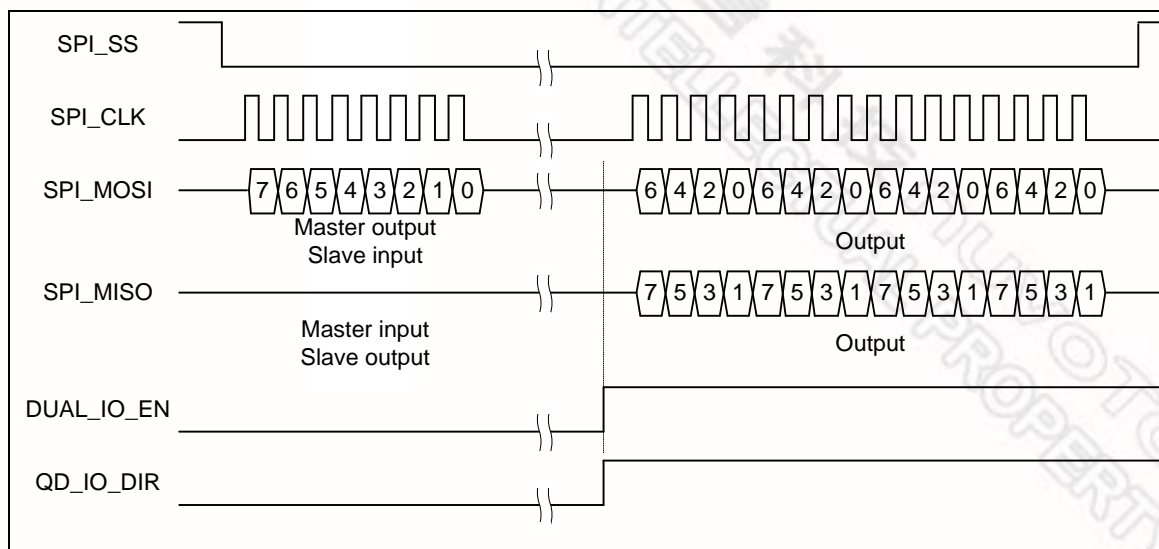
Figure 5-43Bit Sequence of Dual Output Mode
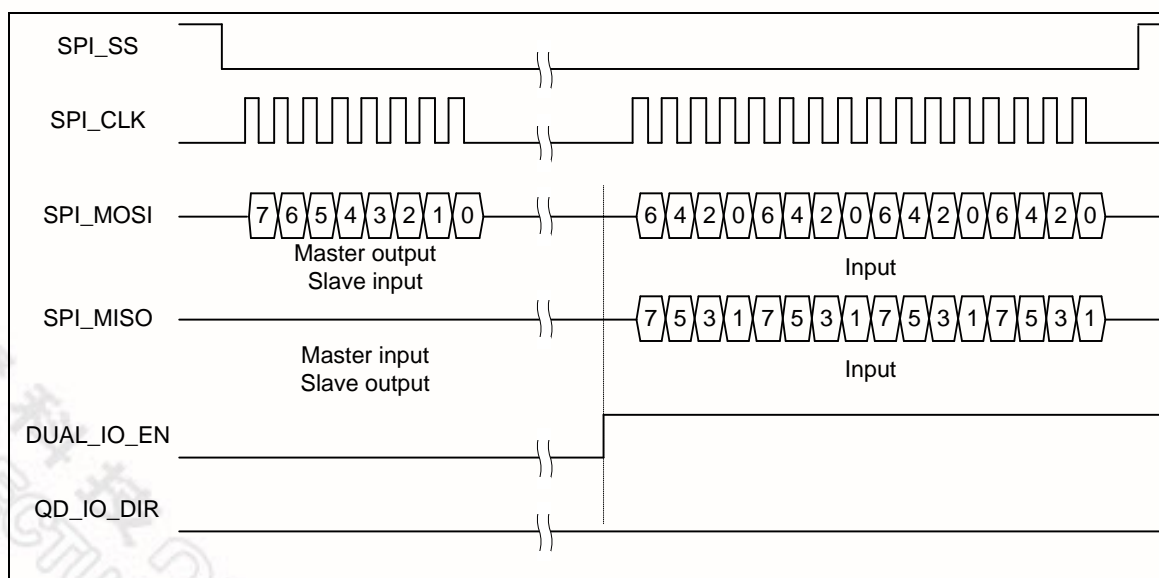
Figure 5-44Bit Sequence of Dual Input Mode

For Quad I/O mode, if both the QUAD_IO_EN and QD_IO_DIR bits are set as 1, the SPI_MOSI0 and SPI_MOSI1 are the even bit data output and the SPI_MISO0 and SPI_MISO1 will be set as the odd bit data output. If the QUAD_IO_EN is set as 1 and QD_IO_DIR is set as 0, both the SPI_MISO0, SPI_MISO1, SPI_MOSI0 and SPI_MOSI1 will be set as data input ports.
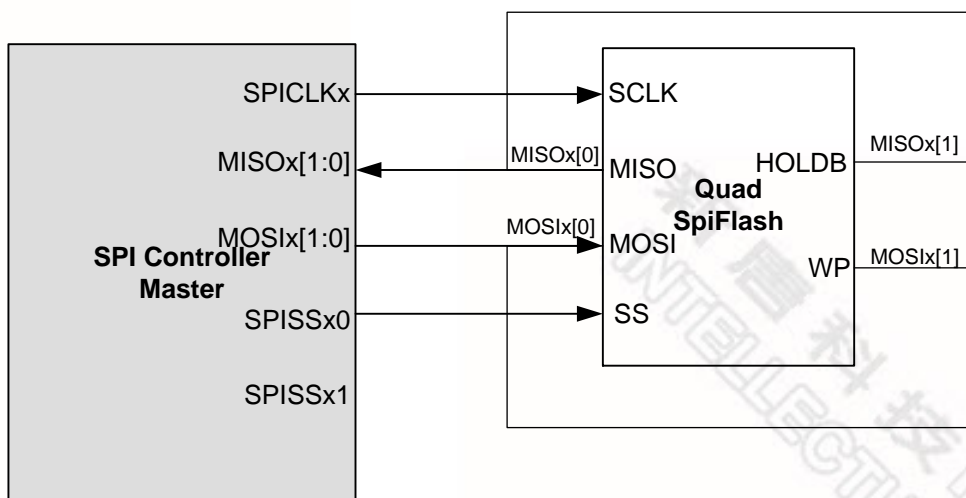
Figure 5-45Quad Mode System Architecture
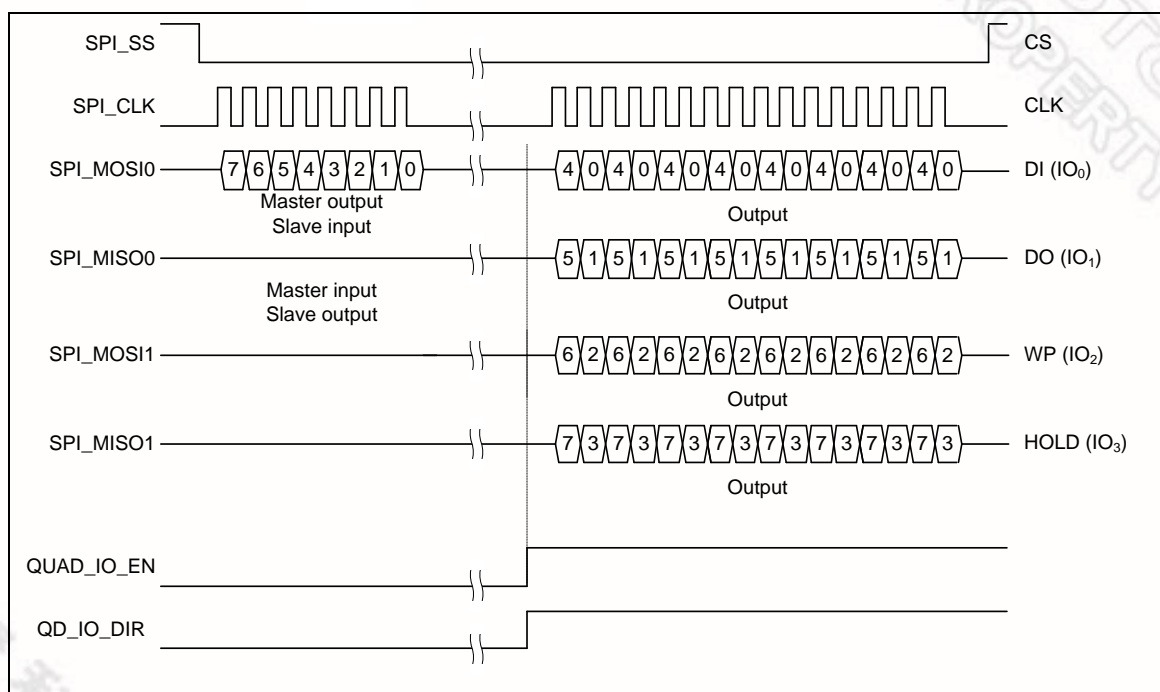
Figure 5-46Bit Sequence of Quad Output Mode

8-Level FIFO Buffer

The SPI controller is equipped with eight 32-bit wide transmit and receive FIFO buffers.

The transmit FIFO buffer is an 8-level depth, 32-bit wide, first-in, first-out register buffer. 8 words of data can be written to the transmit FIFO buffer in advance through software by writing the SPI_TX register. The data stored in the transmit FIFO buffer will be read and sent out by the transmission control logic. If the 8-level transmit FIFO buffer is full, the TX_FULL bit will be set to 1. When the SPI transmission logic unit draws out the last datum of the transmit FIFO buffer, so that the 8-level transmit FIFO buffer is empty, the TX_EMPTY bit will be set to 1. Notice that the TX_EMPTY flag is set to 1

while the last transaction is still in progress. In Master mode, both the BUSY bit (SPI_STATUS[0]) and TX_EMPTY bit should be checked by software to make sure whether the SPI is idle or not.

The received FIFO buffer is also an 8-level depth, 32-bit wide, first-in, first-out register buffer. The receive control logic will store the received data to this buffer. The FIFO buffer data can be read from SPI_RX register by software. There are FIFO related status bits, like RX_EMPTY and RX_FULL, to indicate the current status of FIFO buffer.

The transmitting and receiving threshold can be set through software by setting the TX_THRESHOLD, SPI_FFCTL[30:24], and RX_THRESHOLD, SPI_FIFOCTL[26:24], settings. When the count of valid data stored in transmit FIFO buffer is less than or equal to TX_THRESHOLD setting, the TXTH_INTSTS, SPI_STATUS[18], bit will be set to 1. When the count of valid data stored in receive FIFO buffer is larger than RX_THRESHOLD setting, the RXTH_INTSTS, SPI_STATUS[10], bit will be set to 1.
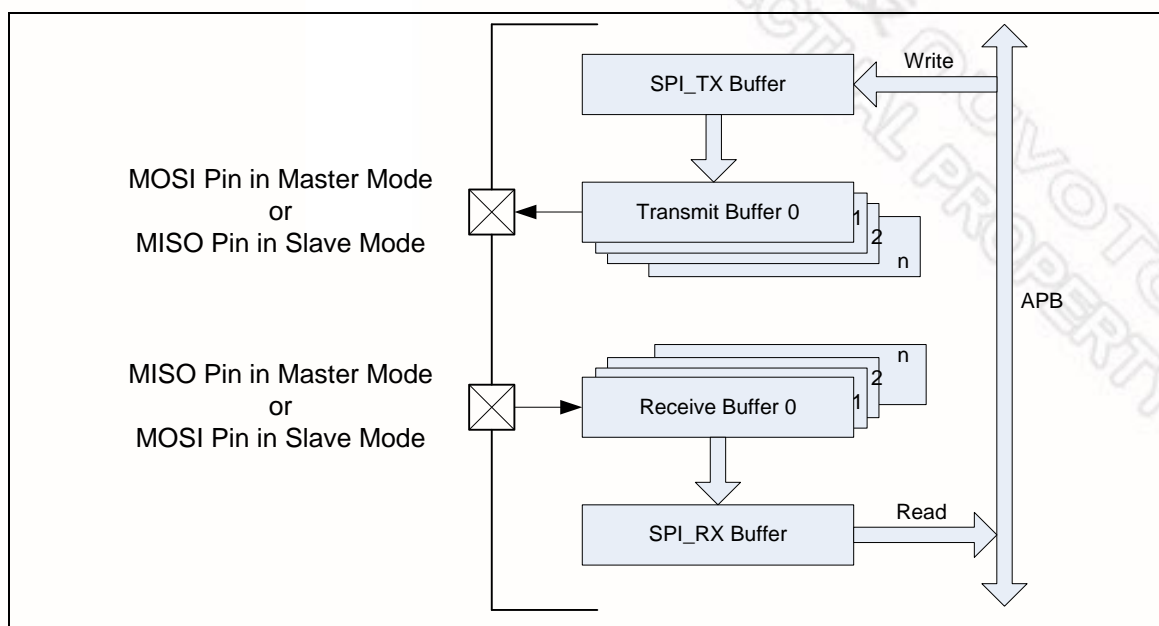
Figure 5-47FIFO Mode Block Diagram

In Master mode, the first datum is written to the SPI_TX register, the TX_EMPTY flag will be cleared to 0.The transmission immediately starts as long as the transmit FIFO buffer is not empty. User can write the next data into SPI_TX register immediately. The SPI controller will insert a suspend interval between two successive transactions and the period of suspend interval is decided by the setting of SP_CYCLE (SPI_CTL [7:4]). User can write data into SPI_TX register as long as the TX_FULL flag is 0.

The subsequent transactions will be triggered automatically if the transmitted data are updated in time. If the SPI_TX register is not updated after data transfer is done, the transfer will stop.

In Master mode, during receive operation, the serial data is received from SPI_MISO0/1 pin and stored to receive FIFO buffer. The RX_EMPTY flag will be cleared to 0 while the receive FIFO buffer contains unread data. The received data can be read by software from SPI_RX register as long as the RX_EMPTY flag is 0. If the receive FIFO buffer contains 8 unread data, the RX_FULL flag will be set to 1.The SPI controller will stop receiving data until the SPI_RX register is read by software.

In Slave mode, during transmission operation, when data is written to the SPI_TX register by software, the data will be loaded into transmit FIFO buffer and the TX_EMPTY flag will beset to 0.   The

transmission will start when the slave device receives clock signal from master. Data can be written to SPI_TX register as long as the TX_FULL flag is 0. After all data have been drawn out by the SPI transmission logic unit and the SPI_TX register is not updated by software, the TX_EMPTY flag will be set to 1.

If there is no any data is written to the SPI_TX register, the under-run event, TXUDR_INTSTS (SPI_STATUS[19]) will active when the slave select active and the serial clock input this controller. Under the previous condition, the Slave mode error 1, SLVER1_INTSTS, SPI_STATUS[7], will be set to 1 when SS goes to inactive state and transmit under-run occurs.

In Slave mode, during receiving operation, the serial data is received from SPI_MOSI0/1 pin and stored to SPI_RXregister. The reception mechanism is similar to Master mode reception operation. If the receive FIFO buffer contains 8 unread data, the RX_FULL flag will be set to 1 and the RXOVR_INTSTS will be set 1 if there is more serial data is received from SPIMOSI and follow-up data will be dropped. If the receive bit counter mismatch with the DWIDTH when the slave select line goes to inactive state, the Slave mode error 0, SLVER0_INTSTS, SPI_STATUS[6], will be set to 1.

When the Slave select is active and the value of SLVTOPRD is not 0, the Slave time-out counter in the SPI controller logic will start after the serial clock input. This counter will be clear after one transaction done or the SLVTOPRD is set to 0. If the value of the time-out counter greater or equal than the value of SLVTOPRD before one transaction done, the slave time-out event occurs and the SLVTO_INTSTS, SPI_STATUS[5], will be set to 1.

A receive time-out function is built-in in this controller. When the receive FIFO is not empty and no read operation in receive FIFO over 64 SPI clock period in Master mode or over 576 SPI engine clock period in Slave mode, the receive time-out occurs and the SLVTO_INTSTS be set to 1. When the receive FIFO is read by user, the time-out status will be cleared automatically.

DMA Receive Mode

The SPI controller supports DMA access to the transmit and receive FIFOs. When the DMA transmit interface is active, DMA sub-system fills the TX FIFO to trigger SPI interface. When only DMA receive function is required, an additional mode is provided to inform the SPI system of the number of transfers desired so that SPI system can read ahead of DMA requests. When SPI0->CNTRL.RX_TRANS_CNT_EN is set, the register SPI0->RX_TRANS_CNT holds the number of SPI transactions (total number of bytes is determined by CNTRL.DWIDTH value).

5.9.5   SPI Timing Diagram

In master/slave mode, the device address/slave select (SPI_SSB0/1) signal can be configured as active low or active high by the SPI->SSR.SS_LVL bit. In slave mode, the SPI-SSR.SS_LTRIG will determine whether the slave select signal is treated as a level triggered or edge triggered signal.

The serial clock phase and polarity is controlled by CLKP, RX_NEG and TX_NEG bits. The bit length of a transfer word is configured by the DWIDTH parameter. Whether data transmission is MSB first or LSB first is controlled by the SPI->CNTRL.LSB bit. Four examples of SPI timing diagrams for master/slave operations and the related settings are shown as below.
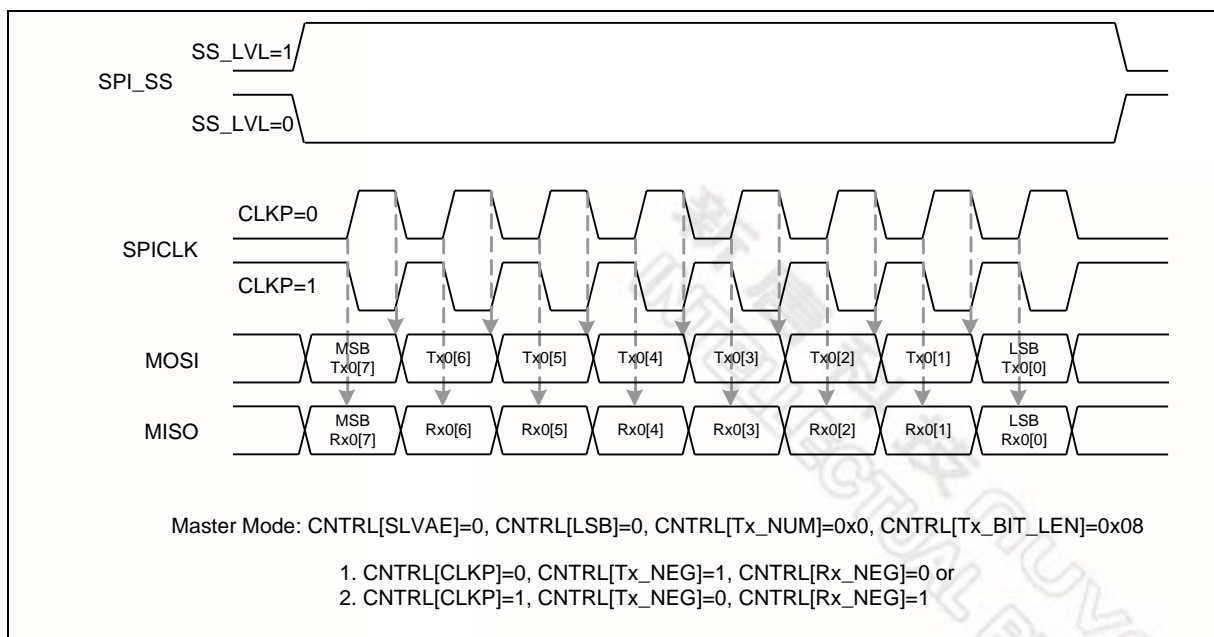
Figure 5-48 SPI Timing in Master Mode
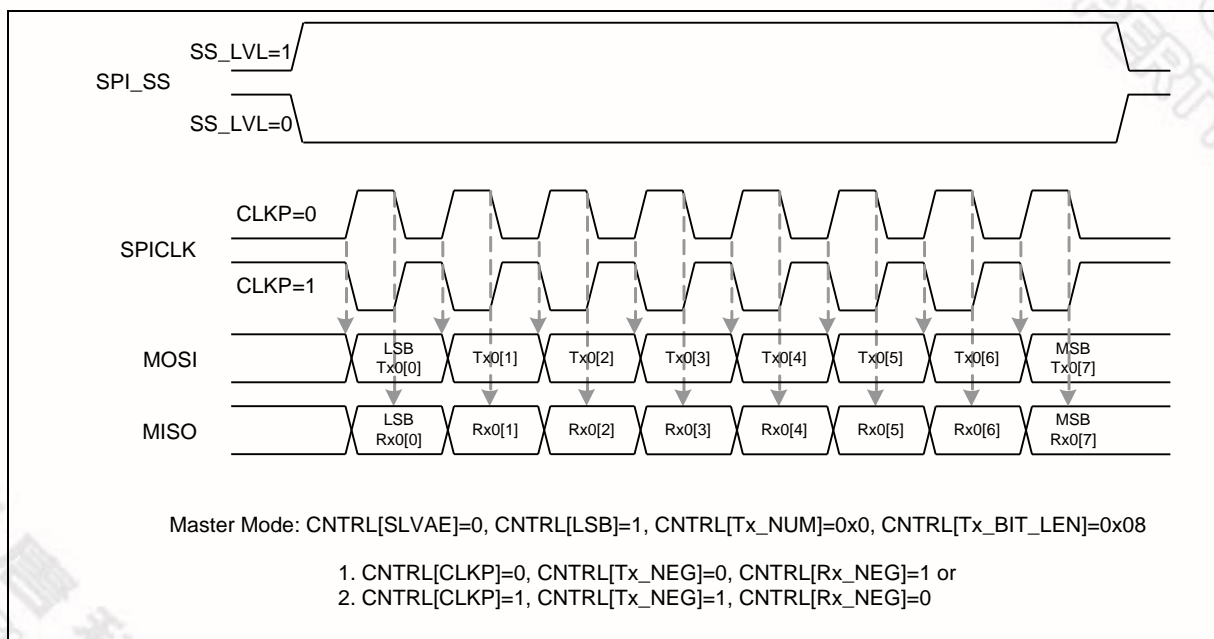
Figure 5-49 SPI Timing in Master Mode (Alternate Phase of SPICLK)
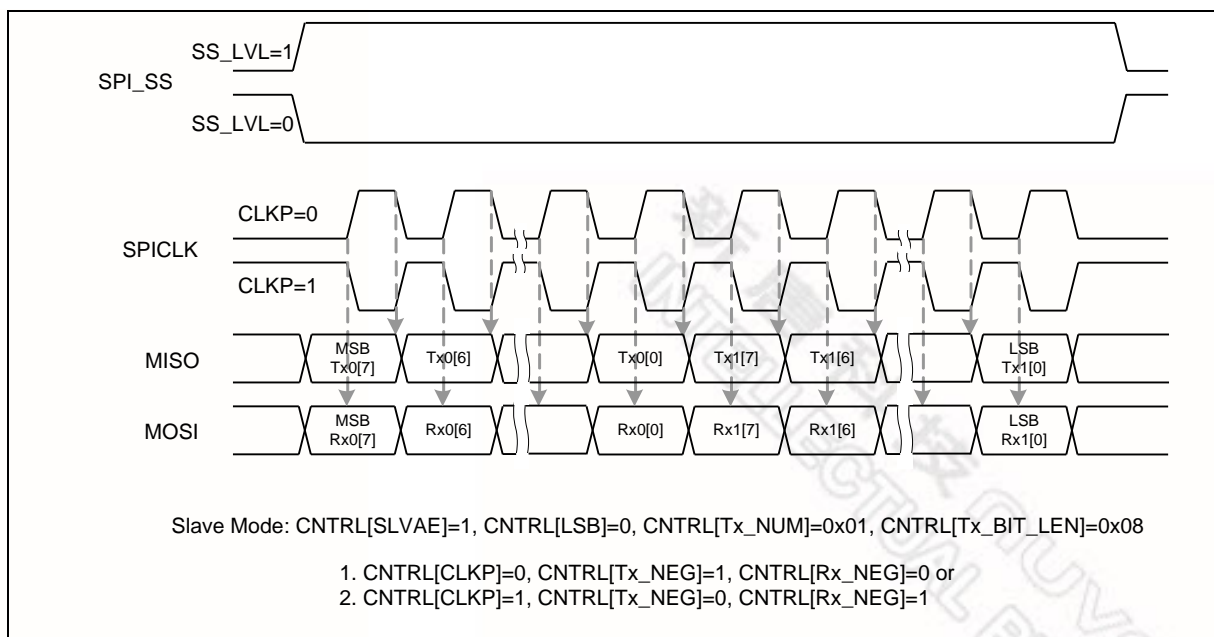
Figure 5-50 SPI Timing in Slave Mode
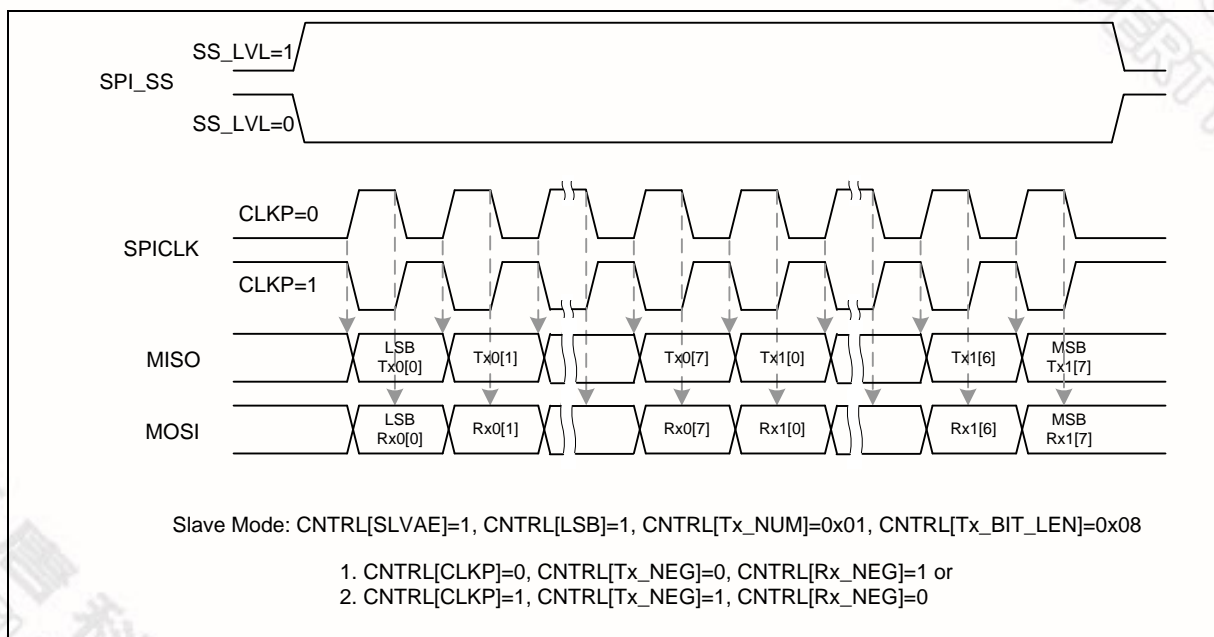
Figure 5-51 SPI Timing in Slave Mode (Alternate Phase of SPICLK)

5.9.6 SPI Configuration Examples

■ Example 1, SPI controller is set as a master to access an off-chip slave device with following specifications:

- Data bit latched on positive edge of serial clock
- Data bit driven on negative edge of serial clock
- Data be transferred from MSB first
- SCLK low in idle state
- Only one byte data be transmitted/received in a transfer
- Slave select signal is active low
- SCLK frequency is 10MHz

To configure the SPI interface to the above specifications perform the following steps:

1) Write a divisor into the SPI->DIVIDER register to determine the output frequency of serial clock. Driver function DrvSPI_SetClock(0,10000000,0) can be used to achieve this.

2) Configure the SPI->SSR register to address device. For example to manually address, set SPI-SSR.ASS=0, SPI->SSR.SSR_LVL=0 for active low SS. When software wishes to address device it will set SPI->SSR.SSR=1 to output an active SS on SPI_SSB0 pin.

3) Configure the SPI->CNTRL register. Set SPI->CNTRL.SLAVE=0 for master mode, set SPI->CNTRL.CLKP=0 for SCLK polarity normally low, set SPI->CNTRL.TX_NEG=1 so that data changes on falling edge of SCLK, set SPI->CNTRL.RX_NEG=0 so that data is latched into device on positive edge of SCLK, set SPI->CNTRL.DWIDTH=8 and SPI->CNTRL.TX_NUM=0 for a single byte transfer and finally set SPI->CNTRL.LSB=0 for MSB first transfer.

4) If manually selecting slave device set SPI->SSR.SSR=1.

5) To transmit one byte of data, write data to SPI->TX register. If only doing a receive, write a dummy byte to SPI->TX register.

6) Enable the SPI->CNTRL.SPIEN bit to start the data transfer over the SPI interface.

7) Wait for SPI transfer to finish. Can be interrupt driven (if the interrupt enable SPI->CNTRL.IE bit is set) or by polling the GO_BUSY bit which will be cleared to 0 by hardware automatically at end of transmission.--

8) Read out the received one byte data from SPI-> RX

9) Go to 5) to continue another data transfer or set SPI->SSR.SSR=0 to deactivate the off-chip slave devices.

■ Example 2, SPI controller is set as a slave device that controlled by an off-chip master device with the following characteristics:

- Data bit latched on positive edge of serial clock
- Data bit driven on negative edge of serial clock
- Data be transferred from LSB first
- SCLK high in idle state
- Only one byte data be transmitted/received in a transfer
- Slave select signal is active high level trigger

To configure the SPI interface to the above specifications perform the following steps:

1) Configure the SPI->SSR register. SPI->SSR.SS_LVL=1 for active high slave select, SPI->SSR.SSR.SS_LTRIG=1 for level sensitive trigger.

2) Configure the SPI->CNTRL register. Set SPI->CNTRL.SLAVE=1 for slave mode, set SPI->CNTRL.CLKP=1 for SCLK polarity idle high, set SPI->CNTRL.TX_NEG=1 so that data changes on falling edge of SCLK, set SPI->CNTRL.RX_NEG=0 so that data is latched into device on positive edge of SCLK, set SPI->CNTRL.DWIDTH=8 and SPI->CNTRL.TX_NUM=0 for a single byte transfer and finally set SPI->CNTRL.LSB=1 for LSB first transfer.

3) If SPI slave is to transmit one byte of data to the off-chip master device, write first byte to TX register. If no data to be transmitted write a dummy byte.

4) Enable the SPIEN bit to wait for the slave select trigger input and serial clock input from the off-chip master device to start the data transfer at the SPI interface.

5) Wait for SPI transfer to finish. Can be interrupt driven (if the interrupt enable SPI->CNTRL.IE bit is set) or by polling the GO_BUSY bit which will be cleared to 0 by hardware automatically at end of transmission.--

6) Read out the received data from RX register.

7) Go to 3) to continue another data transfer or disable the GO_BUSY bit to stop data transfer.

5.9.7 SPI Serial Interface Control Register Map

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

Register	Offset	R/W	Description	Reset Value	Reference
SPI0_BA = 0x4003_0000					
CNTRL	SPI0_BA + 0x00	R/W	Control and Status Register	0x0000_0034	
DIVIDER	SPI0_BA + 0x04	R/W	Clock Divider Register	0x0000_0000	
SSR	SPI0_BA + 0x08	R/W	Slave Select Register	0x0000_0000	Table 5-86
DMA	SPI0_BA + 0x0C	R/W	DMA Control Register	0x0000_0000	Table 5-87
FIFO	SPI0_BA + 0x10	R/W	FIFO Control/Status Register	0x4400_0000	Table 5-88
STATUS	SPI0_BA + 0x14	R/W	Status Register	0x0005_0110	Table 5-89
RX_TRANS_CNT	SPI0_BA + 0x18	R/W	Receive Transaction Count Register	0x0000_0000	Table 5-90
TX	SPI0_BA + 0x20	W	FIFO Data Transmit Register	0x0000_0000	Table 5-91
RX	SPI0_BA + 0x30	R	FIFO Data Receive Register	0x0000_0000	Table 5-92
VER_NUM	SPIx_BA + 0x50	R	IP Version Number Register	0x0201_0001	

5.9.8 SPI Control Register Description

<u>SPI Control and Status Register (CNTRL)</u>

Register	Offset	R/W	Description	Reset Value
CNTRL	SPI0_BA + 0x00	R/W	Control and Status Register	0x0000_0004

31	30	29	28	27	26	25	24
Reserved							RX_MODE
23	22	21	20	19	18	17	16
RX_TRANS_CNT_EN	QUAD_IO_EN	DUAL_IO_EN	QD_IO_DIR	REORDER	SLAVE	UNIT_INTEN	TWOB
15	14	13	12	11	10	9	8
RESERVED		LSB	DWIDTH				
7	6	5	4	3	2	1	0
SP_CYCLE				CLKP	TX_NEG	RX_NEG	SPIEN

Table 5-84 SPI Control and Status Register (CNTRL, address 0x4003_0000)

Bits	Descriptions	
[31:25]	**Reserved**	Reserved
[24]	**RX_MODE_EN**	**FIFO Receive Mode Enable** 1 = Enable FIFO receive mode. In this mode SPI transactions will be continuously performed while RX_FULL is not active. To stop transactions, set RX_MODE_EN to 0. 0 = Disable function.
[23]	**RX_TRANS_CNT_EN**	**DMA Receive Transaction Count Enable** 1 = Enable transaction counter for DMA receive only mode. SPI will perform the number of transfers specified in the RX_TRANS_CNT register, allowing the SPI interface to read ahead of DMA controller. 0 = Disable function.
[22]	**QUAD_IO_EN**	**Quad I/O Mode Enable** 1 = Quad I/O mode Enabled. 0 = Quad I/O mode Disabled.
[21]	**DUAL_IO_EN**	**Dual I/O Mode Enable** 1 = Dual I/O mode Enabled. 0 = Dual I/O mode Disabled.
[20]	**QD_IO_DIR**	**Quad or Dual I/O Mode Direction Control** 1 = Quad or Dual Output mode. 0 = Quad or Dual Input mode.

[19]	REORDER	**Byte Reorder Function Enable**
		1 = Byte reorder function Enabled. A byte suspend interval will be inserted between each byte. The period of the byte suspend interval depends on the setting of SP_CYCLE.
		0 = Byte reorder function Disabled.
		Note:
		1. Byte reorder function is only available if DWIDTH is defined as 16, 24, and 32 bits.
[18]	SLAVE	**Master Slave Mode Control**
		0 = Master mode.
		1 = Slave mode.
[17]	UNIT_INTEN	**Unit Transfer Interrupt Enable**
		0 = Disable SPI Unit Transfer Interrupt.
		1 = Enable SPI Unit Transfer Interrupt to CPU.
[16]	TWOB	**Two Bits Transfer Mode**
		1 = Enable two-bit transfer mode.
		0 = Disable two-bit transfer mode.
		When 2-bit mode is enabled, the first serial transmitted bit data is from the first FIFO buffer data, and the 2nd serial transmitted bit data is from the second FIFO buffer data. As the same as transmitted function, the first received bit data is stored into the first FIFO buffer and the 2nd received bit data is stored into the second FIFO buffer at the same time.
[15:14]	RESEVED	
[13]	LSB	**LSB First**
		0 = The MSB is transmitted/received first (which bit in TX and RX FIFO depends on the DWIDTH field).
		1 = The LSB is sent first on the line (bit 0 of TX FIFO]), and the first bit received from the line will be put in the LSB position in the SPI*n*->RX FIFO (bit 0 SPI*n*->RX).
[12:8]	DWIDTH	**DWIDTH – Data word Bit Length**
		This field specifies how many bits are transmitted in one transmit/receive. Up to 32 bits can be transmitted.
		DWIDTH = 0x01 … 1 bit
		DWIDTH = 0x02 … 2 bits
		……
		DWIDTH = 0x1f … 31 bits
		DWIDTH = 0x00 … 32 bits

[7:4]	SP_CYCLE	**Suspend Interval(Master Only)**
		
		The four bits provide configurable suspend interval between two successive transmit/receive transactions in a transfer. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value is 0x3. The period of the suspend interval is obtained according to the following equation.
		
		    (SP_CYCLE[3:0] + 0.5) * period of SPICLK clock cycle
		
		Example:
		
		    SP_CYCLE = 0x0 … 0.5 SPICLK clock cycle
		
		    SP_CYCLE = 0x1 … 1.5 SPICLK clock cycle
		
		    ……
		
		    SP_CYCLE = 0xE … 14.5 SPICLK clock cycle
		
		    SP_CYCLE = 0xF … 15.5 SPICLK clock cycle

[3]	CLKP	**Clock Polarity** 0 = SCLK idle low. 1 = SCLK idle high.
[2]	TX_NEG	**Transmit At Negative Edge** 0 = The transmitted data output signal is changed at the rising edge of SCLK. 1 = The transmitted data output signal is changed at the falling edge of SCLK.
[1]	RX_NEG	**Receive At Negative Edge** 0 = The received data input signal is latched at the rising edge of SCLK. 1 = The received data input signal is latched at the falling edge of SCLK.
[0]	SPIEN	**SPI Transfer Enable** 0 = Disable SPI Transfer. 1 = Enable SPI Transfer. In Master mode, the transfer will start when there is data in the FIFO buffer after this is set to 1. In Slave mode, the device is ready to receive data when this bit is set to 1. **Note:** All configurations should be set before writing 1 to this SPIEN bit. (e.g.: TX_NEG, RX_NEG, DWIDTH, LSB, CLKP, and so on).

SPI Divider Register (DIVIDER)

Register	Offset	R/W	Description	Reset Value
DIVIDER	SPI0_BA + 0x04	R/W	Clock Divider Register (Master Only)	0x0000_0000

Table 5-85 SPI Clock Divider Register (DIVIDER, address 0x4003_0004)

Bits	Descriptions	
[7:0]	DIVIDER	**Clock Divider Register** The value in this field is the frequency divider for generating the SPI engine clock, f_{spi_eclk}, and the SPI serial clock of SPI master. The frequency is obtained according to the following equation. $$f_{spi\_eclk} = \frac{f_{spi\_clock\_src}}{(DIVIDER+1)}$$ where $f_{spi\_clock\_src}$ is the SPI engine clock source, which is defined in the clock control, CLK_SEL1 register.

SPI Slave Select Register (SSR)

Register	Offset	R/W	Description	Reset Value
SSR	SPI0_BA + 0x08	R/W	Slave Select Register	0x0000_0000

31	30	29	28	27	26	25	24
SLVTPPRD[15:8]							
23	22	21	20	19	18	17	16
SLVTPPRD[7:0]							
15	14	13	12	11	10	9	8
Reserved		SSINA_INTEN	SSACT_INTEN	Reserved		SLVER1_INTEN	SLVER0_INTEN
7	6	5	4	3	2	1	0
Reserved	SLVTO_FFCLR	SLVTO_INTEN	SLV3WIRE	AUTOSS	SS_LVL	SSR	

Table 5-86 SPI Slave Select Register (SSR, address 0x4003_0008)

Bits	Description	
[31:16]	**SLVTOPRD**	**Slave Mode Time-Out Period** In Slave mode, these bits indicate the time out period when there is serial clock input during slave select active. The clock source of the time out counter is Slave engine clock. If the value is 0, it indicates the slave mode time-out function is disabled.
[15:14]	**Reserved**	Reserved
[13]	**SSINA_INTEN**	**Slave Select Inactive Interrupt Enable** 1 = Slave select inactive interrupt Enable 0 = Slave select inactive interrupt Disable
[12]	**SSACT_INTEN**	**Slave Select Active Interrupt Enable** 1 = Slave select active interrupt Enable 0 = Slave select active interrupt Disable
[11:10]	**Reserved**	Reserved
[9]	**SLVER1_INTEN**	**Slave Mode Error 1 Interrupt Enable** 1 = Slave mode error 1 interrupt Enable 0 = Slave mode error 1 interrupt Disable
[8]	**SLVER0_INTEN**	**Slave Mode Error 0 Interrupt Enable** 1 = Slave mode error 0 interrupt Enable 0 = Slave mode error 0 interrupt Disable
[7]	**Reserved**	Reserved
[6]	**SLVTO_FFCLR**	**Slave Mode Time-Out FIFO Clear** 1 = Both the FIFO clear function, TX_CLR and RX_CLR, are activated automatically when there is a slave mode time-out event.

		0 = Function disabled.
[5]	**SLVTO_INTEN**	**Slave Mode Time-Out Interrupt Enable** 1 = Slave mode time-out interrupt Enabled. 0 = Slave mode time-out interrupt Disabled.
[4]	**SLV3WIRE**	**Slave 3-Wire Mode Enable** This is used to ignore the slave select signal in Slave mode. The SPI controller can work with 3-wire interface consisting of SPI_CLK, SPI_MISO, and SPI_MOSI. 1 = 3-wire bi-directional interface. 0 = 4-wire bi-directional interface.
[3]	**AUTOSS**	**Automatic Slave Select Function Enable (Master Only)** 1 = If this bit is set, SPI_SS0/1 signals will be generated automatically. It means that device/slave select signal, which is set in SPI_SSCR[1:0], will be asserted by the SPI controller when transmit/receive is started, and will be de-asserted after each transmit/receive is finished. 0 = If this bit is cleared, slave select signals will be asserted/de-asserted by setting/clearing the corresponding bits of SPI_SSCR[1:0].
[2]	**SS_LVL**	**Slave Select Active Level** This bit defines the active status of slave select signal (SPI_SS0/1). 1 = The slave select signal SPI_SS0/1 is active on high-level/rising-edge. 0 = The slave select signal SPI_SS0/1 is active on low-level/falling-edge.
[1:0]	**SSR**	**Slave Select Control Bits (Master Only)** If AUTOSS bit is cleared, writing 1 to any bit of this field sets the proper SPISSx0/1 line to an active state and writing 0 sets the line back to inactive state. If the AUTOSS bit is set, writing 0 to any bit location of this field will keep the corresponding SPI_SS0/1 line at inactive state; writing 1 to any bit location of this field will select appropriate SPISSx0/1 line to be automatically driven to active state for the duration of the transmit/receive, and will be driven to inactive state for the rest of the time. The active state of SPI_SS0/1 is specified in SS_LVL. **Note:**SPI_SS0 is defined as the slave select input in Slave mode.

SPI DMA Control Register (DMA)

Register	Offset	R/W	Description	Reset Value
DMA	SPIx_BA+0x0C	R/W	SPI PDMA Control Register	0x0000_0000

Table 5-87 SPI DMA Control Register (address SPI_BA + 0x0C)

7	6	5	4	3	2	1	0
Reserved					PDMA _RST	RX_DMA_GO	TX_DMA_GO

Bits	Description	
[31:3]	Reserved	Reserved
[2]	PDMA_RST	**PDMA Reset** 1 = Reset the PDMA control logic of the SPI controller. This bit will be cleared to 0 automatically. 0 = No effect.
[1]	RX_DMA_GO	**Receive PDMA Enable** Setting this bit to 1 will start the receive PDMA process. The SPI controller will issue request to PDMA controller automatically when the SPI receive buffer is not empty. This bit will be cleared to 0 by hardware automatically after PDMA transfer is done.
[0]	TX_DMA_GO	**Transmit DMA Enable** Setting this bit to 1 will start the transmit PDMA process. SPI controller will issue request to PDMA controller automatically. Hardware will clear this bit to 0 automatically after PDMA transfer done.

SPI FIFO Control Register (FIFO_CTRL)

Register	Offset	R/W	Description	Reset Value
FIFO_CTRL	SPIx_BA+0x10	R/W	SPI FIFO Control Register	0x4400_0000

Table 5-88 SPI FIFO Control Register FIFO_CTRL (address SPI_BA + 0x10)

31	30	29	28	27	26	25	24
Reserved		TX_THRESHOLD		Reserved		RX_THRESHOLD	
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
TXUDR_INTEN	TXUDR_DO	RXOV_INTEN	RXTO_INTEN	TX_INTEN	RX_INTEN	TX_CLR	RX_CLR

Bits	Description	
[31]	**Reserved**	Reserved
[30:28]	**TX_THRESHOLD**	**Transmit FIFO Threshold** If the valid data count of the transmit FIFO buffer is less than or equal to the TX_THRESHOLD setting, the TXTH_INTSTS bit will be set to 1, else the TXTH_INTSTS bit will be cleared to 0.
[27]	**Reserved**	Reserved
[26:24]	**RX_THRESHOLD**	**Receive FIFO Threshold** If the valid data count of the receive FIFO buffer is larger than the RX_THRESHOLD setting, the RXTH_INTSTS bit will be set to 1, else the RXTH_INTSTS bit will be cleared to 0.
[23:8]	**Reserved**	Reserved
[7]	**TXUDR_INTEN**	**Slave Transmit Under Run Interrupt Enable** 1 = Slave Transmit FIFO under-run interrupt Enabled. 0 = Slave Transmit FIFO under-run interrupt Disabled.
[6]	**TXUDR_DO**	**Transmit Under-run Data Out** 1 = The SPI data out is keep 1 if there is transmit under-run event in Slave mode. 0 = The SPI data out is keep 0 if there is transmit under-run event in Slave mode. Note: The under run event is active after the serial clock input and the hardware synchronous, so that the first 1~3 bit (depending on the relation between system clock and the engine clock) data out will be the last transaction data. Note: If the frequency of system clock approach to engine clock, they may need 3-bit time to report the transmit under-run data out.
[5]	**RXOVR_INTEN**	**Receive FIFO Overrun Interrupt Enable** 1 = Receive FIFO overrun interrupt Enabled. 0 = Receive FIFO overrun interrupt Disabled.
[4]	**RXTO_INTEN**	**Slave Receive Time-out Interrupt Enable** 1 = Receive time-out interrupt Enabled. 0 = Receive time-out interrupt Disabled.
[3]	**TXTH_INTEN**	**Transmit FIFO Threshold Interrupt Enable** 1 = TX FIFO threshold interrupt Enabled. 0 = TX FIFO threshold interrupt Disabled.
[2]	**RXTH_INTEN**	**Receive FIFO Threshold Interrupt Enable** 1 = RX FIFO threshold interrupt Enabled. 0 = RX FIFO threshold interrupt Disabled.
[1]	**TX_CLR**	**Clear Transmit FIFO Buffer** 1 = Clear transmit FIFO buffer. The TX_FULL bit will be cleared to 0 and the TX_EMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 3 system clocks + 3 SPI engine clock after it is set to 1.

		0 = No effect. Note: If there is slave receive time out event, the TX_CLR will be set 1 when the SLVTO_FFCLR, SPI_SSCR[6], is enabled.
[0]	RX_CLR	**Clear Receive FIFO Buffer** 1 = Clear receive FIFO buffer. The RX_FULL bit will be cleared to 0 and the RX_EMPTY bit will be set to 1. This bit will be cleared to 0 by hardware about 3 system clocks + 3 SPI engine clock after it is set to 1. 0 = No effect. Note: If there is slave receive time out event, the RX_CLR will be set 1 when the SLVTO_FFCLR, SPI_SSCR[6], is enabled.

SPI Status Register (STATUS)

Register	Offset	R/W	Description	Reset Value
STATUS	SPIx_BA+0x14	R/W	SPI Status Register	0x0005_0110

Table 5-89 SPI Status Register STATUS (address SPI_BA + 0x14)

31	30	29	28	27	26	25	24
TX_FIFO_COUNT				RX_FIFO_COUNT			
23	22	21	20	19	18	17	16
FFCLR_STS	Reserved			TXUDR_INTSTS	TXTH_INTSTS	TX_FULL	TX_EMPTY
15	14	13	12	11	10	9	8
SPIEN_STS	Reserved		RXTO_INTSTS	RXOVR_INTSTS	RXTH_INTSTS	RX_FULL	RX_EMPTY
7	6	5	4	3	2	1	0
SLVER1_INTSTS	SLVER0_INTSTS	SLVTO_INTSTS	SS_LINE	SSINA_INTSTS	SSAC_INTSTS	UNIT_INTSTS	BUSY

Bits		Description
[31:28]	TX_FIFO_COUNT	**Transmit FIFO Data Count(Read Only)** This bit field indicates the valid data count of transmit FIFO buffer.
[27:24]	RX_FIFO_COUNT	**Receive FIFO Data Count(Read Only)** This bit field indicates the valid data count of receive FIFO buffer.
[23]	FFCLR_STS	**FIFO CLR Status(Read Only)** 1 = Doing the FIFO buffer clear function of TX_CLR or RX_CLR. 0 = Done the FIFO buffer clear function of TX_CLR and RX_CLR. **Note:** Both the TX_CLR, RX_CLR, need 3 system clock + 3 engine clock , the status of this bit allows the user to monitor whether the clear function is busy or done.
[22:20]	Reserved	Reserved

[19]	TXUDR_INTSTS	**Slave Transmit FIFO Under-Run Interrupt Status(Read Only)** When the transmit FIFO buffer is empty and further serial clock pulses occur, data transmitted will be the value of the last transmitted bit and this under-run bit will be set. **Note:** This bit will be cleared by writing 1 to itself.
[18]	TXTH_INTSTS	**Transmit FIFO Threshold Interrupt Status (Read Only)** 1 = The valid data count of the transmit FIFO buffer is less than or equal to the setting value of TX_THRESHOLD. 0 = The valid data count of the transmit FIFO buffer is larger than the setting value of TX_THRESHOLD. **Note:** If TXTH_INTEN = 1 and TXTH_INTSTS = 1, the SPI controller will generate a SPI interrupt request.
[17]	TX_FULL	**Transmit FIFO Buffer Full Indicator(Read Only)** 1 = Transmit FIFO buffer is full. 0 = Transmit FIFO buffer is not full.
[16]	TX_EMPTY	**Transmit FIFO Buffer Empty Indicator(Read Only)** 1 = Transmit FIFO buffer is empty. 0 = Transmit FIFO buffer is not empty.
[15]	SPIEN_STS	**SPI Enable Bit Status(Read Only)** 1 = Indicate the transfer control bit is active. 0 = Indicate the transmit control bit is disabled. **Note:** The clock source of SPI controller logic is engine clock, it is asynchronous with the system clock. In order to make sure the function is disabled in SPI controller logic, this bit indicates the real status of SPIEN in SPI controller logic for user.
[14:13]	**Reserved**	Reserved
[12]	RXTO_INTSTS	**Receive Time-out Interrupt Status** 1 = Receive FIFO buffer is not empty and no read operation on receive FIFO buffer over 64 SPI clock period in Master mode or over 576 SPI engine clock period in Slave mode. When the received FIFO buffer is read by software, the time-out status will be cleared automatically. 0 = No receive FIFO time-out event. **Note:** This bit will be cleared by writing 1 to itself.
[11]	RXOVR_INTSTS	**Receive FIFO Overrun Status** When the receive FIFO buffer is full, the follow-up data will be dropped and this bit will be set to 1. **Note:** This bit will be cleared by writing 1 to itself.
[10]	RXTH_INTSTS	**Receive FIFO Threshold Interrupt Status (Read Only)** 1 = The valid data count within the receive FIFO buffer is larger than the setting value of RX_THRESHOLD. 0 =The valid data count within the Rx FIFO buffer is smaller than or equal to the setting value of RX_THRESHOLD. **Note:** If RX_INTEN = 1 and RX_INTSTS = 1, the SPI controller will generate a SPI interrupt request.
[9]	RX_FULL	**Receive FIFO Buffer Empty Indicator(Read Only)** 1 = Receive FIFO buffer is empty. 0 = Receive FIFO buffer is not empty.
[8]	RX_EMPTY	**Receive FIFO Buffer Empty Indicator(Read Only)** 1 = Receive FIFO buffer is empty.

		0 = Receive FIFO buffer is not empty.
[7]	SLVER1_INTSTS	**Slave Mode Error 1 Interrupt Status(Read Only)** In Slave mode, transmit under-run occurs when the slave select line goes to inactive state. 1 = Slave mode error 1 occurs. 0 = No Slave mode error 1 event.
[6]	SLVER0_INTSTS	**Slave Mode Error 0 Interrupt Status(Read Only)** In Slave mode, there is bit counter mismatch with DWIDTH when the slave select line goes to inactive state. 1 = Slave mode error 0 occurs. 0 = No Slave mode error 0 event. **Note:** If the slave select active but there is no any serial clock input, the SLVER0_INTSTS also active when the slave select goes to inactive state.
[5]	SLVTO_INTSTS	**Slave Time-out Interrupt Status (Read Only)** When the Slave Select is active and the value of SLVTOPRD is not 0 and the serial clock input, the slave time-out counter in SPI controller logic will be start. When the value of time-out counter greater or equal than the value of SLVTOPRD, SPI_SSCR[31:16], during before one transaction done, the slave time-out interrupt event will active. 1 = Slave time-out is active. 0 = Slave time-out is not active **Note:** If the DWIDTH is set 16, one transaction is equal 16 bits serial clock period.
[4]	SS_LINE	**Slave Select Line Bus Status(Read Only)** 1 = Indicates the slave select line bus status is 1. 0 = Indicates the slave select line bus status is 0. **Note:** If SS_LVL, SPI_SSCR[2], is set 0, and the SS_LINE is 1, the SPI slave select is in inactive status.
[3]	SSINA_INTSTS	**Slave Select Inactive Interrupt Status** 1 =Slave select inactive interrupt event has occur. 0 = Slave select inactive interrupt is clear or not occur. **Note:** This bit will be cleared by writing 1 to itself.
[2]	SSACT_INTSTS	**Slave Select Active Interrupt Status** 1 =Slave select active interrupt event has occur. 0 = Slave select active interrupt is clear or not occur. **Note:** This bit will be cleared by writing 1 to itself.
[1]	UNIT_INTSTS	**Unit Transfer Interrupt Status** 1 = SPI controller has finished one unit transfer. 0 = No transaction has been finished since this bit was cleared to 0. **Note:** This bit will be cleared by writing 1 to itself.
[0]	BUSY	**SPI Unit Bus Status (Read Only)** 1 = SPI controller unit is in busy state. 0 = No transaction in the SPI bus. The following listing are the bus busy conditions: a. SPIEN = 1 and the TX_EMPTY = 0 b. For SPI Master, the TX_EMPTY = 1 but the current transaction is not finished yet. c. For SPI Slave receive mode, the SPIEN = 1 and there is serial clock input into the SPI core logic when slave select is active. d. For SPI Slave transmit mode, the SPIEN = 1 and the transmit buffer is not empty in SPI core

			logic event if the slave select is inactive.	

SPI Receive Transaction Count (RX_TRANS_CNT)

Register	Offset	R/W	Description	Reset Value
RX_TRANS_CNT	SPIx_BA+0x18	R/W	Receive Transaction Count	0x0000_0000

Table 5-90 SPI Receive Transaction Count (RX_TRANS_CNT, address SPIx_BA + 0x18)

Bits	Description	
[16:0]	**RX_TRANS_CNT**	**DMA Receive Transaction Count** When using DMA to receive SPI data without transmitting data, this register can be used in conjunction with the control bit CTRL.RX_TRANS_CNT_EN to set number of transactions to perform. Without this, the SPI interface will only initiate a transaction when it receives a request from the DMA system, resulting in a lower achievable data rate.

SPI Data Transmit Register (TX)

Register	Offset	R/W	Description	Reset Value
TX	SPIx_BA+0x20	W	Data Transmit FIFO	0x0000_0000

Table 5-91 SPI Data Transmit Register (TX, address SPIx_BA + 0x20)

Bits	Description	
[31:0]	**TX**	**Data Transmit Register** A write to the data transmit register pushes data onto into the 8-level transmit FIFO buffer. The number of valid bits depends on the setting of transmit bit width field of the SPI_CTL register. For example, if DWIDTH is set to 0x08, the bits TX[7:0] will be transmitted. If DWIDTH is set to 0x00, the SPI controller will perform a 32-bit transfer.

SPI Data Receive Register (RX)

Register	Offset	R/W	Description	Reset Value
RX	SPIx_BA+0x30	R	Data Receive FIFO Register	0x0000_0000

Table 5-92SPI Data Receive Register (RX, address SPIx_BA + 30)

Bits	Description	
[31:0]	**RX**	**Data Receive Register** A read from this register pops data from the 8-level receive FIFO. Valid data is present if the RX_EMPTY bit,

		SPI->STATUS[8],is not set to 1. This is a read-only register.

5.10 Timer Controller

5.10.1 General Timer Controller

The ISD93xx provides two general 24bit timer modules, TIMER0 and TIMER1. They allow the user to implement event counting or provide timing control for applications. The timer can perform functions such as frequency measurement, event counting, interval measurement, clock generation and delay timing. The timer can generates an interrupt signal upon timeout and provide the current value of count during operation.

5.10.2 Features

• Independent clock source for each channel(TMR0_CLK, TMR1_CLK).

• Time out period = (Period of timer clock input) * (8-bit prescale + 1) * (24-bit TCMP)

• Maximum count cycle time = (1 / TMR_CLK) * (2^8) * (2^24).

• Internal 24-bit up counter is readable through TDR (Timer Data Register).

5.10.3 Timer Controller Block Diagram

Each channel is equipped with an 8-bit pre-scale counter, a 24-bit up-counter, a 24-bit compare register and an interrupt request signal. Refer to Figure 5-52 for the timer controller block diagram. There are five options of clock source for each channel, Figure 5-53illustrate the clock source control function.
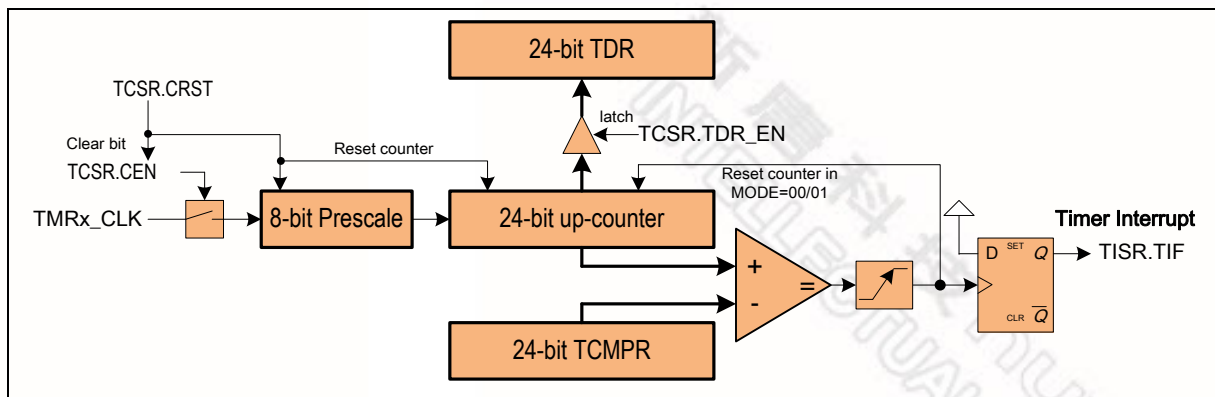
Figure 5-52 Timer Controller Block Diagram
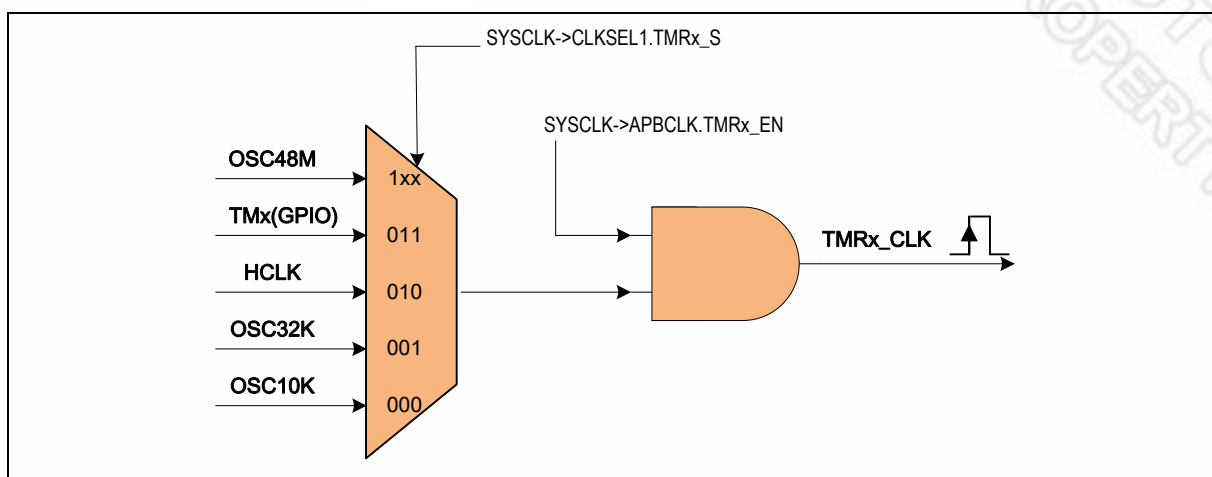
Figure 5-53 Clock Source of Timer Controller

5.10.4 Timer Controller Register Map
R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value	Reference
TMR0_BA = 0x4001_0000					
TMR1_BA = 0x4001_0020					
TCSR	TMRx_BA+00	R/W	Timer0 Control and Status Register	0x0000_0005	Table 5-93
TCMPR	TMRx_BA+04	R/W	Timer0 Compare Register	0x0000_0000	Table 5-94
TISR	TMRx_BA+08	R/W	Timer0 Interrupt Status Register	0x0000_0000	Table 5-95
TDR	TMRx_BA+0C	R	Timer0 Data Register	0x0000_0000	Table 5-96

Timer Control Register (TCSR)

Register	Offset	R/W	Description	Reset Value
TCSR	TMRx_BA+000	R/W	Timer Control and Status Register	0x0000_0005

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

Table 5-93 Timer Control and Status Register (TCSR, address 0x4001_0000 + x *0x20).

Bits	Descriptions	
[31]	Reserved	Reserved
[30]	CEN	**Counter Enable Bit** 0 = Stops/Suspends counting 1 = Starts counting **Note1**: Setting CEN=1 enables 24-bit counter. It continues count from last value. **Note2**: This bit is auto-cleared by hardware in one-shot mode (MODE=00b) when the timer interrupt is generated (IE=1b).
[29]	IE	**Interrupt Enable Bit** 0 = Disable TIMER Interrupt. 1 = Enable TIMER Interrupt. If timer interrupt is enabled, the timer asserts its interrupt signal when the count is equal to TCMPR.

		Timer Operating Mode	
[28:27]	**MODE**	MODE	Timer Operating Mode
		00	The timer is operating in the one-shot mode. The associated interrupt signal is generated once (if IE is enabled) and CEN is automatically cleared by hardware.
		01	The timer is operating in the periodic mode. The associated interrupt signal is generated periodically (if IE is enabled).
		10	Reserved.
		11	The timer is operating in continuous counting mode. The associated interrupt signal is generated when TDR = TCMPR (if IE is enabled); however, the 24-bit up-counter counts continuously without reset.
[26]	**CRST**	**Counter Reset Bit** Set this bit will reset the timer counter, prescale and also force CEN to 0. 0 = No effect. 1 = Reset Timer's prescale counter, internal 24-bit up-counter and CEN bit.	
[25]	**CACT**	**Timer Active Status Bit (Read only)** This bit indicates the counter status of timer. 0 = Timer is **not** active. 1 = Timer is active.	
[24:17]	Reserved	Reserved	
[16]	**TDR_EN**	**Data Latch Enable** When TDR_EN is set, TDR (Timer Data Register) will be updated continuously with the 24-bit up-counter value as the timer is counting. 1 = **Timer Data Register** update enable. 0 = **Timer Data Register** update disable.	
[15:8]	Reserved	Reserved	
[7:0]	**PRESCALE**	**Pre-scale Counter** Clock input is divided by PRESCALE+1 before it is fed to the counter. If PRESCALE =0, then there is no scaling.	

Timer Compare Register (TCMPR)

Register	Offset	R/W	Description	Reset Value
TCMPR	TMRx_BA+004	R/W	Timer Compare Register	0x0000_0000

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

Table 5-94 Timer Compare Register (TCMPR, address 0x4001_0004 + x * 0x20)

Bits	Descriptions	
[24:0]	**TCMP**	**Timer Comparison Value** TCMP is a 24-bit comparison register. When the 24-bit up-counter is enabled and its value is equal to TCMP value, a Timer Interrupt is requested if the timer interrupt is enabled with TCSR.IE=1. The TCMP value defines the timer cycle time. Time out period = (Period of timer clock input) * (8-bit PRESCALE + 1) * (24-bit TCMP) **NOTE1:** Never set TCMP to 0x000 or 0x001. Timer will not function correctly. **NOTE2:** Regardless of CEN state, whenever a new value is written to this register, TIMER will restart counting using this new value and abort previous count.

Timer Interrupt Status Register (TISR)

Register	Offset	R/W	Description	Reset Value
TISR	TMRx_BA+08	R/W	Timer Interrupt Status Register	0x0000_0000

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

Table 5-95Timer Interrupt Status Register (TISR, address 0x4001_0008 + x * 0x20)

Bits	Descriptions	
[31:1]	Reserved	Reserved
[0]	**TIF**	**Timer Interrupt Flag** This bit indicates the interrupt status of Timer. TIF bit is set by hardware when the 24-bit counter matches the timer comparison value (TCMP). It is cleared by writing 1.

Timer Data Register (TDR)

Register	Offset	R/W	Description	Reset Value
TDR	TMRx_BA+0C	R/W	Timer Data Register	0x0000_0000

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

Table 5-96Timer Data Register (TDR, address 0x4001_000C + x *0x20).

Bits	Descriptions	
[31:24]	Reserved	Reserved
[23:0]	**TDR**	**Timer Data Register** When TCSR.TDR_EN is set to 1, the internal 24-bit timer up-counter value will be latched into TDR. User can read this register for the up-counter value.

5.11 Watchdog Timer

The purpose of Watchdog Timer is to perform a system reset if software is not responding as designed. This prevents system from hanging for an infinite period of time. The watchdog timer includes a 18-bit free running counter with programmable time-out intervals.

Setting WTE enables the watchdog timer and the WDT counter starts counting up. When the counter reaches the selected time-out interval, Watchdog timer interrupt flag WTIF will be set immediately to request a WDT interrupt if the watchdog timer interrupt enable bit WTIE is set, in the meantime, a specified delay time follows the time-out event. User must set WTR (Watchdog timer reset) high to reset the 18-bit WDT counter to prevent Watchdog timer reset before the delay time expires. WTR bit is auto cleared by hardware after WDT counter is reset. There are eight time-out intervals with specific delay time which are selected by Watchdog timer interval select bits WTIS. If the WDT counter has not been cleared after the specific delay time expires, the watchdog timer will set Watchdog Timer Reset Flag (WTRF) high and reset CPU. This reset will last 64 WDT clocks then CPU restarts executing program from reset vector (0x0000 0000). WTRF will not be cleared by Watchdog reset. User may poll WTFR by software to recognize the reset source.

If the application uses any sleep modes (calling WFI or WFE instructions), the watchdog reset may not fully reset the M0 core due to parts of the core being un-clocked. In this case application should detect the WTRF in boot sequence and perform a Deep Power Down (DPD) to ensure complete reset. See the Timer driver sample code for example.

Table 5-97Watchdog Timeout Interval Selection

WTIS	Interrupt Timeout	Watchdog Reset Timeout	WTR Timeout Interval (WDT_CLK=49.152 MHz)	WTR Timeout Interval (WDT_CLK=32kHz)
000	2^4 WDT_CLK	$(2^4 + 1024)$ WDT_CLK	21.2us	31.7 ms
001	2^6 WDT_CLK	$(2^6 + 1024)$ WDT_CLK	22.1 us	33.2 ms
010	2^8 WDT_CLK	$(2^8 + 1024)$ WDT_CLK	26.0 us	39 ms
011	2^{10} WDT_CLK	$(2^{10} + 1024)$ WDT_CLK	41.7 us	64 ms
100	2^{12} WDT_CLK	$(2^{12} + 1024)$ WDT_CLK	104.2 us	160 ms
101	2^{14} WDT_CLK	$(2^{14} + 1024)$ WDT_CLK	354.2 us	544 ms
110	2^{16} WDT_CLK	$(2^{16} + 1024)$ WDT_CLK	1.4 ms	2080 ms
111	2^{18} WDT_CLK	$(2^{18} + 1024)$ WDT_CLK	5.4 ms	8224 ms
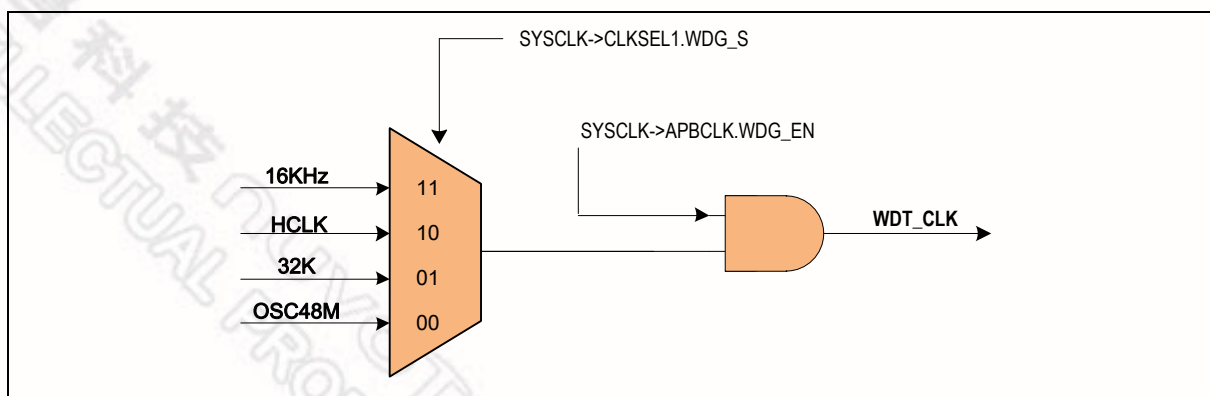
Figure 5-54 Watchdog Timer Clock Control
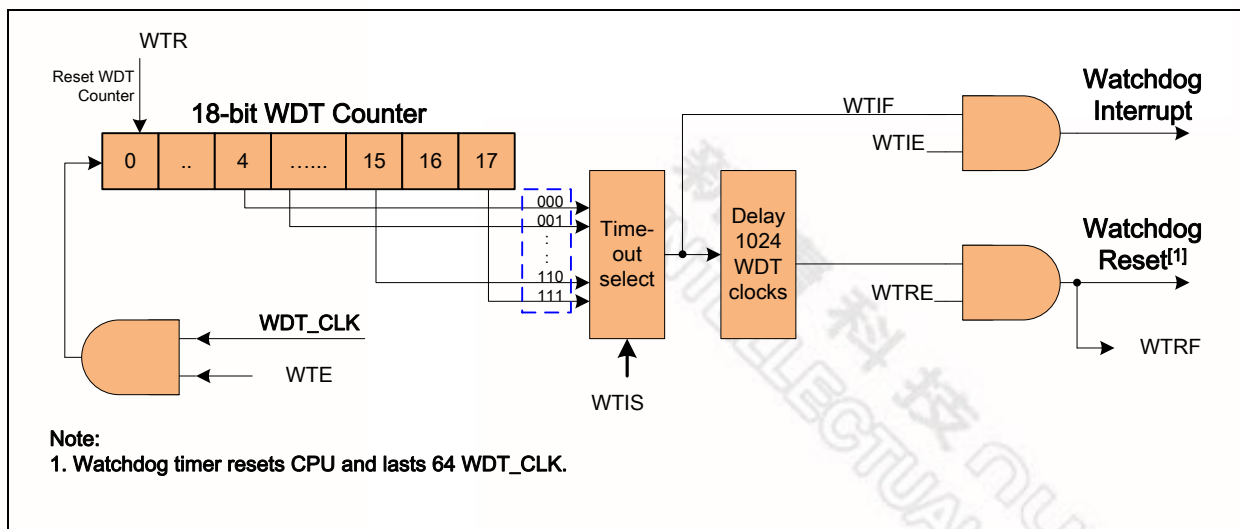
Figure 5-55 Watchdog Timer Block Diagram

5.11.1 Watchdog Timer Control Registers Map

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

Register	Offset	R/W	Description	Reset Value
WDT_BA = 0x4000_4000				
WTCR	WDT_BA+00	R/W	Watchdog Timer Control Register	0x0000_0700

Watchdog Timer Control Register (WTCR)

This is a protected register, to write to register, first issue the unlock sequence (see Protected Register Lock Key Register (REGLOCK)). Only flag bits, WTIF and WTRF are unprotected and can be write-cleared at any time.

Register	Offset	R/W	Description	Reset Value
WTCR	WDT_BA+000	R/W	Watchdog Timer Control Register	0x0000_0700

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					WTIS		
7	6	5	4	3	2	1	0
WTE	WTIE	Reserved		WTIF	WTRF	WTRE	WTR

Bits	Descriptions	
[31:11]	Reserved	Reserved
[10:8]	**WTIS**	**Watchdog Timer Interval Select** These three bits select the timeout interval for the Watchdog timer, a watchdog reset will occur 1024 clock cycles later if WDG not reset. The timeout is given by: $$\text{Interrupt Timeout} = 2^{(2 \times WTIS+4)} \times WDT\_CLK$$ $$\text{Reset Timeout} = (2^{(2 \times WTIS+4)} + 1024) \times WDT\_CLK$$ Where WDT_CLK is the period of the Watchdog Timer clock source.
[7]	**WTE**	**Watchdog Timer Enable** 0 : Disable the Watchdog timer (This action will reset the internal counter) 1 : Enable the Watchdog timer

[6]	**WTIE**	**Watchdog Timer Interrupt Enable** 0 : Disable the Watchdog timer interrupt 1 : Enable the Watchdog timer interrupt
[3]	**WTIF**	**Watchdog Timer Interrupt Flag** If the Watchdog timer interrupt is enabled, then the hardware will set this bit to indicate that the Watchdog timer interrupt has occurred. If the Watchdog timer interrupt is not enabled, then this bit indicates that a timeout period has elapsed. 0 : Watchdog timer interrupt has not occurred. 1 : Watchdog timer interrupt has occurred. **NOTE**: This bit is cleared by writing 1 to this bit.
[2]	**WTRF**	**Watchdog Timer Reset Flag** When the Watchdog timer initiates a reset, the hardware will set this bit. This flag can be read by software to determine the source of reset. Software is responsible to clear it manually by writing 1 to it. If **WTRE** is disabled, then the Watchdog timer has no effect on this bit. 0 : Watchdog timer reset has not occurred. 1: Watchdog timer reset has occurred. **NOTE**: This bit is cleared by writing 1 to this bit.
[1]	**WTRE**	**Watchdog Timer Reset Enable** Setting this bit will enable the Watchdog timer reset function. 0 : Disable Watchdog timer reset function 1: Enable Watchdog timer reset function
[0]	**WTR**	**Clear Watchdog Timer** Set this bit will clear the Watchdog timer. 0 : Writing 0 to this bit has no effect 1 : Reset the contents of the Watchdog timer **NOTE**: This bit will auto clear after few clock cycle

5.12 UART Interface Controller

The ISD93xx includes a Universal Asynchronous Receiver/Transmitter (UART). The UART supports high speed operation and flow control functions as well as protocols for Serial Infrared (IrDA) and Local interconnect Network (LIN).

5.12.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) performs a serial-to-parallel conversion on data received from the peripheral, and a parallel-to-serial conversion on data transmitted from the CPU. The UART controller also supports LIN(Local Interconnect Network) master mode function and IrDA SIR (Serial Infrared)function. The UART channel supports seven types of interrupts including transmitter FIFO empty interrupt(THRE_INT), receiver threshold level interrupt (RDA_INT), line status interrupt (overrun error or parity error or framing error or break interrupt) (RLS_INT), time out interrupt (TOUT_INT), MODEM status interrupt (MODEM_INT), Buffer error interrupt (BUF_ERR_INT) and LIN receiver break field detected interrupt.

The UART has a 8-byte transmit FIFO (TX_FIFO) and a 8-bytereceive FIFO (RX_FIFO) that reduces the number of interrupts presented to the CPU. The CPU can read the status of the UART at any time during the operation. The reported status information includes the type and condition of the transfer operations being performed by the UART, as well as 4 error conditions (parity error, overrun error, framing error and break interrupt) that can occur while receiving data. The UART includes a programmable baud rate generator that is capable of dividing master clock input by divisors to produce the baud rate clock. The baud rate equation is Baud Rate = UART_CLK / M * [BRD + 2], where M and BRD are defined in Baud Rate Divider Register (UART0->BAUD). Table 5-98lists the equations under various conditions.

The UART controller supports auto-flow control function that uses two active-low signals,/CTS (clear-to-send) and /RTS (request-to-send), to control the flow of data transfer between the UART and external devices (e.g. Modem). When auto-flow is enabled, the UART will not receive data until the UART asserts /RTS to external device. When the number of bytes in the Rx FIFO equals the value ofUART0->FCR.RTS_TRIG_LEVEL, the /RTS is de-asserted. The UART sends data out when UART controller detects /CTS is asserted from external device. If /CTS is not detected the UART controller will not send data out.

The UART controller also provides Serial IrDA (SIR, Serial Infrared) function (UART0->FUNSEL.IrDA_EN =1 to enable IrDA function). The SIR specification defines a short-range infrared asynchronous serial transmission mode with one start bit, 8 data bits, and 1 stop bit. The maximum data rate is 115.2 Kbps (half duplex). The IrDA SIR block contains an IrDA SIR Protocol encoder/decoder. The IrDA SIR protocol is half-duplex only. So it cannot transmit and receive data at the same time. The IrDA SIR physical layer specifies a minimum 10ms transfer delay between transmission and reception. This delay must be implemented by software.

The alternate function of UART controller is LIN (Local Interconnect Network) function. The LIN mode is selected by setting the UART0->FUNSEL.LIN_EN bit. In LIN mode, one start bit, 8-bit data format with 1-bit stop bit are generated in accordance with the LIN standard.

Table 5-98 UART Baud Rate Equation

Mode	DIVX_EN	DIVX_ONE	DIVX[3:0]	BRD[15:0]	Baud rate equation
0	0	0	B	A	UART_CLK / [16 * (A+2)]
1	1	0	B	A	UART_CLK / [(B+1) * (A+2)] , B ≥ 8
2	1	1	Don't care	A	UART_CLK / (A+2), A ≥ 3

Table 5-99 UART Baud Rate Setting Table

System clock = 49.152MHz						
Baud rate	Mode0	%err	Mode1	%err	Mode2	%err
921600	x		A=4,B=8	1.2	A=51	-0.6
460800	x		A=10,B=8	1.2	A=104	0.3
230400	x		A=22,B=8 A=7,B=11	1.2 1.2	A=211	-0.2
115200	A=25	1.2	A=37,B=10 A=31,B=12	0.5 0.5	A=425	0.1
57600	A=51	-0.6	A=59,B=13 A=93,B=8	0.1 0.2	A=851	0.0
38400	A=78	0.0	A=126,B=9 A=78,B=15	0.0 0.0	A=1278	0.0
19200	A=158	0.0	A=254,B=9 A=158,B=15	0.0 0.0	A=2558	0.0
9600	A=318	0.0	A=510,B=9 A=318,B=15	0.0 0.0	A=5118	0.0
4800	A=638	0.0	A=1022,B=9 A=638,B=15	0.0 0.0	A=10238	0.0

5.12.2 Features of UART controller

- UART supports 8 byte FIFO for receive and transmit data payloads.

- PDMA access support.

- Auto flow control function (/CTS, /RTS) supported.

- Programmable baud-rate generator.

- Fully programmable serial-interface characteristics:

    o 5-, 6-, 7-, or 8-bit character.

    o Even, odd, or no-parity bit generation and detection.

    o 1-, 1&1/2, or 2-stop bit generation.

    o Baud rate generation.

    o False start bit detection.

- IrDA SIR Function.

- LIN master mode.

5.12.3 Block Diagram

The UART clock control and block diagram are shown as following.
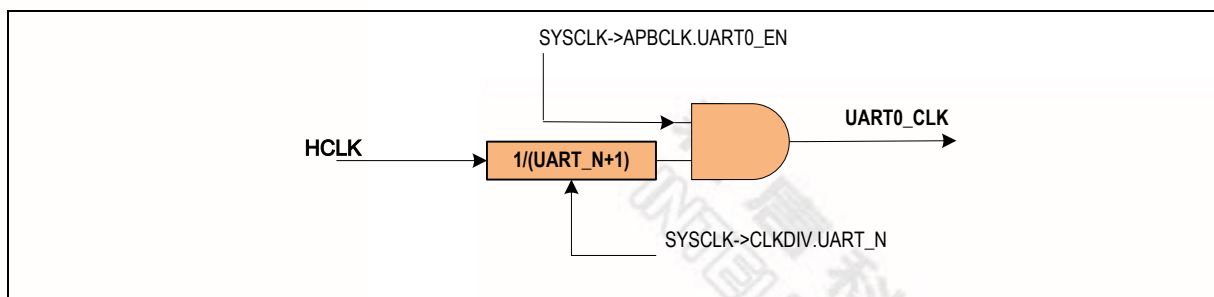
Figure 5-56 UART Clock Control Diagram
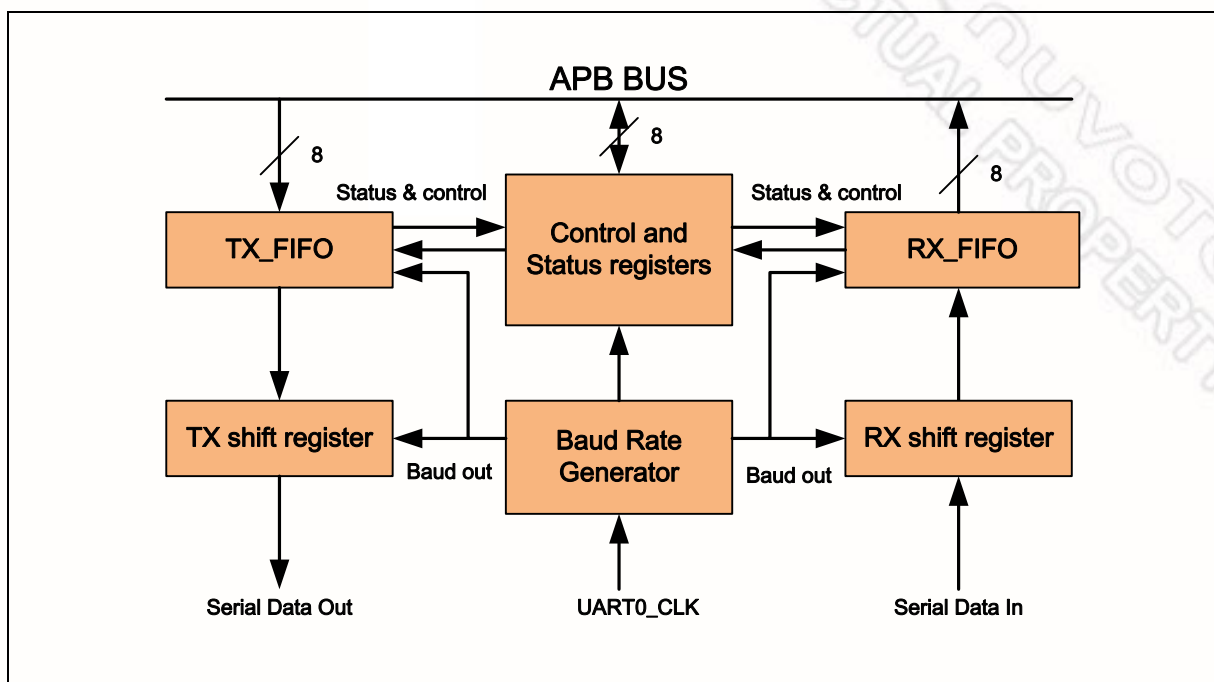
Figure 5-57 UART Block Diagram

TX_FIFO
The transmitter is buffered with an 8 byte FIFO to reduce the number of interrupts presented to the CPU.

RX_FIFO
The receiver is buffered with an 8 byte FIFO (plus three error bits per byte) to reduce the number of interrupts presented to the CPU.

TX shift Register
Shifts the transmit data out serially

RX shift Register
Shifts the receive data in serially

Modem Control Register
This register controls the interface to the MODEM or data set (or a peripheral device emulating a MODEM).

Baud Rate Generator
Divides the UART0_CLK clock by the divisor to get the desired baud rate clock. Refer to Table 5-98for the baud rate equation.

Control and Status Register
This is a register set, including the FIFO control registers (FCR), FIFO status registers (FSR), and line control register (LCR) for transmitter and receiver. The time out control register (TOR) identifies the condition of time out interrupt. This register set also includes the interrupt enable register (IER) and interrupt status register (ISR) to enable or disable the responding interrupt and to identify the occurrence of the responding interrupt. There are six types of interrupts, transmitter FIFO empty interrupt(THRE_INT), receiver threshold level reaching interrupt (RDA_INT), line status interrupt (overrun error or parity error or framing error or break interrupt) (RLS_INT) , time out interrupt (TOUT_INT), MODEM status interrupt (MODEM_INT) and Buffer error interrupt (BUF_ERR_INT).

Figure 5-58 demonstrates the auto-flow control block diagram.
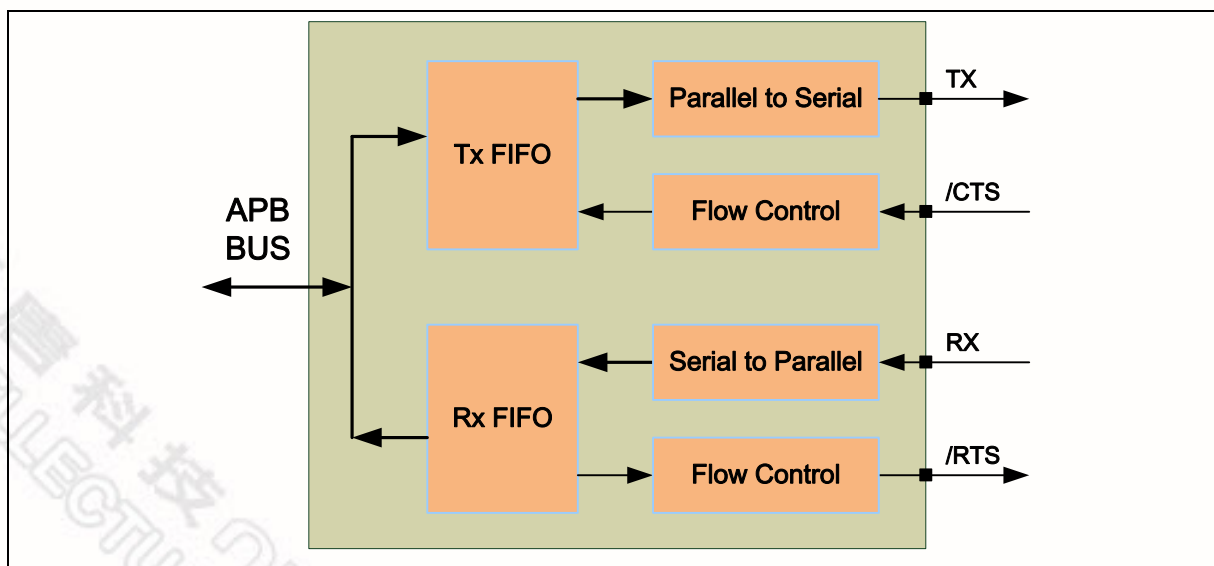
Figure 5-58 Auto Flow Control Block Diagram

5.12.4 IrDA Mode

The UART supports IrDA SIR (Serial Infrared) Transmit Encoder and Receive Decoder. IrDA mode is selected by setting the UART0->FUNSEL.IrDA_ENbit.

When in IrDA mode, the UART0->BAUD.DIVX_EN register must be zero and baud rate is given by:

Baud Rate = UART_CLK / (16 * BRD), where BRD is Baud Rate Divider in the UART0->BAUD.BRD register.
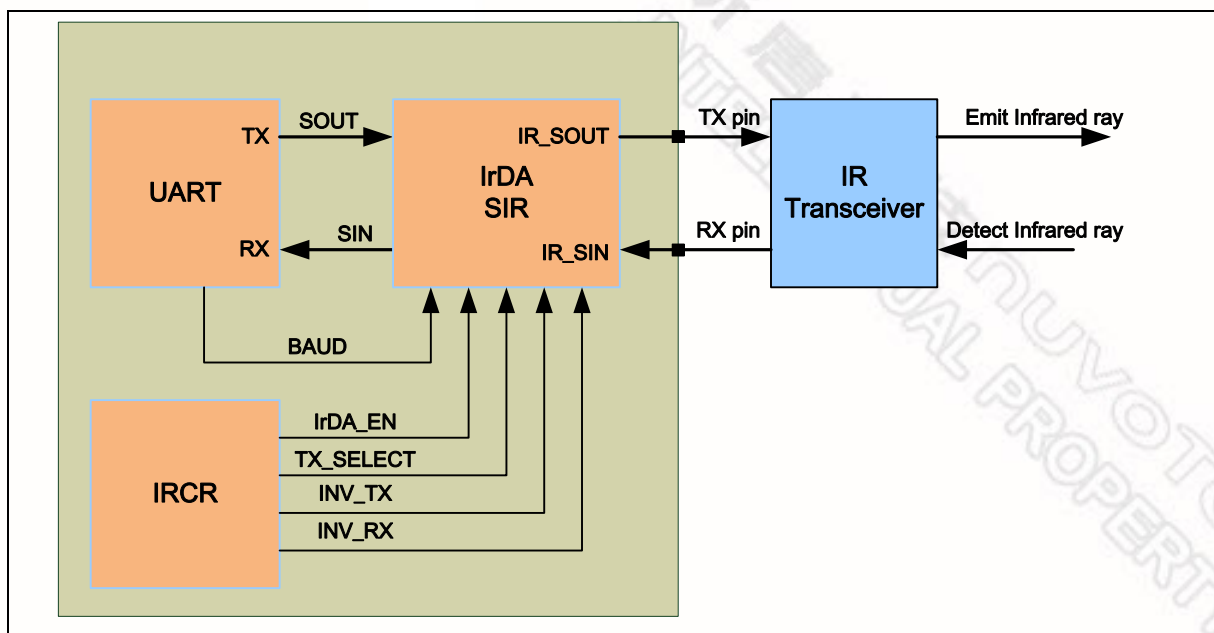
Figure 5-59 IrDA Block Diagram

5.12.4.1 *IrDA SIR Transmit Encoder*

The IrDA SIR Transmit Encoder modulates Non-Return-to Zero (NRZ) transmission bit stream from UART serial output. The IrDA SIR physical layer specifies use of Return-to-Zero, Inverted (RZI) modulation scheme which represents logic 0 as an infrared light pulse. The modulated output pulse stream is transmitted to an external output driver and infrared LED (Light Emitting Diode).In normal mode, the transmitted pulse width is specified as 3/16the period of the baud rate.

5.12.4.2 *IrDA SIR Receive Decoder*

The IrDA SIR Receive Decoder demodulates the return-to-zero bit stream from the input detector and outputs the NRZ serial bit stream to the UART received data input. The IR_SIN decoder input is normally high in the idle state. Because of this, IRCR.RX_INV_EN should be set 1 by default). A start bit is detected when the IR_SIN decoder input is LOW.

5.12.4.3 *IrDA SIR Operation*

The IrDA SIR Encoder/decoder provides functionality which converts between UART data stream and half duplex serial SIR interface. Figure 5-60 shows the IrDA encoder/decoder waveform:
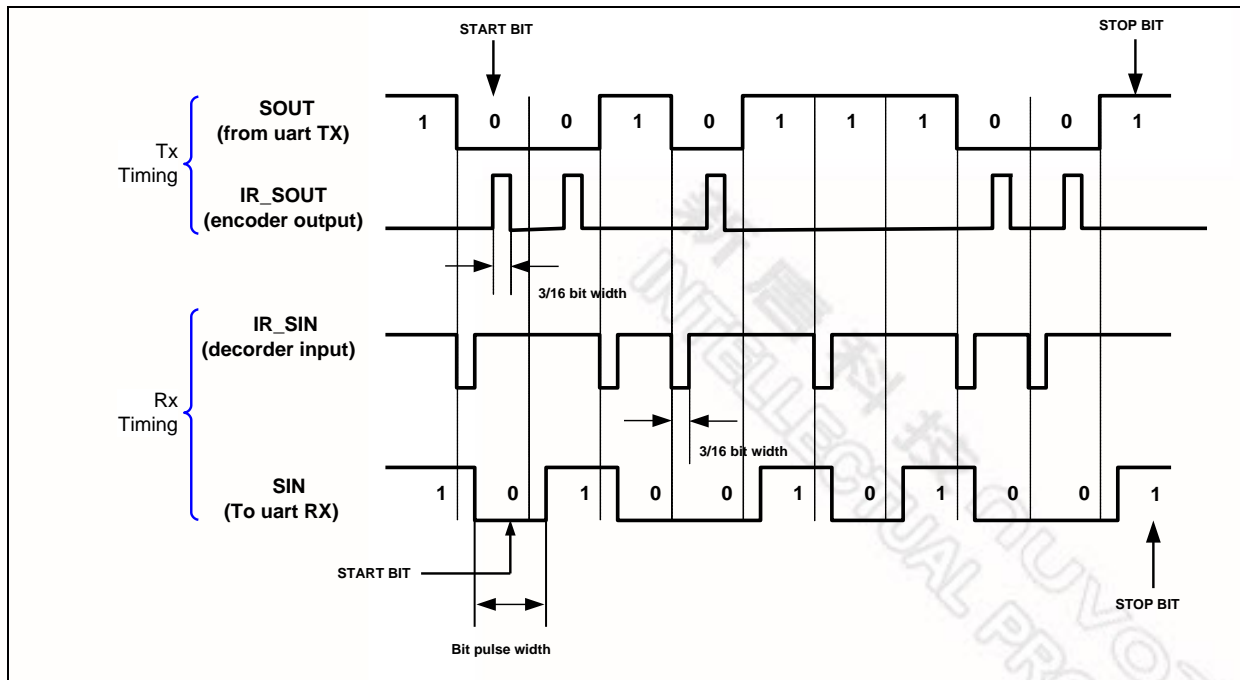
Figure 5-60 IrDA Tx/Rx Timing Diagram

5.12.5 LIN (Local Interconnection Network) mode

The UART supports a Local Interconnection Network (LIN) function. LIN mode is selected by setting the UART0->FUNSEL.LIN_EN bit. In LIN mode, each byte field is initiated by a start bit with value zero (dominant), followed by 8 data bits (LSB is first) and ended by 1 stop bit with value one (recessive) in accordance with the LIN standard (http://www.lin-subbus.org/).
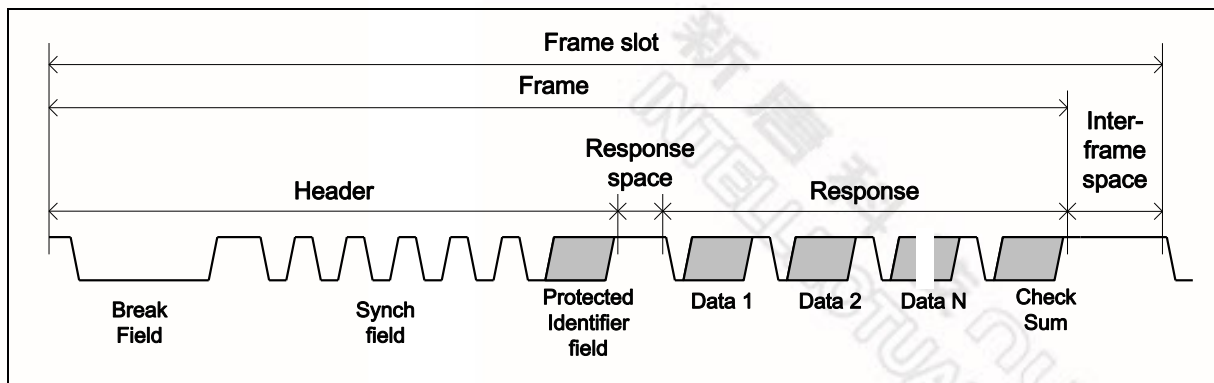
Figure 5-61Structure of LIN Frame

The program flow of LIN Bus Transmit transfer (Tx) is shown as following

1. Set the UART0->FUNSEL.LIN_EN bit to enable LIN Bus mode.

2. SetUART0->LINCON.LINBCNTto choose break field length. The break field length is LINBCNT+2.

3. Fill 0x55 to UART0->DATA to request synch field transmission.

4. Request Identifier Field transmission by writing the protected identifier value toUART0->DATA

5. Set the UART0->LINCON.LINTX_EN bit to start transmission (When break filed operation is finished, LINTX_EN will be cleared automatically).

6. When the STOP bit of the last byte UART0->DATAhas been sent to bus, hardware will set flag UART0->FSR.TE to 1.

7. Fill N bytes data and Checksum to UART0->DATA then repeat step 5 and 6 to transmit the data.

The program flow of LIN Bus Receiver transfer (Rx) is show as following

1. Set the UART0->FUNSEL.LIN_EN bit to enable LIN Bus mode.

2. Set the UART0->LINCON.LINRX_EN bit register to enable LIN Rx mode.

3. Wait for the flagUART0->ISR.LIN_Rx_Break_IF to indicate Rx received Break field or not.

4. Wait for the flagUART0->ISR.RDA_IF read back the UART0->DATA register.

5.12.6 UART Interface Control Register Map

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

Register	Offset	R/W	Description	Reset Value	Reference
UART0_BA= 0x4005_0000					
DATA	UART0_BA + 0x00	R	UART0 Receive FIFO Register.	Undefined	Table 5-100
DATA	UART0_BA + 0x00	W	UART0 Transmit FIFO Register.	Undefined	Table 5-101
IER	UART0_BA + 0x04	R/W	UART0 Interrupt Enable Register.	0x0000_0000	Table 5-102
FCR	UART0_BA + 0x08	R/W	UART0 FIFO Control Register.	0x0000_0000	Table 5-103
LCR	UART0_BA + 0x0C	R/W	UART0 Line Control Register.	0x0000_0000	Table 5-104
MCR	UART0_BA + 0x10	R/W	UART0 Modem Control Register.	0x0000_0000	Table 5-105
MSR	UART0_BA + 0x14	R/W	UART0 Modem Status Register.	0x0000_0000	Table 5-106
FSR	UART0_BA + 0x18	R/W	UART0 FIFO Status Register.	0x1040_4000	Table 5-107
ISR	UART0_BA + 0x1C	R/W	UART0 Interrupt Status Register.	0x0000_0002	Table 5-108
TOR	UART0_BA + 0x20	R/W	UART0 Time Out Register	0x0000_0000	Table 5-111
BAUD	UART0_BA + 0x24	R/W	UART0 Baud Rate Divisor Register	0x0F00_0000	Table 5-112
IRCR	UART0_BA + 0x28	R/W	UART0 IrDA Control Register.	0x0000_0040	Table 5-114
LINCON	UART0_BA + 0x2C	R/W	UART0 LIN Control Register.	0x0000_0000	Table 5-115
FUNSEL	UART0_BA + 0x30	R/W	UART0 Function Select Register.	0x0000_0000	Table 5-116

5.12.7 UART Interface Control Register Description

Receive FIFO Data Register (DATA)

Register	Offset	R/W	Description	Reset Value
DATA	UART0_BA + 0x00	R	UART0 Receive FIFO Register.	Undefined

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

Table 5-100 UART Receive FIFO Data Register (DATA, address 0x4005_0000)

Bits	Descriptions	
[7:0]	**8-bit Received Data**	**Receive FIFO Register** Reading this register will return data from the receive data FIFO. By reading this register, the UART will return the 8-bit data received from Rx pin (LSB first).

Transmit FIFO Data Register (DATA)

Register	Offset	R/W	Description	Reset Value
DATA	UART0_BA + 0x00	W	UART0 Transmit FIFO Data Register.	Undefined

31	30	29	28	27	26	25	24
Reserved								

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
8-bit Transmit Data							

Table 5-101UART Transmit FIFO Data Register (DATA, address 0x4005_0000)

Bits	Descriptions	
[7:0]	**8-bit Transmit Data**	**Transmit FIFO Data Register** By writing to this register, transmit data will be pushed onto the transmit FIFO. The UART will send out an 8-bit data through the Tx pin (LSB first).

Interrupt Enable Register (IER)

Register	Offset	R/W	Description	Reset Value
IER	UART0_BA + 0x04	R/W	UART0 Interrupt Enable Register.	0x0000_0000

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

Table 5-102UART Interrupt Enable Register (IER, address 0x4005_0004)

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15]	**DMA_RX_EN**	**Receive DMA Enable** If enabled, the UART will request DMA service when data is available in receive FIFO.
[14]	**DMA_TX_EN**	**Transmit DMA Enable** If enabled, the UART will request DMA service when space is available in transmit FIFO.
[13]	**AUTO_CTS_EN**	**CTS Auto Flow Control Enable** 1 = Enable,0 = Disable CTS auto flow control. When CTS auto-flow is enabled, the UART will send data to external device when CTS input is asserted (UART will not send data to device until CTS is asserted).
[12]	**AUTO_RTS_EN**	**RTS Auto Flow Control Enable** 1 = Enable, 0 = Disable RTS auto flow control. When RTS auto-flow is enabled, if the number of bytes in the Rx FIFO equals FCR.RTS_TRIG_LEVEL, the UART will de-assert the RTS signal.
[11]	**TOC_EN**	**Time-Out Counter Enable** 1 = Enable, 0 = Disable Time-out counter.
[10:9]	Reserved	Reserved
[8]	**LIN_RX_BRK_IEN**	**LIN RX Break Field Detected Interrupt Enable** 0 = Mask off Lin bus Rx break field interrupt. 1 = Enable Lin bus Rx break field interrupt.

[7:6]	Reserved	Reserved
[5]	**BUF_ERR_IEN**	**Buffer Error Interrupt Enable** 0 = Mask off BUF_ERR_INT 1 = Enable IBUF_ERR_INT
[4]	**RTO_IEN**	**Receive Time out Interrupt Enable** 0 = Mask off TOUT_INT 1 = Enable TOUT_INT
[3]	**MS_IEN**	**Modem Status Interrupt Enable** 0 = Mask off MODEM_INT 1 = Enable MODEM_INT
[2]	**RLS_IEN**	**Receive Line Status Interrupt Enable** 0 = Mask off RLS_INT 1 = Enable RLS_INT
[1]	**THRE_IEN**	**Transmit FIFO Register Empty Interrupt Enable** 0 = Mask off THRE_INT 1 = Enable THRE_INT
[0]	**RDA_IEN**	**Receive Data Available Interrupt Enable**. 0 = Mask off RDA_INT 1 = Enable RDA_INT

FIFO Control Register (FCR)

Register	Offset	R/W	Description	Reset Value
FCR	UART0_BA + 0x08	R/W	UART0 FIFO Control Register.	0x0000_0000

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

Table 5-103UART FIFO Control Register (FCR, address 0x4005_0008)

Bits	Descriptions	
[31:20]	Reserved	Reserved
[19:16]	RTS_TRIG_LEVEL	**RTS Trigger Level for Auto-flow Control** Sets the FIFO trigger level when auto-flow control will de-assert RTS (request-to-send). | RTS_Tri_Lev	Trigger Level (Bytes)
---	---
0000	1
0001	4
0010	8	
[7:4]	RFITL	**Receive FIFO Interrupt (RDA_INT) Trigger Level** When the number of bytes in the receive FIFO equals the RFITL then the RDA_IF will be set and, if enabled, an RDA_INT interrupt will generated. | RFITL	INTR_RDA Trigger Level (Bytes)
---	---
0000	1
0001	4
0010	8	
[3]	Reserved	Reserved

[2]	TFR	**Transmit FIFO Reset** When TFR is set, all the bytes in the transmit FIFO are cleared and transmit internal state machine is reset. 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the transmit internal state machine and pointers. **Note:** This bit will auto-clear after 3 UART engine clock cycles.
[1]	RFR	**Receive FIFO Reset** When RFR is set, all the bytes in the receive FIFO are cleared and receive internal state machine is reset. 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the receive internal state machine and pointers. **Note:** This bit will auto-clear after 3 UART engine clock cycles.
[0]	Reserved	Reserved

Line Control Register (LCR)

Register	Offset	R/W	Description	Reset Value
LCR	UART0_BA + 0x0C	R/W	UART0 Line Control Register.	0x0000_0000

7	6	5	4	3	2	1	0
Reserved	BCB	SPE	EPE	PBE	NSB	WLS	

Table 5-104 UART Line Control Register (LCR, address 0x4005_000C)

Bits	Descriptions	
[31:7]	Reserved	Reserved
[6]	BCB	**Break Control Bit** When this bit is set to logic 1, the serial data output (Tx) is forced to the 'Space' state (logic 0). Normal condition is serial data output is 'Mark' state. This bit acts only on Tx and has no effect on the transmitter logic.
[5]	SPE	**Stick Parity Enable** 0 = Disable stick parity 1 = When bits PBE and SPE are set 'Stick Parity' is enabled. If EPE=0 the parity bit is transmitted and checked as always set, if EPE=1, the parity bit is transmitted and checked as always cleared.
[4]	EPE	**Even Parity Enable** 0 = Odd number of logic 1's are transmitted or checked in the data word and parity bits. 1 = Even number of logic 1's are transmitted or checked in the data word and parity bits. This bit has effect only when PBE (parity bit enable) is set.
[3]	PBE	**Parity Bit Enable** 0 = Parity bit is not generated (transmit data) or checked (receive data) during transfer. 1 = Parity bit is generated or checked between the "last data word bit" and "stop bit" of the serial data.
[2]	NSB	**Number of STOP bits** 0= One "STOP bit" is generated after the transmitted data 1= Two "STOP bits" are generated when 6-, 7- and 8-bit word length is selected; One and a half "STOP bits" are generated in the transmitted data when 5-bit word length is selected;
[1:0]	WLS	**Word Length Select** <table><tr><td>WLS[1:0]</td><td>Character length</td></tr><tr><td>00</td><td>5 bits</td></tr><tr><td>01</td><td>6 bits</td></tr><tr><td>10</td><td>7 bits</td></tr><tr><td>11</td><td>8 bits</td></tr></table>

MODEM Control Register (MCR)

Register	Offset	R/W	Description	Reset Value
MCR	UART0_BA + 0x10	R/W	UART0 Modem Control Register.	0x0000_0000

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

Table 5-105UART Modem Control Register (MCR, address 0x4005_0010)

Bits	Descriptions	
[31:14]	Reserved	Reserved
[13]	RTS_ST	**RTS Pin State(read only)** This bit is the pin status of RTS.
[12:10]	Reserved	Reserved
[9]	RTS_ACT	**Request-to-Send (RTS)Active Trigger Level** This bit can change the RTS trigger level. 0=RTS is active low level. 1=RTS is active high level
[8:5]	Reserved	Reserved
[4]	LBME	Loopback Mode Enable.
[3:2]	Reserved	Reserved
[1]	RTS_SET	**RTS (Request-To-Send) Signal** If IER.AUTO_RTS_EN=0, this bit controls whether RTS pin is active or not. 0: Drive RTS inactive (=~RTS_ACT). 1: Drive RTS active (=RTS_ACT).

Modem Status Register (MSR)

Register	Offset	R/W	Description	Reset Value
MSR	UART0_BA + 0x14	R/W	UART0 Modem Status Register.	0x0000_0000

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

Table 5-106UART Modem Status Register (MSR, address 0x4005_0014)

Bits	Descriptions	
[31:9]	Reserved	Reserved
[8]	**CTS_ACT**	**Clear-to-Send (CTS)Active Trigger Level** This bit can change the CTS trigger level. 0= CTS is active low level. 1= CTS is active high level
[7:5]	Reserved	Reserved
[4]	**CTS_ST**	**CTS Pin Status (read only)** This bit is the pin status of CTS.
[3:1]	Reserved	Reserved
[0]	**DCTSF**	**Detect CTS State Change Flag** This bit is set whenever CTS input has state change. It will generate Modem interrupt to CPU when IER.MS_IEN=1 **NOTE:** This bit is cleared by writing 1 to itself.

FIFO Status Register (FSR)

Register	Offset	R/W	Description	Reset Value
FSR	UART0_BA + 0x18	R/W	UART0 FIFO Status Register.	0x1040_4000

31	30	29	28	27	26	25	24
Reserved			TE	Reserved			TX_OVF_IF
23	22	21	20	19	18	17	16
TX_FULL	TX_EMPTY	TX_POINTER					
15	14	13	12	11	10	9	8
RX_FULL	RX_EMPTY	RX_POINTER					
7	6	5	4	3	2	1	0
Reserved	BIF	FEF	PEF	Reserved			RX_OVF_IF

Table 5-107UART FIFO Status Register (FSR, address 0x4005_0018)

Bits	Descriptions	
[31:29]	Reserved	Reserved
[28]	TE	**Transmitter Empty (Read Only)** Bit is set by hardware when Tx FIFO is empty and the STOP bit of the last byte has been transmitted. Bit is cleared automatically when Tx FIFO is not empty or the last byte transmission has not completed. NOTE: This bit is read only.
[27:25]	Reserved	Reserved
[24]	TX_OVF_IF	**Tx Overflow Error Interrupt Flag** If the Tx FIFO (UART0->DATA) is full, an additional write to UART0->DATA will cause an overflow condition and set this bit to logic 1. It will also generate a BUF_ERR_IF event and interrupt if enabled. NOTE: This bit is cleared by writing 1 to itself.
[23]	TX_FULL	**Transmit FIFO Full(Read Only)** This bit indicates whether the Tx FIFO is full or not. This bit is set when TxFIFO is full; otherwise it is cleared by hardware. TX_FULL=0 indicates there is room to write more data to Tx FIFO.
[22]	TX_EMPTY	**Transmit FIFO Empty(Read Only)** This bit indicates whether the Tx FIFOis empty or not. When the last byte of Tx FIFO has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared after writing data to FIFO (Tx FIFO not empty).

[21:16]	TX_POINTER	**Tx FIFO Pointer (Read Only)** This field returns the Tx FIFO buffer pointer. When CPU writes a byte into the TxFIFO, TX_POINTER is incremented. When a byte from Tx FIFO is transferred to the Transmit Shift Register, TX_POINTER is decremented.
[15]	RX_FULL	**Receive FIFO Full(Read Only)** This bit indicates whether the Rx FIFO is full or not. This bit is set when RxFIFO is full; otherwise it is cleared by hardware.
[14]	RX_EMPTY	**Receive FIFO Empty(Read Only)** This bit indicates whether the Rx FIFO is empty or not. When the last byte of Rx FIFO has been read by CPU, hardware sets this bit high. It will be cleared when UART receives any new data.
[13:8]	RX_POINTER	**Rx FIFO pointer (Read Only)** This field returns the Rx FIFO buffer pointer. It is the number of bytes available for read in the Rx FIFO. When UART receives one byte from external device, RX_POINTER is incremented. When one byte of Rx FIFO is read by CPU, RX_POINTER is decremented.
[7]	Reserved	Reserved
[6]	BIF	**Break Interrupt Flag** This bit is set to a logic 1 whenever the receive data input(Rx) is held in the "space" state (logic 0) for longer than a full word transmission time (that is, the total time of start bit + data bits + parity + stop bits). It is reset whenever the CPU writes 1 to this bit.
[5]	FEF	**Framing Error Flag** This bit is set to logic 1 whenever the received character does not have a valid "stop bit" (that is, the stop bit following the last data bit or parity bit is detected as a logic 0),and is reset whenever the CPU writes 1 to this bit.
[4]	PEF	**Parity Error Flag** This bit is set to logic 1 whenever the received character does not have a valid "parity bit", and is reset whenever the CPU writes 1 to this bit.
[3:1]	Reserved	Reserved
[0]	RX_OVF_IF	**Rx Overflow Error Interrupt Flag** If the Rx FIFO (UART0->DATA) is full, and an additional byte is received by the UART, an overflow condition will occur and set this bit to logic 1. It will also generate a BUF_ERR_IF event and interrupt if enabled. NOTE: This bit is cleared by writing 1 to itself.

Interrupt Status Register (ISR)

Register	Offset	R/W	Description	Reset Value
ISR	UART0_BA + 0x1C	R/W	UART0 Interrupt Status Register.	0x0000_0002

31	30	29	28	27	26	25	24
DMA_LIN_Rx_Break_INT	Reserved	DMA_BUF_ERR_INT	DMA_TOUT_INT	DMA_MODEM_INT	DMA_RLS_INT	Reserved	
23	22	21	20	19	18	17	16
DMA_LIN_Rx_Break_IF	Reserved	DMA_BUF_ERR_IF	DMA_TOUT_IF	DMA_MODEM_IF	DMA_RLS_IF	Reserved	
15	14	13	12	11	10	9	8
LIN_Rx_Break_INT	Reserved	BUF_ERR_INT	TOUT_INT	MODEM_INT	RLS_INT	THRE_INT	RDA_INT
7	6	5	4	3	2	1	0
LIN_Rx_Break_IF	Reserved	BUF_ERR_IF	TOUT_IF	MODEM_IF	RLS_IF	THRE_IF	RDA_IF

Table 5-108 UART Interrupt Status Register (ISR, address 0x4005_001C)

Bits	Descriptions	
[31:16]	DMA mode Bits	DMA mode equivalent following interrupt indicators and flags. See Table 5-109and normal mode descriptions below. In DMA mode (either DMA transmit or receive requests are active) these bits are generated rather than the normal use bits below.
[15]	**LIN_Rx_Break_INT**	**LIN Bus Rx Break Field Detected Interrupt Indicator to Interrupt Controller** Logical AND of IER.LIN_RX_BRK_IEN and LIN_Rx_Break_IF
[14]	Reserved	Reserved
[13]	**BUF_ERR_INT**	**Buffer Error Interrupt Indicator to Interrupt Controller** Logical AND of IER.BUF_ERR_IEN and BUF_ERR_IF
[12]	**TOUT_INT**	**Time Out Interrupt Indicator to Interrupt Controller** Logical AND of IER.RTO_IEN and TOUT_IF
[11]	**MODEM_INT**	**MODEM Status Interrupt Indicator to Interrupt** Logical AND of IER.MS_IEN and MODEM_IF
[10]	**RLS_INT**	**Receive Line Status Interrupt Indicator to Interrupt Controller** Logical AND of IER.RLS_IEN and RLS_IF
[9]	**THRE_INT**	**Transmit Holding Register Empty Interrupt Indicator to Interrupt Controller** Logical AND of IER.THRE_IEN and THRE_IF

[8]	RDA_INT	**Receive Data Available Interrupt Indicator to Interrupt Controller**
		Logical AND of IER.RDA_IEN and RDA_IF
[7]	LIN_Rx_Break_IF	**LIN Bus Rx Break Field Detected Flag**
		This bit is set when LIN controller detects a break field. This bit is cleared by writing a 1.
[6]	Reserved	Reserved
[5]	BUF_ERR_IF	**Buffer Error Interrupt Flag (Read Only)**
		This bit is set when either the Tx or Rx FIFO overflows (FSR.TX_OVF_IF or FSR.RX_OVF_IF is set). When BUF_ERR_IF is set, the serial transfer may be corrupted. If IER.BUF_ERR_IEN is enabled a CPU interrupt request will be generated.
		NOTE: This bit is cleared when both FSR.TX_OVF_IF and FSR.RX_OVF_IF are cleared.
[4]	TOUT_IF	**Time Out Interrupt Flag (Read Only)**
		This bit is set when the Rx FIFO is not empty and no activity occurs in the Rx FIFO and the time out counter equal to TOIC. If IER.TOUT_IEN is enabled a CPU interrupt request will be generated.
		NOTE: This bit is read only and user can read FIFO to clear it.
[3]	MODEM_IF	**MODEM Interrupt Flag (Read Only)**
		This bit is set when the CTS pin has changed state (MSR.DCTSF=1). If IER.MS_IEN is enabled, a CPU interrupt request will be generated.
		NOTE: This bit is read only and reset when bit MSR.DCTSF is cleared by a write 1.
[2]	RLS_IF	**Receive Line Status Interrupt Flag (Read Only).**
		This bit is set when the Rx receive data has a parity, framing or break error (at least one of, FSR.BIF, FSR.FEF and FSR.PEF, is set). If IER.RLS_IEN is enabled, the RLS interrupt will be generated.
		NOTE: This bit is read only and reset to 0 when all bits of BIF, FEF and PEF are cleared.
[1]	THRE_IF	**Transmit Holding Register Empty Interrupt Flag (Read Only).**
		This bit is set when the last data of Tx FIFO is transferred to Transmitter Shift Register. If IER.THRE_IEN is enabled, the THRE interrupt will be generated.
		NOTE: This bit is read only and it will be cleared when writing data into the Tx FIFO.
[0]	RDA_IF	**Receive Data Available Interrupt Flag (Read Only).**
		When the number of bytes in the Rx FIFO equals FCR.RFITL then the RDA_IF will be set. If IER.RDA_IEN is enabled, the RDA interrupt will be generated.
		NOTE: This bit is read only and it will be cleared when the number of unread bytes of Rx FIFO drops below the threshold level (RFITL).

When the DMA controller is used to transmit or receive data to the UART, an alternate set of flags and interrupt indicators are generated. These are equivalent to the normal mode set above and are summarized in Table 5-109.

Table 5-109 UART Interrupt Sources and Flags Table In DMA Mode

UART Interrupt Source	Interrupt Enable Bit	Interrupt Indicator to Interrupt Controller	Interrupt Flag	Flag Cleared by
LIN RX Break Field Detected interrupt	LIN_RX_BRK_IEN	DMA_LIN_Rx_Break_INT	**DMA_LIN_Rx_Break_IF**	Write '1' to LIN_Rx_Break_IF
Buffer Error Interrupt BUF_ERR_INT	BUF_ERR_IEN	DMA_BUF_ERR_INT	**DMA_BUF_ERR_IF** = (TX_OVF_IF or RX_OVF_IF)	Write '1' to TX_OVF_IF/ RX_OVF_IF
Rx Timeout Interrupt TOUT_INT	RTO_IEN	DMA_TOUT_INT	**DMA_TOUT_IF**	Read data FIFO
Modem Status Interrupt MODEM_INT	MS_IEN	DMA_MODEM_INT	**DMA_MODEM_IF** = (DCTSF)	Write '1' to DCTSF
Receive Line Status Interrupt RLS_INT	RLS_IEN	DMA_RLS_INT	**DMA_RLS_IF** = (BIF or FEF or PEF)	Write '1' to BIF/FEF/PEF
Transmit Holding Register Empty Interrupt THRE_INT	THRE_IEN	DMA_THRE_INT	**DMA_THRE_IF**	Write data FIFO
Receive Data Available Interrupt RDA_INT	RDA_IEN	DMA_RDA_INT	**DMA_RDA_IF**	Read data FIFO

Table 5-110 UART Interrupt Sources and Flags Table In Software Mode

UART Interrupt Source	Interrupt Enable Bit	Interrupt Indicator to Interrupt Controller	Interrupt Flag	Flag Cleared by
LIN RX Break Field Detected interrupt	LIN_RX_BRK_IEN	LIN_Rx_Break_INT	**LIN_Rx_Break_IF**	Write '1' to LIN_Rx_Break_IF
Buffer Error Interrupt BUF_ERR_INT	BUF_ERR_IEN	BUF_ERR_INT	**BUF_ERR_IF** = (TX_OVF_IF or RX_OVF_IF)	Write '1' to TX_OVF_IF/ RX_OVF_IF
Rx Timeout Interrupt TOUT_INT	RTO_IEN	TOUT_INT	**TOUT_IF**	Read data FIFO
Modem Status Interrupt MODEM_INT	MS_IEN	MODEM_INT	**MODEM_IF** = (DCTSF)	Write '1' to DCTSF
Receive Line Status Interrupt RLS_INT	RLS_IEN	RLS_INT	**RLS_IF** = (BIF or FEF or PEF)	Write '1' to BIF/FEF/PEF
Transmit Holding Register Empty Interrupt THRE_INT	THRE_IEN	THRE_INT	**THRE_IF**	Write data FIFO
Receive Data Available Interrupt RDA_INT	RDA_IEN	RDA_INT	**RDA_IF**	Read data FIFO

Time Out Register (TOR)

Register	Offset	R/W	Description	Reset Value
TOR	UART0_BA + 0x20	R/W	UART0 Time Out Register	0x0000_0000

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

Table 5-111UART Time Out Register (TOR, address 0x4005_0020)

Bits	Descriptions	
[31:7]	Reserved	Reserved
[6:0]	**TOIC**	**Time Out Interrupt Comparator** The time out counter resets and starts counting whenever the Rx FIFO receives a new data word. Once the content of time out counter (TOUT_CNT) is equal to that of time out interrupt comparator (TOIC), a receiver time out interrupt (TOUT_INT) is generated if IER.RTO_IEN is set. A new incoming data word or RX FIFO empty clears TOUT_IF. The period of the time out counter is the baud rate.

Baud Rate Divider Register (BAUD)

Register	Offset	R/W	Description	Reset Value
BAUD	UART0_BA + 0x24	R/W	UART0 Baud Rate Divisor Register	0x0F00_0000

The baud rate generator takes the UART master clock UART_CLK and divides it to produce the baud rate (bit rate) clock. The divider has two division stages controlled by BRD and DIVX fields. These are configured in three modes depending on the selections of DIVX_EN and DIVX_ONE. These modes and the baud rate equations for them are described in Table 5-113.

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

Table 5-112 UART Baud Rate Divider Register (BAUD, address 0x4005_0024)

Bits	Descriptions	
[31:30]	Reserved	Reserved
[29]	DIVX_EN	**Divider X Enable** The baud rate equation is: **Baud Rate = UART_CLK / [M * (BRD + 2)]** ; The default value of M is 16. 0 = Disable divider X (M = 16) 1 = Enable divider X (M = DIVX+1, with DIVX ≥8). Refer to Table 5-113for more information. **NOTE**: When in IrDA mode, this bit must disabled.
[28]	DIVX_ONE	**Divider X equal 1** 0:M = DIVX+1, with restriction DIVX ≥8. 1: M = 1, with restriction BRD[15:0]≥ 3. Refer toTable 5-113for more information.
[27:24]	DIVX	**Divider X** The baud rate divider M = DIVX+1.
[23:16]	Reserved	Reserved
[15:0]	BRD	**Baud Rate Divider.** Refer to Table 5-113for more information.

Table 5-113Baud Rate Equations.

Mode	DIVX_EN	DIVX_ONE	DIVX[3:0]	BRD[15:0]	Baud rate equation
0	0	0	B	A	UART_CLK / [16 * (A+2)]
1	1	0	B	A	UART_CLK / [(B+1) * (A+2)] , requires B≥ 8
2	1	1	Don't care	A	UART_CLK / (A+2), requires A ≥ 3

IrDA Control Register (IRCR)

Register	Offset	R/W	Description	Reset Value
IRCR	UART0_BA + 0x28	R/W	UART0 IrDA Control Register.	0x0000_0040

7	6	5	4	3	2	1	0
Reserved	RX_INV_EN	TX_INV_EN	Reserved		LOOPBACK	TX_SELECT	Reserved

Table 5-114UART IrDA Control Register (IRCR, address 0x4005_0028)

Bits	Descriptions	
[31:7]	Reserved	Reserved
[6]	**RX_INV_EN**	**Receive Inversion Enable** 1= Invert Rx input signal 0= No inversion
[5]	**TX_INV_EN**	**Transmit inversion enable** 1= Invert Tx output signal 0= No inversion
[4:3]	Reserved	Reserved
[2]	**LOOPBACK**	IrDA Loopback test mode. Loopback Tx to Rx.
[1]	**TX_SELECT**	**Transmit/Receive Selection** 1: Enable IrDA transmitter 0: Enable IrDA receiver
[0]	Reserved	Reserved

UART LIN Network Control Register (LINCON)

Register	Offset	R/W	Description	Reset Value
LINCON	UART0_BA + 0x2C	R/W	UART0 LIN Network Control/Status Register.	0x0000_0000

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

Table 5-115UART LIN Network Control Register (LINCON, address 0x4005_002C)

Bits	Descriptions	
[31:8]	Reserved	Reserved
[7]	LIN_TX_EN	**LIN TX Break Mode Enable** 1 = Enable LIN Tx Break Mode. 0 = Disable LIN Tx Break Mode. NOTE: When Tx break field transfer operation finished, this bit will be cleared automatically.
[6]	LIN_RX_EN	**LIN RX Enable** 1 = Enable LIN Rx mode. 0 = Disable LIN Rx mode.
[3:0]	LINBCNT	**UART LIN Break Field Length Count** This field indicates a 4-bit LIN Tx break field count. NOTE: This break field length is LINBCNT + 2

UART Function Select Register (FUNSEL)

Register	Offset	R/W	Description	Reset Value
FUNSEL	UART0_BA + 0x30	R/W	UART0 Function Select Register.	0x0000_0000

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

Table 5-116UART Function Select Register (FUNSEL, address 0x4005_0030)

Bits	Descriptions	
[31:2]	Reserved	Reserved
[1]	**IrDA_EN**	**Enable IrDA Function.** 0 = UART Function. 1 = Enable IrDA Function.
[0]	**LIN_EN**	**Enable LIN Function.** 0 = UART Function. 1 = Enable LIN Function. Note that IrDA and LIN functions are mutually exclusive: both cannot be active at same time.

5.13 I2S Audio PCM Controller

5.13.1 Overview

The I2Scontrolleris a peripheral for serial transmission and reception of audio PCM (Pulse-Code Modulated) signals across a 4-wire bus. The bus consists of a bit clock (I2S_BCLK) a frame synchronization clock (I2S_FS) and serial data in (I2S_SDI) and out (I2S_SDO) lines. This peripheral allows communication with an external audio CODEC or DSP. The peripheral is capable of mono or stereo audio transmission with 8-32bit word sizes. Audio data is buffered in 8 word deep FIFO buffers and has DMA capability.

5.13.2 Features

- I2S can operate as either master or slave

- Master clock generation for slave device synchronization.

- Capable of handling 8, 16, 24 and 32 bit word sizes.

- Mono and stereo audio data supported.

- I2S and MSB justified data format supported.

- 8 word FIFO data buffers for transmit and receive.

- Generates interrupt requests when buffer levels crosses programmable boundary.

- Two DMA requests, one for transmit and one for receive.

5.13.3 I2S Block Diagram
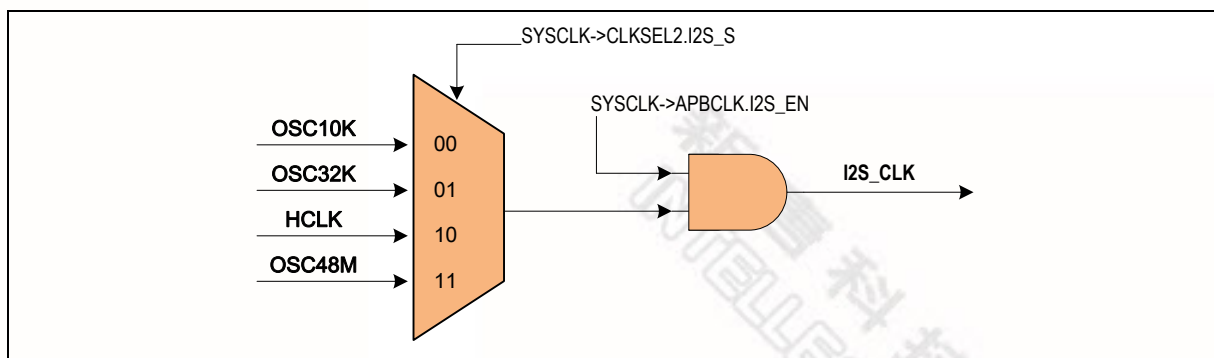
Figure 5-62 I2S Clock Control Diagram
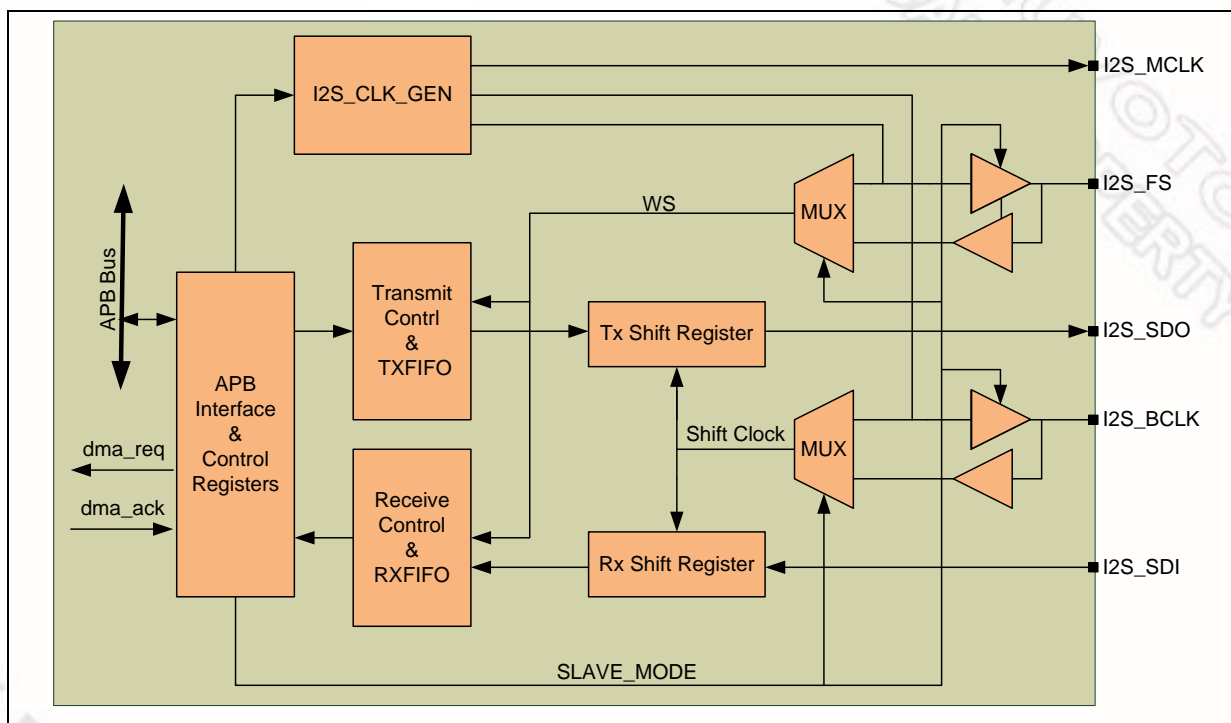
Figure 5-63 I2S Controller Block Diagram

5.13.4 I2S Operation
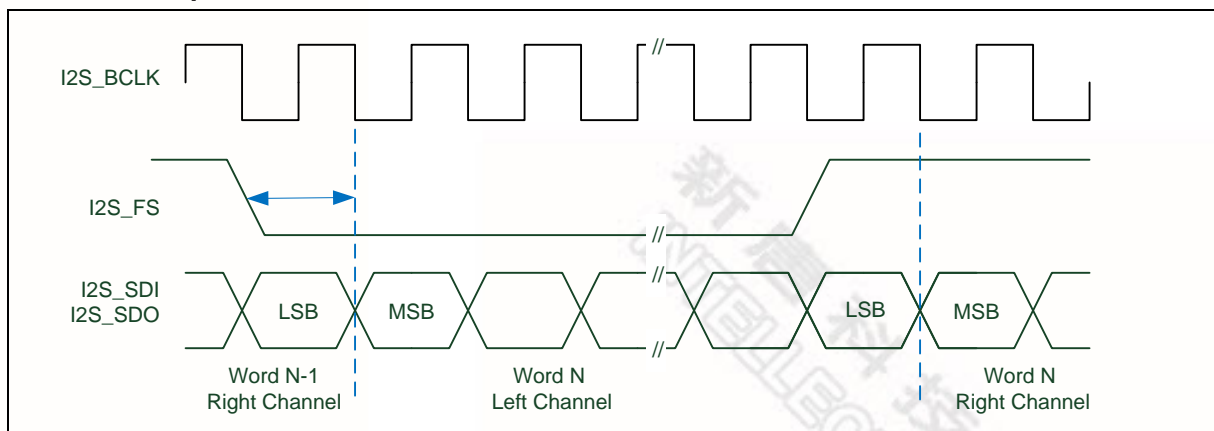
Figure 5-64I2S Bus Timing Diagram (Format =0)
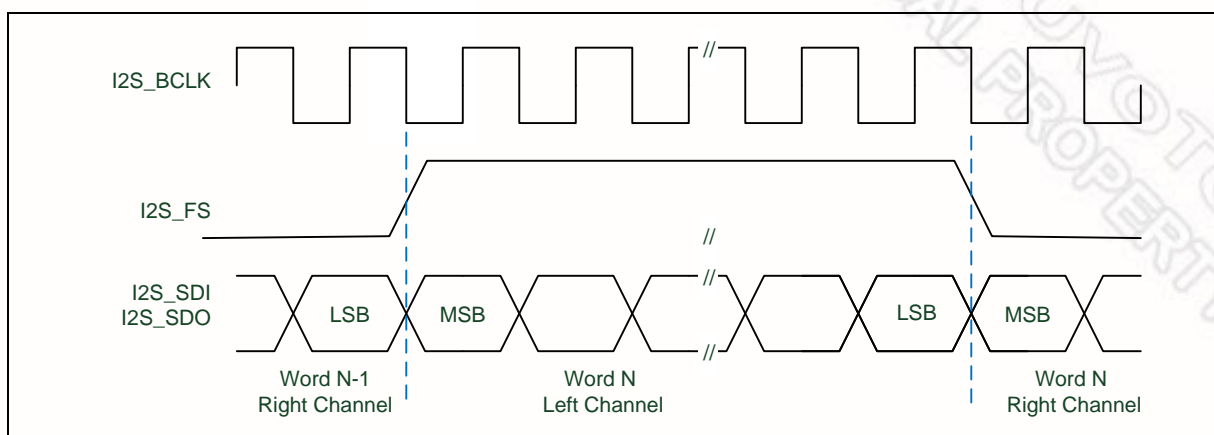
Figure 5-65MSB Justified Timing Diagram (Format=1)

5.13.5 FIFO operation
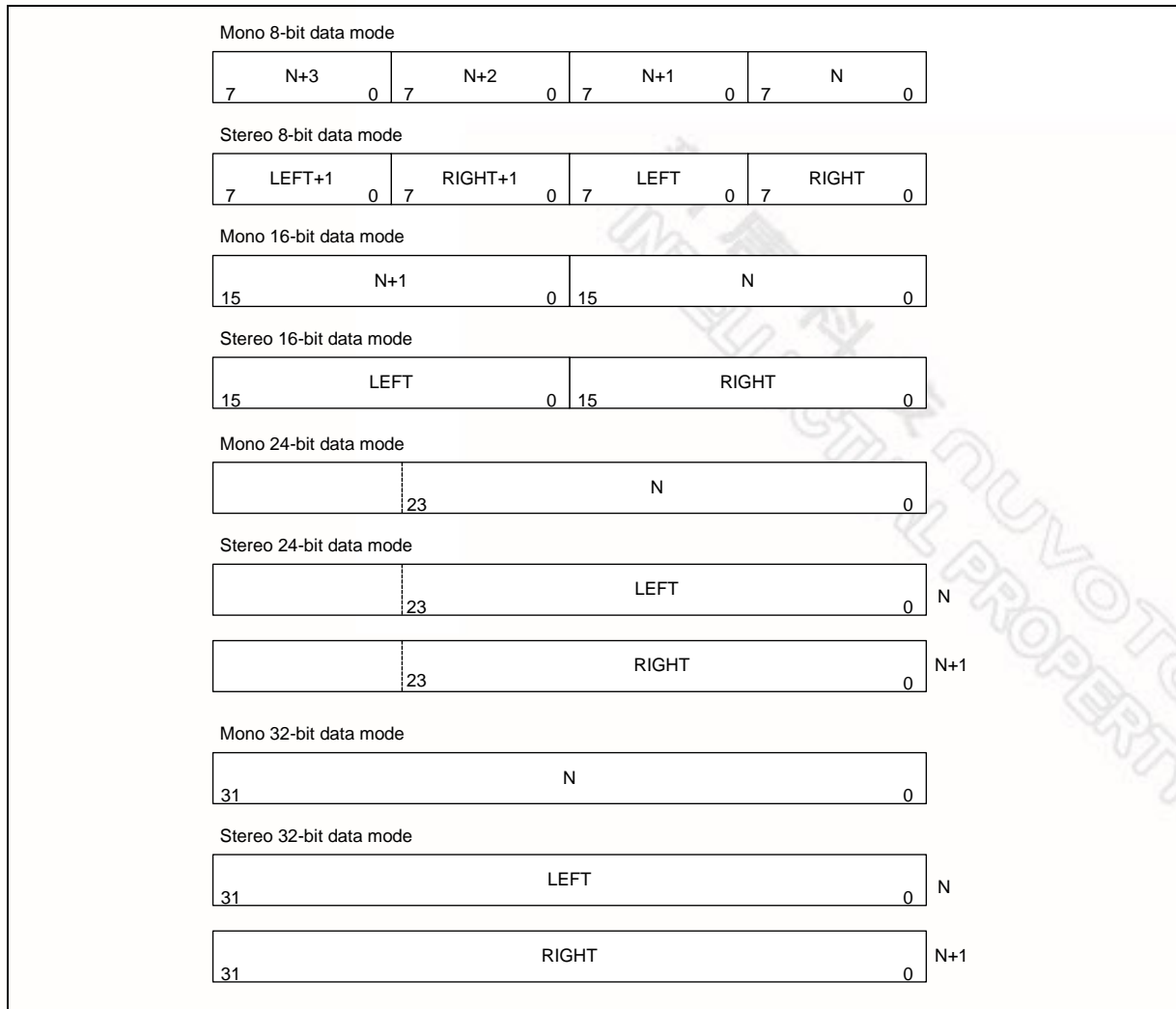
Figure 5-66FIFO contents for various I2S modes

5.13.6 I2S Control Register Map

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

Register	Offset	R/W	Description	Reset Value	Reference
I2S_BA = 0x400A_0000					
ICON	I2S_BA+0x00	R/W	I2S Control Register	0x0000_0000	Table 5-117
CLKDIV	I2S_BA+0x04	R/W	I2S Clock Divider Register	0x0000_0000	Table 5-118
IE	I2S_BA+0x08	R/W	I2S Interrupt Enable Register	0x0000_0000	Table 5-119
STATUS	I2S_BA+0x0C	R/W	I2S Status Register	0x0014_1000	Table 5-120
TXFIFO	I2S_BA+0x10	W	I2S Transmit FIFO Register	0x0000_0000	Table 5-121
RXFIFO	I2S_BA+0x14	R	I2S Receive FIFO Register	0x0000_0000	Table 5-122

5.13.7 I2S Control Register Description

I2S Control Register (CON)

Register	Offset	R/W	Description	Reset Value
CON	I2S_BA + 0x00	R/W	I2S Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	Reserved	RXDMA	TXDMA	CLR_RXFIFO	CLR_TXFIFO	LCHZCEN	RCHZCEN
15	14	13	12	11	10	9	8
MCLKEN	RXTH[2:0]			TXTH[2:0]			SLAVE
7	6	5	4	3	2	1	0
FORMAT	MONO	WORDWIDTH[1:0]		MUTE	RXEN	TXEN	I2SEN

Table 5-117 I2S Control Register (CON, address 0x400A_0000)

Bits	Descriptions	
[31:22]	Reserved	Reserved
[21]	**RXDMA**	**Enable Receive DMA** When RX DMA is enabled, I2S requests DMA to transfer data from receive FIFO to SRAM if FIFO is not empty. 1 = Enable RX DMA 0 = Disable RX DMA
[20]	**TXDMA**	**Enable Transmit DMA** When TX DMA is enables, I2S request DMA to transfer data from SRAM to transmit FIFO if FIFO is not full. 1 = Enable TX DMA 0 = Disable TX DMA
[19]	**CLR_RXFIFO**	**Clear Receive FIFO** Write 1 to clear receive FIFO, internal pointer is reset to FIFO start point, and RXFIFO_LEVEL[3:0] returns to zero and receive FIFO becomes empty. This bit is cleared by hardware automatically when clear operation complete.
[18]	**CLR_TXFIFO**	**Clear Transmit FIFO** Write 1 to clear transmit FIFO, internal pointer is reset to FIFO start point, and TXFIFO_LEVEL[3:0] returns to zero and transmit FIFO becomes empty. Data in transmit FIFO is not changed. This bit is cleared by hardware automatically when clear operation complete.

[17]	LCHZCEN	**Left channel zero cross detect enable** If this bit is set to 1, when left channel data sign bit changes, or data bits are all zero, the LZCF flag in I2S_STATUS register will be set to 1. 1 = Enable left channel zero cross detect 0 = Disable left channel zero cross detect
[16]	RCHZCEN	**Right channel zero cross detect enable** If this bit is set to 1, when right channel data sign bit changes, or data bits are all zero, the RZCF flag in I2S_STATUS register will be set to 1. 1 = Enable right channel zero cross detect 0 = Disable right channel zero cross detect
[15]	MCLKEN	**Master clock enable** The ISD93xx can generate a master clock signal to an external audio CODEC to synchronize the audio devices. If audio devices are not synchronous, then data will be periodically corrupted. Software needs to implement a way to drop/repeat or interpolate samples in a jitter buffer if devices are not synchronized. The master clock frequency is determined by the CLKDIV.MCLK_DIV[2:0] register. 1 = Enable master clock 0 = Disable master clock
[14:12]	RXTH[2:0]	**Receive FIFO threshold level** When received data word(s) in buffer is equal or higher than threshold level then RXTHI flag is set. Threshold = RXTH+1 words of data in receive FIFO.
[11:9]	TXTH[2:0]	**Transmit FIFO threshold level** If remaining data words in transmit FIFO less than or equal to the threshold level then TXTHI flag is set. Threshold = TXTH words remaining in transmit FIFO
[8]	SLAVE	**Slave mode** I2S can operate as a master or slave. For master mode, I2S_BCLK and I2S_FS pins are outputs and send bit clock and frame sync from ISD93xx. In slave mode, I2S_BCLK and I2S_FS pins are inputs and bit clock and frame sync are received from external audio device. 1 = Slave mode 0 = Master mode
[7]	FORMAT	**Data format** 1 = MSB justified data format 0 = I2S data format See Figure 5-64 and Figure 5-65 for timing differences.
[6]	MONO	**Monaural data** This parameter sets whether mono or stereo data is processed. See Figure 5-66 for details of how data is formatted in transmit and receive FIFO. 1 = Data is monaural format 0 = Data is stereo format

[5:4]	WORDWIDTH[1:0]	**Word width** This parameter sets the word width of audio data. See Figure 5-66 for details of how data is formatted in transmit and receive FIFO. 00 = data is8 bit 01 = data is 16 bit 10 = data is 24 bit 11 = data is 32 bit
[3]	MUTE	**Transmit mute enable** 1= Transmit channel zero 0 = Transmit data is shifted from FIFO
[2]	RXEN	**Receive enable** 1 = Enable data receive 0 = Disable data receive
[1]	TXEN	**Transmit enable** 1 = Enable data transmit 0 = Disable data transmit
[0]	I2SEN	**Enable I2S controller** 1 = Enable 0 = Disable

I2S Clock Divider (CLKDIV)

Register	Offset	R/W	Description	Reset Value
CLKDIV	I2S_BA + 0x04	R/W	I2S Clock Divider Control Register	0x0000_0000

15	14	13	12	11	10	9	8
			BCLK_DIV [7:0]				
7	6	5	4	3	2	1	0
	Reserved				MCLK_DIV[2:0]		

Table 5-118 I2S Clock Divider Register (CLKDIV, address 0x400A_0004)

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:8]	**BCLK_DIV [7:0]**	**Bit Clock Divider** If I2S operates in master mode, bit clock is provided by ISD93xx. Software can program these bits to generate bit clock frequency for the desired sample rate. For sample rate F_s, the desired bit clock frequency is: $$F_{BCLK} = F_s \times \text{Word\_width\_in\_bytes} \times 16$$ For example if Fs=16kHz, and word width is 2-bytes (16bit) then desired bit clock frequency is 512kHz. The bit clock frequency is given by: $$F_{BCLK} = \frac{F_{I2S\_CLK}}{2 \times (\text{BCLK\_DIV} + 1)}$$ Or, $$\text{BCLK\_DIV} = \frac{F_{I2S\_CLK}}{2 \times F_{BCLK}} - 1$$ So if $F_{I2S\_CLK} = HCLK = 49.152\text{MHz}$, desired $F_{BCLK} = 512\text{kHz}$ then $\text{BCLK\_DIV} = 47$
[7:3]	Reserved	Reserved
[2:0]	**MCLK_DIV[2:0]**	**Master Clock Divider** ISD93xx can generate a master clock to synchronously drive an external audio device. If MCLK_DIV is set to 0, MCLK is the same as I2S_CLK clock input, otherwise MCLK frequency is given by: $$F_{MCLK} = \frac{F_{I2S\_CLK}}{2 \times \text{MCLK\_DIV}}$$ Or, $$\text{MCLK\_DIV} = \frac{F_{I2S\_CLK}}{2 \times F_{MCLK}}$$ If the desired MCLK frequency is $256F_s$ and $F_s = 16\text{kHz}$ then $\text{MCLK\_DIV} = 6$

I2S Interrupt Enable Register (IE)

Register	Offset	R/W	Description	Reset Value
IE	I2S_BA + 0x08	R/W	I2S Interrupt Enable Register	0x0000_0000

15	14	13	12	11	10	9	8
Reserved			LZCIE	RZCIE	TXTHIE	TXOVFIE	TXUDFIE
7	6	5	4	3	2	1	0
Reserved					RXTHIE	RXOVFIE	RXUDFIE

Table 5-119 I2S Interrupt Enable Register (IE, address 0x400A_0008)

Bits		Descriptions
[12]	LZCIE	**Left channel zero cross interrupt enable** Interrupt will occur if this bit is set to 1 and left channel has zero cross event 1 = Enable interrupt, 0 = Disable interrupt
[11]	RZCIE	**Right channel zero cross interrupt enable** Interrupt will occur if this bit is set to 1 and right channel has zero cross event 1 = Enable interrupt, 0 = Disable interrupt
[10]	TXTHIE	**Transmit FIFO threshold level interrupt enable** Interrupt occurs if this bit is set to 1 and data words in transmit FIFO is less than TXTH[2:0]. 1 = Enable interrupt, 0 = Disable interrupt
[9]	TXOVFIE	**Transmit FIFO overflow interrupt enable** Interrupt occurs if this bit is set to 1 and transmit FIFO overflow flag is set to 1 1 = Enable interrupt, 0 = Disable interrupt
[8]	TXUDFIE	**Transmit FIFO underflow interrupt enable** Interrupt occur if this bit is set to 1 and transmit FIFO underflow flag is set to 1. 1 = Enable interrupt , 0 = Disable interrupt
[2]	RXTHIE	**Receive FIFO threshold level interrupt** Interrupt occurs if this bit is set to 1 and data words in receive FIFO is greater than or equal to RXTH[2:0]. 1 = Enable interrupt, 0 = Disable interrupt
[1]	RXOVFIE	**Receive FIFO overflow interrupt enable** 1 = Enable interrupt, 0 = Disable interrupt
[0]	RXUDFIE	**Receive FIFO underflow interrupt enable** If software read receive FIFO when it is empty then RXUDF flag in I2SSTATUS register is set to 1. 1 = Enable interrupt, 0 = Disable interrupt

I2S Status Register (STATUS)

Register	Offset	R/W	Description	Reset Value
STATUS	I2S_BA + 0x0C	R/W	I2S Status Register	0x0014_1000

31	30	29	28	27	26	25	24
TX_LEVEL[3:0]				RX_LEVEL[3:0]			
23	22	21	20	19	18	17	16
LZCF	RZCF	TXBUSY	TXEMPTY	TXFULL	TXTHF	TXOVF	TXUDF
15	14	13	12	11	10	9	8
Reserved			RXEMPTY	RXFULL	RXTHF	RXOVF	RXUDF
7	6	5	4	3	2	1	0
Reserved				RIGHT	I2STXINT	I2SRXINT	I2SINT

Table 5-120I2S Status Register (STATUS, address 0x400A_000C)

Bits	Descriptions	
[31:28]	TX_LEVEL	**Transmit FIFO level (Read Only)** TX_LEVEL= number of words in transmit FIFO.
[27:24]	RX_LEVEL	**Receive FIFO level (Read Only)** RX_LEVEL = number of words in receive FIFO.
[23]	LZCF	**Left channel zero cross flag (write '1' to clear, or clear LCHZCEN)** 1 = Left channel zero cross is detected 0 = No zero cross detected.
[22]	RZCF	**Right channel zero cross flag (write '1' to clear, or clear RCHZCEN)** 1 = Right channel zero cross is detected 0 = No zero cross
[21]	TXBUSY	**Transmit Busy (Read Only)** This bit is cleared when all data in transmit FIFO and Tx shift register is shifted out. It is set when first data is loaded to Tx shift register. 1 = Transmit shift register is busy 0 = Transmit shift register is empty
[20]	TXEMPTY	**Transmit FIFO Empty (Read Only)** This is set when transmit FIFO is empty. 1 = Empty 0 = Not empty

[19]	**TXFULL**	**Transmit FIFO Full (Read Only)** This bit is set when transmit FIFO is full. 1 = Full. 0 = Not full.
[18]	**TXTHF**	**Transmit FIFO Threshold Flag (Read Only)** When data word(s) in transmit FIFO is less than or equal to the threshold value set in TXTH[2:0] the TXTHF bit becomes to 1. It remains set until transmit FIFO level is greater than TXTH[2:0]. Cleared by writing to TXFIFO register until threshold exceeded. 1 = Data word(s) in FIFO is less than or equal to threshold level 0 = Data word(s) in FIFO is greater than threshold level
[17]	**TXOVF**	**Transmit FIFO Overflow Flag (Write '1' to clear)** This flag is set if data is written to transmit FIFO when it is full. 1 = Overflow 0 = No overflow
[16]	**TXUDF**	**Transmit FIFO underflow flag (Write '1' to clear)** This flag is set if I2S controller requests data when transmit FIFO is empty. 1 = Underflow 0 = No underflow
[15:13]	Reserved	Reserved
[12]	**RXEMPTY**	**Receive FIFO empty (Read Only)** This is set when receive FIFO is empty. 1 = Empty 0 = Not empty
[11]	**RXFULL**	**Receive FIFO full (Read Only)** This bit is set when receive FIFO is full. 1 = Full. 0 = Not full.
[10]	**RXTHF**	**Receive FIFO Threshold Flag (Read Only)** When data word(s) in receive FIFO is greater than or equal to threshold value set in RXTH[2:0] the RXTHF bit becomes to 1. It remains set until receive FIFO level is less than RXTH[2:0]. It is cleared by reading RXFIFO until threshold satisfied. 1 = Data word(s) in FIFO is greater than or equal to threshold level 0 = Data word(s) in FIFO is less than threshold level
[9]	**RXOVF**	**Receive FIFO Overflow Flag (Write '1' to clear)** This flag is set if I2S controller writes to receive FIFO when it is full. Audio data is lost. 1 = Overflow 0 = No overflow

[8]	RXUDF	**Receive FIFO Underflow Flag (Write '1' to clear)** This flag is set if attempt is made to read receive FIFO while it is empty. 1 = Underflow 0 = No underflow
[7:4]	Reserved	Reserved
[3]	RIGHT	**Right Channel Active (Read Only)** This bit indicates current data being transmitted/received belongs to right channel 1 = Right channel 0 = Left channel
[2]	I2STXINT	**I2S Transmit Interrupt (Read Only)** This indicates that there is an active transmit interrupt source. This could be TXOVF, TXUDF, TXTHF, LZCF or RZCF if corresponding interrupt enable bits are active. To clear interrupt the corresponding source(s) must be cleared. 1 = Transmit interrupt occurred. 0 = No transmit interrupt
[1]	I2SRXINT	**I2S Receive Interrupt (Read Only)** This indicates that there is an active receive interrupt source. This could be RXOVF, RXUDF or RXTHF if corresponding interrupt enable bits are active. To clear interrupt the corresponding source(s) must be cleared. 1 = Receive interrupt occurred 0 = No receive interrupt
[0]	I2SINT	**I2S Interrupt (Read Only)** This bit is set if any enabled I2S interrupt is active. 1 = I2S interrupt active 0 = No I2S interrupt

I2S Transmit FIFO (TXFIFO)

Register	Offset	R/W	Description	Reset Value
TXFIFO	I2S_BA + 0x10	W	I2S Transmit FIFO	0x0000_0000

Table 5-121I2S Transmit FIFO Register (TXFIFO, address 0x400A_0010)

Bits	Descriptions	
[31:0]	**TXFIFO**	**Transmit FIFO Register (Write Only)** A write to this register pushes data onto the transmit FIFO. The transmit FIFO is eight words deep. The number of words currently in the FIFO can be determined by reading I2S->STATUS.TX_LEVEL.

I2S Receive FIFO (I2S_RXFIFO)

Register	Offset	R/W	Description	Reset Value
I2S_RXFIFO	I2S_BA + 0x14	R	I2S Receive FIFO	0x0000_0000

Table 5-122I2S Receive FIFO Register (RXFIFO, address 0x400A_0014)

Bits	Descriptions	
[31:0]	**RXFIFO**	**Receive FIFO Register (Read Only)** A read of this register will pop data from the receive FIFO. The receive FIFO is eight words deep. The number of words currently in the FIFO can be determined by reading I2S->STATUS.RX_LEVEL.

5.14 PDMA Controller

5.14.1 Overview

The ISD93xx series incorporates a Peripheral Direct Memory Access (PDMA) controller that transfers data between SRAM and APB devices. The PDMA has four channels of DMA PDMA CH0~CH3). PDMA transfers are unidirectional and can be Peripheral-to-SRAM, SRAM-to-Peripheral or SRAM-to-SRAM.

The peripherals available for PDMA transfer are SPI, UART, I2S, ADC and DPWM.

PDMA operation is controlled for each channel by configuring a source and destination address and specifying a number of bytes to transfer. Source and destination addresses can be fixed, automatically increment by the transfer size, update by an arbitrary value (span mode) or wrap around a circular buffer. When PDMA operation is complete, controller can be configured to provide CPU with an interrupt.

The DMA controller contains a cyclic redundancy check (CRC) generator that can perform CRC calculation with programmable polynomial settings.

5.14.2 Features

- Provides access to SPI, UART, I2S, ADC and DPWM peripherals.

- AMBA AHB master/slave interface, transfers can occur concurrently with CPU access to flash memory.

- PDMA source and destination addressing modes allow fixed, incrementing, wrap-around and spanned addressing.

- Cyclic Redundancy Check (CRC)

  - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
    - CRC-CCITT: $X16 + X12 + X5 + 1$
    - CRC-8: $X8 + X2 + X + 1$
    - CRC-16: $X16 + X15 + X2 + 1$
    - CRC-32: $X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4 + X2 + X + 1$

  - Programmable seed value

  - Supports programmable order reverse setting for input data and CRC checksum

  - Supports programmable 1's complement setting for input data and CRC checksum.

  - Supports CPU PIO mode or DMA transfer mode

  - Supports 8/16/32-bit of data width in CPU PIO mode
    - 8-bit write mode: 1-AHB clock cycle operation
    - 16-bit write mode: 2-AHB clock cycle operation
    - 32-bit write mode: 4-AHB clock cycle operation

  - Supports byte alignment transfer length in CRC DMA mode
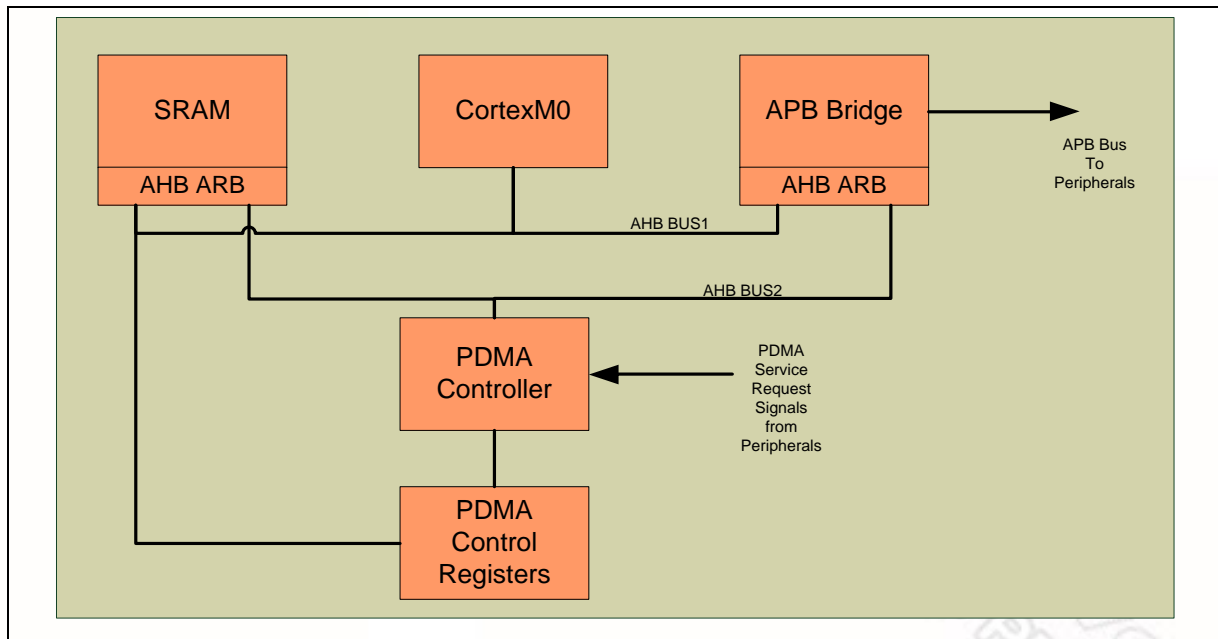
5.14.3 Block Diagram

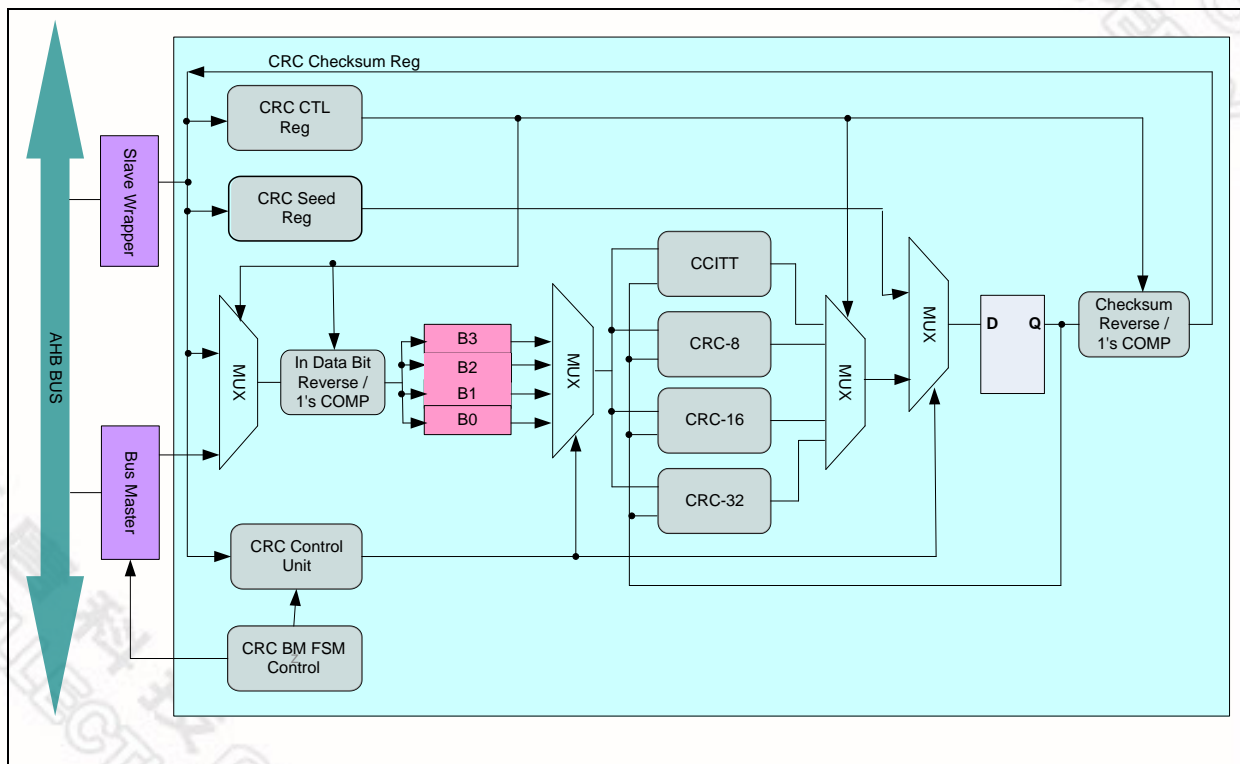Figure 5-67 PDMA Controller Block Diagram

Figure 5-68CRCGeneratorBlock Diagram

5.14.4 Function Description

The PDMA controller has four channels of DMA, each channel can be configured to one of the following transfer types: Peripheral-to-SRAM SRAM-to-Peripheral or SRAM-to-SRAM. The SRAM and the AHB-APB bus bridge each have an AHB bus arbiter that allows AHB bus access to occur either from the CPU or the PDMA controller. The PDMA controller requests bus transfers over the AHB bus from one address into a single word buffer within the PDMA controller then writes this buffer to another address over the AHB bus. Peripherals with PDMA capability generate control signals to the PDMA block requesting service when they need data (Rx request) or have data to transfer (Tx request). The PDMA control registers reside in address space on the AHB bus.

Transfer completion can be determined by polling of status registers or by generation of PDMA interrupt to CPU. A transfer is set up as a specified number of bytes from a source address to a destination address. Both source and destination address can be configured as a fixed address, an incrementing address or a wrap-around buffer address.

The general procedure to operate a DMA channel is as follows:

- Enable PDMA channel *n* clock by setting PDMA->HCLK*n*_EN
- Enable PDMA channel *n* by setting PDMA->channel[*n*].CSR.PDMACEN
- Set source address in PDMA->channel[*n*].SAR
- Set destination address in PDMA->channel[*n*].DAR
- Set the transfer count in PDMA->channel[*n*].BCR
- Set transfer mode and address increment mode in PDMA->channel[*n*].CSR
- Route peripheral PDMA request signal to channel *n* in service selection register.
- Trigger transfer PDMA->channel[*n*].CSR.TRIG_EN

If the source or destination address is not in wraparound mode, the PDMA will continue the transfer until PDMA->channel[*n*].CBCR decrements to zero (CBCR is initialized to BCR, in wraparound mode, CBCR will reload and continue until PDMACEN is disabled). If an error occurs during the PDMA operation, the channel stops until software clears the error condition and sets the PDMA->channel[*n*].CSR.SW_RST bit to reset the PDMA channel. After reset the PDMACEN and TRIG_EN bits would need to be set to start a new operation.

The DMA controller contains a cyclic redundancy check (CRC) generator that can perform CRC calculations with programmable polynomial settings. Available operation polynomials are CRC-CCITT, CRC-8, CRC-16 and CRC-32; Software can choose the operation polynomial mode by setting CRC_MODE fields in CRC_CTL register.

The CRC engine support CPU PIO mode (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 0) and DMA transfer mode (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 1).

Procedure when operating in CPU PIO mode:

- Enable CRC engine by setting CRCCEN bit in CRC_CTL register.
- Initial Setting. Set the data format (WDATA_RVS, CHECKSUM_RVS, WDATA_COM and CHECKSUM_COM by setting CRC_CTL register), initial seed value (CRC_SEED) and select the data length by setting CRC_CTL [CPU_WDLEN] register.
- Set CRC_RST to load the initial seed value to CRC circuit.
- Write data to CRC_WDATA to perform CRC calculation.
- Get the CRC checksum result by reading CRC_CHECKSUM register.

Procedure when operation in CRC DMA mode:

- Enable CRC engine by setting CRCCEN bit in CRC_CTL register.
- Initial Setting. Set the data format (WDATA_RVS, CHECKSUM_RVS, WDATA_COM and CHECKSUM_COM by setting CRC_CTL register), initial seed value (CRC_SEED).
- Give a valid source address and transfer count by setting CRC_DMASAR and CRC_DMABCR.
- Enable CRC_CTL [TRIG_EN] and then hardware will reset the seed value and then read

memory data to perform CRC calculation.
- Wait CRC DMA transfer and CRC calculation done and then retrieve the CRC checksum result by reading CRC_CHECKSUM register.

5.14.5 PDMA Controller Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Offset	R/W	Description	Reset Value	Reference
PDMA_BA = 0x5000_8000					
PDMA_BA_ch0 = 0x5000_8000 **PDMA_BA_ch1** = 0x5000_8100 **PDMA_BA_ch2** = 0x5000_8200					
PDMA_BA_ch3 = 0x5000_8300					
CSR	PDMA_BA_ch*x* + 0x00	R/W	PDMA Control Register	0x0000_0000	Table 5-123
SAR	PDMA_BA_chx + 0x04	R/W	PDMA Source Address Register	0x0000_0000	Table 5-124
DAR	PDMA_BA_chx + 0x08	R/W	PDMA Destination Address Register	0x0000_0000	Table 5-125
BCR	PDMA_BA_chx + 0x0C	R/W	PDMA Transfer Byte Count Register	0x0000_0000	Table 5-126
POINT	PDMA_BA_chx + 0x10	R	PDMA Internal buffer pointer	0xXXXX_0000	Table 5-127
CSAR	PDMA_BA_chx + 0x14	R	PDMA Current Source Address Register	0x0000_0000	Table 5-128
CDAR	PDMA_BA_chx + 0x18	R	PDMA Current Destination Address Register	0x0000_0000	Table 5-129
CBCR	PDMA_BA_chx + 0x1C	R	PDMA Current Transfer Byte Count Register	0x0000_0000	Table 5-130
IER	PDMA_BA_chx + 0x20	R/W	PDMA Interrupt Enable Register	0x0000_0001	Table 5-133
ISR	PDMA_BA_chx + 0x24	R/W	PDMA Interrupt Status Register	0x0000_0000	Table 5-134
SPIR	PDMA_BA_chx + 0x34	R/W	Span Increment Register	0x0000_0000	Table 5-131
CSPIR	PDMA_BA_chx + 0x38	R	Current Span Increment Register	0x0000_0000	Table 5-132
CRC_BA= 0x5000_8E00					
CRC_CTL	CRC_BA+0x00	R/W	CRC Control Register	0x2000_0000	265
CRC_DMASAR	CRC_BA+0x04	R/W	CRC DMA Transfer Source Address Register	0x0000_0000	268
CRC_DMABCR	CRC_BA+0x0C	R/W	CRC DMA Transfer Byte Count Register	0x0000_0000	269
CRC_DMACSAR	CRC_BA+0x14	R	CRC DMA Current Source Address Register	0x0000_0000	270
CRC_DMACBCR	CRC_BA+0x1C	R	CRC DMA Current Transfer Byte Count Register	0x0000_0000	271
CRC_DMAIER	CRC_BA+0x20	R/W	CRC DMA Interrupt Enable Control Register	0x0000_0001	272
CRC_DMAISR	CRC_BA+0x24	R/W	CRC DMA Interrupt Status Register	0x0000_0000	273
CRC_WDATA	CRC_BA+0x80	R/W	CRC Write Data Register	0x0000_0000	273
CRC_SEED	CRC_BA+0x84	R/W	CRC Seed Register	0xFFFF_FFFF	275
CRC_CHECKSUM	CRC_BA+0x88	R	CRC Checksum Register	0x0000_0000	276
CRC_CTL	CRC_BA+0x00	R/W	CRC Control Register	0x2000_0000	265
PDMA_GCR_BA= 0x5000_8F00					
GCR	PDMA_GCR_BA + 0x00	R/W	PDMA Global Control Register	0x0000_0000	Table 5-135

PDSSR0	PDMA_GCR_BA+ 0x04	R/W	PDMA Service Selection Control Register	0XFFFF_FFFF	Table 5-136
GCRISR	PDMA_GCR_BA+ 0x0C	R/W	PDMA Global Interrupt Register	0x0000_0000	Table 5-137

5.14.6 PDMA Control Register Description

PDMA Control and Status Register (CSR)

Register	Offset	R/W	Description	Reset Value
CSR	PDMA_BA+*n**0x100 + 0x000	R/W	PDMA Control and Status Register Channel *n*	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
TRIG_EN	Reserved		APB_TWS		Reserved		
15	14	13	12	11	10	9	8
WRA_INT_SEL				Reserved			
7	6	5	4	3	2	1	0
DAD_SEL		SAD_SEL		MODE_SEL		SW_RST	PDMACEN

Table 5-123 PDMA Control and Status Register (CSR, address 0x5000_8000 + *n* * 0x100)

Bits	Descriptions	
[23]	**TRIG_EN**	**Trigger Enable – Start a PDMA operation.** 0 = Write: no effect. Read: Idle/Finished. 1 = Enable PDMA data read or write transfer. **Note:** When PDMA transfer completed, this bit will be cleared automatically. If a bus error occurs, all PDMA transfer will be stopped. Software must reset PDMA channel, and then trigger again.
[20:19]	**APB_TWS**	**Peripheral Transfer Width Select.** This parameter determines the data width to be transferred each PDMA transfer operation. 00 = One word (32 bits) is transferred for every PDMA operation. 01 = One byte (8 bits) is transferred for every PDMA operation. 10 = One half-word (16 bits) is transferred for every PDMA operation. 11 = Reserved. **Note:** This field is meaningful only when **MODE_SEL** is IP to Memory mode (APB-to-Memory) or Memory to IP mode (Memory-to-APB).

[15:12]	WRA_INT_SEL	**Wrap Interrupt Select** x1xx: If this bit is set, and wraparound mode is in operation a Wrap Interrupt can be generated when half each PDMA transfer is complete. For example if BCR=32 then an interrupt could be generated when 16 bytes were sent. xxx1: If this bit is set, and wraparound mode is in operation a Wrap Interrupt can be generated when each PDMA transfer is wrapped. For example if BCR=32 then an interrupt could be generated when 32 bytes were sent and PDMA wraps around. x1x1: Both half and w interrupts generated.
[7:6]	DAD_SEL	**Destination Address Select** This parameter determines the behavior of the current destination address register with each PDMA transfer. It can either be fixed, incremented or wrapped. 00 = Transfer Destination Address is incremented. 01 = Transfer Destination Address is Spanned. 10 = Transfer Destination Address is fixed (Used when data transferred from multiple addresses to a single destination such as peripheral FIFO input). 11 = Transfer Destination Address is wrapped. For Span mode: CDAR is incremented by Span Increment Register (SPIR) each transfer. For Wrap mode: When CBCR (Current Byte Count) equals zero, the CDAR (Current Destination Address) and CBCR registers will be reloaded from the DAR (Destination Address) and BCR (Byte Count) registers automatically and PDMA will start another transfer. Cycle continues until software sets PDMA_EN=0. When PDMA_EN is disabled, the PDMA will complete the active transfer but the remaining data in the SBUF will not be transferred to the destination address.
[5:4]	SAD_SEL	**Source Address Select** This parameter determines the behavior of the current source address register with each PDMA transfer. It can either be fixed, incremented or wrapped. 00 = Transfer Source address is incremented. 01 = Transfer Source address is Spanned. 10 = Transfer Source address is fixed 11 = Transfer Source address is wrapped. For Span mode: CSAR is incremented by Span Increment Register (SPIR) each transfer. For Wrap mode: When CBCR (Current Byte Count) equals zero, the CSAR (Current Source Address) and CBCR registers will be reloaded from the SAR (Source Address) and BCR (Byte Count) registers automatically and PDMA will start another transfer. Cycle continues until software sets PDMA_EN=0. When PDMA_EN is disabled, the PDMA will complete the active transfer but the remaining data in the SBUF will not be transferred to the destination address.
[3:2]	MODE_SEL	**PDMA Mode Select** This parameter selects to transfer direction of the PDMA channel. Possible values are: 00 = Memory to Memory mode (SRAM-to-SRAM). 01 = IP to Memory mode (APB-to-SRAM). 10 = Memory to IP mode (SRAM-to-APB).

[1]	**SW_RST**	**Software Engine Reset** 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal state machine and pointers. The contents of the control register will not be cleared. This bit will auto clear after a few clock cycles.
[0]	**PDMACEN**	**PDMA Channel Enable** Setting this bit to 1 enables PDMA's operation. If this bit is cleared, PDMA will ignore all PDMA request and force Bus Master into IDLE state. **Note:** SW_RST will clear this bit.

PDMA Transfer Source Address Register (SAR)

Register	Offset	R/W	Description	Reset Value
SAR	PDMA_BA+ *n**0x100 + 0x004	R/W	PDMA Transfer Source Address Register Channel *n*	0x0000_0000

Table 5-124PDMA Source Address Register (SAR, address 0x5000_8004 + *n**0x100)

Bits	Descriptions	
[31:0]	**SAR**	**PDMA Transfer Source Address Register** This register holds the initial Source Address of PDMA transfer. **Note:** The source address must be word aligned.

PDMA Transfer Destination Address Register (DAR)

Register	Offset	R/W	Description	Reset Value
DAR	PDMA_BA+ *n**0x100 + 0x008	R/W	PDMA Transfer Destination Address Register Channel *n*	0x0000_0000

Table 5-125PDMA Destination Address Register (DAR, address 0x5000_8008 + *n**0x100)

Bits	Descriptions	
[31:0]	**DAR**	**PDMA Transfer Destination Address Register** This register holds the initial Destination Address of PDMA transfer. **Note:** The destination address must be word aligned.

PDMA Transfer Byte Count Register (BCR)

Register	Offset	R/W	Description	Reset Value
BCR	PDMA_BA+ *n**0x100 + 0x00C	R/W	PDMA Transfer Byte Count Register Channel *n*	0x0000_0000

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

Table 5-126PDMA Transfer Byte Count Register (BCR, address 0x5000_800C + *n**0x100)

Bits	Descriptions	
[31:24]	Reserved	Reserved
[15:0]	**BCR**	**PDMA Transfer Byte Count Register** This register controls the transfer byte count of PDMA. Maximum value is 0xFFFF. **Note:** When in memory-to-memory (CSR.MODE_SEL = 00b) mode, the transfer byte count must be word aligned, that is multiples of 4bytes.

PDMA Internal Buffer Pointer Register (POINT)

Register	Offset	R/W	Description	Reset Value
POINT	PDMA_BA+ *n**0x100 + 0x010	R	PDMA Internal Buffer Pointer Register Channel *n*	0xXXXX_XX00

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

Table 5-127PDMA Internal Buffer Point Register (POINT, address 0x5000_8010 + *n**0x100)

Bits	Descriptions	
[31:4]	Reserved	Reserved
[3:0]	**POINT**	**PDMA Internal Buffer Pointer Register (Read Only)** A PDMA transaction consists of two stages, a read from the source address and a write to the destination address. Internally this data is buffered in a 32bit register. If transaction width between the read and write transactions are different, this register tracks which byte/half-word of the internal buffer is being processed by the current transaction.

PDMA Current Source Address Register (CSAR)

Register	Offset	R/W	Description	Reset Value
CSAR	PDMA_BA+ n*0x100 + 0x014	R	PDMA Current Source Address Register Channel n	0x0000_0000

Table 5-128PDMA Current Source Address Register (CSAR, address 0x5000_8014 + n*0x100)

Bits	Descriptions	
[31:0]	CSAR	**PDMA Current Source Address Register (Read Only)** This register returns the source address from which the PDMA transfer is occurring. This register is loaded from SAR when PDMA is triggered or when a wraparound occurs.

PDMA Current Destination Address Register (CDAR)

Register	Offset	R/W	Description	Reset Value
CDAR	PDMA_BA+ n*0x100 + 0x018	R	PDMA Current Destination Address Register Channel n	0x0000_0000

Table 5-129PDMA Current Destination Address Register (CDAR, address 0x5000_8018 + n*0x100)

Bits	Descriptions	
[31:0]	CDAR	**PDMA Current Destination Address Register (Read Only)** This register returns the destination address to which the PDMA transfer is occurring. This register is loaded from DAR when PDMA is triggered or when a wraparound occurs.

PDMA Current Byte Count Register (CBCR)

Register	Offset	R/W	Description	Reset Value
CBCR	PDMA_BA+ n*0x100 + 0x01C	R	PDMA Current Byte Count Register Channel n	0x0000_0000

Table 5-130PDMA Current Byte Count Register (CBCR, address 0x5000_801C + n*0x100)

Bits	Descriptions	
[31:16]	Reserved	Reserved
[15:0]	CBCR	**PDMA Current Byte Count Register (Read Only)** This field indicates the current remaining byte count of PDMA transfer. This register is initialized with BCR register when PDMA is triggered or when a wraparound occurs

PDMA Span Increment Register (SPIR)

Register	Offset	R/W	Description	Reset Value
SPIR	PDMA_BA+ *n**0x100 + 0x034	R	PDMA Span Increment Register	0x0x0x_0000

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

Table 5-131 PDMA Span Increment Register (ISR, address 0x5000_8034 + *n**0x100)

Bits	Descriptions	
[8:0]	**SPIR**	**Span Increment Register.** This is a signed number in range [-128,127] for use in spanned address mode. If destination or source addressing mode is set as spanned, then this number is added to the address register each transfer. The size of the transfer is determined by the APB_TW setting. Note that span increment must be a multiple of the transfer width otherwise a memory addressing HardFault will occur. Also SPIR may be a negative number.

PDMA Current Span Increment Register (CSPIR)

Register	Offset	R/W	Description	Reset Value
CSPIR	PDMA_BA+ *n**0x100 + 0x038	R/W	PDMA Current Span Increment Register	0x0x0x_0000

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

Table 5-132 PDMA Current Span Increment Register (ISR, address 0x5000_8038 + *n**0x100)

Bits	Descriptions	
[15:0]	**CSPIR**	**Current Span Increment Register.** This is a signed read only register for use in spanned address mode. It provides the current address offset from SAR or DAR if either is set to span mode.

PDMA Interrupt Enable Control Register (IER)

Register	Offset	R/W	Description	Reset Value
IER	PDMA_BA+ *n**0x100 + 0x020	R/W	PDMA Interrupt Enable Control Register Channel *n*	0x0000_0001

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

Table 5-133 PDMA Interrupt Enable Control Register(IER, address 0x5000_8020 + *n**0x100)

Bits	Descriptions	
[31:3]	Reserved	Reserved
[2]	**WAR_IE**	**Wraparound Interrupt Enable** If enabled, and channel source or destination address is in wraparound mode, the PDMA controller will generate a WRAP interrupt to the CPU according to the setting of CSR.WRA_INT_SEL. This can be interrupts when the transaction has finished and has wrapped around and/or when the transaction is half way in progress. This allows the efficient implementation of circular buffers for DMA. 0 = Disable Wraparound PDMA interrupt generation. 1 = Enable Wraparound interrupt generation.
[1]	**BLKD_IE**	**PDMA Transfer Done Interrupt Enable** If enabled, the PDMA controller will generate and interrupt to the CPU when the requested PDMA transfer is complete. 0 = Disable PDMA transfer done interrupt generation. 1 = Enable PDMA transfer done interrupt generation.

[0]	**TABORT_IE**	**PDMA Read/Write Target Abort Interrupt Enable** If enabled, the PDMA controller will generate and interrupt to the CPU whenever a PDMA transaction is aborted due to an error. If a transfer is aborted, PDMA channel must be reset to resume DMA operation. 0 = DisablePDMA transfer target abort interrupt generation. 1 = Enable PDMA transfer target abort interrupt generation.

PDMA Interrupt Status Register (ISR)

Register	Offset	R/W	Description	Reset Value
ISR	PDMA_BA+ *n**0x100 + 0x024	R/W	PDMA Interrupt Status Register Channel *n*	0x0x0x_0000

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

Table 5-134PDMA Interrupt Enable Status Register (ISR, address 0x5000_8024 + *n**0x100)

Bits	Descriptions	
[31]	**INTR**	**Interrupt Pin Status (Read Only)** This bit is the Interrupt pin status of PDMA channel.
[30:12]	Reserved	Reserved
[11:8]	**WAR_IF**	**Wrap around transfer byte count interrupt flag.** These flags are set whenever the conditions for a wraparound interrupt (complete or half complete) are met. They are cleared by writing one to the bits. 0001 = Current transfer finished flag (CBCR==0). 0100 = Current transfer half complete flag (CBCR==BCR/2).

[1]	**BLKD_IF**	**Block Transfer Done Interrupt Flag** This bit indicates that PDMA block transfer complete interrupt has been generated. It is cleared by writing 1 to the bit. 0 = Transfer ongoing or Idle. 1 = Transfer Complete.
[0]	**TABORT_IF**	**PDMA Read/Write Target Abort Interrupt Flag** This flag indicates a Target Abort interrupt condition has occurred. This condition can happen if attempt is made to read/write from invalid or non-existent memory space. It occurs when PDMA controller receives a bus error from AHB master. Upon occurrence PDMA will stop transfer and go to idle state. To resume, software must reset PDMA channel and initiate transfer again. 0 = No bus ERROR response received. 1 = Bus ERROR response received. **NOTE:** This bit is cleared by writing 1 to itself.

CRC Control Register (CRC_CTL)

Register	Offset	R/W	Description	Reset Value
CRC_CTL	CRC_BA+0x00	R/W	CRC Control Register	0x2000_0000

31	30	29	28	27	26	25	24
CRC_MODE		CPU_WDLEN		CHECKSUM_COM	WDATA_COM	CHECKSUM_RVS	WDATA_RVS
23	22	21	20	19	18	17	16
TRIG_EN	Reserved						
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						CRC_RST	CRCCEN

Bits	Description	
[31:30]	**CRC_MODE**	**CRC Polynomial Mode** 00 = CRC-CCITT Polynomial mode 01 = CRC-8 Polynomial mode 10 = CRC-16 Polynomial mode 11 = CRC-32 Polynomial mode

[29:28]	CPU_WDLEN	**CPU Write Data Length** When operation in CPU PIO mode (CRCCEN= 1, TRIG_EN= 0), this field indicates the write data length. 00 = Data length is 8-bit mode 01 = Data length is 16-bit mode 1x = Data length is 32-bit mode **Note1:** This field is used for CPU PIO mode. **Note2:** When the data length is 8-bit mode, the valid data is CRC_WDATA [7:0]; if the data length is 16-bit mode, the valid data is CRC_WDATA [15:0].
[27]	CHECKSUM_COM	**Checksum Complement** 1 = 1's complement for CRC checksum. 0 = No 1's complement for CRC checksum.
[26]	WDATA_COM	**Write Data Complement** 1 = 1's complement for CRC write data in. 0 = No 1's complement for CRC write data in.
[25]	CHECKSUM_RVS	**Checksum Reverse** 1 = Bit order reverse for CRC checksum. 0 = No bit order reverse for CRC checksum. **Note:** If the checksum data is 0XDD7B0F2E, the bit order reversed for CRC checksum is 0x74F0DEBB.
[24]	WDATA_RVS	**Write Data Order Reverse** 1 = Bit order reversed for CRC write data in (per byte) 0 = No bit order reversed for CRC write data in. **Note:** If the write data is 0xAABBCCDD, the bit order reverse for CRC write data in is 0x55DD33BB
[23]	TRIG_EN	**TRIG_EN** 1 = CRC DMA data read or write transfer Enabled. 0 = No effect. **Note1:** If this bit assert indicates the CRC engine operation in CRC DMA mode, do not fill in any data in CRC_WDATA register. **Note2:** When CRC DMA transfer is completed, this bit will be cleared automatically. **Note3:** If the bus error occurs, all CRC DMA transfer will be stopped. Software must reset all DMA channel, and then trigger again.
[22:2]	Reserved	Reserved
[1]	CRC_RST	**CRC Engine Reset** 0 = No effect. 1 = Reset the internal CRC state machine and internal buffer. The contents of control register will not be cleared. This bit will automatically be cleared after few clock cycles. **Note:** When operated in CPU PIO mode, setting this bit will reload the initial seed value.

[0]	CRCCEN	**CRC Channel Enable**
		Setting this bit to 1 enables CRC's operation.
		When operation in CRC DMA mode (TRIG_EN = 1), if user clears this bit, the DMA operation will be continuous until all CRC DMA operation is done, and the TRIG_EN bit will asserted until all CRC DMA operation done. But in this case, the CRC_DMAISR [BLKD_IF] flag will inactive, user can read CRC result by reading CRC_CHECKSUM register when TRIG_EN = 0
		When operation in CRC DMA mode (TRIG_EN = 1), if user wants to stop the transfer immediately, user can write 1 to CRC_RST bit to stop the transmission.

CRCDMA Transfer Source Address Register (CRC_DMASAR)

Register	Offset	R/W	Description	Reset Value
CRC_DMASAR	CRC_BA+0x04	R/W	CRC DMA Source Address Register	0x0000_0000

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

Bits		Description
[31:0]	**CRC_DMASAR**	**CRC DMA Transfer Source Address Register** This field indicates a 32-bit source address of CRC DMA. **Note:** The source address must be word alignment.

CRCDMA Transfer Byte Count Register (CRC_DMABCR)

Register	Offset	R/W	Description	Reset Value
CRC_DMABCR	CRC_BA+0x0C	R/W	CRC DMA Transfer Byte Count Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved
[15:0]	**CRC_DMABCR**	**CRC DMA Transfer Byte Count Register** This field indicates a 16-bit transfer byte count number of CRC DMA.

CRC DMA Current Source Address Register (CRC_DMACSAR)

Register	Offset	R/W	Description	Reset Value
CRC_DMACSAR	CRC_BA+0x14	R	CRC DMA Current Source Address Register	0x0000_0000

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

Bits	Description	
[31:0]	**CRC_DMACSAR**	**CRC DMA Current Source Address Register (Read Only)** This field indicates the source address where the CRC DMA transfer just occurs.

CRC DMA Current Transfer Byte Count Register(CRC_DMACBCR)

Register	Offset	R/W	Description	Reset Value
CRC_DMACBCR	CRC_BA+0x1C	R	CRC DMA Current Transfer Byte Count Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved
[15:0]	**CRC_DMACBCR**	**CRC DMA Current Byte Count Register (Read Only)** This field indicates the current remained byte count of CRC_DMA. **Note:** CRC_RST will clear this register value.

CRC DMA Interrupt Enable Control Register (CRC_DMAIER)

Register	Offset	R/W	Description	Reset Value
CRC_DMAIER	CRC_BA+0x20	R/W	CRC DMA Interrupt Enable Register	0x0000_0001

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

Bits	Description	
[31:2]	**Reserved**	Reserved
[1]	**BLKD_IE**	**CRC DMA Transfer Done Interrupt Enable** 1 = Interrupt generator Enabled when CRC DMA transfer is done. 0 =Interrupt generator Disabled when CRC DMA transfer is done.
[0]	**TABORT_IE**	**CRC DMA Read/Write Target Abort Interrupt Enable** 1 = Target abort interrupt generation Enabled during CRC DMA transfer. 0 = Target abort interrupt generation Disabled during CRC DMA transfer.

CRC DMA Interrupt Status Register (CRC_DMAISR)

Register	Offset	R/W	Description	Reset Value
CRC_DMAISR	CRC_BA+0x24	R/W	CRC DMA Interrupt Status Register	0x0000_0000

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

Bits	Description	
[31:2]	**Reserved**	Reserved
[1]	**BLKD_IF**	**Block Transfer Done Interrupt Flag** This bit indicates that CRC DMA has finished all transfer. 1 = Done 0 = Not finished. Software can write 1 to clear this bit to zero.
[0]	**TABORT_IF**	**CRC DMA Read/Write Target Abort Interrupt Flag** 1 = Bus ERROR response received. 0 = No bus ERROR response received. Software can write 1 to clear this bit to zero.

Note: CRC_DMAISR [TABORT_IF] indicate if bus master received ERROR response or not. If bus master received ERROR response, it means that target abort is happened. DMA will stop transfer and respond this event to software then go to IDLE state. When target abort occurred, software must reset DMA, and then transfer those data again.

CRC Write Data Register(CRC_WDATA)

Register	Offset	R/W	Description	Reset Value

CRC_WDATA	CRC_BA+0x80	R/W	CRC Write Data Register	0x0000_0000

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

Bits		Description
[31:0]	CRC_WDATA	**CRC Write Data Register** When operated in CPU PIO (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 0) mode, software can write data to this field to perform CRC operation; When operated in CRC DMA mode (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 0), this field will be used for DMA internal buffer. **Note1:** When operated in CRC DMA mode, so don't filled any data in this field. **Note2:** The CRC_CTL [WDATA_COM] and CRC_CTL [WDATA_RVS] bit setting will affect this field; for example, if WDATA_RVS = 1, if the write data in CRC_WDATA register is 0xAABBCCDD, the read data from CRC_WDATA register will be 0x55DD33BB.

CRC Seed Register(CRC_SEED)

Register	Offset	R/W	Description	Reset Value
CRC_SEED	CRC_BA+0x84	R/W	CRC Seed Register	0xFFFF_FFFF

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

Bits		Description
[31:0]	**CRC_SEED**	**CRC Seed Register** This field indicates the CRC seed value.

CRC Checksum Register(CRC_CHECKSUM)

Register	Offset	R/W	Description	Reset Value
CRC_CHECKSUM	CRC_BA+0x88	R	CRC Checksum Register	0x0000_0000

31	30	29	28	27	26	25	24
CRC_CHECKSUM [31:24]								

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

Bits	Description	
[31:0]	**CRC_CHECKSUM**	**CRC Checksum Register** This field indicates the CRC checksum.

PDMA Global Control Register (GCR)

Register	Offset	R/W	Description	Reset Value
GCR	PDMA_BA_GCR + 0x000	R/W	PDMA Global Control Register	0x0000_0000

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

Table 5-135PDMA Global Control Register (GCR, address 0x5000_8F00)

Bits	Descriptions	
[31:17]	Reserved	Reserved

[11:8]	HCLK_EN[3:0]	**PDMA Controller Channel Clock Enable Control** To enable clock for channel n HCLK_EN[n] must be set. HCLK_EN[n]=1: Enable Channel n clock HCLK_EN[n]=0: Disable Channel n clock
[7:1]	Reserved	Reserved
[0]	PDMA_RST	**PDMA Software Reset** 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal state machine and pointers. The contents of control register will not be cleared. This bit will auto clear after several clock cycles. **Note:** This bit can reset all channels (global reset).

PDMA Service Selection Control Register (PDSSR)

Register	Offset	R/W	Description	Reset Value
PDSSR	PDMA_BA_GCR +0x04	R/W	PDMA Service Selection Control Register	0xFFFF_FFFF

PDMA peripherals have transmit and/or receive request signals to control dataflow during PDMA transfers. These signals must be connected to the PDMA channel assigned by software for use with that peripheral. For instance if PDMA Channel 3 is to be used to transfer data from memory to DPWM peripheral, then DPWM_TXSEL should be set to 3. This will route the DPWM transmit request signal to PDMA channel 3, whenever DPWM has space in FIFO it will request transmission of data from PDMA. When not used the selection should be set to 0xFF.

31	30	29	28	27	26	25	24
I2S_TXSEL				I2S_RXSEL			

31	30	29	28	27	26	25	24
I2S_TXSEL				I2S_RXSEL			
23	22	21	20	19	18	17	16
UART0_TXSEL				UART0_RXSEL			
15	14	13	12	11	10	9	8
DPWM_TXSEL				ADC_RXSEL			
7	6	5	4	3	2	1	0
SPI0_TXSEL				SPI0_RXSEL			

Table 5-136PDMA Service Selection Control Register (PDSSR, address 0x5000_8F04)

Bits	Descriptions	
[31:28]	I2S_TXSEL	**PDMA I2S Transmit Selection** This field defines which PDMA channel is connected to I2S peripheral transmit (PDMA destination) request.
[27:24]	I2S_RXSEL	**PDMA I2S Receive Selection** This field defines which PDMA channel is connected to I2S peripheral receive (PDMA source) request.
[23:20]	UART0_TXSEL	**PDMA UART0 Transmit Selection** This field defines which PDMA channel is connected to UART0peripheral transmit (PDMA destination) request.
[19:16]	UART0_RXSEL	**PDMA UART0 Receive Selection** This field defines which PDMA channel is connected to UART0peripheral receive (PDMA source) request.
[15:12]	DPWM_TXSEL	**PDMA DPWM Transmit Selection** This field defines which PDMA channel is connected to DPWM peripheral transmit (PDMA destination) request.

[11:8]	ADC_RXSEL	**PDMA ADC Receive Selection** This field defines which PDMA channel is connected to ADC peripheral receive (PDMA source) request.
[7:4]	SPI0_TXSEL	**PDMA SPI0 Transmit Selection** This field defines which PDMA channel is connected to SPI0 peripheral transmit (PDMA destination) request.
[3:0]	SPI0_RXSEL	**PDMA SPI0 Receive Selection** This field defines which PDMA channel is connected to SPI0peripheral receive (PDMA source) request.

PDMA Global Interrupt Status Register (GCRISR)

Register	Offset	R/W	Description	Reset Value
GCRISR	PDMA_BA_GCR + 0x00C	R	PDMA Global Interrupt Status Register	0x0000_0000

Table 5-137PDMA Global Interrupt Status Register (GCRISR, address 0x5000_8F0C)

Bits	Descriptions	
[3:0]	**GCRISR**	**Interrupt Pin Status (Read Only)** GCRISR[*n*] is the interrupt status of PDMA channel *n.*

6 FLASH MEMORY CONTROLLER (FMC)

6.1 Overview

The ISD93xxseries is available with 141Kbytes of on-chip embedded Flash EEPROM for application program and data flash memory. The memory can be updated through procedures for In-Circuit Programming (ICP) through the ARM Serial-Wire Debug (SWD) port or via In-System Programming (ISP) functions under software control. In-System Programming (ISP) functions enable user to update program memory when chip is soldered onto PCB.

Main flash memory is divided into two partitions: Application Program ROM (APROM) and Data flash (DATAF). In addition there are two other partitions, a 4K Byte Boot Loader ROM (LDROM),and Configuration ROM (CONFIG).

Upon chip power-on, the Cortex-M0 CPU fetches code from APROM or LDROM determined by a boot select configuration in CONFIG.

The boundary between APROM and user DATA Flash can be configured to any sector address boundary. Erasable sector size is 1K Byte. This boundary is also specified in the CONFIG memory.

LDROM is a fixed 4K Byte in size, but if not required can be incorporated into the APROM address space of the 141K Byte device for a total device memory of 145K Byte.

6.2 Features

- AHB interface compatible

- Runs up to 98 MHz with zero wait-state for continuous address read access

- Mini-cache to reduce flash access and power consumption.

- 141KB application program memory (APROM)

- 4KB in system programming (ISP) boot loader program memory (LDROM)

- Configurable data flash with 1k Bytes sector erase unit

- Programmable data flash start address.

- In System Program (ISP) capability to update on chip Flash EEPROM

6.3 Flash Memory Controller Block Diagram

The flash memory controller consist of AHB slave interface, ISP control logic, writer interface and flash macro interface timing control logic. The block diagram of flash memory controller is shown as following:
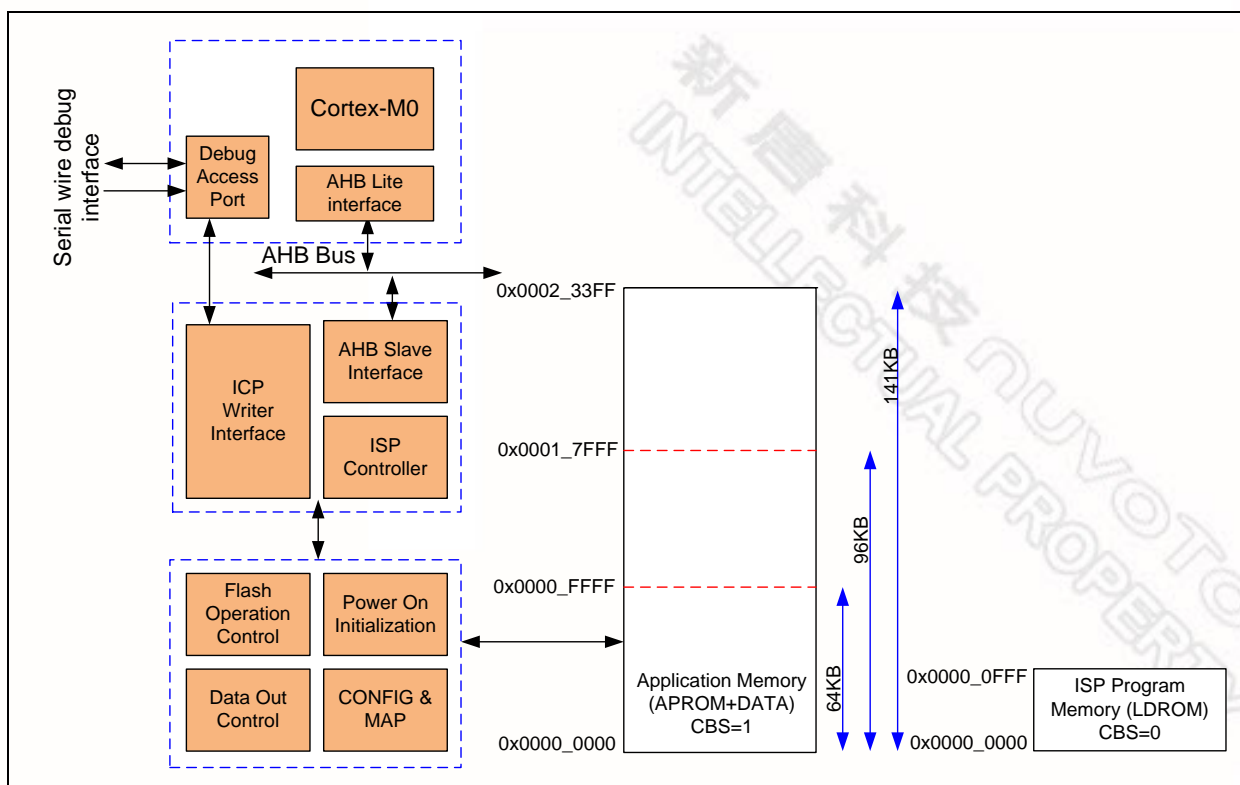

Figure 6-1 Flash Memory Control Block Diagram

6.4 Flash Memory Organization

The ISD93xx flash memory consists of Application Program (APROM) memory (141KB), data flash (DATAF), ISP boot loader (LDROM) program memory (4KB), user configuration (CONFIG). User configuration block provides 2 words that control system configuration, like flash security lock, boot select, brown out voltage level and data flash base address. An additional 504Bytes are available in CONFIG memory for the user to store custom configuration data. The first two CONFIG words are loaded from CONFIG memory at power-on into device control registers to initialize certain chip functions. The data flash start address (DFBADR) is defined in CONFIG memory and determines the relative size of the APROM and DATAF partitions.

Table 6-1 Memory Address Map

Block Name	Size	Start Address	End Address
APROM	141 KB	0x0000_0000	0x0002_33FF (141KB) OR DFBADR-1 if DFEN!=0
DATAF	User Configurable	DFBADR	0x0002_33FF (141KB)
LDROM	4 KB	0x0010_0000	0x0010_0FFF
CONFIG	512B	0x0030_0000	0x0030_01FF

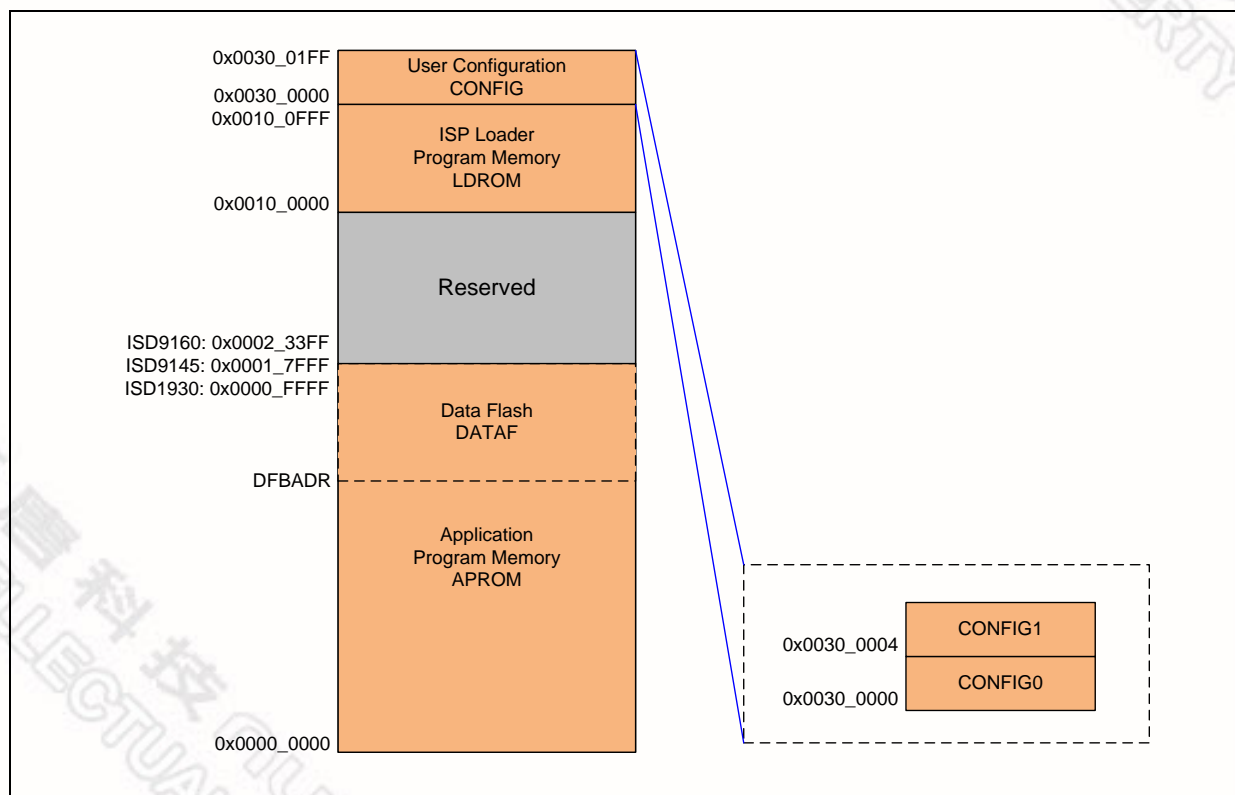The Flash memory organization is shown as below:

Figure 6-2 Flash Memory Organization

6.5 Boot Selection

The ISD93xx provides an in-system programming (ISP) feature to enable user to update the

application program memory when the chip is mounted on a PCB. A dedicated 4KBboot loader program memory is used to store ISP firmware. The user customizes this firmware to implement a protocol specific to their system to download updated application code. This firmware could utilize device peripherals such as UART, SPI or I2C to fetch new application code. The memory area from which the ISD93xx boots is controlled by the CBS bit in Config0 register.

6.6 Data Flash (DATAF)

The ISD93xx provides a data flash partition for user to store non-volatile data such as audio recordings. It accessed through ISP procedures via the Flash Memory Controller (FMC). The size of each erasable sector is 1Kbyte and minimum write size is one word (4Bytes).An erase operation resets all memory in sector to value 0xFF. A write operation can only change a '1' bit to a '0' bit. If a subset of the sector needs to be changed, the entire 1KB sector must be copied to another page or into SRAM in advance as entire sector must be erased before modification. Data flash and application program memory share the same memory space. If DFENB bit in Config0 is enabled ('0'), the data flash base address is defined by DFBADR and application program memory size is (X-N)KB and data flash size is N KB, where X is the total device memory size (141K) and N is number of Kbytes (sectors) reserved for data flash. In addition, for the 141KB device, the LDROM partition can be disabled and included in APROM/DATAF memory by setting the LDROM_EN configuration bit low allowing a total of 145KB of memory available to APROM/DATAF.
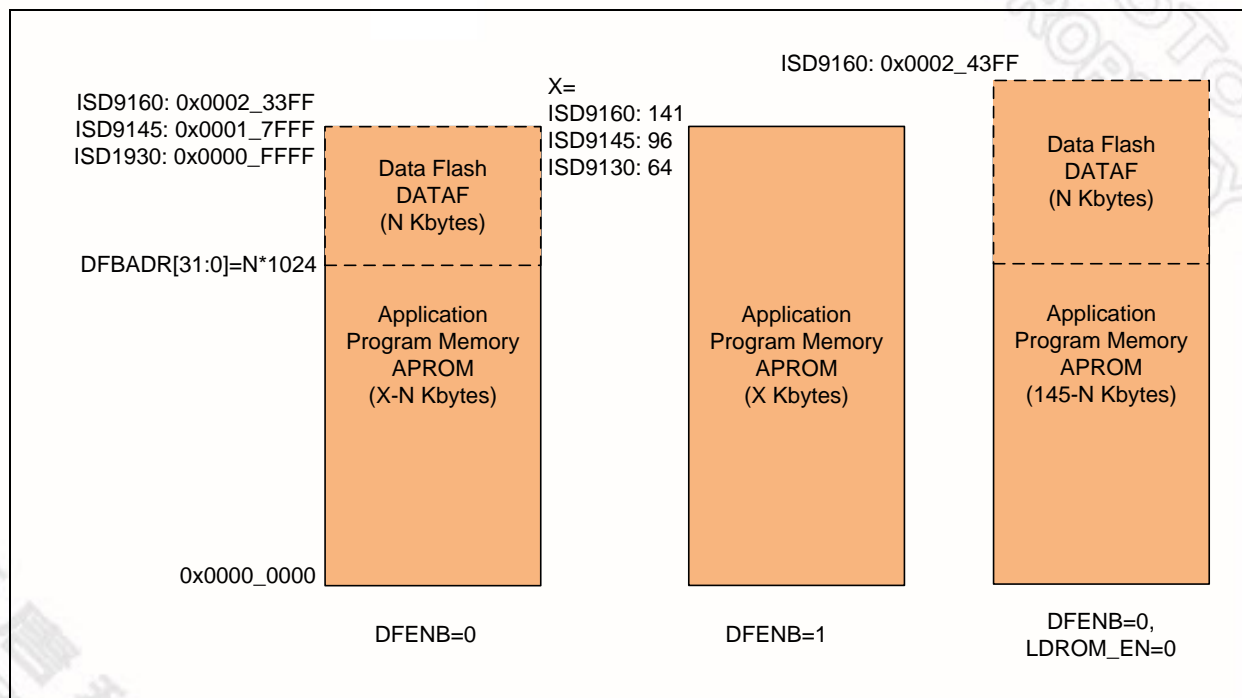
Figure 6-3 Flash Memory Structure

6.7 User Configuration (CONFIG)

Config0 (ISP Address = 0x0030_0000)

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

Table 6-2 User Configuration Register 0 (Config0, address 0x0030_0000 accessible through ISP only)

Bits	Descriptions	
[31:23]	Reserved	Reserved for future use
[23]	**CBODEN**	**Brown Out Detector Enable** If set to '0' the Brown Out Detector (BOD) will be enabled after power up. It will be configured at lowest voltage (2.1V) and if brown out condition detected will trigger the NMI interrupt to processor. 0= Enable, 1=Disable brown out detect after power on
[22:8]	Reserved	Reserved for future use
[7]	**CBS**	**Configuration Boot Selection** 0 = Chip will boot from LDROM, 1 = Chip will boot from APROM
[6:2]	Reserved	Reserved for future use
[1]	**LOCK**	**Security Lock** 0 = Flash data is locked, 1 = Flash data is not locked. When flash data is locked, only device ID, Config0 and Config1 can be read by ICP through serial debug interface. Other data is locked as 0xFFFFFFFF. Once locked no SWD debugging is possible. ISP can read data anywhere regardless of LOCK bit value.
[0]	**DFENB**	**Data Flash Enable Bar** When data flash is enabled, flash memory is partitioned between APROM and DATAF memory depending on the setting of data flash base address in Config1 register. If set to '0' then no DATAF partition exists. 0 = Enable data flash 1 = Disable data flash

Config1 (Address = 0x0030_0004)

Table 6-3User Configuration Register 1 (Config1, address 0x0030_0004 accessible through ISP only)

Bits	Descriptions	
[31:20]	Reserved	Reserved It is mandatory to program 0x00 to these Reserved bits
[19:0]	**DFBADR**	**Data Flash Base Address** This pointer sets the address for the start of data flash memory. Address must be on a 1KB sector boundary so DFBADR[9:0] must be 0x000.

6.8 In-System Programming (ISP)

The program and data flash memory support both in hardware In-Circuit Programming (ICP) and firmware based In-System programming (ISP). Hardware ICP programming mode uses the Serial-Wire Debug (SWD) port to program chip. Dedicated ICE Debug hardware or ICP gang-writers are available to reduce programming and manufacturing costs. For firmware updates in the field, the ISD93xx provides an ISP mode allowing a device to be reprogrammed under software control.

ISP is performed without removing the device from the system. Various interfaces enable LDROM firmware to fetch new program code from an external source. A common method to perform ISP would be via a UART controlled by firmware in LDROM. In this scenario, a PC could transfer new APROM code through a serial port. The LDROM firmware receives it and re-programs APROM through ISP commands. An alternative might be to fetch new firmware from an attached SD-Card via the SPI interface.

6.8.1 ISP Procedure

The ISD93xx will boot from APROM or LDROM from a power-on reset as defined by user configuration bit CBS. If user desires to update application program in APROM, the ISPCON.BS can be set to '1' and a software reset issued. This will cause the chip to boot from LDROM. An example flow diagram of the ISP sequence is shown in Figure 6-5.

The ISPCON register is a protected register, user must first follow the unlock sequence (see Protected Register Lock Key Register (REGLOCK)) to gain access. This procedure is to protect the flash memory from unintentional access.

To enable ISP functionality software must first ensure the ISP clock (AHBCLK.ISP_EN) is present then set the ISPCON.ISPEN bit.

Several error conditions are checked after software writes the ISPTRIG register. If an error condition occurs, ISP operation is not started and the ISP fail flag (ISPCON.ISPFF) will be set instead. The ISPFF flag will remain set until it is cleared by software. Subsequent ISP procedure can be started even if ISPFF is set. It is recommended that software check ISPFF bit and clear it after each ISP operation if set.

When ISPTRIG register is set, the CoretxM0 CPU will wait for ISP operation to finish, during this period; peripherals operate as usual. If any interrupt requests occur, CPU will not service them until ISP operation finishes. As the ISP functions affect the operation of the flash memory M0 instruction pipeline should be flushed with an ISB (Instruction Synchronization Barrier) instruction after the ISP is triggered.
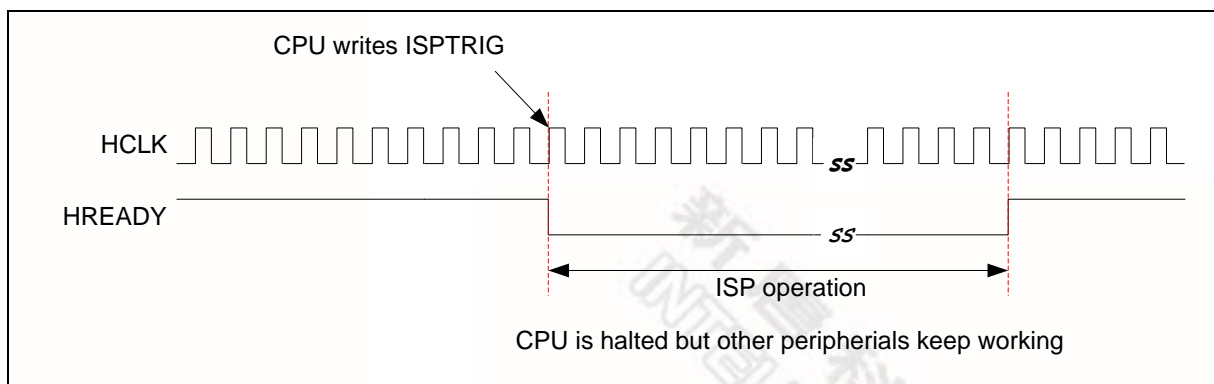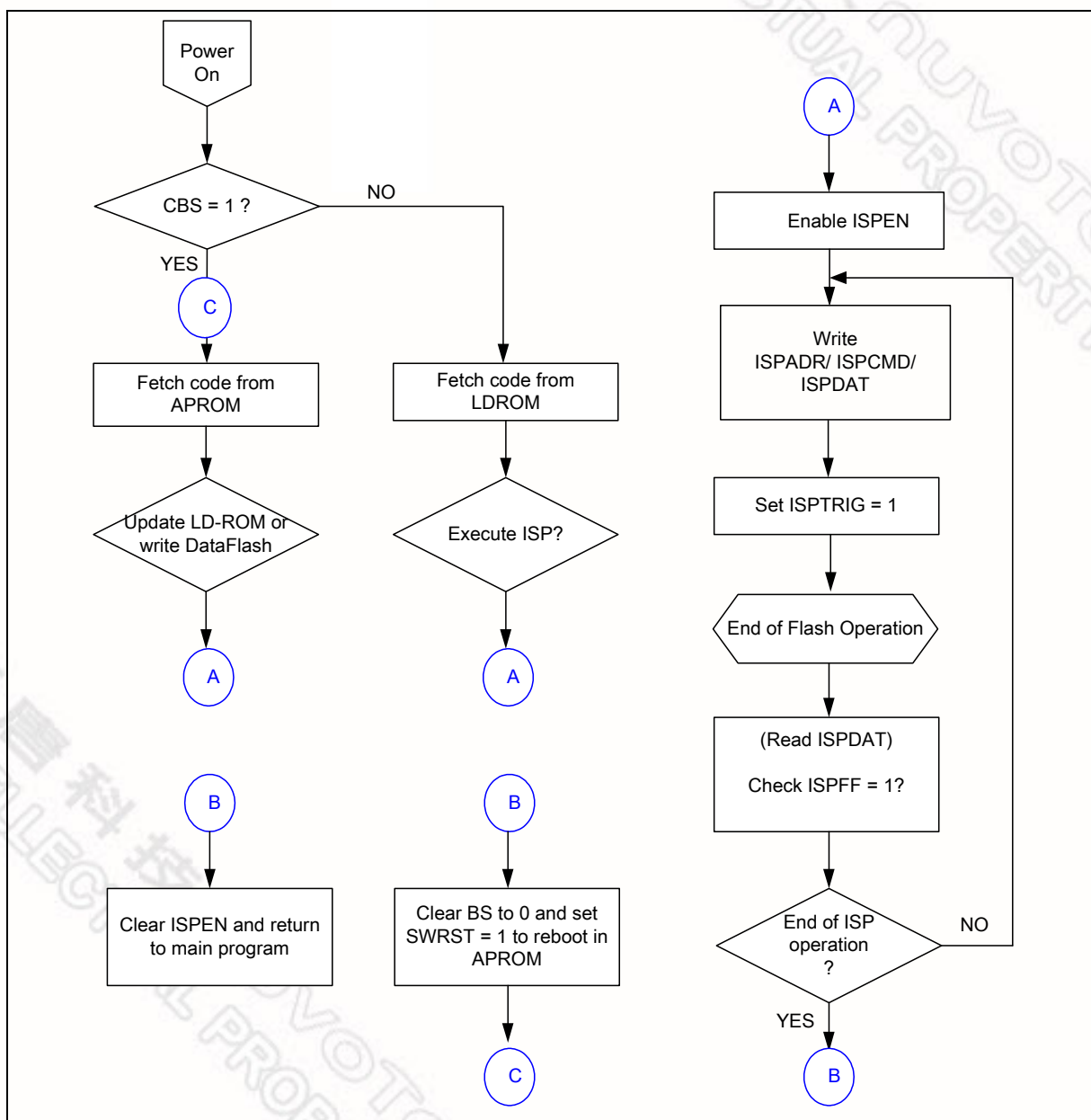
Figure 6-4 ISP Operation Timing

Figure 6-5 Boot Sequence and ISP Procedure

The ISP command set is shown in Table 6-4. Three registers determine the action of a command: ISPCMD is the command register and accepts commands for reading ID registers and read/write/erase of flash memory. The ISPADR is the address register where the flash memory address for access is written. ISPDAT is the data register that input data is written to and return data read from. An ISP command is executed by setting ISPCMD, ISPDAT and ISPADR then writing to the trigger register ISPTRIG.

There is an ISP command to read the device ID register. This register returns a code that reports the memory configuration of the ISD93xx series part as given in Table 6-5.

Table 6-4 ISP Command Set

ISP Mode	ISPCMD	ISPADR			ISPDAT
	ISPCMD[5:0]	A21	A20	A[19:0]	D[31:0]
Standby	0x3x	x	x	x	x
Read Company ID	0x0B	x	x	x	Returns 0x0000_00DA
Read Device ID	0x0C	x	x	0x00000	0x1D00_02nn. See Table 6-5
FLASH Page Erase	0x22	0	A[20]	A[19:0]	x
FLASH Program	0x21	0	A[20]	A[19:0]	Data input
FLASH Read	0x00	0	A[20]	A[19:0]	Data output
CONFIG Page Erase	0x22	1	1	A[19:0]	x
CONFIG Program	0x21	1	1	A[19:0]	Data input
CONFIG Read	0x00	1	1	A[19:0]	Data output

Table 6-5 Device ID Memory Size

DID[7:0]	RAM Size (KB)		DID[15:8]	Flash Size (KB)
1	4		5	32
2	8		6	64
3	12		7	96
4	16		8	145

6.9 Flash Control Register Map

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

Register	Offset	R/W	Description	Reset Value	Reference
Base Address (FMC_BA) : 0x5000_C000					
ISPCON	FMC_BA+0x000	R/W	ISP Control Register	0x0000_0000	Table 6-6
ISPADR	FMC_BA+0x004	R/W	ISP Address Register	0x0000_0000	Table 6-7
ISPDAT	FMC_BA+0x008	R/W	ISP Data Register	0x0000_0000	Table 6-8
ISPCMD	FMC_BA+0x00C	R/W	ISP Command Register	0x0000_0000	Table 6-9
ISPTRG	FMC_BA+0x010	R/W	ISP Trigger Register	0x0000_0000	Table 6-10
DFBADR	FMC_BA+0x014	R	Data Flash Start	0x0000_0000	Table 6-11

6.10 Flash Control Register Description

ISP Control Register (ISPCON)

The ISPCON register is a protected register, user must first follow the unlock sequence (see Protected Register Lock Key Register (REGLOCK)) to gain access.

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

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

Table 6-6 ISP Control Register (ISPCON, address 0x5000_C000)

Bits	Descriptions	
[21]	CACHE_DIS	**Cache Disable** When set to 1, caching of flash memory reads is disabled.
[18:16]	WAIT_CFG	**Flash Access Wait State Configuration** For M and H speed grade parts this sets the access speed to the flash memory. 0x00: Three wait states. HCLK > 72MHz 0x01: Two wait states. 72MHz > HCLK > 50MHz 0x02: One wait state. HCLK <= 50MHz Before changing WAIT_CFG, ensure HCLK speed is < 50MHz.
[7]	SWRST	**Software Reset** Writing 1 to this bit will initiate a software reset. It is cleared by hardware after reset.
[6]	ISPFF	**ISP Fail Flag** This bit is set by hardware when a triggered ISP meets any of the following conditions: (1) APROM writes to itself. (2) LDROM writes to itself. (3) Destination address is illegal, such as over an available range. Write 1 to clear.

[5]	**LDUEN**	**LDROM Update Enable** LDROM update enable bit. 1 = LDROM can be updated when the MCU runs in APROM. 0 = LDROM cannot be updated
[4]	**CFGUEN**	**CONFIG Update Enable** 1 = Enable, 0 = Disable When enabled, ISP functions can access the CONFIG address space and modify device configuration area.
[3:2]	Reserved	Reserved
[1]	**BS**	**Boot Select** 1: LDROM, 0: APROM Modify this bit to select which ROM next boot is to occur. This bit also functions as MCU boot status flag, which can be used to check where MCU booted from. **This bit is initialized after power-on reset with the inverse of CBS in Config0; It is not reset for any other reset event.**
[0]	**ISPEN**	**ISP Enable** 1 = Enable ISP function 0 = Disable ISP function

ISP Address Register (ISPADR)

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

Table 6-7 ISP Address Register (ISPADR, address 0x5000_C004)

Bits	Descriptions	
[31:0]	**ISPADR**	**ISP Address Register** This is the memory address register that a subsequent ISP command will access. ISP operation are carried out on 32bit words only, consequently ISPARD[1:0] must be 00b for correct ISP operation.

ISP Data Register (ISPDAT)

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

Table 6-8 ISP Data Register (ISPDAT, address 0x5000_C008)

Bits	Descriptions	
[31:0]	**ISPDAT**	**ISP Data Register** Write data to this register before an ISP program operation. Read data from this register after an ISP read operation

ISP Command (ISPCMD)

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

Table 6-9ISP Data Register (ISPCMD, address 0x5000_C00C)

Bits	Descriptions	
[31:6]	Reserved	Reserved
[5:0]	**ISPCMD**	**ISP Command** | Operation Mode	ISPCMD
----------------	--------
Standby	0x3X
Read	0x00
Program	0x21
Page Erase	0x22
Read CID	0x0B
Read DID	0x0C	

ISP Trigger Control Register (ISPTRG)

The ISPTRG register is a protected register, user must first follow the unlock sequence (see Protected Register Lock Key Register (REGLOCK)) to gain access.

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

Table 6-10ISP Trigger Control Register (ISPTRG, address 0x5000_C010)

Bits	Descriptions	
[31:1]	Reserved	Reserved
[0]	**ISPGO**	**ISP Start Trigger** Write 1 to start ISP operation. This will be cleared to 0 by hardware automatically when ISP operation is finished. 1 = ISP is on going 0 = ISP operation is finished After triggering an ISP function M0 instruction pipeline should be flushed with a ISB instruction to guarantee data integrity. This is a protected register, user must first follow the unlock sequence (see Protected Register Lock Key Register (REGLOCK)) to gain access.

Data Flash Base Address Register (DFBADR)

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

Table 6-11 Data Flash Base Address Register (DFBADR, address 0x5000_C014)

Bits	Descriptions	
[31:0]	**DFBA**	**Data Flash Base Address** This register reports the data flash starting address. It is a read only register. Data flash size is defined by user configuration; register content is loaded from Config1 when chip is reset.

7   ANALOG SIGNAL PATH BLOCKS

This section describes the functional blocks that perform analog signal functions on the ISD93xx. This includes the ADC, DPWM Speaker Driver, PGA Gain Amplifier, Automatic Gain Control and a variety of auxiliary analog functional blocks.

7.1   Audio Analog-to-Digital Converter (ADC)

7.1.1   Functional Description

The ISD93xxseries includes a 2nd Order Delta-Sigma Audio Analog-to-Digital converter providing SNR >85dB and THD >68dB.The converter can run at sampling rates up to 6.144MHz while a configurable decimation filter allows oversampling ratios of 64/128/192 and 384. This provides support for standard audio sampling rates from 8kHz to 48kHz.

7.1.2   Features

- Front-end PGA providing gain range of -12dB – 35dB.

- Boost Gain stage of 0dD or 26dB.

- Configurable OSR (Over Sampling Ratio) of 64/128/192/384

- Configurable clock rate through master oscillator integer division.

- Decimation signal can be used directly or passed to bi-quad filter for further filtering.

- Audio data buffered to 8 word FIFO, accessible via APB and PDMA.
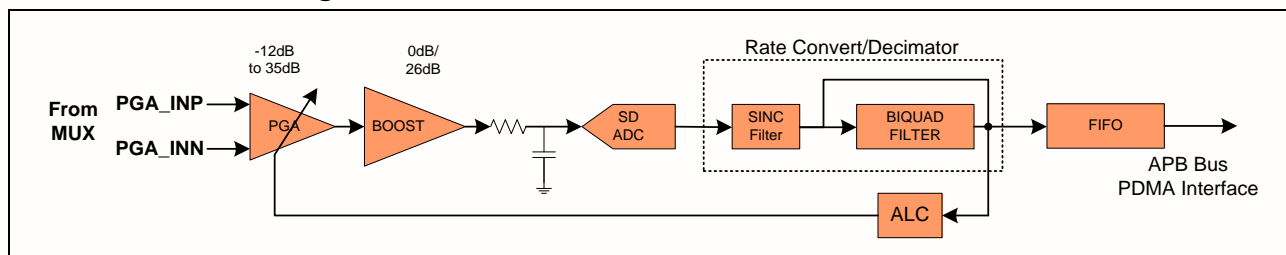
7.1.3   Block Diagram

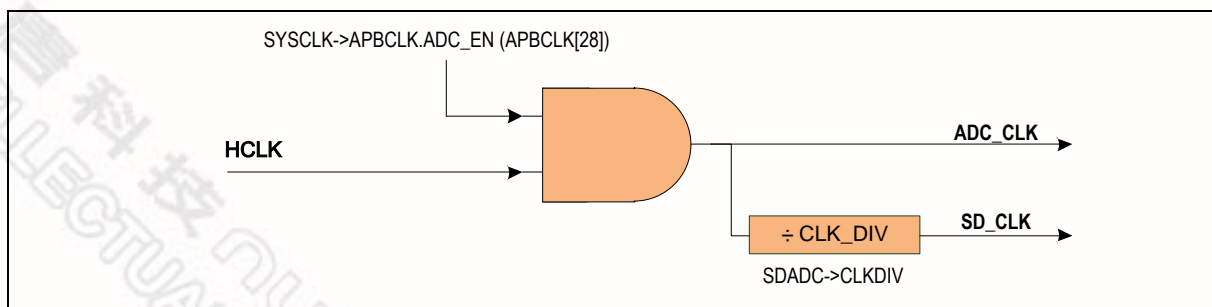Figure 7-1 ADC Signal Path Block Diagram

Figure 7-2 ADC Clock Control

7.1.4 Operation

The ADC is an Audio Delta-Sigma converter that operates by oversampling the analog input at low resolution and decimating the result by an over-sampling ratio to obtain a high resolution output which is pushed into the FIFO. The ultimate data rate is determined by the converter clock frequency SDCLK, and the oversampling ratio.

The data stream generated by the ADC is most conveniently handled by PDMA which can load data into a streaming audio buffer for further processing. Alternatively an interrupt driven approach can be used to monitor the FIFO.

If FIFO is not serviced then oldest data is over-written such that the FIFO always contains the eight most recent samples.

7.1.4.1 Determining Sample Rate

The maximum clock rate of the Delta-Sigma Converter is 6.144MHz. Best performance is gained with clocks rates between 1.024MHz and 4.096MHz. Sample rate is given by the following formula:

$$F_s = HCLK \div CLK\_DIV \div OSR$$

Tables of common audio sample rates are provided below.

Table 7-1 Sample Rates for HCLK=49.152MHz

HCLK=49.152MHz	SD_CLK	Sample Rate (Hz) for OSR			
ADC CLKDIV		64	128	192	384
8	6,144,000	96,000	48,000	32,000	16,000
16	3,072,000	48,000	24,000	16,000	8,000
24	2,048,000	32,000	16,000	10,667	5,333
32	1,536,000	24,000	12,000	8,000	4,000
48	1,024,000	16,000	8,000	5,333	2,667

Table 7-2Sample Rates for HCLK=32.768MHz

HCLK=32.768MHz	SD_CLK	Sample Rate (Hz) for OSR			
ADC CLKDIV		64	128	192	384
8	4,096,000	64,000	32,000	21,333	10,667
16	2,048,000	32,000	16,000	10,667	5,333
24	1,365,333	21,333	10,667	7,111	3,556
32	1,024,000	16,000	8,000	5,333	2,667

Table 7-3 Sample Rates for HCLK=24.576MHz

HCLK=24.576MHz ADC CLKDIV	SD_CLK	Sample Rate (Hz) for OSR			
		64	128	192	384
8	3,072,000	48,000	24,000	16,000	8,000
12	2,048,000	32,000	16,000	10,667	5,333
16	1,536,000	24,000	12,000	8,000	4,000
24	1,024,000	16,000	8,000	5,333	2,667

Table 7-4 Sample Rates for HCLK=16.384MHz

HCLK=16.384MHz ADC CLKDIV	SD_CLK	Sample Rate (Hz) for OSR			
		64	128	192	384
4	4,096,000	64,000	32,000	21,333	10,667
8	2,048,000	32,000	16,000	10,667	5,333
16	1,024,000	16,000	8,000	5,333	2,667
24	682,667	10,667	5,333	3,556	1,778
32	512,000	8,000	4,000	2,667	1,333

7.1.4.2 *Configuring Analog Path*

To operate the ADC the entire analog path from analog input to ADC needs to be configured for correct operation. This involves:

- Selecting and powering up VMID reference.
- Powering up modulator and reference buffers.
- Selecting an input source with the analog MUX.
- Configure sample rate and ADC clock source.

7.1.4.3 *Interrupt Sources*

The ADC can be configured to generate an interrupt when the data level in the FIFO exceeds a defined threshold. The interrupt condition is only cleared by disabling the interrupt or reading values from the FIFO. In addition two comparators can monitor the ADC FIFO output to generate interrupts when set levels are exceeded.
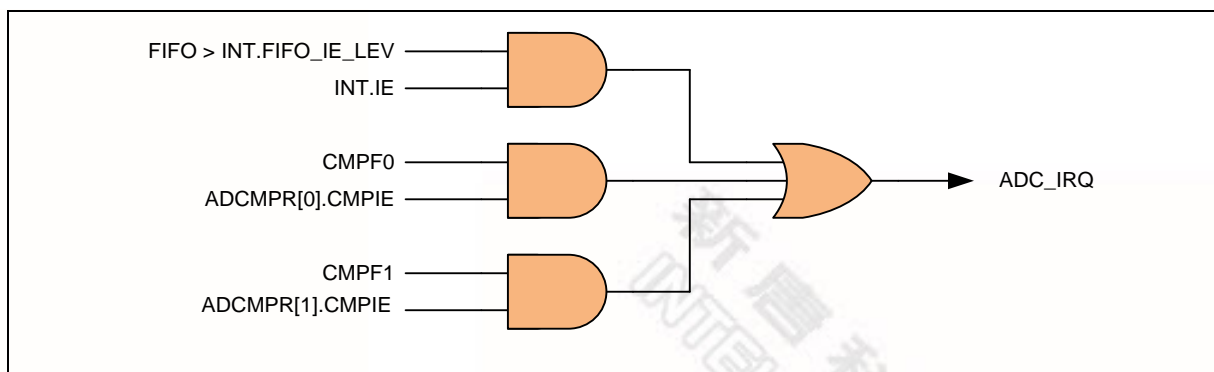
Figure 7-3 SDADC Controller Interrupt

7.1.4.4 *Peripheral DMA Request*

Normal use of the ADC is with PDMA. In this mode ADC requests PDMA service whenever data is in FIFO. PDMA channel will copy this data to a buffer and alert the CPU when buffer is full. In this way an entire buffer of data can be collected without any CPU intervention.

7.1.5 ADC Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Offset	R/W	Description	Reset Value	Reference
ADC_BA = 0x400E_0000					
ADCOUT	ADC_BA+0x00	R	FIFO Data Out.	0x0000_XXXX	Table 7-5
EN	ADC_BA+0x04	R/W	ADC Enable Register	0x0000_0000	Table 7-6
CLK_DIV	ADC_BA+0x08	R/W	ADC Clock Divider Register	0x0000_0000	Table 7-7
DEC	ADC_BA+0x0C	R/W	ADC Decimation Control Register	0x0000_0000	Table 7-8
INT	ADC_BA+0x10	R/W	ADC Interrupt Control Register	0x0000_0000	Table 7-9
ADCPDMA	ADC_BA+0x14	R/W	ADC PDMA Control Register	0x0000_0000	Table 7-10
ADCMPR[0]	ADC_BA+0x18	R/W	ADC Comparator 0 Control Register	0x0000_0000	Table 7-11
ADCMPR[1]	ADC_BA+0x1C	R/W	ADC Comparator 1 Control Register	0x0000_0000	Table 7-11

7.1.6   ADC Register Description

FIFO Audio Data Register (ADCOUT)

Register	Offset	R/W	Description	Reset Value
ADCOUT	ADC_BA+0x00	R	ADC Audio Data FIFO Read	0x0000_XXXX

Table 7-5 FIFO Audio Data Register (ADCOUT, address 0x400E_0000)

Bits	Descriptions	
[31:16]	Reserved	**0x0000**
[15:0]	**ADCOUT**	**ADC Audio Data FIFO read.** A read of this register will read data from the audio FIFO and increment the read pointer. A read past empty will repeat the last data. Can be used with FIFO_IE_LEV interrupt to determine if valid data is present in FIFO.

ADC Enable Register (EN)

Register	Offset	R/W	Description	Reset Value
EN	ADC_BA+0x04	R/W	ADC Enable Register	0x0000_0000

Table 7-6 ADC Enable Register (EN, address 0x400E_0004)

Bits	Descriptions	
[31:1]	Reserved	-
[0]	**EN**	**ADC Enable** 1 = ADC Conversion enabled. 0 = Conversion stopped and ADC is reset including FIFO pointers.

ADC Clock Division Register (CLK_DIV)

Register	Offset	R/W	Description	Reset Value
CLK_DIV	ADC_BA+0x08	R/W	ADC Converter Clock Divider	0x0000_0000

Table 7-7 ADC Clock Divider Register (CLK_DIV, address 0x400E_0008)

Bits	Descriptions	
[31:8]	Reserved	-
[7:0]	**CLK_DIV**	**ADC Clock Divider** This register determines the clock division ration between the incoming ADC_CLK (=HCLK by default) and the Delta-Sigma sampling clock of the ADC. This together with the over-sampling ratio (OSR) determines the audio sample rate of the converter. CLK_DIV should be set to give a SD_CLK frequency in the range of 1.024-6.144MHz. CLK_DIV must be greater than 2. SD_CLK frequency = HCLK ÷ CLK_DIV

ADC Decimation Control Register (DEC)

Register	Offset	R/W	Description	Reset Value
DEC	ADC_BA+0x0C	R/W	ADC Decimation Control Register	0x0000_0000

Table 7-8 ADC Decimation Control Register (DEC, address 0x400E_000C)

Bits	Descriptions	
[19:16]	**GAIN**	**CIC Filter additional Gain.** This should normally remain default 0. Can be set to non-zero values to provide additional digital gain from the decimation filter. An additional gain is applied to signal of GAIN÷2.
[3:0]	**OSR**	**Decimation Over-Sampling Ratio** This term determines the over-sampling ratio of the decimation filter. Valid values are: 0: OSR=64 1: OSR=128 2: OSR=192 3: OSR=384

ADC Interrupt Control Register (INT)

Register	Offset	R/W	Description	Reset Value
INT	ADC_BA+0x10	R/W	ADC Interrupt Control Register	0x0000_0000

Table 7-9ADC Interrupt Control Register (INT, address 0x400E_0010)

Bits	Descriptions	
[31]	**IE**	**Interrupt Enable** If set to '1' an interrupt is generated whenever FIFO level exceeds that set in FIFO_IE_LEV.
[2:0]	**FIFO_IE_LEV**	**FIFO Interrupt Level** Determines at what level the ADC FIFO will generate a servicing interrupt to the CPU. Interrupt will be generated when number of words present in ADC FIFO is > FIFO_IE_LEV.

ADC PDMA Control Register (ADCPDMA)

Register	Offset	R/W	Description	Reset Value
ADCPDMA	ADC_BA+0x14	R/W	ADC PDMA Control Register	0x0000_0000

Table 7-10ADC PDMA Control Register (ADCPDMA, address 0x400E_0014)

Bits	Descriptions	
[0]	**RxDmaEn**	**Enable ADC PDMA Receive Channel** Enable ADC PDMA. If set, then ADC will request PDMA service when data is available.

A/D Compare Register 0/1 (ADCMPR0/1)

Register	Offset	R/W	Description	Reset Value
ADCMPR[0]	ADC_BA+0x18	R/W	A/D Compare Register 0	0x0000_0000
ADCMPR[1]	ADC_BA+0x1C	R/W	A/D Compare Register 1	0x0000_0000

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

Table 7-11 ADC Comparator Control Registers (ADCMPR[n], address 0x400E_0018, 0x400E_001C)

Bits	Descriptions	
[31:16]	**CMPD**	**Comparison Data** 16 bit value to compare to FIFO output word.
[11:8]	**CMPMATCNT**	**Compare Match Count** When the A/D FIFO result matches the compare condition defined by CMPCOND, the internal match counter will increase by 1. When the internal counter reaches the value to (CMPMATCNT +1), the CMPF bit will be set.
[7]	**CMPF**	**Compare Flag** When the conversion result meets condition in ADCMPR0 this bit is set to 1. It is cleared by writing 1 to self.
[2]	**CMPCOND**	**Compare Condition** 1= Set the compare condition that result is greater or equal to CMPD 0= Set the compare condition that result is less than CMPD **Note:** When the internal counter reaches the value (CMPMATCNT +1), the CMPF bit will be set.
[1]	**CMPIE**	**Compare Interrupt Enable** 1 = Enable compare function interrupt.0 = Disable compare function interrupt. If the compare function is enabled and the compare condition matches the setting of CMPCOND and CMPMATCNT, CMPF bit will be asserted, if CMPIE is set to 1, a compare interrupt request is generated.
[0]	**CMPEN**	**Compare Enable** 1 = Enable compare.0 = Disable compare. Set this bit to 1 to enable compare CMPD with FIFO data output.

7.2   Audio Class D Speaker Driver (DPWM)

7.2.1   Functional Description

The ISD93xxseries includes a differential Class D (PWM) speaker driver capable of delivering 1W into an 8Ω load at 5V supply voltage. The driver works by up-sampling and modulating a PCM input to differentially drive the SPK+ and SPK- pins. The speaker driver operates from its own independent supply VCCSPK and VSSSPK. This supply should be well decoupled as peak currents from speaker driver are large.

7.2.2   Features

•   Differential Bridge-Tied-Load structure to directly drive 8Ω Speaker.

•   Power delivery up to 1W @5V into 8Ω.

•   Power efficiency of up to 85%.

•   Configurable input sample rate.

•   16 Sample FIFO for audio output.

•   PDMA data channel for streaming of PCM audio data.
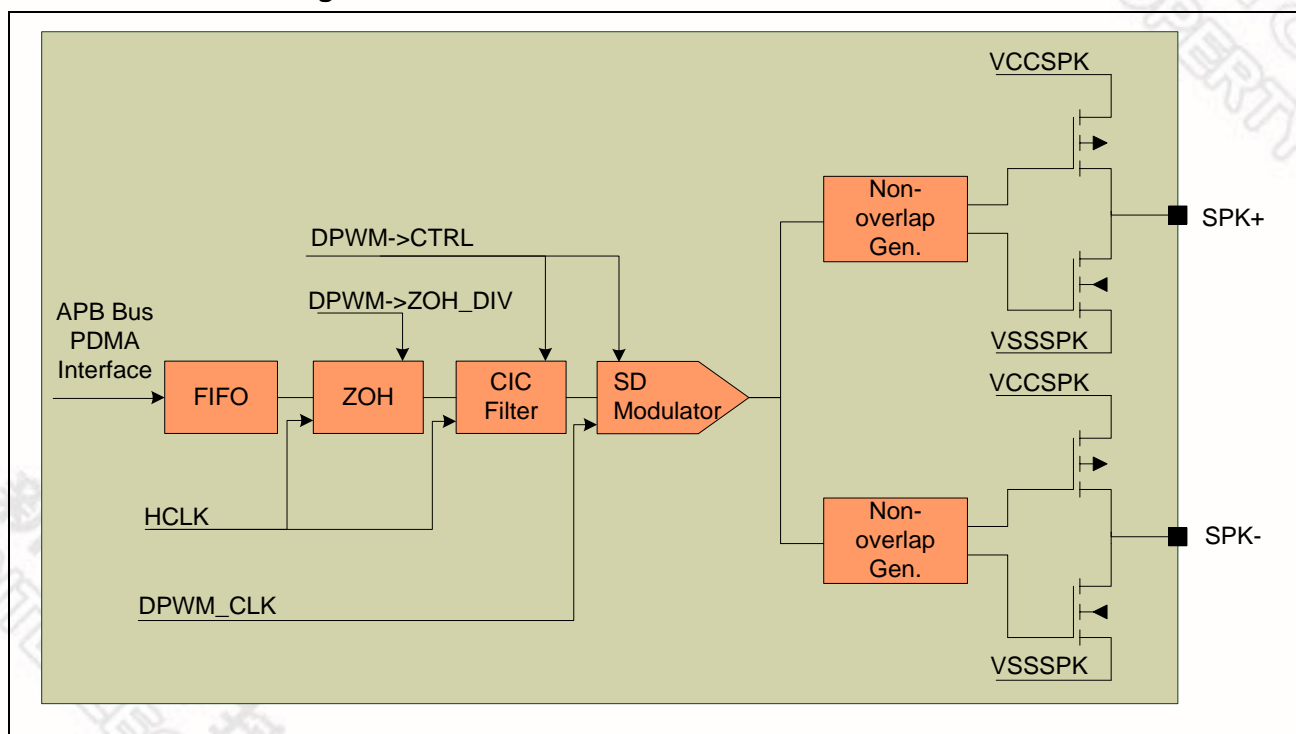
7.2.3   Block Diagram

Figure 7-4DPWM Block Diagram

7.2.4   Operation

The DPWM block receives audio data by writing 16bit PCM audio to the FIFO. FIFO is accessed through PDMA for ease of streaming. The audio stream is sampled by a zero-order hold and fed to an up-sampling Cascaded Integrator Comb (CIC) filter with an up-sampling ratio of 64. The signal is then modulated and sent to the driver stage through a non-overlap circuit. Master clock rate of the Delta-Sigma modulator is controlled by DPWM_CLK. This clock is generated by the internal oscillator

(OSC48M) and operates at the frequency of OSC48M or 2x the frequency of OSC48M (See CLKSEL1 register Table 5-36). Ultimate SNR (Signal-to-Noise Ratio) is determined by the time resolution of the master clock.

7.2.4.1   *Determining Sample Rate*

The sample rate at which the DPWM block consumes audio data is given by:

$$F_s = HCLK \div ZOH\_DIV \div 64$$

Where HCLK is the master CPU clock rate and ZOH_DIV is the divider control register. A table of common audio sample rates is provided below.

Table 7-12 DPWM Sample Rates for Various HCLK

HCLK (MHz)	ZOH_DIV	Sample Rate (Hz)
49.152	24	32,000
49.152	48	16,000
49.152	96	8,000
32.768	16	32,000
32.768	32	16,000
32.768	64	8,000
24.576	12	32,000
24.576	24	16,000
24.576	48	8,000

7.2.4.2   *Configuring Speaker Driver*

To operate the speaker driver the following configuration is recommended:

- Enable DPWM clock source (APBCLK.DPWM_EN Table 5-33, CLKSEL1.DPWM_S Table 5-36).
- Reset DPWM IP block. (IPRSTC2.DPWM_RST Table 5-4)
- Select sample rate based on current HCLK frequency.
- Setup PDMA channel to provide data to DPWM.
- Enable PDMA Request.
- Enable Driver.

7.2.4.3   *Peripheral DMA Request*

Normal use of the DPWM is with PDMA. In this mode DPWM requests PDMA service whenever there is space in FIFO. PDMA channel will copy data from a streaming buffer to the DPWM and alert the CPU when buffer is empty. In this way an entire buffer of data can be sent to DPWM without any CPU intervention.

7.2.5   DPWM Register Map

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

Register	Offset	R/W	Description	Reset Value	Reference
DPWM_BA = 0x4007_0000					
CTRL	DPWM_BA+0x00	R/W	DPWM Control Register.	0x0000_0000	Table 7-13
STAT	DPWM _BA+0x04	R	DPWM FIFO Status	0x0000_0002	Table 7-14
DMA	DPWM _BA+0x08	R/W	DPWM PDMA Control Register	0x0000_0000	Table 7-15
FIFO	DPWM _BA+0x0C	W	DPWM FIFO Input	0x0000_0000	Table 7-16
ZOH_DIV	DPWM _BA+0x10	R/W	DPWM Zero Order Hold Division Register	0x0000_0030	Table 7-17

7.2.6 DPWM Register Description

DPWM Control Register (CTRL)

Register	Offset	R/W	Description	Reset Value
CTRL	DPWM_BA+0x00	R/W	DPWM Control Register	0x0000_0000

7	6	5	4	3	2	1	0
Reserved	Enable	Dither		Deadtime	Freq		

Table 7-13 DPWM Control Register (DPWM->CTRL, address 0x4007_0000)

Bits	Descriptions	
[6]	**Enable**	**DPWM Enable.** 1: Enable DPWM, SPK pins are enabled and driven, data is taken from FIFO. 0: Disable DPWM, SPK pins are tri-state, CIC filter is reset, FIFO pointers are reset (FIFO data is not reset).
[5:4]	**Dither**	**DPWM Signal Dither Control** To prevent structured noise on PWM output due to DC offsets in the input signal it is possible to add random dither to the PWM signal. These bits control the dither: 0: No dither. 1: ±1 bit dither 3: ±2 bit dither
[3]	**Deadtime**	**DPWM Driver Deadtime Control.** Enabling this bit will insert an additional clock cycle deadtime into the switching of PMOS and NMOS driver transistors.
[2:0]	**Freq**	**DPWM Modulation Frequency.** This parameter controls the carrier modulation frequency of the PWM signal as a proportion of DPWM_CLK. | Freq	DPWM_CLK Division	Frequency for DPWM_CLK = 98.304MHz
---	---	---
0	228	431,158
1	156	630,154
2	76	1,293,474
3	52	1,890,462
4	780	126,031
5	524	187,603
6	396	248,242
7	268	366,806	

DPWM FIFO Status Register (STAT)

Register	Offset	R/W	Description	Reset Value
STAT	DPWM_BA+0x04	R	DPWM FIFO Status Register	0x0000_0002

Table 7-14DPWM FIFO Status Register (STAT, address 0x4007_0004)

Bits	Descriptions	
[1]	**EMPTY**	FIFO Empty 1= FIFO is empty 0= FIFO is not empty
[0]	**FULL**	**FIFO Full** 1 = FIFO is full. 0 = FIFO is not full.

DPWM PDMA Control Register(DMA)

Register	Offset	R/W	Description	Reset Value
DMA	DPWM_BA+0x08	R/W	DPWM PDMA Control Register	0x0000_0000

Table 7-15DPWM PDMA Control Register (DMA, address 0x4007_0008)

Bits	Descriptions	
[31:8]	Reserved	-
[0]	**EnablePDMA**	**Enable DPWM DMA Interface.** 1= Enable PDMA. Block will request data from PDMA controller whenever FIFO is not empty. 0= Disable PDMA. No requests will be made to PDMA controller.

DPWM FIFO Input (FIFO)

Register	Offset	R/W	Description	Reset Value
FIFO	DPWM_BA+0x0C	W	DPWM FIFO Input	0x0000_0000

Table 7-16DPWM FIFO Input (FIFO, address 0x4007_000C)

Bits	Descriptions	
[15:0]	**FIFO**	**DPWM FIFO Audio Data Input** A write to this register pushes data onto the DPWM FIFO and increments the write pointer. This is the address that PDMA writes audio data to.

DPWM ZOH Division (ZOH_DIV)

Register	Offset	R/W	Description	Reset Value
ZOH_DIV	DPWM_BA+0x10	R/W	DPWM Zero Order Hold Division Register	0x0000_0030

Table 7-17DPWM Zero Order Hold Division Register (FIFO, address 0x4007_0010)

Bits	Descriptions	
[7:0]	**ZOH_DIV**	**DPWM Zero Order Hold, down-sampling divisor.** The input sample rate of the DPWM is set by HCLK frequency and the divisor set in this register by the following formula: $$F_s = HCLK \div ZOH_{DIV} \div 64$$ Valid range is 1,..,255. Default is 48, which gives a sample rate of 16kHz for a 49.152MHz (default) HCLK.

7.3   Analog Comparator

7.3.1   Functional Description

ISD93xx series contains two analog comparators. The comparator output is a logical one when positive input greater than negative input, otherwise the output is a zero. Each comparator can be configured to cause an interrupt when the comparator output value changes. The block diagram is shown in Figure 7-5.

Note that the analog input port pins must be configured as input type or analog alternate function before Analog Comparator function is enabled.

7.3.2   Features

● Analog input voltage range: 0~5.0V

● Comparator 0 multiplexed to all analog enabled GPIO (GPIOB[7:0]).

● Comparator 0 can compare against VBG or VMID.

● Comparator 1 can compare GPIOB[7] to GPIOB[6] or VBG.

● Single comparator interrupt requested by either comparator.

● Can be used in conjunction with CapSense block for Capacitive sensing.
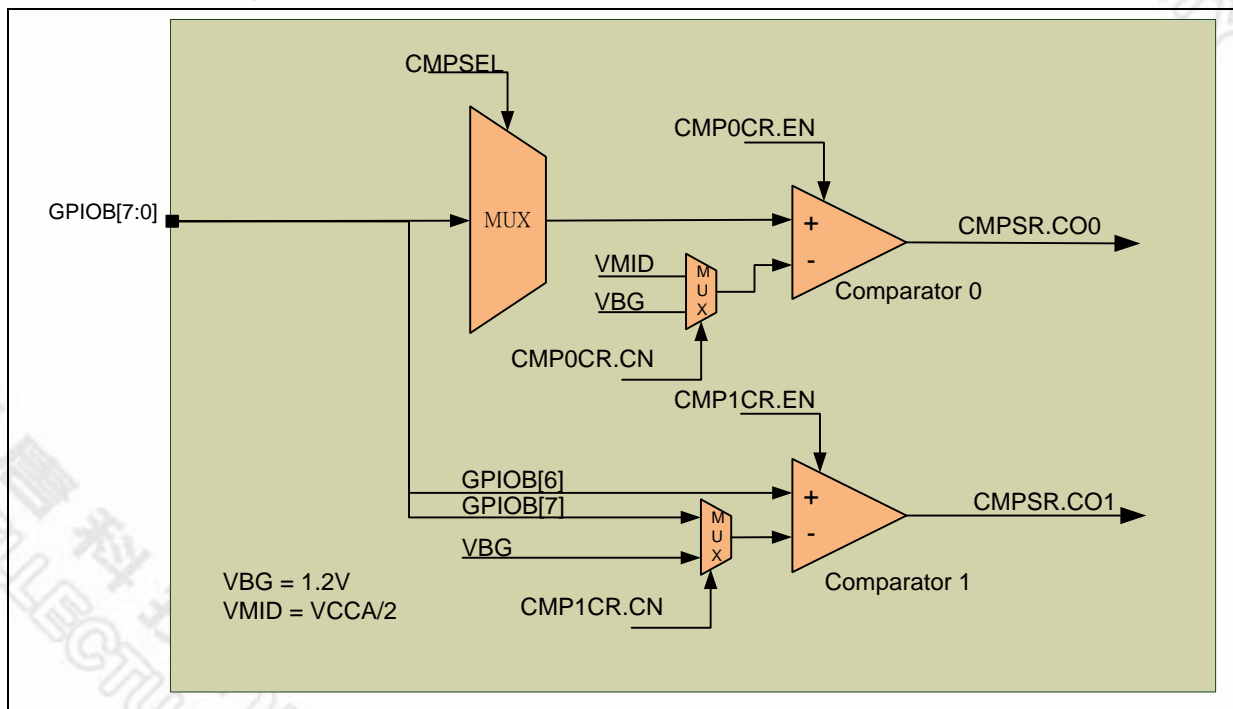
7.3.3   Block Diagram

Figure 7-5 Analog Comparator Block Diagram

7.3.4　Operational Procedure

Setup Procedure

To use the Analog Comparator block, use the following sequence:

1. Configure GPIO for use as analog input by setting type to input.
2. Enable the peripheral clock (SYSCLK->APBCLK.ACMP_EN)
3. Reset the Comparator block (SYS->IPRSTC2.ACMP_RST, Table 5-4)
4. If using VMID ensure that VMID block is powered up (Section 7.4.4)
5. Select comparison sources with CMPnCR and CMPSEL.
6. Enable comparators and appropriate interrupts with CMPnCR.
7. Enables system interrupt if appropriate (e.g. NVIC_EnableIRQ(ACMP_IRQn);)

Interrupt Sources

The comparator generates an output COn (n=0,1) which is reported in CMPSR register. If CMPnCR.IEbit is set then a state change on the comparator output COn will cause comparator flag CMPFn to go high and the comparator interrupt is requested. Software can write a one to CMPF$n$to clear flag and interrupt request.
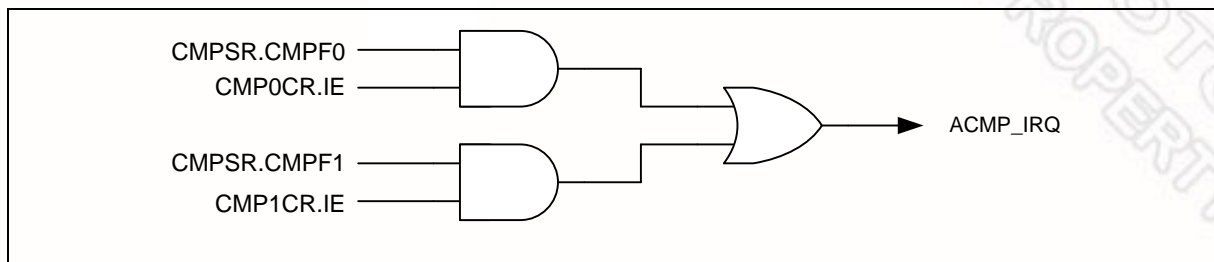
Figure 7-6 Comparator Controller Interrupt Sources

7.3.5　Register Map

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

Register	Offset	R/W	Description	Reset Value	Reference
ACMP_BA = 0x400D_0000					
CMPCR[0]	ACMP_BA+0x00	R/W	CMP0 Control Register	0x0000_0000	Table 7-18
CMPCR[1]	ACMP_BA+0x04	R/W	CMP1 Control Register	0x0000_0000	Table 7-19
CMPSR	ACMP_BA+0x08	R/W	CMP Status Register	0x0000_00XX	Table 7-20
CMPSEL	ACMP_BA+0x0C	R/W	CMP Select Register	0x0000_0000	Table 7-21

7.3.6 Register Description

<u>**Comparator 0 Control Register (CMPCR[0])**</u>

Register	Offset	R/W	Description	Reset Value
CMPCR[0]	CMP_BA+0x00	R/W	Comparator 0 Control Register	0x0000_0000

7	6	5	4	3	2	1	0
-	-	-	CMPCN	-		CMPIE	CMPEN

Table 7-18 Comparator 0 Control Register (CMPCR[0], address 0x400D_0000).

Bits	Descriptions	
[4]	**CMPCN**	**Comparator0 negative input select** 1 = VMID reference voltage = VCCA/2 0 = VBG, Bandgap reference voltage = 1.2V
[1]	**CMPIE**	**CMP0 Interrupt Enable** 1 = Enable CMP0 interrupt function 0 = Disable CMP0interrupt function
[0]	**CMPEN**	**Comparator Enable** 1 = Enable 0 = Disable

Comparator 1 Control Register (CMPCR[1])

Register	Offset	R/W	Description	Reset Value
CMPCR[1]	CMP_BA+0x04	R/W	Comparator 1 Control Register	0x0000_0000

7	6	5	4	3	2	1	0
-	-	-	CMPCN	-		CMPIE	CMPEN

Table 7-19 Comparator 1 Control Register (CMPCR[1], address 0x400D_0004).

Bits	Descriptions	
[4]	**CMPCN**	**Comparator1 negative input select** 1 = VBG, Bandgap reference voltage = 1.2V 0 = GPIOB[7]
[1]	**CMPIE**	**CMP1 Interrupt Enable** 1 = Enable CMP1 interrupt function 0 = Disable CMP1interrupt function
[0]	**CMPEN**	**Comparator Enable** 1 = Enable 0 = Disable

CMP Status Register (CMPSR)

Register	Offset	R/W	Description	Reset Value
CMPSR	CMP_BA+0x08	R/W	CMP Status Register	undefined

7	6	5	4	3	2	1	0
Reserved				CO1	CO0	CMPF1	CMPF0

Table 7-20 CMP Status Register (CMPSR, address 0x400D_0008).

Bits	Descriptions	
[3]	CO1	**Comparator1 Output** Synchronized to the APB clock to allow reading by software. Cleared when the comparator is disabled (CMP1EN = 0).
[2]	CO0	**Comparator0 Output** Synchronized to the APB clock to allow reading by software. Cleared when the comparator is disabled (CMP0EN = 0).
[1]	CMPF1	**Compare 1 Flag** This bit is set by hardware whenever the comparator output changes state. This bit will cause a hardware interrupt if enabled. **This bit is cleared by writing 1 to itself.**
[0]	CMPF0	**Compare 0 Flag** This bit is set by hardware whenever the comparator output changes state. This bit will cause a hardware interrupt if enabled. **This bit is cleared by writing 1 to itself.**

CMP Select Register (CMPSEL)

Register	Offset	R/W	Description	Reset Value
CMPSEL	CMP_BA+0x0C	R/W	CMP Select Register	0x0000_0000

Table 7-21 CMP Select Register (CMPSEL, address 0x400D_000C).

Bits	Descriptions	
[2:0]	CMPSEL	**Comparator0 GPIO Selection** GPIOB[CMPSEL] is the active analog GPIO input selected to Comparator 0 positive input.

7.4 Analog Functional Blocks

7.4.1 Overview

The ISD93xx contains a variety of analog functional blocks that facilitate audio processing, enable analog GPIO functions (current source, relaxation oscillator, and comparator), adjust and measure internal oscillator and provide voltage regulation. These blocks are controlled by registers in the analog block address space. This section describes these functions and registers.

7.4.2 Features

- VMID reference voltage generation.

- Current source generation for AGPIO (Analog enabled GPIO).

- LDO control for GPIOA[7:0] power domain and external device use.

- Microphone Bias generator.

- Analog Multiplexor.

- Programmable Gain Amplifier (PGA).

- OSC48M Frequency Control.

- CapSense Relaxation Oscillator.

- Oscillator Frequency Measurement block.

7.4.3 Register Map

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

Register	Offset	R/W	Description	Reset Value	Reference
ANA_BA = 0x4008_0000					
VMID	ANA_BA+0x00	R/W	VMID Reference Control Register	0x0000_0007	Table 7-22
ISRC	ANA_BA+0x08	R/W	Current Source Control Register	0x0000_0000	Table 7-23
LDOSET	ANA_BA+0x20	R/W	LDO Voltage Select Register	0x0000_0000	Table 7-24
LDOPD	ANA_BA+0x24	R/W	LDO Power Down Register	0x0000_0001	Table 7-25
MICB*SEL*	ANA_BA+0x28	R/W	Microphone Bias Select Register	0x0000_0000	Table 7-26
MICBEN	ANA_BA+0x2C	R/W	Microphone Bias Enable Register	0x0000_0000	Table7-27
VSET	ANA_BA+0x38	R/W	Logic Power Control Register	0x0000_0000	Do Not Modify
TMPOWER	ANA_BA+0x3C	R/W	Power Test Mode Register	0x0000_0000	Do Not Modify
TMANALOG	ANA_BA+0x40	R/W	Analog Test Mode Register	0x0000_0000	Do Not Modify
AMUX	ANA_BA+0x50	R/W	Analog Multiplexer Control Register	0x0000_0000	Table 7-28
PGAEN	ANA_BA+0x60	R/W	PGA Enable Register	0x0000_0000	Table 7-31
SIGCTRL	ANA_BA+0x64	R/W	Signal Path Control Register	0x0000_0000	Table 7-32
PGASEL	ANA_BA+0x68	R/W	PGA Gain Select Register	0x0000_0010	Table 7-33
CAPS_CTRL	ANA_BA+0x8C	R/W	CapSense Relaxation Oscillator Control	0x0000_0000	Table 7-34
CAPS_CNT	ANA_BA+0x90	R	CapSense Relaxation Oscillator Count	0x0000_0000	Table 7-35

FREQ_CTRL	ANA_BA+0x94	R/W	Oscillator Frequency Measurement Control	0x0000_0001	Table 7-37
FREQ_CNT	ANA_BA+0x98	R	Oscillator Frequency Measurement Count	0x0000_0000	Table 7-38

7.4.4 VMID Reference Voltage Generation

The analog path and blocks require a low noise, mid-rail, Voltage reference for operation, the VMID generation block provides this. Control of this block allows user to power down the block, select its power down condition and control over the reference impedance. The block consists of a switchable resistive divider connected to the device VMID pin. A 4.7µF capacitor should be placed on this pin and returned to analog ground (VSSA) as shown in Figure 7-7.

Before using the ADC, PGA or other analog blocks, the VMID reference needs to be enabled. A low impedance option allows fast charging of the external noise de-coupling capacitor, while a higher impedance options provides lower power consumption. A pulldown option allows the reference to be discharged when off.
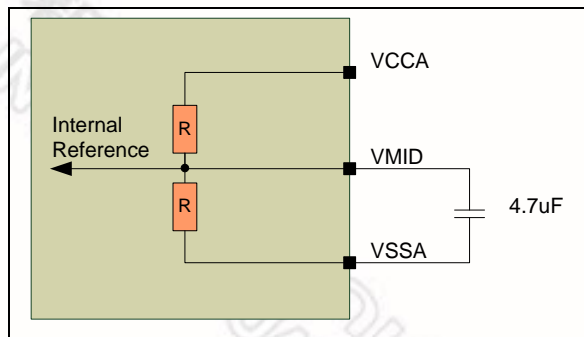

Figure 7-7VMID Reference Generation

VMID Control Register (ANA->VMID)

Register	Offset	R/W	Description	Reset Value
VMID	ANA_BA+0x00	R/W	VMID Control Register.	0x0000_0007

7	6	5	4	3	2	1	0
Reserved					PDHIRES	PDLORES	PULLDOWN

Table 7-22 VMID Control Register (VMID, address 0x4008_0000).

Bits	Descriptions	
[2]	PDHIRES	**Power Down High (360kΩ) Resistance Reference.** 1: The High Resistance reference is disconnected from VMID. Default power down and reset condition. 0: Connect the High Resistance reference to VMID. Use this setting for minimum power consumption.
[1]	PDLORES	**Power Down Low (4.8kΩ) Resistance Reference.** 1: The Low Resistance reference is disconnected from VMID. Default power down and reset condition. 0: Connect the Low Resistance reference to VMID. Use this setting for fast power up of VMID. Can be turned off after 50ms to save power.
[0]	PULLDOWN	**VMID Pulldown.** 1: Pull VMID pin to ground. Default power down and reset condition. 0: Release VMID pin for reference operation.

7.4.5 GPIO Current Source Generation

The GPIOB port consists of analog enabled GPIO. One of the features of these pins is the ability to route a current source to the pin. This is useful for a variety of purposes such as providing a current load to a sensor such as a photo-transistor or CDS cell. It can also be used to do capacitive sensing in combination with the relaxation oscillator control circuit.

The current generation block consists of a programmable current source controlled by ISRC.VAL and individual switches to each of the GPIOB pins as shown in Figure 7-8. Power control for this block is merged with the analog comparator, this block must be enabled to use current source (ACMP->CMP0CR.EN=1).

Analog peripheral clock must be enabled before register can be written. At least one of the analog comparators must be enabled to enable current source.
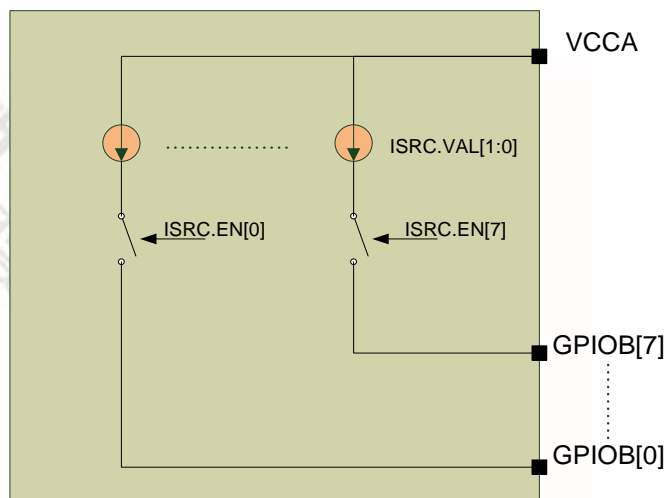

Figure 7-8 GPIOB Current Source Generation

Current Source Control Register (ANA->ISRC)

Register	Offset	R/W	Description	Reset Value
ISRC	ANA_BA+0x08	R/W	GPIO Current Source Control Register.	0x0000_0000

15	14	13	12	11	10	9	8
Reserved						VAL	
7	6	5	4	3	2	1	0
EN[7:0]							

Table 7-23 GPIO Current Source Control Register (ISRC, address 0x4008_0008).

Bits	Descriptions	
[31:10]	Reserved	Reserved
[7:0]	**EN[x]**	**Enable Current Source to GPIOB[x]** Individually enable current source to GPIOB pins. Each GPIOB pin has a separate current source. 1: Enable current source to pin GPIOB[x], 0: Disable
[9:8]	**VAL**	**Current Source Value.** Select master current for source generation 3=5μA. , 2= 2.5μA, 1=1μA, 0= 0.5μA

7.4.6　LDO Power Domain Control

The ISD93xx provides a Low Dropout Regulator (LDO) that provides power to the I/O domain of GPIOA[7:0]. Using this regulator device can operate from a 5V supply rail and generate a 2.4-3.3V regulated supply to operate the GPIOA[7:0] domain and external loads up to 30mA. The supply pin for the LDO is the VCCLDO pin which should be connected to VCCD. If the LDO is not used, both VCCLDO and VD33 should be tied to VCCD. Upon POR or reset the default condition of the LDO is off, meaning supply will be high impedance. Software must configure the LDO before GPIOA[7:0] is usable (unless VD33=VCCD).
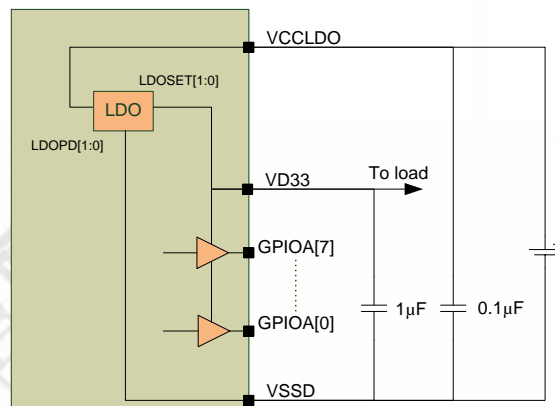

Figure 7-9 LDO Power Domain

LDO Voltage Control Register (ANA->LDOSET)

Register	Offset	R/W	Description	Reset Value
LDOSET	ANA_BA+0x20	R/W	LDO Voltage Control Register	0x0000_0000

Table 7-24 LDO Voltage Control Register (LDOSET, address 0x4008_0020).

Bits	Descriptions	
[31:2]	Reserved	Reserved
[1:0]	LDOSET	**Select LDO Output Voltage.** Note that maximum I/O pad operation speed only specified for voltage >2.4V. 3= 3.3V , 2= 2.4V, 1=1.8V, 0= 3.0V

LDO Power Down Register (ANA->LDOPD)

Register	Offset	R/W	Description	Reset Value
LDOPD	ANA_BA+0x24	R/W	LDO Power Down Control Register	0x0000_0001

7	6	5	4	3	2	1	0
Reserved						DISCH	PD

Table 7-25 LDO Power Down Control Register (LDOPD, address 0x4008_0024).

Bits	Descriptions	
[1]	DISCH	1: Switch discharge resistor to VD33. 0: No load on VD33
[0]	PD	**Power Down LDO.** When powered down no current delivered to VD33. 1= Power Down. 0: Enable LDO

7.4.7 Microphone Bias Generator

The ISD93xx provides a microphone bias generator (MICBIAS) for improved recording quality. The MICBIAS can provide a maximum current of 1mA with a -60dB power supply rejection. The MICBIAS output voltage can be configured with MICBSEL[1:0] to select bias voltages from 50% to90% of the VCCA supply voltage (see description below). The user should consider the microphone manufacturers specification in deciding on the optimum MICBIAS voltage to use. Generally, a microphone will require a current of 0.1mA to a maximum 0.5mA and a voltage of 1V to 3V across it.

Referring to the application diagram of Figure 7-11, external resistor R_1 and R_2 values are selected to limit the current to a maximum that can be provided by MICBIAS; 1mA. On the ISD93XX, the minimum total resistance $(R_1 + R_2)$ is 4Kohms. MICBIAS output voltage should be such that the following condition is met:

$$V_{MICBIAS} > V_S + (R_1 + R_2) \times I_{MIC}$$

where V_S is the desired voltage across the microphone from specification and I_{MIC} is the current through the microphone (0.1-0.5mA)

From Figure 7-11, MIC_IN1 and MIC_IN2 are AC coupled to the ISD93XX MIC+ and MIC- respectively for differential inputs. In single-ended operation, MIC_IN1 should go to MIC- of the ISD93XX. C_1 and C_2 are AC coupling capacitors. In single-ended application, R_2 can be removed and R_1 increased to at least 4Kohms. For improved performance, it is recommended to keep R_2toprovide additional rejection from ground noise.
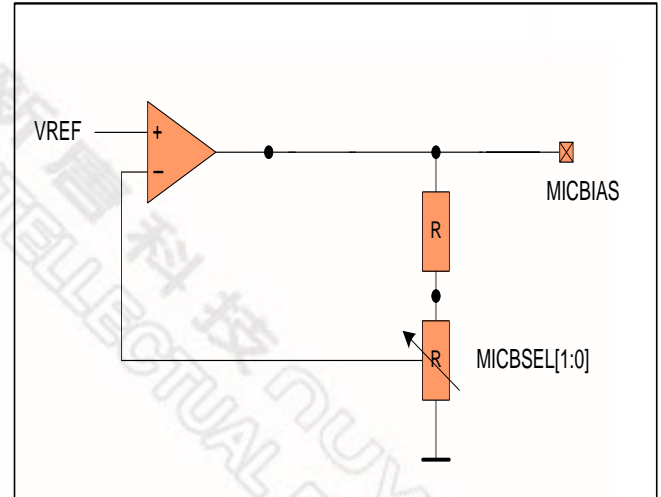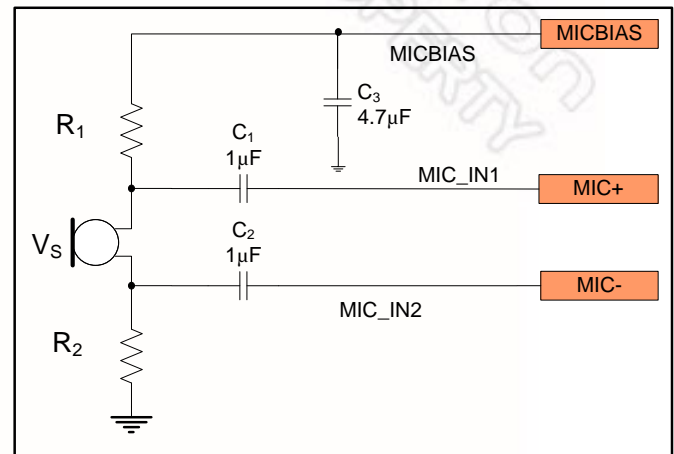

Figure 7-10 MICBIAS Block Diagram

Figure 7-11 MICBIAS Application Diagram

Microphone Bias Select (ANA->MICBSEL)

Register	Offset	R/W	Description	Reset Value
MICBSEL	ANA_BA+0x28	R/W	Microphone Bias Selection Register	0x0000_0000

Table 7-26 Microphone Bias Selection Register (MICBSEL, address 0x4008_0028).

Bits	Descriptions		
[31:3]			
[2]	**REF**	Select reference source for MICBIAS generator. VMID provides superior noise performance for MICBIAS generation and should be used unless fixed voltage is absolutely necessary, then noise performance can be sacrificed and bandgap voltage used as reference. 0: VMID = VCCA/2 is reference source. 1: VBG (bandgap voltage reference) is reference source.	
[1:0]	**SEL**	Select Microphone Bias Voltage.	
		REF=0	REF=1	
		0: 90% VCCA 1: 65% VCCA 2: 75% VCCA 3: 50% VCCA	0: 2.4V 1: 1.7V 2: 2.0V 3: 1.3V	

Microphone Bias Enable Register (ANA->MICBEN)

Register	Offset	R/W	Description	Reset Value
MICBEN	ANA_BA+0x2C	R/W	Microphone Bias Enable Register	0x0000_0000

7	6	5	4	3	2	1	0
Reserved							EN

Table7-27Microphone Bias Enable Register (MICBEN, address 0x4008_002C)

Bits	Descriptions	
[0]	**MICBEN**	Enable Microphone Bias Generator. 1: Enabled. 0: Powered Down.

7.4.8 Analog Multiplexer

The ISD93xx provides an analog multiplexer (AMUX) which allows the PGA input to be switched from the dedicated MICP/MICN analog inputs to any of the analog enabled GPIO (GPIOB[7:0]).   The negative input of the PGA connects to GPIOB[7:0], while the positive PGA input connects to the odd numbered GPIOB[7:1].   Figure 7-12 shows the multiplexer block diagram and Table 7-28 shows the multiplexer control signals.
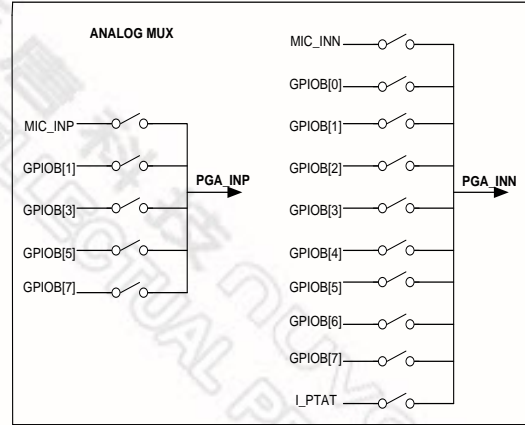

Figure 7-12 Analog Multiplexer Block Diagram

Analog Multiplexer Control Register (ANA->AMUX)

Register	Offset	R/W	Description	Reset Value
AMUX	ANA_BA+0x50	R/W	Analog Multiplexer Control Register	0x0000_0000

15	14	13	12	11	10	9	8
Reserved	EN	MIC_SEL	TEMP_SEL	MUXP_SEL			

7	6	5	4	3	2	1	0
MUXN_SEL							

Table 7-28Analog Multiplexer Control Register (AMUX, address 0x4008_0050).

Bits	Symbol	Descriptions
[31:15]	Reserved	Reserved
[14]	EN	Enable the analog multiplexer. EN=0: All channels disabled EN=1: Selection determined by register setting.
[13]	MIC_SEL	Select MICP/MICN to PGA inputs
[12]	TEMP_SEL	Select PTAT current, I_PTAT, to PGA_INN, negative input to PGA, for temperature measurement.
[11:8]	MUXP_SEL[3:0]	Selects connection of GPIOB[7,5,3,1] to PGA_INP, positive input of PGA. 1000b: GPIOB[7] connected to PGA_INP

		0100b: GPIOB[5] connected to PGA_INP
		0010b: GPIOB[3] connected to PGA_INP
		0001b: GPIOB[1] connected to PGA_INP
[7:0]	MUXN_SEL[7:0]	Selects connection of GPIOB[7:0] to PGA_INN, negative input of PGA.
		If MUXN_SEL[n]=1 then GPIOB[n] is connected to PGA_INN.

Temperature Sensor Measurement

In addition, the multiplexer can route a PTC (positive temperature coefficient) current, PTAT current, to the ADC to perform temperature measurements. To configure the signal path to do temperature measurement, configure the ADC path as follows:

1) Enable the multiplexer, PGA, IPBOOST, and sigma-delta modulator. (See Section 7.4.9, Section7.1).

2) Have the multiplexer select I_PTAT current as input and choose VBG (bandgap voltage) as reference (REF_SEL).

3) Set the 6-bit PGA_GAIN[5:0] gain value to hex 0x17 and choose 0dB gain setting for IPBOOST gain block.

4) The temperature can be inferred by the information given in Table 7-29 and equation below.

$$T\ (°C) = 27 + (ADC\_VAL-0x42EA)/50. \quad \text{(Equation 7-1)}$$

The settings corresponding to this configuration are:

ANA->SIGCTRL=0x1E, ANA->PGAEN=0x07, ANA->AMUX=0x5000, ANA->PGA_GAIN=0x17

Table 7-29 Temperature Sensor Measurement.

Parameter	Specification (Reference)				Test Condition
	Min.	Typ.	Max.	Unit	
Temperature Sensor Output		0x42EA		Code	At 27° C
Temperature Sensor Delta Coefficient (number of bits per degree ^{o}C)**		50		LSB/^{o}C	Relative to 27°C

**LSB is the least significant bit of a 16-bit ADC with a defined full-scale RMS input voltage of 0.77V

7.4.9 Programmable Gain Amplifier

The ISD93xx provides a Programmable Gain Amplifier (PGA) as the front-end to the ADC to allow the adjustment of signal path gain. It is used in conjunction with the ALC block to provide automatic level control of incoming audio signals. Figure 7-13 shows the signal path diagram. The PGA provides a gain from -12dB to 35.25dB in increments of 0.75dB steps using a 6-bit control, PGA_GAIN[5:0]. The gain is monotonically increasing with 0x00 for lowest gain (-12dB) and 0x3f for the maximum gain (35.25dB). The signal path is enabled by powering up the gain elements (PU_PGA, PU_IPBOOST). The PGA and IPBOOST blocks can be muted with the SIGCTRL register. Input to the PGA can be either differential or single-ended on the PGA_INN input. The Analog MUX controls connection of the signal path to external pins. PGA input impedance varies based on the gain setting. Table 7-30 shows a table of input impedance for different gain setting.

The IPBOOST block can provide 0dB or 26dB of gain to provide a maximum gain of 61dB in the signal path. Front-end anti-alias filtering for the sigma-delta ADC is also provided by PGA/IPBOOST blocks with an attenuation of -45dB at 6MHz frequency. The signal path defaults to have VCCA/2 as the reference voltage.
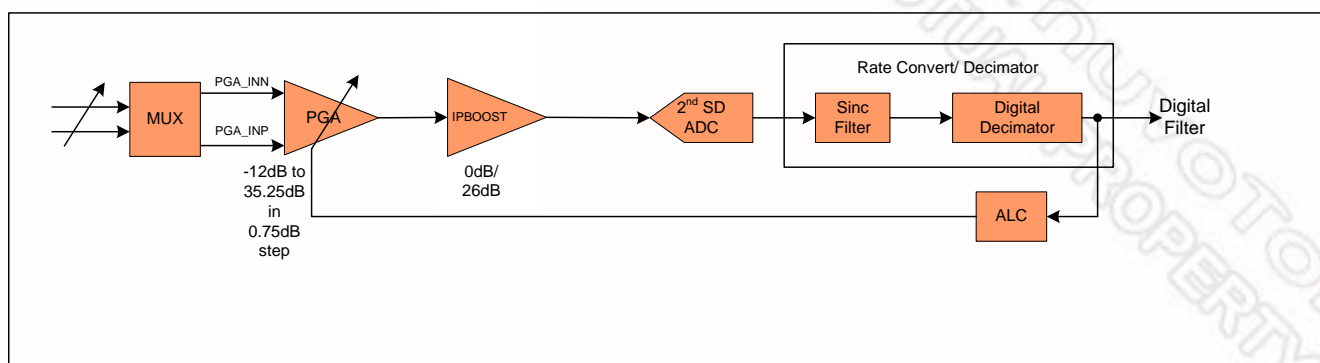

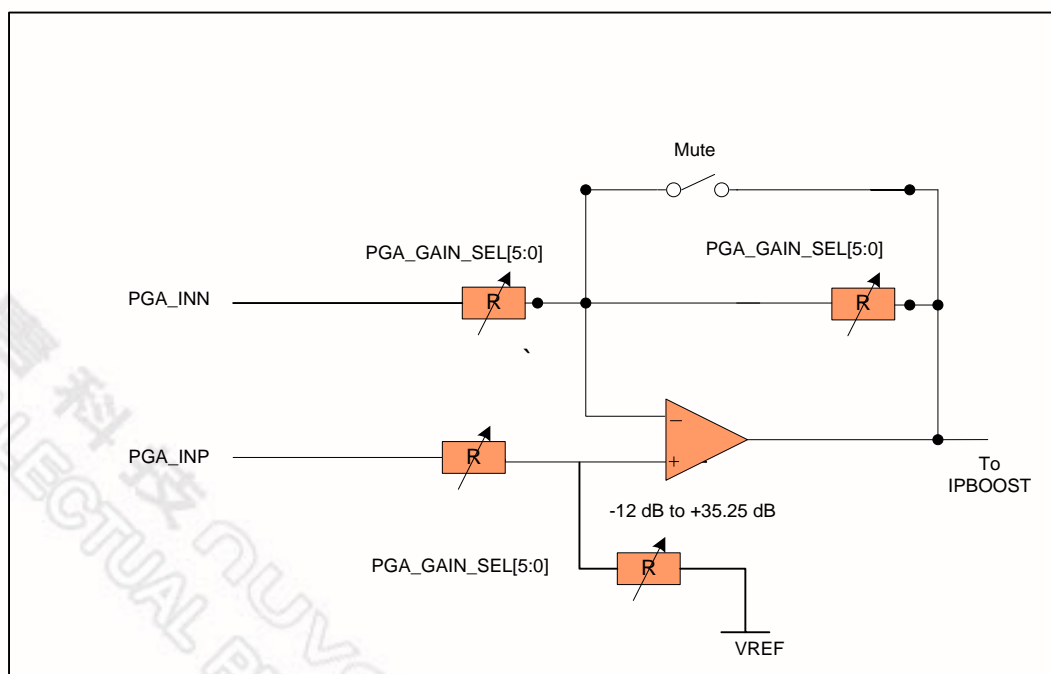Figure 7-13PGA Signal Path Block Diagram

Figure 7-14 PGA Structure

Table 7-30 PGA Input Impedance Variation with Gain Setting

Gain (dB)	-12	-9	-6	-3	0	3	6	9	12	18	30	35.2
MICN Impedance (kΩ)	75	69	63	55	47	35	31	25	19	11	2.9	1.6
MICPImpedance (kΩ)	94	94	94	94	94	94	94	94	94	94	94	94

PGA Enable Register (ANA->PGAEN)

Register	Offset	R/W	Description	Reset Value
PGAEN	ANA_BA+0x60	R/W	PGA Enable and Control Register	0x0000_0000

7	6	5	4	3	2	1	0
Reserved				BOOSTGAIN	PU_BOOST	PU_PGA	REF_SEL

Table 7-31PGA Enable and Control Register (PGAEN, address 0x4008_0060)

Bits	Descriptions	
[3]	BOOSTGAIN	Boost stage gain setting. 1: Gain = 26dB 0: Gain = 0dB.
[2]	PU_BOOST	Power Up control for boost stage amplifier. This amplifier must be powered up for signal path operation. 1: Power up. 0: Power Down.
[1]	PU_PGA	Power Up control for PGA amplifier. This amplifier must be powered up for signal path operation. 1: Power up. 0: Power down.
[0]	REF_SEL	**Select Reference for Analog path.** Signal path is normally referenced to VMID (VCCA/2). To use an absolute reference this can be set to VBG=1.2V. 1: Select Bandgap voltage as analog ground reference. 0: Select VMID voltage as analog ground reference.

Signal Path Control Register (ANA->SIGCTRL)

Register	Offset	R/W	Description	Reset Value
SIGCTRL	ANA_BA+0x64	R/W	Signal Path Control Register	0x0000_0000

7	6	5	4	3	2	1	0
Reserved	MUTE_IPBOOST	MUTE_PGA	PU_MOD	PU_IBGEN	PU_BUFADC	PU_BUFPGA	PU_ZCD

Table 7-32Signal Path Control Register (SIGCTRL, address 0x4008_0064)

Bits	Descriptions	
[6]	**MUTE_IPBOOST**	Boost stage mute control 1: Signal Muted. 0: Normal.
[5]	**MUTE_PGA**	PGA Mute control. 1: Signal Muted. 0: Normal.
[4]	**PU_MOD**	Power up ADC ΣΔ Modulator. This block must be powered up for ADC operation. 1: Power up. 0: Power down.
[3]	**PU_IBGEN**	Power up control for current bias generation. This block must be powered up for signal path operation. 1: Power up. 0: Power down.
[2]	**PU_BUFADC**	Power up control for ADC reference buffer. This block must be powered up for signal path operation. 1: Power up. 0: Power down.
[1]	**PU_BUFPGA**	Power up control for PGA reference buffer. This block must be powered up for signal path operation. 1: Power up. 0: Power down.
[0]	**PU_ZCD**	Power up and enable control for Zero Cross Detect Comparator. When enabled PGA gain settings will only be updated when ADC input signal crosses zero signal threshold. To operate ZCD the ALC peripheral clock (SYSCLK.BIQALC_EN) must also be enabled and BIQ->BIQ_CTRL.RSTn=1 to allow ZCD clocks to be generated. 1: Power up and enable zero cross detection. 0: Power down.

PGA GAIN Control Register (ANA->PGA_GAIN)

Register	Offset	R/W	Description	Reset Value
GAIN	ANA_BA+0x68	R/W	PGA Gain Select Register	0x0000_0010

31	30	29	28	27	26	25	24
Reserved							

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		GAIN_READ					

7	6	5	4	3	2	1	0
Reserved		GAIN					

Table 7-33 PGA Gain Control Register (PGA_GAIN, address 0x4008_0068)

Bits		Descriptions
[13:8]	GAIN_READ	Read Only. Current PGA Gain, may be different from GAIN register when AGC is enabled and is controlling the PGA gain.
[5:0]	GAIN	Selects the PGA gain setting from -12dB to 35.25dB in 0.75dB step size. 0x00 is lowest gain setting at -12dB and 0x3F is largest gain at 35.25dB.

7.4.10 CapSense Relaxation Oscillator/Counter

The ISD93xx provides a functional unit that is used with analog GPIO functions to form a relaxation oscillator. The major application of this function is to measure the capacitive load on a GPIO pin. This measurement allows the user to implement a capacitive touch sensing scheme. With appropriate touch sensor design, the capacitance of the sensor will change appreciably in the presence of a finger, and the CapSense Relaxation Oscillator can measure this.

This block us used in conjunction with the analog comparator block and current source block to form a relaxation oscillator and counter circuit that can sense capacitance changes. A block diagram of the system is shown in Figure 7-15.
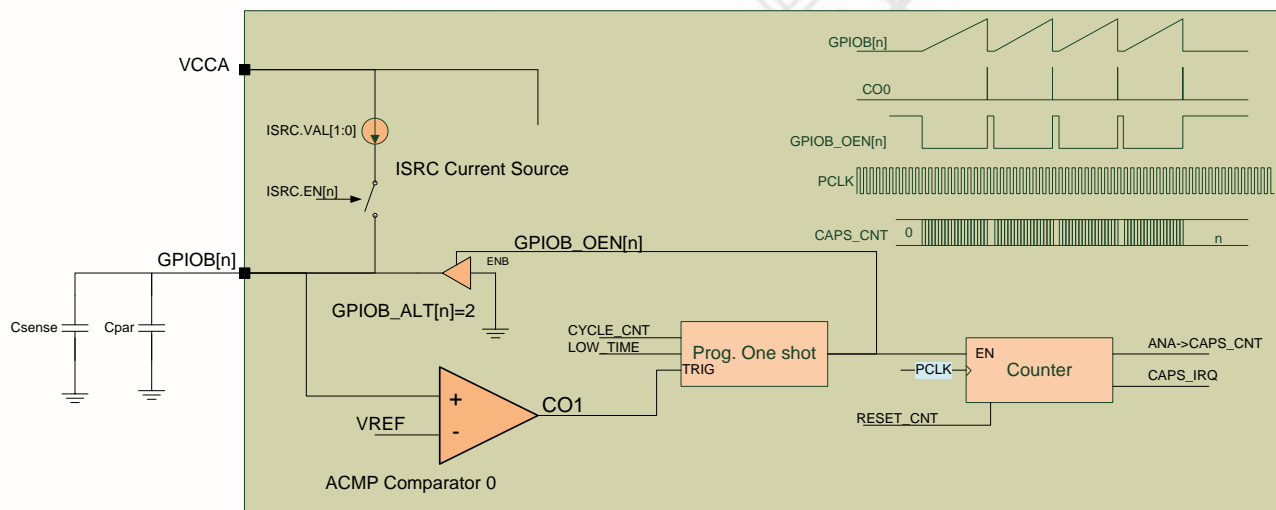

Figure 7-15 CapSense Function Block Diagram

7.4.10.1 *Functional Description*

The principle behind the operation of this block is that a certain capacitance is present on one of the analog enabled GPIO (GPIOB[7:0]). This capacitance consists of a certain parasitic capacitance C_{par} and the capacitor that is to be sensed C_{sense}. The GPIO is configured into the CapSense mode by setting SYS->GPB_ALT.GPBn = 2 and enabling a current source to this pin (ANA->ISRC.EN = 2^n). The Analog Comparator 0 is also setup to compare the voltage at the pin to a reference voltage (ACMP->CMPSEL = n, ACMP->CMP0CR.EN = 1).

In this configuration the circuit will charge the total capacitance with current ANA->ISRC.VAL = 0.5µA-5µA. When the voltage reaches the reference voltage (normally set to VBG=1.2V), the CapSense block will reset the GPIO pin to 0V. The circuit can be configured to do this 2^CYCLE_CNT times before generating an interrupt. While the capacitor is charging, a 24bit counter is also enabled such that the total charge time is recorded. After completion of 2^CYCLES_CNT cycles the software can read the ANA->CAP_CNT register to get a value proportional to the total capacitance on the pin. Once this is done, the count can be reset with RST_CNT and a new measurement started either on the same GPIO or selecting a different GPIO.

7.4.10.2 *Design Considerations*

Selecting parameters for CapSense measurement is a trade-off between speed and accuracy/noise immunity. The higher the current source setting, the faster the oscillation but lower the resolution. The higher the cycle count the slower the measurement but the higher the accuracy and noise immunity.

7.4.10.3   *Register Descriptions*

CapSense Control Register (ANA->CAPS_CTRL)

Register	Offset	R/W	Description	Reset Value
CAPS_CTRL	ANA_BA+0x8C	R/W	CapSense Control Register	0x0000_0000

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

Table 7-34CapSense Control Register (CAPS_CTRL, address 0x4008_008C).

Bits	Symbol	Description
[31]	EN	**Enable**. 1: Enable Block. 0: Disable/Reset block.
[30]	**INT_EN**	**Interrupt Enable**: 1: Enable CAPS_IRQ interrupt. 0: Disable/Reset CAPS_IRQ interrupt.
[29]	**RST_CNT**	Reset Count. 1: Set high to reset CAP_CNT. 0: Release/Activate CAP_CNT
[15:8]	**CLK_DIV**	**Reference clock divider**. Circuit can be used to generate a reference clock output of SDCLK/2/(CLK_DIV+1) instead of a CapSense reset signal.
[5]	**REF_CLK_MD**	**Reference Clock Mode.** 1: Circuit is in Reference clock generation mode. 0: CapSense Mode.
[4:2]	**CYCLE_CNT**	**Number of Relaxation Cycles.** Peripheral performs 2^(CYCLE_CNT) relaxation cycles before generating interrupt.
[1:0]	**LOW_TIME**	**Output Low Time.** Number of PCLK cycles to discharge external capacitor. 0=1cycle, 1=2cycles, 2=8cycles, 3=16cycles.,

CapSense Count Register (ANA->CAPS_CNT)

Register	Offset	R/W	Description	Reset Value
CAPS_CNT	ANA_BA+0x90	R	CapSense Count Register	0x0000_0000

Table 7-35CapSense Count Register (CAPS_CNT, address 0x4008_0090).

Bits	Symbol	Description
[23:0]	**CAPS_CNT**	**Counter read back value of CapSense Block.**

7.4.11 Oscillator Frequency Measurement and Control

The ISD93xx provides a functional unit that can be used to measure PCLK frequency given a reference frequency such as the 32.768kHz crystal or an I2S frame synchronization signal. This is simply a special purpose timer/counter as shown in Figure 7-16.
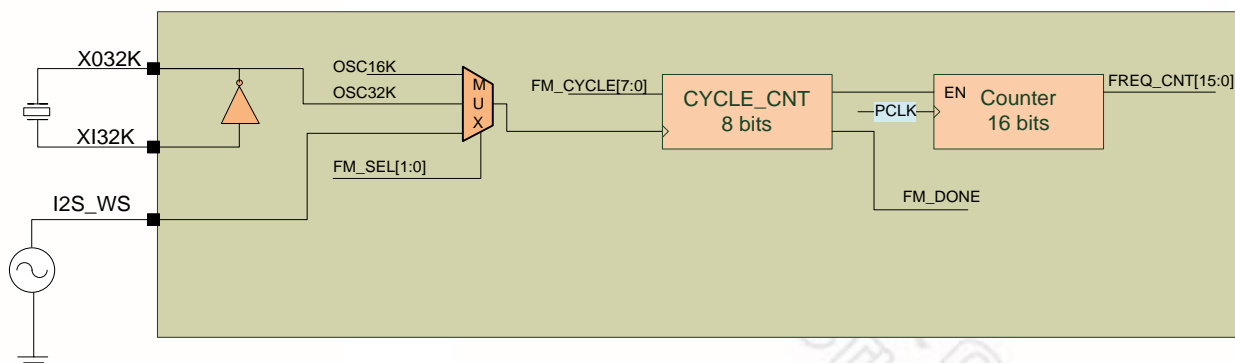

Figure 7-16 Oscillator Frequency Measurement Block Diagram

The block can be used to trim/measure the internal high frequency oscillator to the reference frequency of the 32.768kHz oscillator or an external reference frequency fed in on the I2S frame sync input. With this the internal clock can be set at arbitrary frequencies, other than those trimmed at manufacturing, or can be periodically trimmed to account for temperature variation. The block can also be used to measure the 16kHz oscillator frequency relative to the internal master oscillator.

An example of use would be to measure the internal oscillator with reference to the 32768Hz crystal. To do this:

SYSCLK->APBCLK.ANA_EN   = 1;   /* Turn on analog peripheral clock */

ANA->FREQ_CTRL.FM_SEL = 1;   // Select reference source as 32kHz XTAL input

ANA->FREQ_CTRL.FM_CYCLE = DRVOSC_NUM_CYCLES-1;

ANA->FREQ_CTRL.FM_GO = TRUE;

while((ANA->FREQ_CTRL.FM_DONE != 1) && (Timeout++ < 0x100000));

      if(Timeout >= 0x100000)

           return(E_DRVOSC_MEAS_TIMEOUT);

Freq = ANA->FREQ_CNT;

ANA->FREQ_CTRL.FM_GO = FALSE;

Freq = Freq*32768 /DRVOSC_NUM_CYCLES;

To adjust the oscillator the user can write to the OSCTRIM register (see Table 5-11). In addition, to obtain frequencies in between OSCTRIM trim settings a SUPERFINE function is available. The SUPERFINE function dithers the trim setting between the current setting and FINE trim settings above and below the current setting. An example of how the SUPERFINE trim register can adjust the measured oscillator frequency is shown in the figure below.
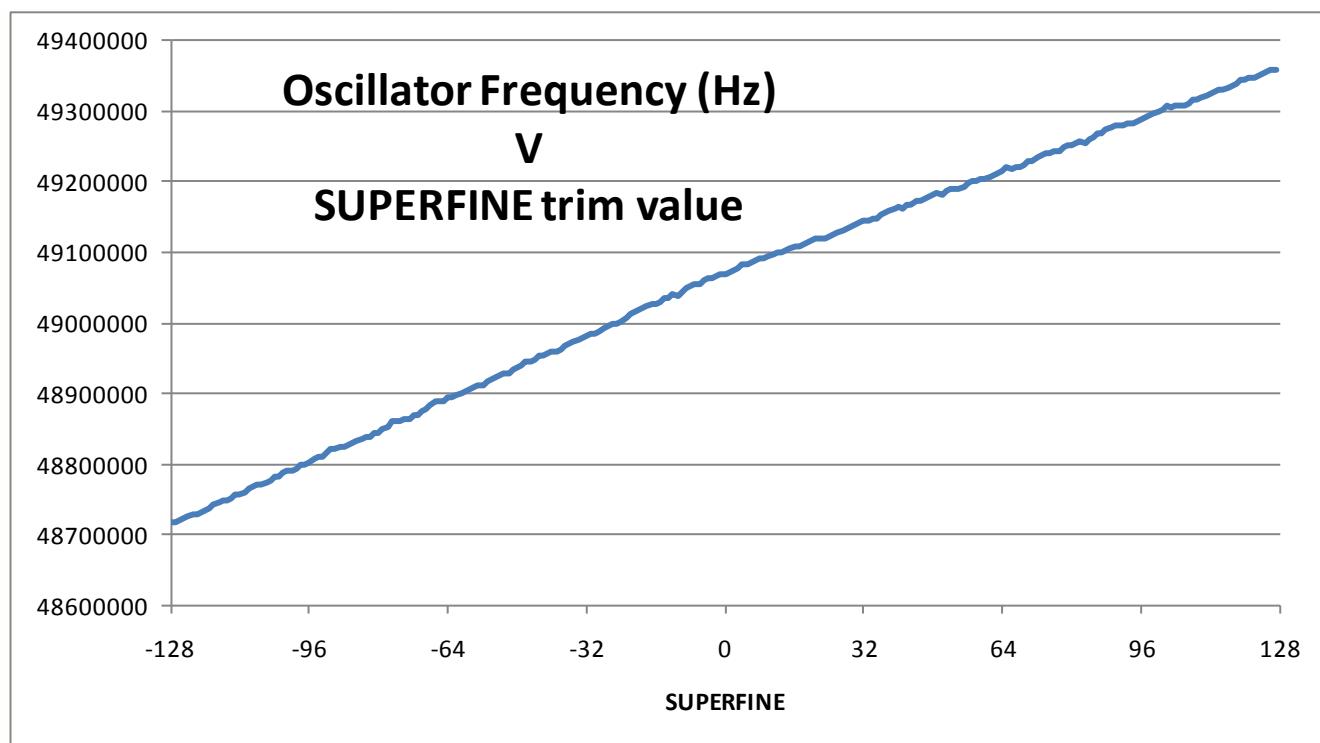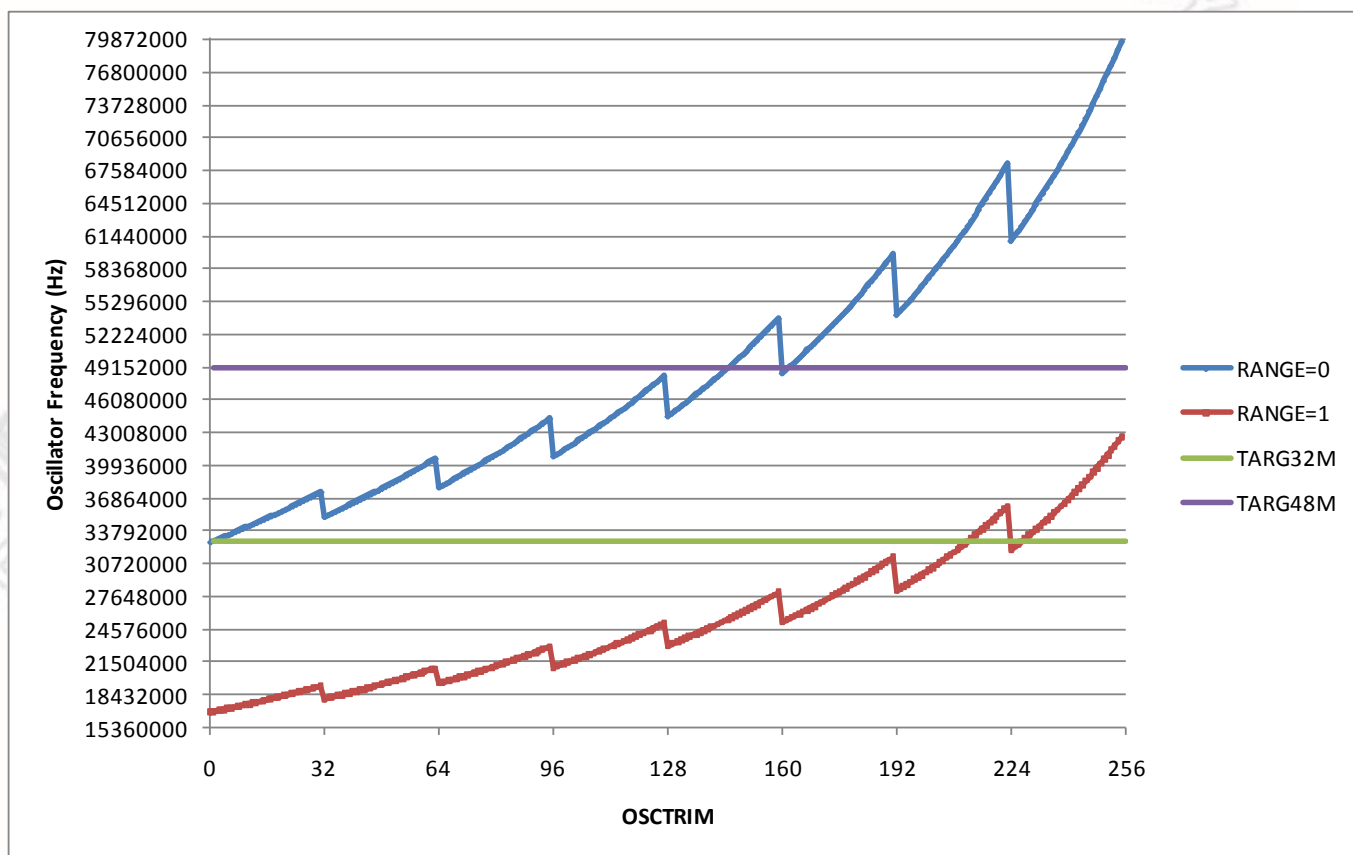
Figure 7-17 Example SUPERFINE Trim Frequency Adjustment.

Figure 7-18 Typical Oscillator Frequency versus OSCTRIM Setting.

Oscillator Trim Register (ANA->TRIM)

Register	Offset	R/W	Description	Reset Value
TRIM	ANA_BA+0x84	R/W	Oscillator Trim Register.	0x0000_XXXX

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

Table 7-36 Oscillator Trim Register (TRIM, address 0x4008_0084).

Bits	Symbol	Description
[23:16]	SUPERFINE	The **SUPERFINE** trim setting is an 8bit signed integer. It adjusts the master oscillator by dithering the FINE trim setting between the current setting and one setting above (values 1,127) or below (values -1, -128) the current trim setting. Each step effectively moves the frequency 1/128th of the full FINE trim step size.
[15:8]	COARSE	**COARSE.** Current COARSE range setting of the oscillator. Read Only
[7:0]	OSCTRIM	**Oscillator Trim**. Reads current oscillator trim setting. Read Only.

Frequency Measurement Control Register (ANA->FREQ_CTRL)

Register	Offset	R/W	Description	Reset Value
FREQ_CTRL	ANA_BA+0x94	R/W	Frequency Measurement Control Register.	0x0000_0001

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

Reserved				FM_DONE	FM_SEL

Table 7-37Frequency Measurement Control Register (FREQ_CTRL, address 0x4008_0094).

Bits	Symbol	Description
[31]	**GO**	**GO**. 1: Start Frequency Measurement. 0: Disable/Reset block.
[23:16]	**FM_CYCLE**	**Frequency Measurement Cycles**: Number of reference clock periods plus one to measure target clock (PCLK). For example if reference clock is OSC32K (T=30.5175µs), FM_CYCLE=7, then measurement period would be 30.5175*(7+1)=244.1µs.
[2]	**FM_DONE**	**Measurement Done.** 1: Measurement Complete. 0: Measurement Ongoing.
[1:0]	**FM_SEL**	**Reference clock source**. 00b: OSC16K, 01b: OSC32K (default), 1xb: I2S_WS – can be GPIOA[4,8,12] according to GPA_ALT register, configure I2S in SLAVE mode to enable.

Frequency Measurement Count (ANA->FREQ_CNT)

Register	Offset	R/W	Description	Reset Value
FREQ_CNT	ANA_BA+0x98	R/W	Frequency Measurement Count Register.	0x0000_0000

Table 7-38Frequency Measurement Count Register (FREQ_CNT, address 0x4008_0098).

Bits	Symbol	Description
[15:0]	**FREQ_CNT**	**Frequency Measurement Count:** When FM_DONE=1 and G0=1, this is number of PCLK periods counted for frequency measurement. The frequency will be PCLK = FREQ_CNT * Fref /(FM_CYCLE+1) Hz Maximum resolution of measurement is Fref /(FM_CYCLE+1)*2 Hz

7.5 Automatic Level Control (ALC)

7.5.1 Overview and Features

The ALC seeks to control the PGA gain such that the PGA output maintains a constant envelope. This helps to prevent clipping at the input of the sigma delta ADC while maximizing the full dynamic range of the ADC. The ALC monitors the output of the ADC bi-quad output when that filter is enabled in the ADC path, or the output of the SINC filter otherwise. The ADC output is fed into a peak detector, which updates the measured peak value whenever the absolute value of the input signal is higher than the current measured peak. The measured peak gradually decays to zero unless a new peak is detected, allowing for an accurate measurement of the signal envelope. Based on a comparison between the measured peak value and the target value, the ALC block adjusts the gain control, which is fed back to the PGA.
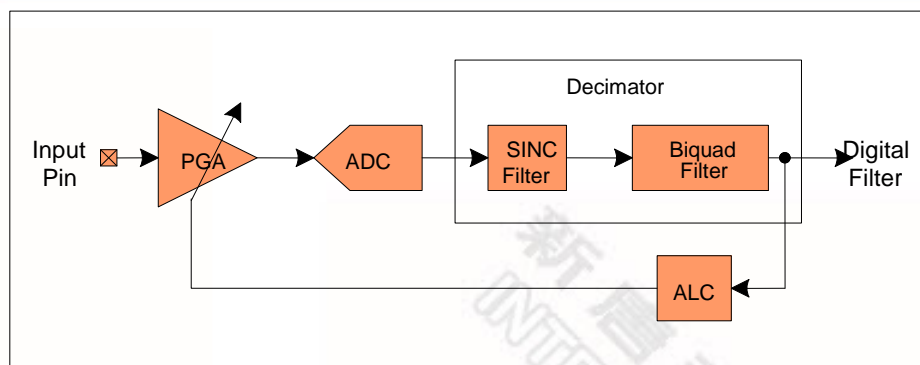
Figure 7-19 ALC Block Diagram

The ALC is enabled by setting ALCSEL. The ALC shares a clock source with the Biquad filter so SYSCLK->APBCLK.BIQALC_EN must be set to operate ALC. The ALC has two functional modes, which is set by ALCMODE.

- Normal mode (ALCMODE = LOW)
- Peak Limiter mode (ALCMODE = HIGH)

When the ALC is disabled, the input PGA returns to the PGA gain setting held in ANA→PGA_GAIN.GAIN. In order to have a smooth transition when disabling the ALC, the user may prefer to fetch the ALC trained gain setting from ANA→PGA_GAIN.GAIN_READ and write that value to ANA→PGA_GAIN.GAIN prior to disabling the ALC. An input gain update must be made by writing to PGASEL[5:0]. A digital peak detector monitors the input signal amplitude and compares it to a register defined threshold level ALCLVL[3:0].

Figure 7-20: ALC Response Graph

The registers listed in the following sections allow configuration of ALC operation with respect to:

- ALC target level
- Gain increment and decrement rates

- Minimum and maximum PGA gain values for ALC operating range
- Hold time before gain increments in response to input signal
- Inhibition of gain increment during noise inputs
- Limiter mode operation

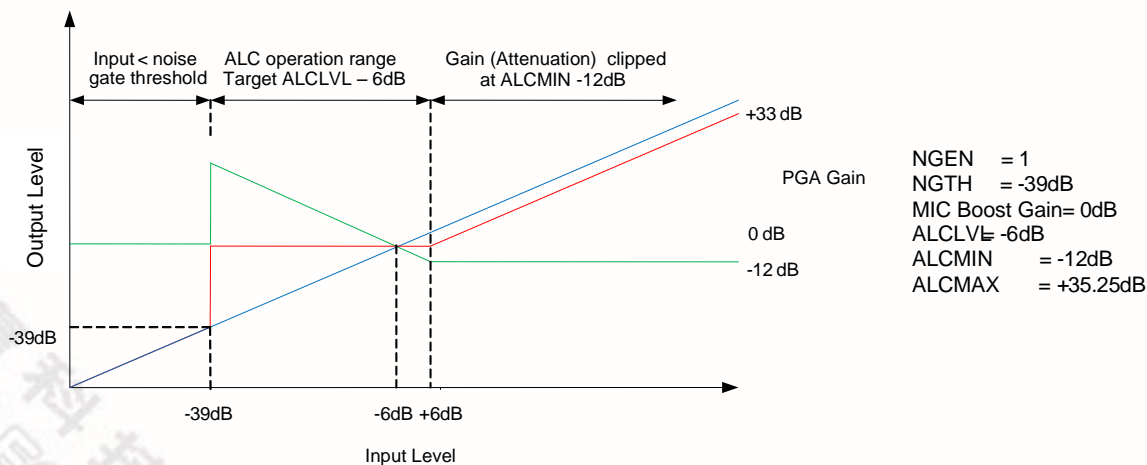The operating range of the ALC is set by ALCMAX and ALCMIN bits such that the PGA gain generated by the ALC is constrained to be between the programmed minimum and maximum levels. When the ALC is enabled, the PGA gain setting from PGASEL has no effect.

In Normal mode, the ALCMAX bits set the maximum level for the PGA but in the Limiter mode ALCMAX has no effect because the maximum level is set by the initial PGA gain setting upon enabling of the ALC.

7.5.1.1 *Normal Mode*

Normal mode is selected when ALCMODE is set LOW and the ALC is enabled by setting ALCSEL HIGH. This block adjusts the PGA gain setting up and down in response to the input level. A peak detector circuit measures the envelope of the input signal and compares it to the target level set by ALCLVL. The ALC increases the gain when the measured envelope is less than (target – 1.5dB) and decreases the gain when the measured envelope is greater than the target. The following waveform illustrates the behavior of the ALC.
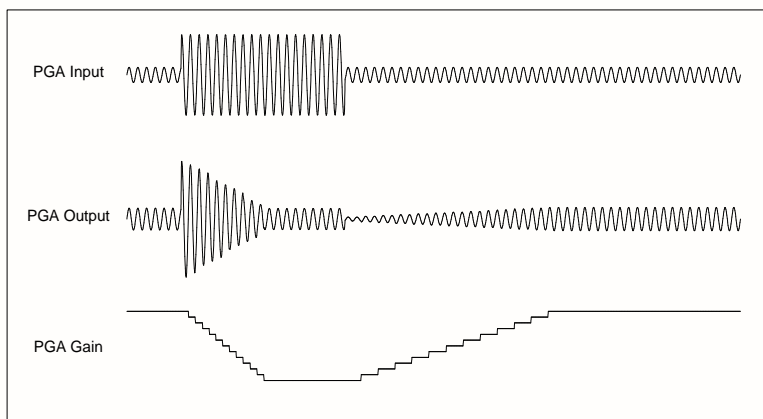
Figure 7-21: ALC Normal Mode Operation

7.5.1.2 *ALC Hold Time (Normal mode only)*

The hold parameter ALCHLD configures the time between detection of the input signal envelope being below the target range and the actual gain increase.

Input signals with different characteristics (e.g., voice vs. music) may require different settings for this parameter for optimal performance. Increasing the ALC hold time prevents the ALC from reacting too quickly to brief periods of silence such as those that may appear in music recordings; having a shorter hold time, on the other hand, may be useful in voice applications where a faster reaction time helps to adjust the volume setting for speakers with different volumes. The waveform below shows the operation of the ALCHLD parameter.
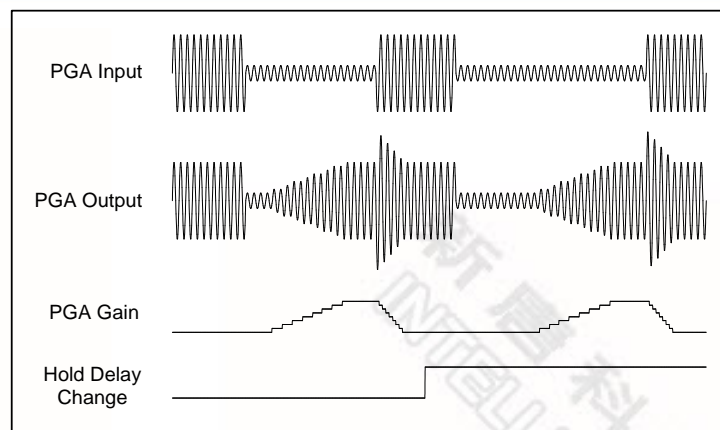
Figure 7-22: ALC Hold Time

7.5.1.3   *Peak Limiter Mode*

Peak Limiter mode is selected when ALCMODE is set to HIGH and the ALC is enabled by setting ALCSEL. In limiter mode, the PGA gain is constrained to be less than or equal to the gain setting at the time the limiter mode is enabled. In addition, attack and decay times are faster in limiter mode than in normal mode as indicated by the different lookup tables for these parameters for limiter mode. The following waveform illustrates the behavior of the ALC in Limiter mode in response to changes in various ALC parameters.
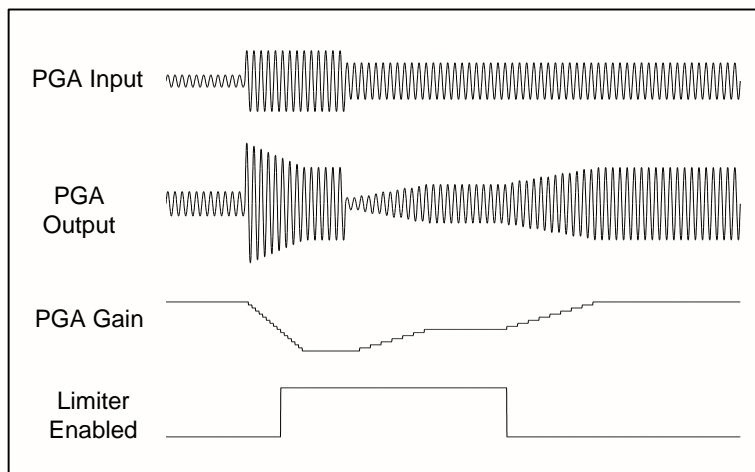
Figure 7-23: ALC Limiter Mode Operations

When the input signal exceeds 87.5% of full scale, the ALC block ramps down the PGA gain at the maximum attack rate (ALCATK=0000) regardless of the mode and attack rate settings until the ADC output level has been reduced below the threshold. This limits ADC clipping if there is a sudden increase in the input signal level.

7.5.1.4   *Attack Time*

When the absolute value of the ADC output exceeds the level set by the ALC threshold, ALCLVL, attack mode is initiated at a rate controlled by the attack rate register ALCATK. The peak detector in the ALC block loads the ADC output value when the absolute value of the ADC output exceeds the current measured peak; otherwise, the peak decays towards zero, until a new peak has been identified. This sequence is continuously running. If the peak is ever below the target threshold, then there is no gain decrease at the next attack timer time; if it is ever above the target-1.5dB, then there is no gain increase at the next decay timer time.

7.5.1.5 *Decay Times*

The decay time ALCDCY is the time constant used when the gain is increasing. In limiter mode, the time constants are faster than in ALC mode.

7.5.1.6 *Noise gate (normal mode only)*

A noise gate is used when there is no input signal or the noise level is below the noise gate threshold. The noise gate is enabled by setting NGEN to HIGH. It does not remove noise from the signal. The noise gate threshold NGTH is set to a desired level so when there is no signal or a very quiet signal (pause), which is composed mostly of noise, the ALC holds the gain constant instead of amplifying the signal towards the target threshold. The noise gate only operates in conjunction with the ALC (ALCSEL HIGH) and ONLY in Normal mode. The noise gate flag is asserted when

(Signal at ADC – PGA gain – MIC Boost gain) < NGTH (dB)

Levels at the extremes of the range may cause inappropriate operation, so care should be taken when setting up the function.
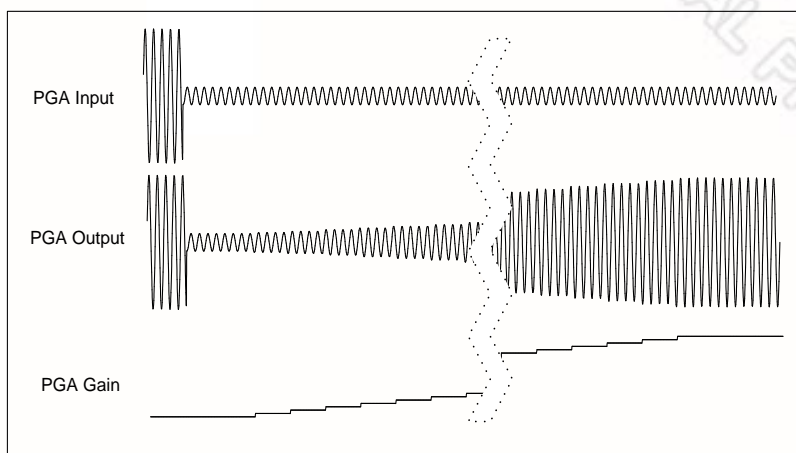
Figure 7-24: ALC Operation with Noise Gate disabled
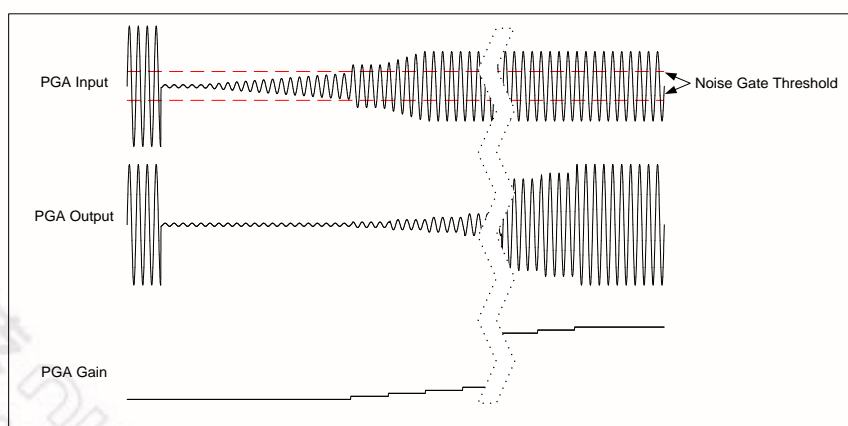
Figure 7-25: ALC Operation with Noise Gate Enabled

7.5.1.7 *Zero Crossing*

The PGA gain comes from either the ALC block when it is enabled or from the PGA gain register setting when the ALC is disabled. Zero crossing detection may be enabled to cause PGA gain changes to occur only at an input zero crossing. Enabling zero crossing detection limits clicks and pops that may occur if the gain changes while the input signal has a high volume.

There are two zero crossing detection enables:

- Register ALCZC – is only relevant when the ALC is enabled.
- Register ANA_EN_ZCD – is only relevant when the ALC is disabled.

If the zero crossing function is enabled (using either register), the zero cross timeout function may take effect. If the zero crossing flag does not change polarity within 0.25 seconds of a PGA gain update (either via ALC update or PGA gain register update), then the gain will update. This backup system prevents the gain from locking up if the input signal has a small swing and a DC offset that prevents the zero crossing flag from toggling.

7.5.2 ALC Control Register Map

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

Register	Offset	R/W	Description	Reset Value	Reference
ALC_BA = 0x400B_0048					
ALC_CTRL	BIQ_BA+0x0	R/W	ALC control	0x0E01_6320	Table 7-39
ALC_STATUS	BIQ_BA+0x4	R	ALC status	0x0000_0000	Table 7-40
ALC_INT	BIQ_BA+0x8	R/W	ALC interrupt	0x0000_0000	Table 7-41
ALC_INTEN	BIQ_BA+0xC	R/W	ALC interrupt enable	0x0000_0000	Table 7-42

7.5.3 ALC Control Register Description

<u>**ALC Control Register (ALC_CTRL)**</u>

Register	Offset	R/W	Description	Reset Value
ALC_CTRL	ALC_BA+0x0	R/W	ALC Control Register	0x0E01_6320

Table 7-39ALC Control Register (ALC_CTRL, address 0x400B_0048)

31	30	29	28	27	26	25	24
ALCPKLIM	ALCPKSEL	ALCNGSEL	ALCSEL	ALCMAX			ALCMIN[2]
23	22	21	20	19	18	17	16
ALCMIN[1:0]		ALCZC	ALCHLD				ALCLVL[3]
15	14	13	12	11	10	9	8
ALCLVL[2:0]			ALCMODE	ALCDCY			
7	6	5	4	3	2	1	0
ALCATK				NGEN	NGTH		

Bits		Descriptions
[31]	**ALCPKLIM**	**ALC peak limiter enable** 0 = enable fast decrement when signal exceeds 87.5% of full scale (default) 1 = disable fast decrement when signal exceeds 87.5% of full scale
[30]	**ALCPKSEL**	**ALC gain peak detector select** 0 = use absolute peak value for ALC training (default) 1 = use peak-to-peak value for ALC training

[29]	ALCNGSEL	**ALC noise gate peak detector select** 0 = use peak-to-peak value for noise gate threshold determination (default) 1 = use absolute peak value for noise gate threshold determination
[28]	ALCSEL	**ALC select** 0 = ALC disabled(default) 1 = ALC enabled
[27:25]	ALCMAX	**ALC maximum gain** Maximum level for ALC operation 0 = -6.75 dB 1 = -.75 dB 2 = +5.25 dB 3 = +11.25 dB … 7 = +35.25 dB
[24:22]	ALCMIN	**ALC minimum gain** Minimum level for ALC operation 0 = -12dB 1 = -6dB 2 = 0dB 3 = +6dB … 7 = +30dB
[21]	ALCZC	**ALC zero crossing** 0 = zero crossing disabled 1 = zero crossing enabled
[20:17]	ALCHLD	**ALC hold time** 0 = 0 ms 1 = 2 ms 2 = 4 ms (doubles every step) … 10+ = 1 s

[16:13]	**ALCLVL**	**ALC target level** 0 = -28.5 dB 1 = -27 dB 2 = -25.5 dB … 14 = -7.5 dB 15 = -6 dB
[12]	**ALCMODE**	**ALC mode** 0 = ALC normal operation mode 1 = ALC limiter mode
[11:8]	**ALCDCY**	**ALC decay time** ALCMODE=0 Range: 125us to 128ms ALCMODE=1 Range: 31us to 32ms (time doubles with every step)
[7:4]	**ALCATK**	**ALC attack time** ALCMODE=0 Range: 500us to 512ms ALCMODE=1 Range: 125us to 128ms(Both ALC time doubles with every step)
[3]	**NGEN**	**Noise gate enable** 0 = Noise gate disabled 1 = Noise gate enabled

[2:0]	NGTH	**Noise gate threshold** Boost disabled: 000 = -81dB 001 = -75dB 010 = -69dB 011 = -63dB 100 = -57dB 101 = -51dB 110 = -45dB 111 = -39dB Boost enabled: 000 = -87dB 001 = -81dB 010 = -75dB 011 = -69dB 100 = -63dB 101 = -57dB 110 = -51dB 111 = -45dB

ALC Status Register (ALC_STATUS)

Register	Offset	R/W	Description	Reset Value
ALC_STATUS	ALC_BA+0x4	R	ALC status	0x0000_0000

Table 7-40ALC Status Register (ALC_STATUS, address 0x400D_004C)

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				PEAK[8:5]			
15	14	13	12	11	10	9	8
PEAK[4:0]					P2P[8:6]		
7	6	5	4	3	2	1	0
P2P[5:0]						NOISE	FAST_DEC

Bits	Descriptions	
[31:19]	**Reserved**	**Reserved**
[18:11]	**PEAK**	**Peak value** 9 MSBs of measured absolute peak value
[10:2]	**P2P**	**Peak-to-peak value** 9 MSBs of measured peak-to-peak value
[1]	**NOISE**	Noise flag asserted when signal level is detected to be below NGTH
[0]	**FAST_DEC**	Clipping flag asserted when signal level is detected to be above 87.5% of full scale

ALC Interrupt Register (ALC_INT)

Register	Offset	R/W	Description	Reset Value
ALC_INT	ALC_BA+0x8	R/W	ALC interrupt	0x0000_0000

Table 7-41ALC Interrupt Register (ALC_INT, address 0x400D_0050)

7	6	5	4	3	2	1	0
Reserved							**ALC_INT**

Bits	Descriptions	
[31:1]	**Reserved**	**Reserved**
[0]	**ALC_INT**	**ALC interrupt** This interrupt flag asserts whenever the interrupt is enabled and the PGA gain is updated, either through an ALC change with the ALC enabled or through a PGA gain write with the ALC disabled. Write a 1 to this register to clear.

ALC Interrupt Enable Register

Register	Offset	R/W	Description	Reset Value
ALC_INTEN	ALC_BA+0xC	R/W	ALC interrupt enable	0x0000_0000

Table 7-42ALC Interrupt Enable Register (ALC_INTEN, address 0x400D_0054)

7	6	5	4	3	2	1	0
Reserved							**ALC_INTEN**

Bits	Descriptions	
[31:1]	**Reserved**	**Reserved**
[0]	**ALC_INTEN**	**ALC Interrupt Enable** 0 = ALC_INT disabled 1 = ALC_INT enabled

7.6 Biquad Filter (BIQ)

7.6.1 Overview and Features

A coefficient programmable 3-stage Biquad filter (6th-Order IIR filter) is available which can be used on either ADC path or DPWM path to further reduce unwanted noise or filter the signal. Each biquad filter has the transfer function as *H(z)* and is implemented in Direct Form II Transpose structure as.

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Upon power on reset or when the BIQ_CTRL.RSTn=0 is released, a set of default coefficients b_{n0}, b_{n1}, b_{n2}, a_{n1}, a_{n2} (n = 1,2,3 which is the stage number of the filter) will be written to the coefficient RAM automatically. And these coefficients can be over-written by the processor for different filter specifications.

Note that the fixed point coefficientshave the format of 3.16 (19 bits) and are stored in the coefficient RAM under normal operation. It takes 32 internal system clocks for the automatic write to finish when the BIQ_CTRL.RSTn bit is released; it is important that the processor has enough delay before start the coefficient programming or enabling biquad (BIQ_CTRL.EN). Attempting to program the coefficients before the auto programming is done will result in unsuccessful programming. The default coefficient setting is a low pass filter with 3db cut-off frequency at 7/16 Fs (Sample Rate).

Biquad is released from reset by setting BIQ_CTRL.RSTn=1. After 32 clock cycles, processor can setup other Biquad parameters or re-program coefficients before enabling filter.

The BIQ_CTRL.SELPWM register bit determines which path the BIQ is going to use. The default value is 0 which is the microphone ADC path, by setting this bit 1, the BIQ will be used in DPWM path.

The operating sample rate of the filter can be setup by the following registers: The default value of BIQ_CTRL.SR_DIV (sample rate divider) is 3071, when the chip is running at HCLK=49.152Mhz, the operating sample rate of BIQ can be calculated by equation HCLK/(SR_DIV+1) = 16Khz. The output sample rate of BIQ in the ADC path further down-sampled by the BIQ_CTRL.UPSR register so final sample rate presented to the FIFO is HCLK/(SR_DIV+1)/(UPSR+1).

If the BIQ is intended to be used in DPWM path, the BIQ can up sample the data rate by programming BIQ_CTRL.UPSR register which has default value at 3. The final BIQ sampling rate for DPWM path is based on both SR_DIV and BIQ_CTRL.UPSR registers which is equal to SR*(BIQ_CTRL.UPSR +1) . So the default DPWM operating sample rate is 16*4 = 64Khz.

The BIQ filter is in reset state in default. To use the BIQ function, the following sequence is recommended:

1. Set BIQ_CTRL.RSTn bit. By releasing the reset, the filter controller will download default coefficients automatically to the RAM.
2. Turn on the BIQ_CTRL.PRGCOEF bit if intending to change the coefficients. Otherwise skip to next step.
3. Setup the BIQ operation sample rate by program UPSR or SR_DIV register bits if necessary.
4. Decide the ADC or DPWM path to be used for the BIQ by programming SELPWM, and turn off PRGCOEF bit (if it was turned on in step #2).
5. Turn on BIQ_CTRL.EN. BIQ will start filter function.

7.6.2   BIQ Control Register Map

7.6.2.1   *BIQ filter coefficients registers*

Register	Offset	R/W	Description	Reset Value
BIQ_BA = 0x400B_0000 1st stage BIQ Coefficients				
BIQ_COEFF[0]	BIQ_BA + 0x00	R/W	Coefficient b0 in H(z) transfer function. (3.16 format)	0x0d010
BIQ_COEFF[1]	BIQ_BA+0x004	RW	Coefficient b1 in H(z) transfer function. (3.16 format)	0x1c020
BIQ_COEFF[2]	BIQ_BA+0x008	RW	Coefficient b2 in H(z) transfer function. (3.16 format)	0x1c020
BIQ_COEFF[3]	BIQ_BA+0x00c	RW	Coefficient a1 in H(z) transfer function. (3.16 format)	0x1ad66
BIQ_COEFF[4]	BIQ_BA+0x010	RW	Coefficient a2 in H(z) transfer function. (3.16 format)	0x0d1dc

Register	Offset	R/W	Description	Reset Value
BIQ_BA = 0x400B_0000 2nd stage BIQ Coefficients				
BIQ_COEFF[5]	BIQ_BA + 0x14	R/W	Coefficient b0 in H(z) transfer function. (3.16 format)	0x0c1d0
BIQ_COEFF[6]	BIQ_BA+0x018	RW	Coefficient b1 in H(z) transfer function. (3.16 format)	0x183a0
BIQ_COEFF[7]	BIQ_BA+0x01c	RW	Coefficient b2 in H(z) transfer function. (3.16 format)	0x0c1d0
BIQ_COEFF[8]	BIQ_BA+0x020	RW	Coefficient a1 in H(z) transfer function. (3.16 format)	0x17445
BIQ_COEFF[9]	BIQ_BA+0x024	RW	Coefficient a2 in H(z) transfer function. (3.16 format)	0x092f6

Register	Offset	R/W	Description	Reset Value
BIQ_BA = 0x400B_0000 3rd stage BIQ Coefficients				
BIQ_COEFF[10]	BIQ_BA + 0x28	R/W	Coefficient b0 in H(z) transfer function. (3.16 format)	0x0b3cc

BIQ_COEFF[11]	BIQ_BA+0x02c	RW	Coefficient b1 in H(z) transfer function. (3.16 format)	0x16798
BIQ_COEFF[12]	BIQ_BA+0x030	RW	Coefficient b2 in H(z) transfer function. (3.16 format)	0x0b3cc
BIQ_COEFF[13]	BIQ_BA+0x034	RW	Coefficient a1 in H(z) transfer function. (3.16 format)	0x1595d
BIQ_COEFF[14]	BIQ_BA+0x038	RW	Coefficient a2 in H(z) transfer function. (3.16 format)	0x075d2

7.6.2.2   *BIQ Control Register (BIQ_CTRL)*

Register	Offset	R/W	Description	Reset Value
BIQ_CTRL	BIQ_BA+0x040	R/W	BIQ control	0x0BFF_0030

Table 7-43 BIQ Control Register (BIQ_CTRL, address 0x400B_0040)

31	30	29	28	27	26	25	24
Reserved			SR_DIV[12:8]				
23	22	21	20	19	18	17	16
SR_DIV[7:0]							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved	UPSR			RSTn	PRGCOEF	SELPWM	EN

Bits	Descriptions	
[28:16]	SR_DIV	**Sample Rate divider** This register is used to program the operating sampling rate of the biquad filter. The sample rate is defined as HCLK/(SR_DIV+1). Default to 3071 so the sampling rate is 16K.

[6:4]	UPSR	**DPWM path up/down sample rate (from SR_DIV result)** If SELPWM is set to 1. The output sample rate of the biquad filter will be (PWM_UPSR +1)*HCLK/(SR_DIV+1). If SELPWM=0 (ADC path) the output sample rate of the biquad filer is HCLK/(SR_DIV+1)/(PWM_UPSR +1) Default value for this register is 3.
[3]	RSTn	**Move BIQ out of reset state** 0 = BIQ filter is in reset state. 1 = When this bit is on, the default coefficients will be downloaded to the coefficient ram automatically in 32 internal system clocks. Processor must delay enough time before changing the coefficients or turn the BIQ on.
[2]	PRGCOEF	0 = Coefficient RAM is in normal mode. 1 = coefficient RAM is under programming mode. This bit must be turned off when BIQEN in on.
[1]	SELPWM	**AC path selection for BIQ** 0 = used in ADC path 1 = used in DPWM path
[0]	EN	**BIQ filter start to run** 0 = BIQ filter is not processing 1 = BIQ filter is on.

8   APPLICATION DIAGRAM

⏚ : Digital ground;   ▽ : Analog ground;
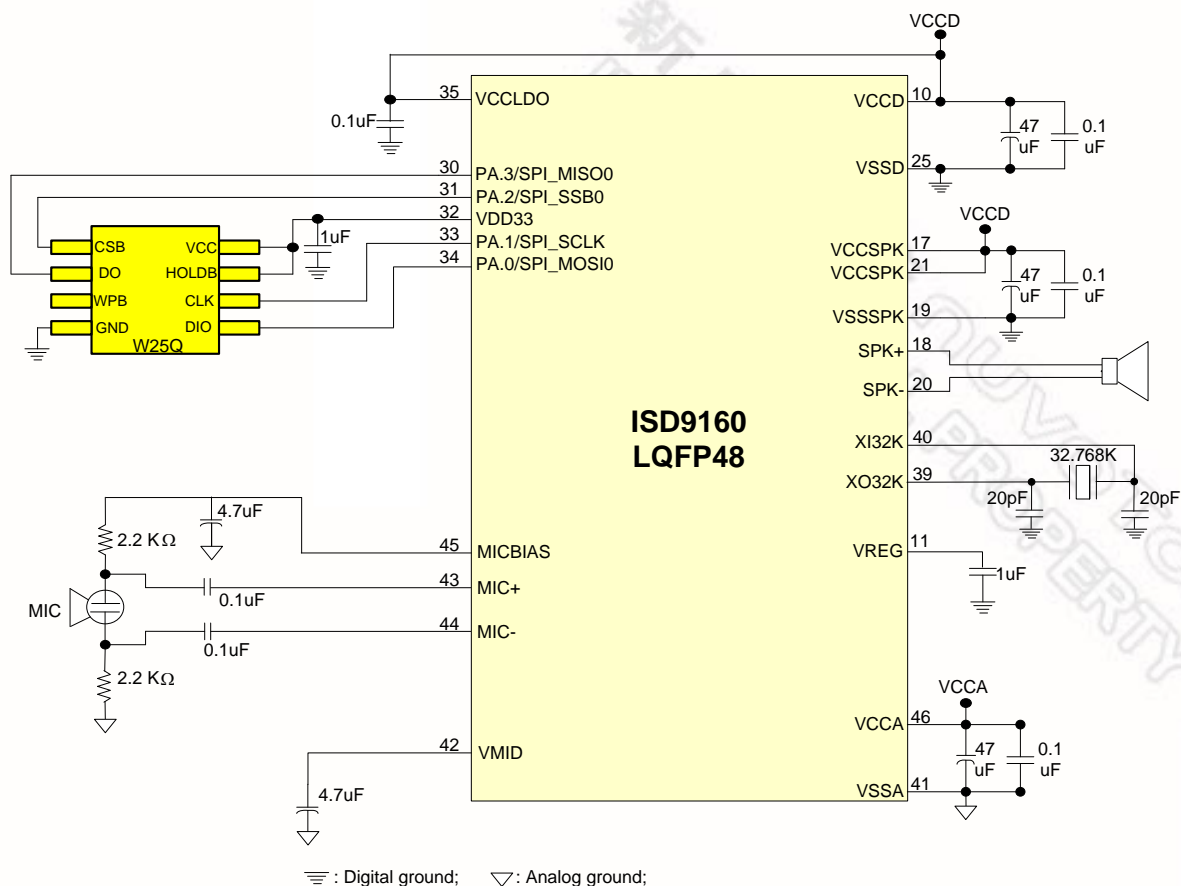
9 ELECTRICAL CHARACTERISTICS

9.1 Absolute Maximum Ratings

SYMBOL	PARAMETER	MIN	MAX	UNIT
DC Power Supply	VDD−VSS	-0.3	+6.0	V
Input Voltage	VIN	VSS-0.3	VDD+0.3	V
Oscillator Frequency	1/t$_{CLCL}$	0	40	MHz
Operating Temperature	TA	-40	+85	°C
Storage Temperature	TST	-55	+150	°C
Maximum Current into V$_{DD}$		-	120	mA
Maximum Current out of V$_{SS}$			120	mA
Maximum Current sunk by a I/O pin			35	mA
Maximum Current sourced by a I/O pin			35	mA
Maximum Current sunk by total I/O pins			100	mA
Maximum Current sourced by total I/O pins			100	mA

Note: Exposure to conditions beyond those listed under absolute maximum ratings may adversely affects the lift and reliability of the device. Functional operation is not implied at these conditions.

9.2 DC Electrical Characteristics

(VDD-VSS=3.3V, TA = 25°C, F$_{OSC}$ = 49.152 MHz unless otherwise specified.)

PARAMETER	SYM.	SPECIFICATION				TEST CONDITIONS
		MIN.	TYP.	MAX.	UNIT	
Operation voltage	V$_{DD}$	2.4		5.5	V	V$_{DD}$ =2.4V ~ 5.5V up to 100 MHz
Power Ground	V$_{SS}$ AV$_{SS}$	-0.3			V	
Analog Operating Voltage	AV$_{DD}$	0		V$_{DD}$	V	
DPWM Speaker Voltage	V$_{DDSPK}$	2.4		5.5	V	
Current Consumption @ Normal Mode @ 49.152 MHz	I$_{DD}$		25		mA	V$_{DD}$= 5.5V
	I$_{DD}$		21		mA	V$_{DD}$ = 3V
Current Consumption @ Sleep Mode	I$_{IDLE1}$		10		mA	V$_{DD}$= 5.5V, Enable all IP (analog and digital)
	I$_{IDLE1}$		9		mA	V$_{DD}$= 3.3V, Enable all IP (analog and digital)
Current Consumption @ Deep Sleep Mode	I$_{IDLE1}$		10		mA	V$_{DD}$=5.5V. *
	I$_{IDLE1}$		9		mA	V$_{DD}$= 3.3V. *
Current Consumption @ Standby Power Down Mode(SPD)	I$_{IDLE1}$		5		uA	V$_{DD}$=3.3V, 32K CRTL running with RTC
	I$_{IDLE1}$		1		uA	V$_{DD}$= 3.3V,OSC16K
Current Consumption @ Deep Power Down Mode(DPD)	I$_{IDLE1}$		400		nA	V$_{DD}$=3.3V, Wakeup by OSC16K timer
	I$_{IDLE1}$		500		nA	V$_{DD}$= 3.3V, wakeup by Wakeup pin

Input Current PA, PB (Quasi-bidirectional mode)	I_{IN1}	-60	-	+15	µA	V_{DD} = 5.5V, V_{IN} = 0V or V_{IN}=V_{DD}
Input Current at /RESET [1]	I_{IN2}	-55	-45	-30	µA	V_{DD} = 3.3V, V_{IN} = 0.45V
Input Leakage Current PA, PB	I_{LK}	-2	-	+2	µA	V_{DD} = 5.5V, 0<V_{IN}<V_{DD}
Logic 1 to 0 Transition Current PA~PB(Quasi-bidirectional mode)	I_{TL} [3]	-650	-	-200	µA	V_{DD} = 5.5V, V_{IN}<2.0V
Input Low Voltage PA, PB(TTL input)	V_{IL1}	-0.3	-	0.8	V	V_{DD} = 4.5V
		-0.3	-	0.6		V_{DD} = 2.5V
Input High Voltage PA, PB (TTL input)	V_{IH1}	2.0	-	V_{DD}+0.2	V	V_{DD} = 5.5V
		1.51	-	V_{DD}+0.2		V_{DD} =3.0V
Input Low Voltage XT1[*2]	V_{IL3}	0	-	0.8	V	V_{DD} = 4.5V
		0	-	0.4		V_{DD} = 3.0V
Input High Voltage XT1[*2]	V_{IH3}	3.5	-	V_{DD}+0.2	V	V_{DD} = 5.5V
		2.4	-	V_{DD}+0.2		V_{DD} = 3.0V
Input Low Voltage X32I[*2]	V_{IL4}	0	-	0.4	V	
Input High Voltage X32I[*2]	V_{IH4}	1.7		2.5	V	
Negative going threshold (Schmitt input), /REST	V_{ILS}	-0.5	-	0.3V_{DD}	V	
Positive going threshold (Schmitt input), /REST	V_{IHS}	0.7V_{DD}	-	V_{DD}+0.5	V	
Hysteresis voltage of PA~PB(Schmitt input)	V_{HY}		0.2V_{DD}		V	

Source Current PA, PB Quasi-bidirectional Mode)	I_{SR11}	-300	-340	-450	µA	$V_{DD} = 4.5V$, $V_S = 2.4V$
	I_{SR12}	-50	-65	-90	µA	$V_{DD} = 2.7V$, $V_S = 2.2V$
	I_{SR12}	-40	-55	-80	µA	$V_{DD} = 2.5V$, $V_S = 2.0V$
Source Current PA, PB (Push-pull Mode)	I_{SR21}	-20	-22	-28	mA	$V_{DD} = 4.5V$, $V_S = 2.4V$
	I_{SR22}	-3.5	-6	-8	mA	$V_{DD} = 2.7V$, $V_S = 2.2V$
	I_{SR22}	-3	-5	-7	mA	$V_{DD} = 2.5V$, $V_S = 2.0V$
Sink Current PA, PB (Quasi-bidirectional and Push-pull Mode)	I_{SK1}	10	12	20	mA	$V_{DD} = 4.5V$, $V_S = 0.45V$
	I_{SK1}	7	9	13	mA	$V_{DD} = 2.7V$, $V_S = 0.45V$
	I_{SK1}	6	7	12	mA	$V_{DD} = 2.5V$, $V_S = 0.45V$

Notes:

1. /REST pin is a Schmitt trigger input.

2. Crystal Input is a CMOS input.

3. Pins of P0, P1, P2, P3 and P4 can source a transition current when they are being externally driven from 1 to 0. In the condition of V_{DD}=5.5V, 5he transition current reaches its maximum value when Vin approximates to 2V.

*: Data value for this item varies depending on different IP modules enabled.

9.2.1 Operating Current Curve (Test condition: run NOP)

Chart: Current (mA) vs CPU Frequency (MHz)

Legend:
- ◆ Iccd (mA)
- ■ Iccd (sleep)
- ▲ Icc (all IP on)
- Linear (Iccd (mA))

Icc(All IP on mA)= 0.2694 * F(MHz) + 7.3795

Iact(mA) = 0.1938 * F(MHz) + 1.1483

Isleep (mA) = 0.0643 * F(MHz) + 1.1533
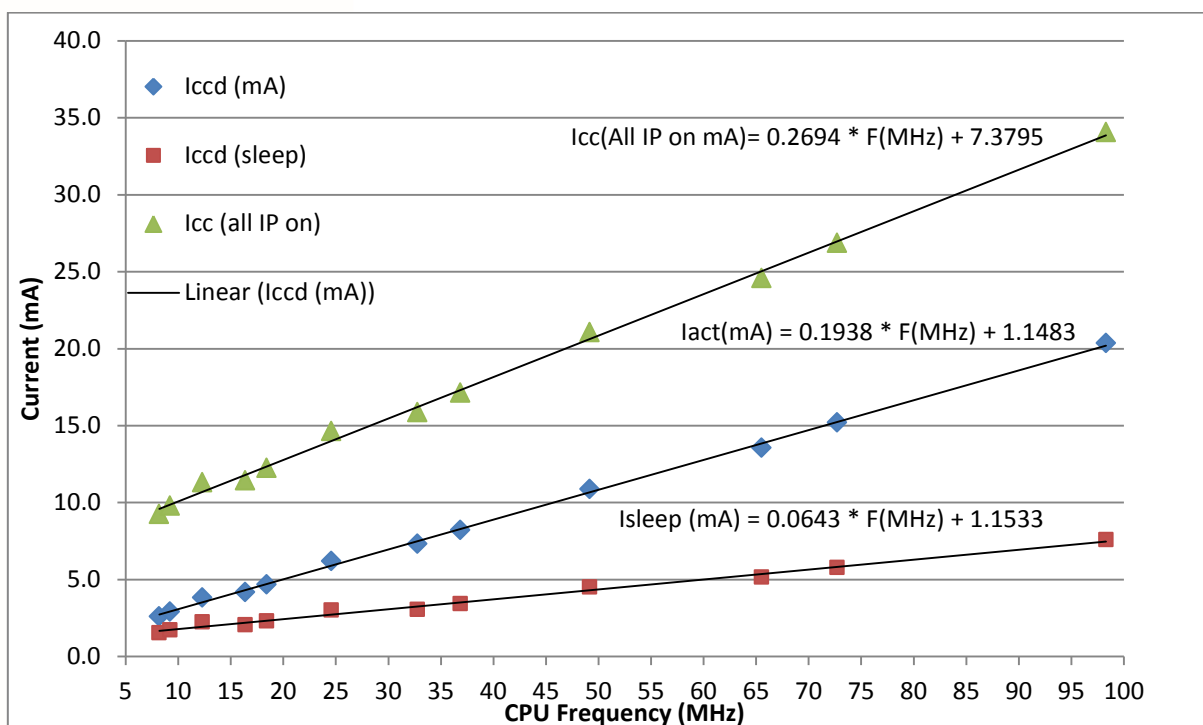
9.2.2 Power Down Current Curve

9.3 AC Electrical Characteristics

9.3.1 External 32kHz XTAL Oscillator

PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
Input clock frequency	External crystal	-	32.768	-	kHz
Temperature	-	-40	-	85	℃
V$_{DD}$	-	2.4	-	5.5	V

9.3.2 Internal 49.152MHz Oscillator

PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
Supply voltage	-	2.4	-	5.5	V
Center Frequency	-	-	49.152		MHz
Calibrated Internal Oscillator Frequency	+25°C; V$_{DD}$ =3V	-1	-	1	%
	-40°C~+85°C; V$_{DD}$=2.4V~5.5V	-4	-	4	%

9.3.3 Internal 16kHz Oscillator

PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
Supply voltage	-	2.4	-	5.5	V
Center Frequency	-	-	16	-	kHz
Calibrated Internal Oscillator Frequency	+25°C; V$_{DD}$ =3V	-10	-	10	%
	-40°C~+85°C; V$_{DD}$=2.4V~5.5V	-20	-	20	%

9.4 Analog Characteristics

9.4.1 Specification of ADC and Speaker Driver

Conditions: VCCD = 3.3V, VCCA = 3.3V, T_A = +25°C, 1kHz signal, fs = 16kHz, 16-bit audio data, unless otherwise stated.

Parameter	Symbol	Comments/Conditions	Min	Typ	Max	Units
Analog to Digital Converter (ADC)						
Full scale input signal [1]	V_{INFS}	PGABST = 0dB PGAGAIN = 0dB		1.0 0		Vrms dBV
Signal-to-noise ratio	SNR	Gain = 0dB, A-weighted		90		dB
Total harmonic distortion [2]	THD+N	Input = -3dB FS input		0.04		%
PWM Speaker Output (8Ω bridge-tied-load)						
Full scale output [4]				VCCSPK / 3.3		V_{rms}
Total harmonic distortion [2]	THD+N	P_o= 200mW, VDDSPK=3.3V		*63		dB
		P_o= 320mW, VDDSPK = 3.3V		-64		dB
		P_o= 860mW, VDDSPK = 5V		-60		dB
		P_o= 1000mW, VDDSPK = 5V		-36		dB
Signal-to-noise ratio	SNR	VDDSPK = 3.3V		90		dB
		VDDSPK=5V		89		dB
Power supply rejection ratio (50Hz - 22kHz)	PSRR	VDDSPK = 3.3V				dB
		VDDSPK = 5V				dB

9.4.2 Specification of PGA and BOOST

Conditions: VCCD = 3.3V, VCCA = 3.3V, T$_A$ = +25℃, 1kHz signal, fs = 16kHz, 16-bit audio data, unless otherwise stated.

Parameter	Symbol	Comments/Conditions	Min	Typ	Max	Units
Microphone Inputs (MICP, MICN) and Programmable Gain Amplifier (PGA)						
Full scale input signal [1]		PGABST = 0dB		1.0		Vrms
		PGAGAIN = 0dB		0		dBV
Programmable gain			-12		35.25	dB
Programmable gain step size		Guaranteed Monotonic		0.75		dB
Mute Attenuation				120		dB
Input resistance		Inverting Input				
		PGA Gain = 35.25dB		1.6		kΩ
		PGA Gain = 0dB		47		kΩ
		PGA Gain = -12dB		75		kΩ
		Non-inverting Input		94		kΩ
Input capacitance				10		pF
PGA equivalent input noise		0 to 20kHz, Gain set to 35.25dB		120		µV
Input Boost						
Gain boost		Boost disabled		0		dB
		Boost enabled		26		dB

9.4.3 Specification of ALC an MICBIAS

Conditions: VCCD = 3.3V, VCCA = 3.3V, T_A = +25°C, 1kHz signal, fs = 16kHz, 16-bit audio data, unless otherwise stated.

Parameter	Symbol	Comments/Conditions	Min	Typ	Max	Units
Automatic Level Control (ALC) & Limiter:						
Target record level			-22.5		-1.5	dBFS
Programmable gain			-12		35.25	dB
Gain hold time [3]	t_{HOLD}	Doubles every gain step, with 16 steps total	0 / 2.67 / 5.33 / … / 43691			ms
Gain ramp-up (decay) [3]	t_{DCY}	ALC Mode ALC = 0	4 / 8 / 16 / … / 4096			ms
		Limiter Mode ALC = 1	1 / 2 / 4 / … / 1024			ms
Gain ramp-down (attack) [3]	t_{ATK}	ALC Mode ALC = 0	1 / 2 / 4 / … / 1024			ms
		Limiter Mode ALC = 1	0.25 / 0.5 / 1 / … / 128			ms
Mute Attenuation				120		dB
Microphone Bias						
Bias voltage	$V_{MICBIAS}$		0.90, 0.65 ,0.75, 0.50, 2.4, 1.7,2.0			VDDA V
Bias current source	$I_{MICBIAS}$			3		mA
Output noise voltage	V_n	1kHz to 20kHz		14		nV/√Hz

Notes

1. Full Scale is relative to the magnitude of VCCA and can be calculated as FS = VDDA/3.3.
2. Distortion is measured in the standard way as the combined quantity of distortion products plus noise. The signal level for distortion measurements is at 3dB below full scale, unless otherwise noted.
3. Time values scale proportionally with HCLK. Complete descriptions and definitions for these values are contained in the detailed descriptions of the ALC functionality.

9.4.4   Specification of LDO & Power management

PARAMETER	MIN	TYP	MAX	UNIT	NOTE
Input Voltage	2.4	5	5.5	V	V_{DD} input voltage
Output Voltage	-10%	1.8	+10%	V	V_{DD}>1.8

Note:

1. It is recommended that a 10uF or higher capacitor and a 100nF bypass capacitor are connected between VCCD and the VSSD pin of the device.

2. To ensure regulator stability, a 1.0uF capacitor must be connected between LDO pin and the VSSD pin of the device. Also a 100nF bypass capacitor between LDO and VSSD will help suppress output noise.

9.4.5 Specification of Brownout Detector

PARAMETER	CONDITION		MIN.	TYP.	MAX.	UNIT
Operation voltage	-		2.2	-	5.5	V
Quiescent current	AVDD=5.5V		-	-	125	µA
Temperature	-		-40	25	85	℃
Brown-out voltage	Range=0	BOV_VL[2:0]=000		2.3		V
		BOV_VL [2:0]=001		2.4		V
		BOV_VL [2:0]=010		2.6		V
		BOV_VL[2:0]=011		2.7		V
		BOV_VL [2:0]=100		2.8		V
		BOV_VL[2:0]=101		3.0		V
		BOV_VL [2:0]=110		3.2		V
		BOV_VL [2:0]=111		4.8		V
	Range=1	BOV_VL[2:0]=000		3.3		V
		BOV_VL [2:0]=001		3.4		V
		BOV_VL [2:0]=010		3.6		V
		BOV_VL[2:0]=011		3.8		V
		BOV_VL [2:0]=100		3.9		V
		BOV_VL[2:0]=101		4.0		V
		BOV_VL [2:0]=110		4.1		V
		BOV_VL [2:0]=111		4.4		V
Hysteresis				–		V

9.4.6 Specification of Power-On Reset (VCCD)

PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNIT
Temperature	-	-40	25	85	℃
Reset voltage	VCC ramping down	-	1.0	-	V
Reset Release voltage	VCC ramping up		1.5		V
Quiescent current	Vin>reset voltage	-	60	-	nA

9.4.7 Specification of Temperature Sensor

PARAMETER	MIN	TYP	MAX	UNIT	CONDITIONS
Supply voltage[1]	2.4	-	5.5	V	
Temperature	-40	-	125	℃	
Current consumption				uA	
Gain				mV/℃	
Offset				mV	Temp=0 ℃

Notes:
1. Internal operation voltage comes from LDO.

9.4.8 Specification of Comparator

PARAMETER	MIN.	TYP.	MAX.	CONDITION
Temperature	-40℃	25 ℃	85℃	-
VCCA	2.4	3	5.5	-
VCCA current	-	20uA	40uA	20uA@VDD=3V
Input offset voltage	-	5mV	15mV	-
Input common mode range	0.1	-	VDD-1.2	-
DC gain	-	70dB	-	-
Propagation delay	-	200ns	-	@VCM=1.2V & VDIFF=0.1V
Comparison voltage	10mV	20mV	-	20mV@VCM=1V 50mV@VCM=0.1V 50mV@VCM=VDD-1.2 @10mV for non-hysteresis
Hysteresis	-	±10mV	-	One bit control W/O & W. hysteresis @VCM=0.4V ~ VDD-1.2V
Wake up time	-	-	2us	@CINP=1.3V CINN=1.2V

10 PACKAGE DIMENSIONS

10.1 64L LQFP (7x7x1.4mm footprint 2.0mm)

	SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS	A	---	---	1.6
STAND OFF	A1	0.05	---	0.15
MOLD THICKNESS	A2	1.35	1.4	1.45
LEAD WIDTH(PLATING)	b	0.13	0.18	0.23
LEAD WIDTH	b1	0.13	0.16	0.19
L/F THICKNESS(PLATING)	c	0.09	---	0.2
L/F THICKNESS	c1	0.09	---	0.16
BODY SIZE	X	D	9 BSC	
	Y	E	9 BSC	
	X	D1	7 BSC	
	Y	E1	7 BSC	
LEAD PITCH	e	0.4 BSC		
FOOTPRINT	L	0.45	0.6	0.75
	L1	1 REF		
	θ	0°	3.5°	7°
	θ1	0°	---	---
	θ2	11°	12°	13°
	θ3	11°	12°	13°
	R1	0.08	---	---
	R2	0.08	---	0.2
	S	0.2	---	---
PACKAGE EDGE TOLERANCE	aaa	0.2		
LEAD EDGE TOLERANCE	bbb	0.2		
COPLANARITY	ccc	0.08		
LEAD OFFSET	ddd	0.07		
MOLD FLATNESS	eee	0.05		

11 ORDERING INFORMATION

<center>I 9 3 x x x x x</center>

ISD Audio Product Family

Product Series
93: Cortex-M0

Flash ROM
6: 145KB

SRAM
1: 16KB

Temperature
I: -40°C ~ +85°C

Package
R: LQFP-64

SW Feature
Blank: Standard
V: VR
I: PSW VR
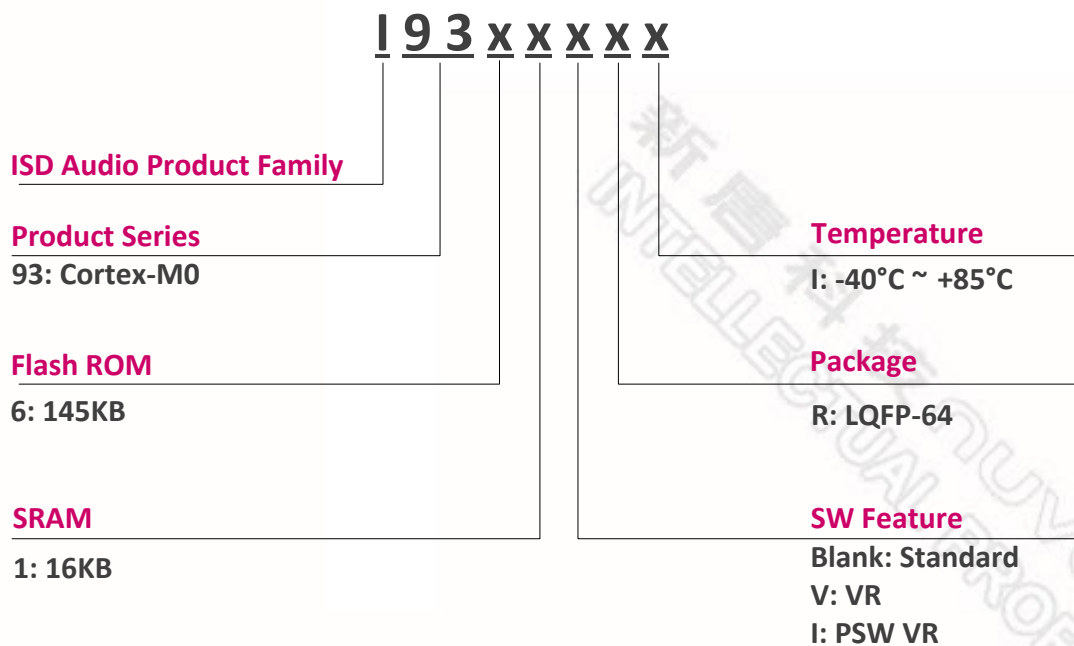
12 REVISION HISTORY

VERSION	DATE	PAGE/ CHAP.	DESCRIPTION
V1.0	Jun 1, 2014	-	First release.

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation where personal injury, death or severe property or environmental damage could occur.

Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

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