AppoTech
卓荣集成

AX107X

Audio Player Microcontroller

User Manual

[AX107X-UM-110-EN]

Versions: 1.1.0

Release Date: 2014-5-16

Unit705-707,7/F IC Development Centre,
No.6 Science Park West Ave,
Hong Kong Science Apark,
Shation,N.T.,HongKong.
香港新界沙田香港科学园科技大道西6号
集成电路开发中心7楼705-707室

Tel 电话: (852)26074090
Fax传真: (852)26074096
info@appotech.com

www.appotech.com www.buildwin.com

Table of Contents

Figure

Table

Register

1 Product Overview

1.1 Outline

AX107X is an MCS-51™ Compatible high performance mixed signal microcontroller. It integrates advanced digital and analog peripherals to suit for Audio playback、MP3 player applications.

1.2 Features

- Compatible with MCS-51TM instruction set;
- Maximum 48MHz operating frequency;
- Supports MP3 decoder;
- Four multi-function 16-bit timers, support Capture and PWM mode;
- Watchdog;
- timer with on-chip RC oscillator;
- One Full-duplex UART;
- SBC-UART;
- one SPI;
- Infrared Remote control decoder;
- SD Card Host controller;
- Support 2 Full speed USB 2.0 HOST/DEVICE controller;
- EQ
- Support 6 Touch-Keys
- Single pin Oscillator for external 32.768K/4M/12M Hz crystal resonator;
- Internal 10/14MHz RC oscillator;
- Internal lower power 2M RC
- PLL with 48MHz frequency output;
- Stereo DAC with 85dB SNR;
- Two channels three groups stereo AUX;
- VCM buffer;
- Mono MIC Amplifier;
- 11 channels 10-bit SARADC;
- Stereo class A/B Amplifier;
- Low Voltage Detector;
- Power-on Reset;

1.3 Internal LDO regulator:3.3V to 1.8V capless, 5V to 3.3VSystem Architecture

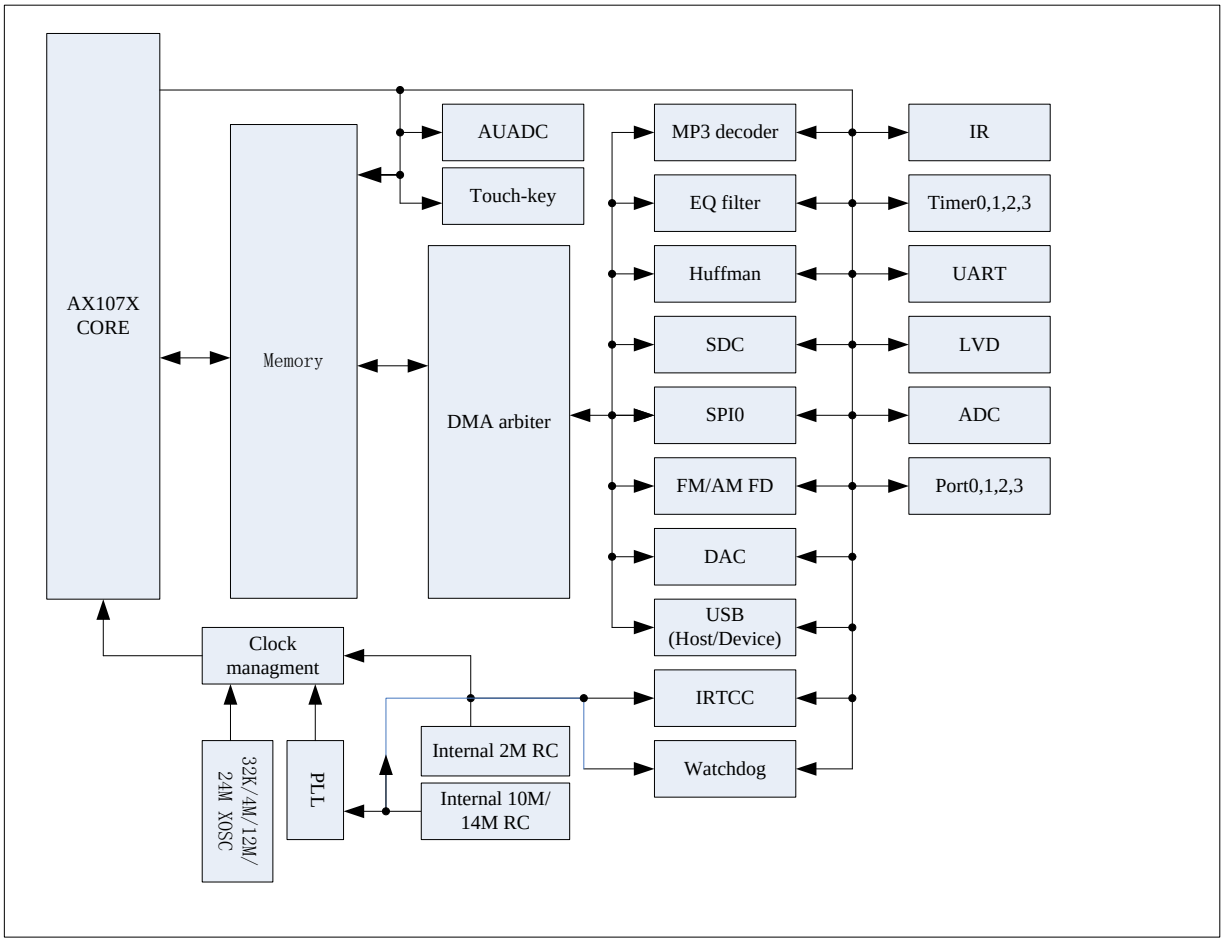


Figure 1-1 AX107X architecture

2 PIN Definitions

2.1 AX1070

2.1.1 Part Number

AX1070

2.1.2 Packages

LQFP48

2.1.3 Pin Assignment

Figure 2-1 shows AX1070 LQFP48 Package.

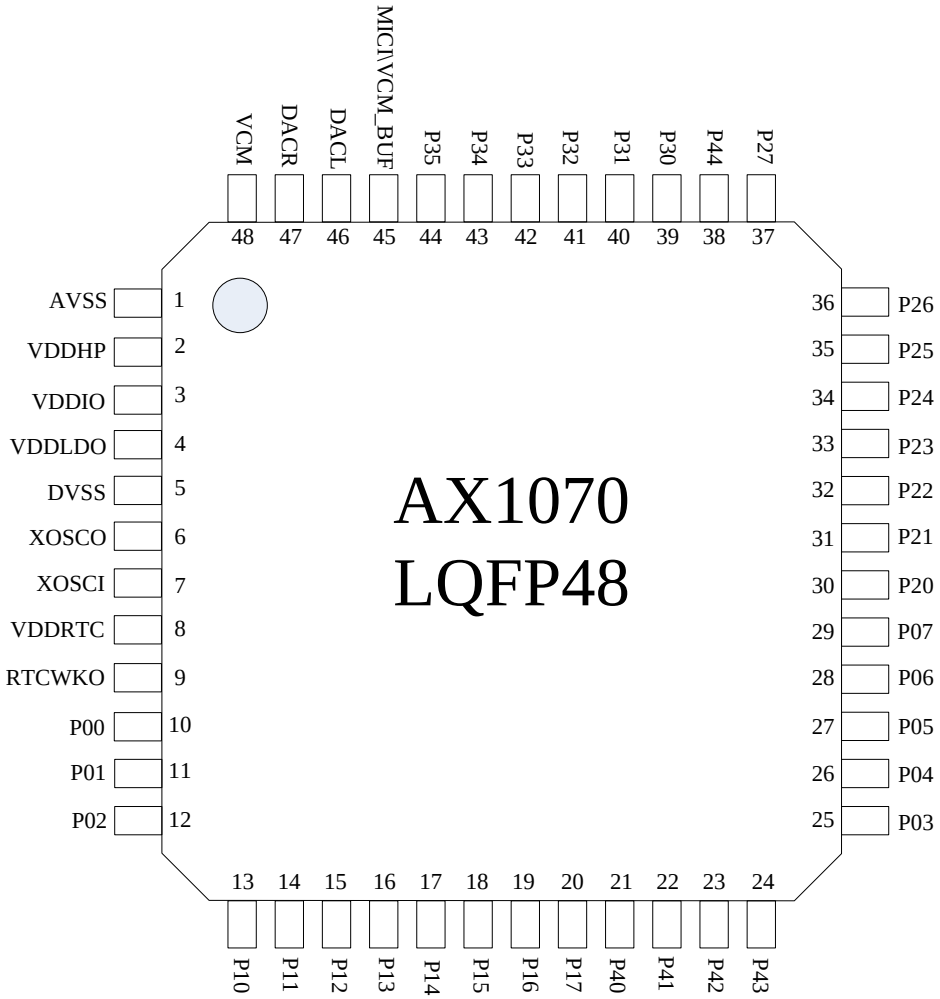


Figure 2-1 Pin assignment for AX1070 LQFP48

2.1.4 Pin Descriptions

Figure 2-1 shows the pin assignments of AX1070 LQFP48 package. *Table 2-1* shows the pin descriptions of AX1070 LQFP48 package.

Table 2-1 AX1070 LQFP48 pin description

Pin No.	Name	Type	Function
1	AVSS	GND	DAC GND
2	VDDHP	PWR	DAC VDDHP POWER
3	VDDIO	PWR	IO Power
4	VDDLDO	PWR	LDO Power
5	DVSS	GND	Digital ground
6	XOSCO	AIO	2-Pin Oscillator output 1-pin Osillator input
7	XOSCI	AI	2-Pin Oscillator input
8	VDDRTC	PWR	IRTC power
9	RTCWKO	I/O	IRTC alarm output IRTC wakeup input
10	P00	I/O	GPIO Mute WKUP TMR3PWM
11	P01	I/O	GPIO ADC Input TMR2CAP WKUP TK5
12	P02	I/O	GPIO ADC Input Ir input TMR0INC WKUP
13	P10	I/O	GPIO ADC Input Ir in TMR2PWM SBC_UART TX WKUP SPICLK TK0
14	P11	I/O	GPIO TMR3INC SBC_UART_RX

Pin No.	Name	Type	Function
			UARTRX WKUP SPIDIO TK1
15	P12	I/O	GPIO SDDAT SPICLK UARTTX WKUP TK2
16	P13	I/O	GPIO SDCLK TMR3CAP TMR2CAP WKUP SPIDIO TK3
17	P14	I/O	GPIO ADC TMR2INC SDCMD SPIDI WKUP TK4
18	P15	I/O	GPIO ADC Input SDDAT SPICLK WKUP Sys_clk_output Pll24M/12M output Rtc32K output
19	P16	I/O	GPIO ADC Input SDCLK SPIDIO Sys_clk_output Pll24M/12M output Rtc32K output WKUP
20	P17	I/O	GPIO VPP

Pin No.	Name	Type	Function
			SDCMD SPIDI
21	P40	I/O	GPIO USBDM1 ADC Input WKUP
22	P41	I/O	GPIO USBDP1 WKUP ADC Input
23	P42	I/O	GPIO SDCLK USBDM2 ADC Input WKUP
24	P43	I/O	GPIO SDDAT USBDP2 ADC Input WKUP
25	P03	I/O	GPIO ADC Input TMR1INC IR WKUP
26	P04	I/O	GPIO TMR1PWM
27	P05	I/O	GPIO TMR0PWM
28	P06	I/O	GPIO TMR0CAP TMR1CAP SDCLK Sys_clk_output Pll24M/12M output Rtc32K output
29	P07	I/O	GPIO TMR1CAP
30	P20	I/O	GPIO
31	P21	I/O	GPIO
32	P22	I/O	GPIO
33	P23	I/O	GPIO

Pin No.	Name	Type	Function
34	P24	I/O	GPIO
35	P25	I/O	GPIO
36	P26	I/O	GPIO
37	P27	I/O	GPIO SBC_UART_RX Ir in ADC Input LVD_DET WKUP
38	P44	I/O	GPIO SBC_UART_TX LCD_BL
39	P30	I/O	GPIO AUXR0 SDCMD
40	P31	I/O	GPIO AUXL0 SDDAT LCD_BL
41	P32	I/O	GPIO AUXR1
42	P33	I/O	GPIO AUXL1
43	P34	I/O	GPIO AUXR2 UARTRX AM
44	P35	I/O	GPIO AUXL2 UARTTX
45	MIC	AIO	MIC AUXL3 VCMBUF (double bonding)
46	DACL	AO	DAC Left Channel AUXR3 GPIO(VDDHP domain)
47	DACR	AO	DAC Right Channel
48	VCM	O	DAC Bandgap voltage reference

2.2 AX1071

2.2.1 Part Number

AX1071

2.2.2 Packages

SOP28

2.2.3 Pin Assignment

Figure 2-2 shows AX1071 SOP28 Package.

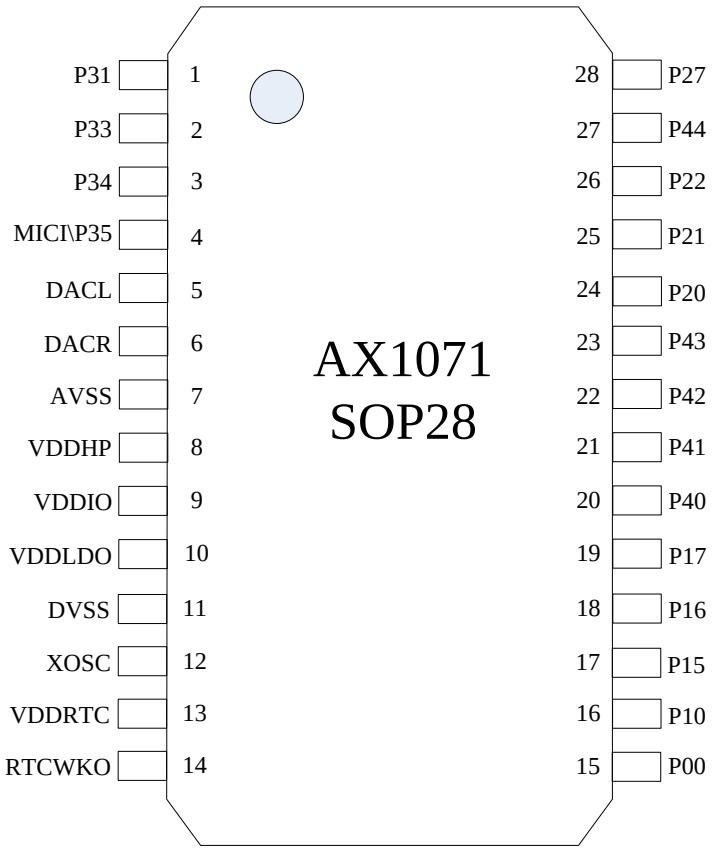


Figure 2-2 Pin Assignment for AX1071 SOP28

2.2.4 Pin Description

Figure 2-2 shows the pin assignments of AX1071 SOP28 package. *Table 2-2* shows the pin descriptions of AX1071 SOP28 package.

Table 2-2 AX1071 SOP28 Pin Description

Pin No.	Name	Type	Function
1	P31	I/O	GPIO

Pin No.	Name	Type	Function
			AUXL0 SDDAT LCD_BL
2	P33	I/O	GPIO AUXL1
3	P34	I/O	GPIO AUXR2 UARTRX AM
4	P35/ MIC	I/O	GPIO AUXL2 UARTTX VCMBUF (double bonding) MIC AUXL3
5	DACL	AO	DAC Left Channel AUXR3 GPIO(VDDHP domain)
6	DACR	AO	DAC Right Channel
7	AVSS	GND	DAC GND
8	VDDHP	PWR	DAC VDDHP POWER
9	VDDIO	PWR	IO Power
10	VDDLDO	PWR	LDO Power
11	DVSS	GND	Digital ground
12	XOSC	AIO	2-Pin Oscillator output 1-pin Osillator input
13	VDDRTC	PWR	IRTC power
14	RTCWKO	I/O	IRTC alarm output IRTC wakeup input
15	P00	I/O	GPIO Mute WKUP TMR3PWM
16	P10	I/O	GPIO ADC Input Ir in TMR2PWM SBC_UART TX WKUP SPICLK TK0
17	P15	I/O	GPIO

Pin No.	Name	Type	Function
			ADC Input SDDAT SPICLK WKUP Sys_clk_output Pll24M/12M output Rtc32K output
18	P16	I/O	GPIO ADC Input SDCLK SPIDIO Sys_clk_output Pll24M/12M output Rtc32K output WKUP
19	P17	I/O	GPIO VPP SDCMD SPIDI
20	P40	I/O	GPIO USBDM1 ADC Input WKUP
21	P41	I/O	GPIO USBDP1 WKUP ADC Input
22	P42	I/O	GPIO SDCLK USBDM2 ADC Input WKUP
23	P43	I/O	GPIO SDDAT USBDP2 ADC Input WKUP
24	P20	I/O	GPIO
25	P21	I/O	GPIO
26	P22	I/O	GPIO
27	P44	I/O	GPIO SBC_UART_TX

Pin No.	Name	Type	Function
			LCD_BL
28	P27	I/O	GPIO SBC_UART_RX Ir in ADC Input LVD_DET WKUP

2.3 AX1073

2.3.1 Part Number

AX1073

2.3.2 Packages

SOP16

2.3.3 Pin Assignment

Figure 2-3 shows AX1073 SOP16 Package.

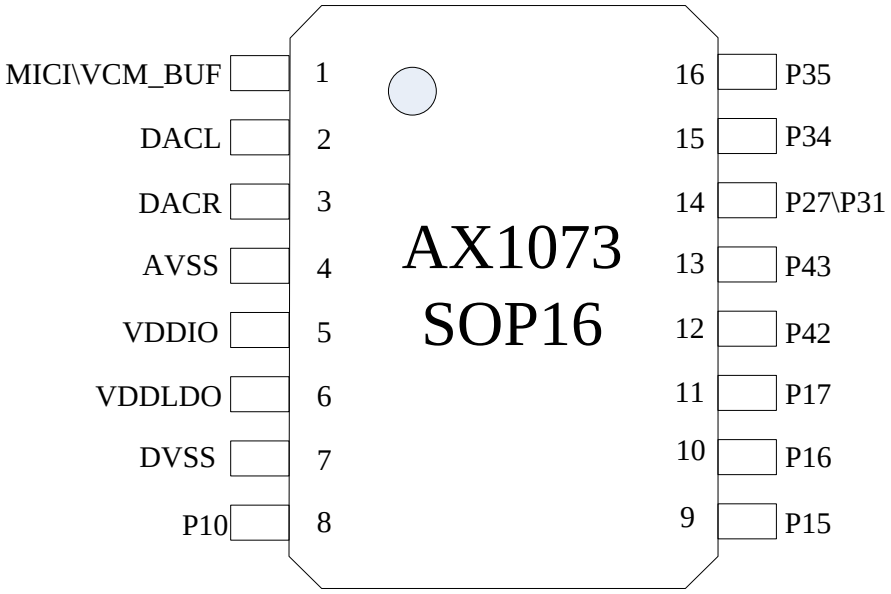


Figure 2-3 Pin assignment for AX1073 SOP16

2.3.4 Pin Description

Table 2-3 shows the pin descriptions of AX1073 SOP16 package.

Table 2-3 AX1073 SOP16 pin description

Pin No.	Name	Type	Function
1	MIC/VCM_BUF	AIO	MIC AUXL3 VCMBUF (double bonding)
2	DACL	AO	DAC Left Channel AUXR3 GPIO(VDDHP domain)
3	DACR	AO	DAC Right Channel
4	AVSS	GND	DAC GND
5	VDDIO	PWR	IO Power
6	VDDLDO	PWR	LDO Power
7	DVSS	GND	Digital ground
8	P10	I/O	GPIO ADC Input Ir in TMR2PWM SBC_UART TX WKUP SPICLK TK0
9	P15	I/O	GPIO ADC Input SDDAT SPICLK WKUP Sys_clk_output Pll24M/12M output Rtc32K output
10	P16	I/O	GPIO ADC Input SDCLK SPIDIO Sys_clk_output Pll24M/12M output Rtc32K output WKUP
11	P17	I/O	GPIO VPP SDCMD SPIDI
12	P42	I/O	GPIO SDCLK USBDM2

Pin No.	Name	Type	Function
			ADC Input WKUP
13	P43	I/O	GPIO SDDAT USBDP2 ADC Input WKUP
14	P27/ P31	I/O	GPIO SBC_UART_RX Ir in ADC Input LVD_DET WKUP AUXL0 SDDAT LCD_BL
15	P34	I/O	GPIO AUXR2 UARTRX AM
16	P35	I/O	GPIO AUXL2 UARTTX

2.4 AX1073B

2.4.1 Part Number

AX1073B

2.4.2 Package

SOP16

2.4.3 Pin Assignment

Figure 2-4 shows AX1073B SOP16 Package.

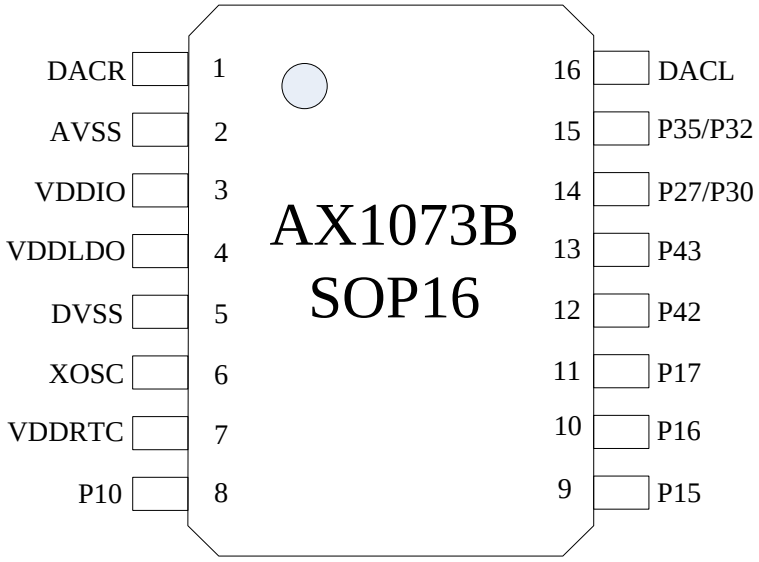


Figure 2-4 Pin assignment for AX1073B SOP16

2.4.4 Pin Description

Table 2-4 shows the pin descriptions of AX1073B SOP16 package.

Table 2-4 AX2073B SOP16 pin description

Pin No.	Name	Type	Function
1	DACR	AO	DAC Right Channel
2	AVSS	GND	DAC GND
3	VDDIO	PWR	IO Power
4	VDDLDO	PWR	LDO Power
5	DVSS	GND	Digital ground
6	XOSC	AIO	2-Pin Oscillator output 1-pin Osillator input
7	VDDRTC	PWR	IRTC power
8	P10	I/O	GPIO ADC Input Ir in TMR2PWM SBC_UART TX WKUP SPICLK TK0
9	P15	I/O	GPIO ADC Input SDDAT SPICLK WKUP

Pin No.	Name	Type	Function
			Sys_clk_output Pll24M/12M output Rtc32K output
10	P16	I/O	GPIO ADC Input SDCLK SPIDIO Sys_clk_output Pll24M/12M output Rtc32K output WKUP
11	P17	I/O	GPIO VPP SDCMD SPIDI
12	P42	I/O	GPIO SDCLK USBDM2 ADC Input WKUP
13	P43	I/O	GPIO SDDAT USBDP2 ADC Input WKUP
14	P27/ P30	I/O	GPIO SBC_UART_RX Ir in ADC Input LVD_DET WKUP AUXR0 SDCMD
15	P32/ P35	I/O	GPIO AUXR1 AUXL2 UARTTX VCMBUF (double bonding)
16	DACL	AO	DAC Left Channel AUXR3 GPIO(VDDHP domain)

2.5 AX1076

2.5.1 Part Number

AX1076

2.5.2 Package

SSOP24

2.5.3 Pin Assignment

Figure 2-5 shows AX1076 SSOP24 Package.

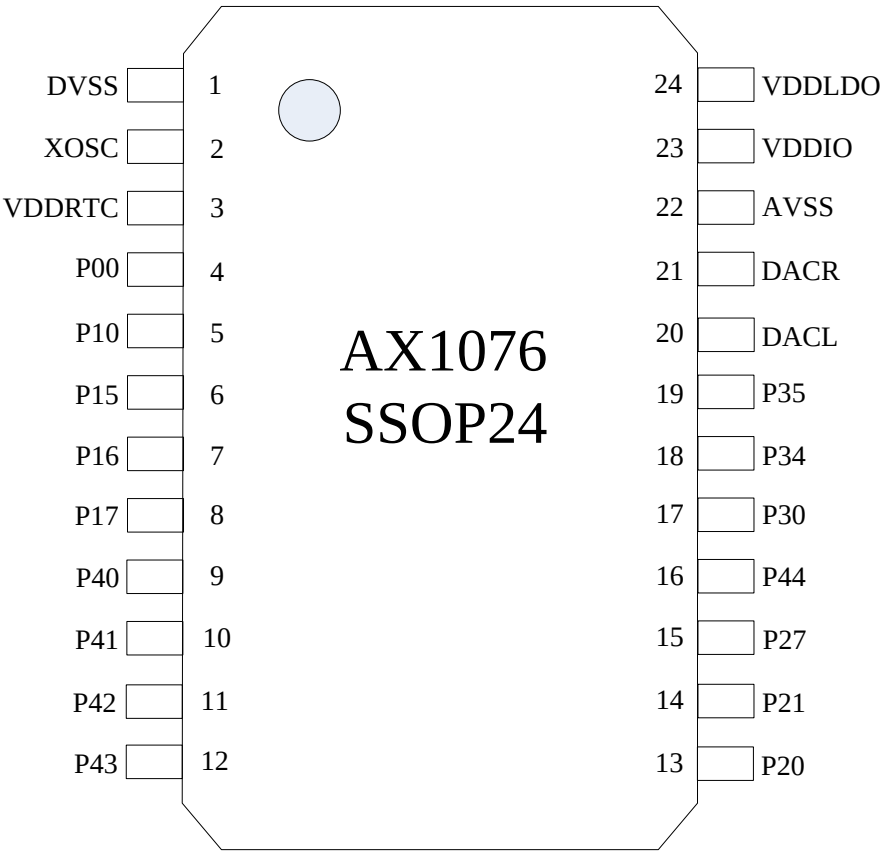


Figure 2-5 Pin Assignment for AX1076 SSOP24

2.5.4 Pin Description

Table 2-5 shows the pin descriptions of AX1076 SSOP24 package.

Table 2-5 AX1076 SSOP24 Pin Description

Pin No.	Name	Type	Function
1	DVSS	GND	Digital ground
2	XOSC	AIO	2-Pin Oscillator output

Pin No.	Name	Type	Function
			1-pin Osillator input
3	VDDRTC	PWR	IRTC power
4	P00	I/O	GPIO Mute WKUP TMR3PWM
5	P10	I/O	GPIO ADC Input Ir in TMR2PWM SBC_UART TX WKUP SPICLK TK0
6	P15	I/O	GPIO ADC Input SDDAT SPICLK WKUP Sys_clk_output Pll24M/12M output Rtc32K output
7	P16	I/O	GPIO ADC Input SDCLK SPIDIO Sys_clk_output Pll24M/12M output Rtc32K output WKUP
8	P17	I/O	GPIO VPP SDCMD SPIDI
9	P40	I/O	GPIO USBDM1 ADC Input WKUP
10	P41	I/O	GPIO USBDP1 WKUP ADC Input

Pin No.	Name	Type	Function
11	P42	I/O	GPIO SDCLK USBDM2 ADC Input WKUP
12	P43	I/O	GPIO SDDAT USBDP2 ADC Input WKUP
13	P20	I/O	GPIO
14	P21	I/O	GPIO
15	P27	I/O	GPIO SBC_UART_RX Ir in ADC Input LVD_DET WKUP
16	P44	I/O	GPIO SBC_UART_TX LCD_BL
17	P30	I/O	GPIO AUXR0 SDCMD
18	P34	I/O	GPIO AUXR2 UARTRX AM
19	P35	I/O	GPIO AUXL2 UARTTX VCMBUF (double bonding)
20	DACL	AO	DAC Left Channel AUXR3 GPIO(VDDHP domain)
21	DACR	AO	DAC Right Channel
22	AVSS	GND	DAC GND
23	VDDIO	PWR	IO Power
24	VDDLDO	PWR	LDO Power

3 CPU Core Information

3.1 Architecture

The AXC51-CORE of AX107X is fully compatible with the MCS-51™ instruction set. The AXC51-CORE employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12MHz. By contrast, the AXC51-CORE executes most of its instructions in 1 system clock cycle. With system clock running at 48 MHz, it has a peak throughput of 48 MIPS running in on-chip SRAM area.

3.2 Instruction Set

The instruction set of the AXC51-CORE is fully compatible with the standard MCS-51™ instruction set; standard 8051 development tools can be used to develop software for the AXC51-CORE. All instructions of AXC51-CORE are the binary and functional equivalent of their MCS-51TM counterparts, including op-codes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

Table 3-1 AXC51-CORE Instruction Set Summary

Number of Bytes	Mnemonic	Operands	Clock Cycles (running in IRAM)
1	NOP		1
2	AJMP	code addr	3
3	LJMP	code addr	3
1	RR	A	1
1	INC	A	1
1	INC	data addr	1
1	INC	@Ri	1
1	INC	Rn	1
3	JBC	bit addr, code addr	1 or 3
2	ACALL	code addr	3
3	LCALL	code addr	3
1	RRC	A	1
1	DEC	A	1
2	DEC	data addr	1
1	DEC	@Ri	1
1	DEC	Rn	1
3	JB	bit addr, code addr	1 or 3
1	RET		4
1	RL	A	1
2	ADD	A, #data	1
2	ADD	A, data addr	1

Number of Bytes	Mnemonic	Operands	Clock Cycles (running in IRAM)
1	ADD	A, @Ri	1
1	ADD	A, Rn	1
3	JNB	bit addr, code addr	1 or 3
1	RETI		4
1	RLC	A	1
2	ADDC	A, #data	1
2	ADDC	A, data addr	1
1	ADDC	A, @Ri	1
1	ADDC	A, Rn	1
2	JC	code addr	1 or 3
2	ORL	data addr, A	1
3	ORL	data addr, #data	1
2	ORL	A, #data	1
2	ORL	A, data addr	1
1	ORL	A, @Ri	1
1	ORL	A, Rn	1
2	JNC	code addr	1 or 3
2	ANL	data addr, A	1
2	ANL	data addr, #data	1
1	ANL	A, @Ri	1
1	ANL	A, Rn	1
2	JZ	code addr	1 or 3
2	XRL	data addr, A	1
3	XRL	data addr, #data	1
2	XRL	A, #data	1
2	XRL	A, data addr	1
1	XRL	A, @Ri	1
1	XRL	A, Rn	1
2	JNZ	code addr	1 or 3
2	ORL	C, bit addr	1
1	JMP	@A+DPTR	3
2	MOV	A, #data	1
3	MOV	data addr, #data	1
2	MOV	@Ri, #data	1
2	MOV	Rn, #data	1
2	SJMP	code addr	3
2	ANL	C, bit addr	1
1	MOVC*	A, @A+PC	1
1	DIV	AB	1
3	MOV	data addr, data addr	1
2	MOV	data addr, @Ri	1

Number of Bytes	Mnemonic	Operands	Clock Cycles (running in IRAM)
2	MOV	data addr, Rn	1
3	MOV	DPTR, #data	1
2	MOV	bit addr, C	1
1	MOVC*	A, @A+DPTR	2
2	SUBB	A, #data	1
2	SUBB	A, data addr	1
1	SUBB	A, @Ri	1
1	SUBB	A, Rn	1
2	ORL	C, bit addr	1
2	MOV	C, bit addr	1
1	INC	DPTR	1
1	MUL	AB	1
2	MOV	@Ri, data addr	1
2	MOV	Rn, data addr	1
2	ANL	C, bit addr	1
2	CPL	bit addr	1
2	CPL	C	1
3	CJNE	A, #data, code addr	1 or 3
3	CJNE	A, data addr, code addr	1 or 3
3	CJNE	@Ri, #data, code addr	1 or 3
3	CJNE	Rn, #data, code addr	1 or 3
2	PUSH	data addr	1
2	CLR	bit addr	1
1	CLR	C	1
1	SWAP	A	1
2	XCH	A, data addr	1
1	XCH	A, @Ri	1
1	XCH	A, Rn	1
2	POP	data addr	1
2	SETB	bit addr	1
1	SETB	C	1
1	DA	A	1
3	DJNZ	data addr, code addr	1 or 3
1	XCHD	A, @Ri	1
2	DJNZ	Rn, code addr	1 or 3
1	MOVX	A, @DPTR	2
1	MOVX	A, @Ri	2
1	CLR	A	1
2	MOV	A, data addr	1
1	MOV	A, @Ri	1
1	MOV	A, Rn	1

Number of Bytes	Mnemonic	Operands	Clock Cycles (running in IRAM)
1	MOVX	@DPTR, A	1
1	MOVX	@Ri, A	1
1	CPL	A	1
2	MOV	data addr, A	1
1	MOV	@Ri, A	1
1	MOV	Rn, A	1

3.3 Memory Mapping

3.3.1 Program Memory

As illustrated in *Figure 3-1*，AX107X program space incorporates 16KB OTP by CC1 = 1 which are mapped to 0x4000 ~ 0x7FFF, and 5376B IRAM which is mapped to 0x0000 ~ 0x14FF.

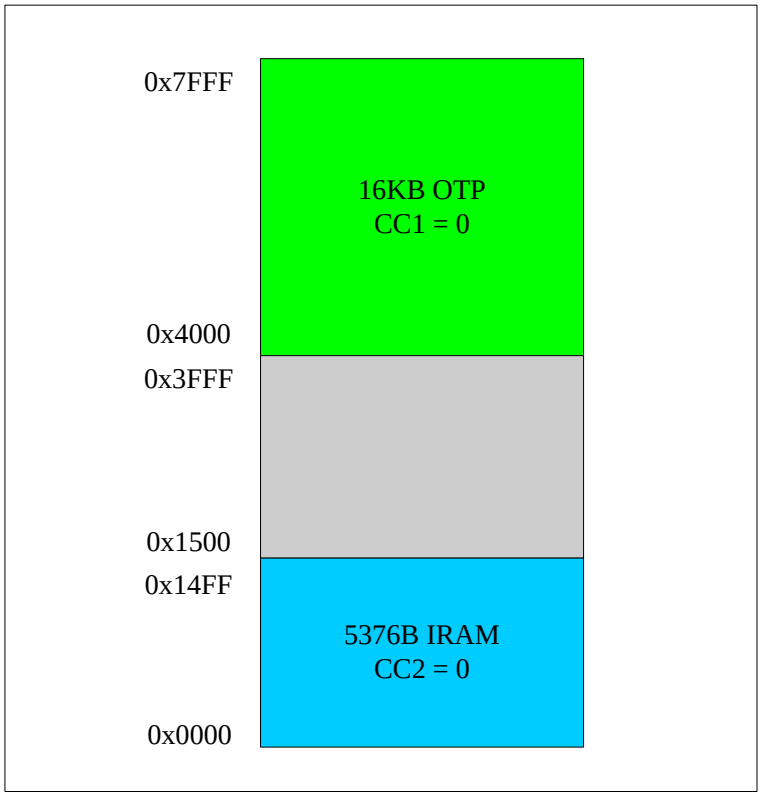


Figure 3-1 Program Memory Organization

3.3.2 External Data Memory Mapping

It also supports about 13K-byte on-chip SRAM (IRAM1, IRAM2 and DECRAM) XDATA Memory. Shows External Data Memory Mapping.

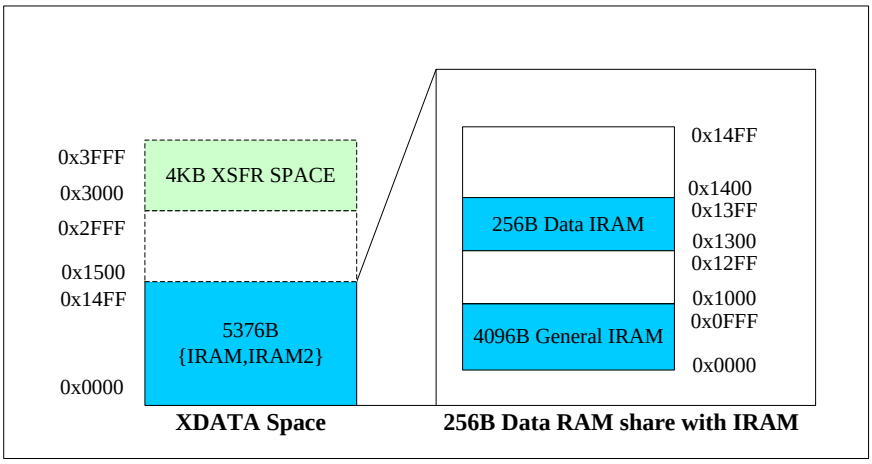


Figure 3-2 External Data Memory Mapping

3.3.3 Internal Data Memory Mapping

Internal data memory locates in SRAM1 at the address from 0x3F00 to 0x3FFF as showed in *Figure 3-2* Internal data memory is mapped in *Figure 3-3.* The memory space is shown divided into three blocks, which are generally referred to as the Lower 128, the Upper 128, and SFR space.

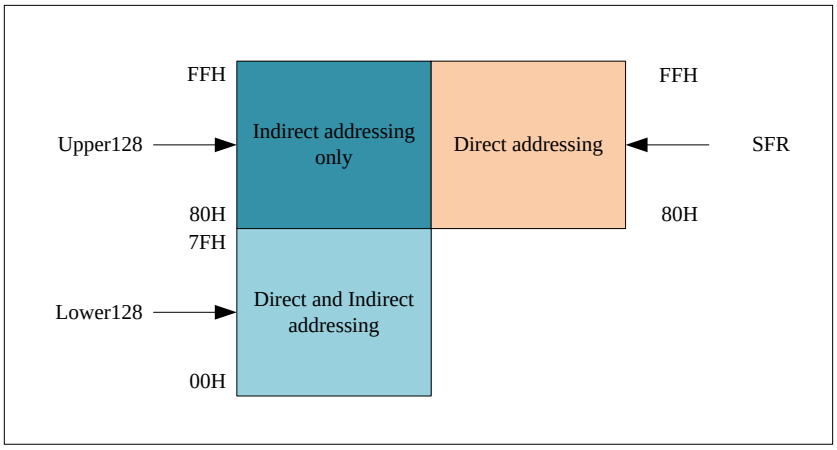


Figure 3-3 Internal data memory mapping

As shown in *Figure 3-4* the Lowest 32 bytes in Lower 128 are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the PSW select which register bank are in use.

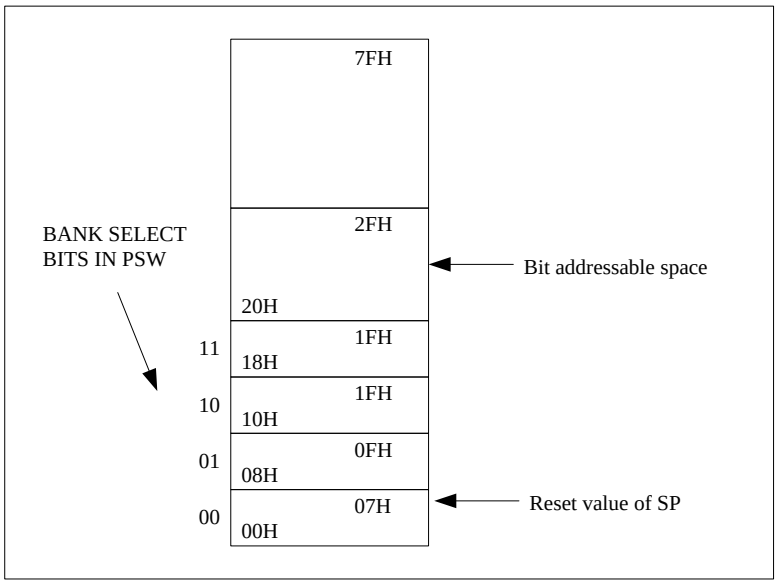


Figure 3-4 Lowest 32 bytes in Internal data memory Lower 128

3.4 Interrupt Processing

3.4.1 Interrupt sources

The AX107X provides 15 interrupt sources. All interrupts are controlled by a series combination of individual enable bits and a global enable (EA) in the interrupt-enable register (IE0.7). Setting EA to logic 1 allows individual interrupts to be enabled. Setting EA to logic 0 disables all interrupts regardless of the individual interrupt-enable settings. The interrupt enables and priorities are functionally identical to those of the 80C52.

The AX107X provides 3 sets of vectors entry addresses, starting from 0x0003, 0x4003. The vector base address is set by DPCON [7:6].*Table 3-2* shows interrupt summary

Table 3-2 Interrupt Summary

Interrupt Sources	Interrupt Vector	Interrupt Number	Natural Order	Interrupt Flag	Interrupt Enable Bit	Priority Control Bit
SINT	0x0003 / 0x4003	0	1	SPMODE.7	IE0.0	IP0.0
Timer 0	0x000B / 0x400B	1	2	TMR0CON.7	IE0.1	IP0.1
Timer 1 IR receiver 3in1 IO ADC	0x0013 / 0x4013	2	3	TMR1CON.7 TMR1CON.6 IRCON[2]	IE0.2 IRCON[1]	IP0.2
Timer 2	0x001B / 0x401B	3	4	TMR2CON.7 TMR2CON.6	IE0.3	IP0.3
MP3	0x0023 / 0x4023	4	5	AUCON7.6 AUCON7.5 AUCON7.4	IE0.4	IP0.4

Interrupt Sources	Interrupt Vector	Interrupt Number	Natural Order	Interrupt Flag	Interrupt Enable Bit	Priority Control Bit
Huffman				AUCON7.3 AUCON7.2 AUCON7.1 AUCON7.0 HFMCON.7 HFMCON.6		
SBC_uart_ovf	0x002B / 0x402B	5	6	UART1STA.1	IE0.5	IP0.5
USBSOF	0x0033 / 0x4033	6	7	USBCON2.1	IE0.6	IP0.6
USBCTL	0x003B / 0x403B	7	8		IE1.0	IP1.0
SDC	0x0043 / 0x4043	8	9	SDCON1.5 SDCON1.4	IE1.1	IP1.1
PORT	0x004B / 0x404B	9	10	WKPND	IE1.2	IP1.2
SPI0	0x0053 / 0x4053	10	11	SPI0CON.7	IE1.3	IP1.3
Timer 3 TK	0x005B / 0x405B	11	12	TMR0CON.7 TKENC[7],TKPND[7]	IE1.4	IP1.4
WDT	0x0063 / 0x4063	12	13	WDTCON.7 SPI1CON.7	IE1.5	IP1.5
RTCC UART Sbc_uart_normal LVD	0x006B / 0x406B	13	14	RTCON.7 UARTSTA.5 UARTSTA.4 UART1STA.2 UART1STA.3 LVDCON.7	IE1.6	IP1.6

3.4.2 Interrupt Priority

There are 2 levels of interrupt priority: Level 1 to 0. All interrupts have individual priority bits in the interrupt priority registers to allow each interrupt to be assigned a priority level from 1 to 0. All interrupts also have a natural hierarchy. In this way, when a set of interrupts has been assigned the same priority, a second hierarchy determines which interrupt is allowed to take precedence. The natural hierarchy is determined by analyzing potential interrupts in a sequential manner with the order listed in *Table 3-2*.

The processor indicates that an interrupt condition occurred by setting the respective flag bit. This bit is set regardless of whether the interrupt is enabled or disabled.

3.5 CPU and Memory related SFR Description

Register 3-1 DPCON – Data Pointer Configure Register

Position	7	6	5	4	3	2	1	0
Name	IA		DPID0	DPID1	DPAID	DPTSL	-	DPSEL
Default	1	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W

IA: Select Interrupt Vector's Base Address

00 = Base address is 0x0003

01 = Base address is 0x4003

1X = Reserved

DPID0: DPTR0 increase direction control

0 = DPTR increase

1 = DPTR decrease

DPID1: DPTR1 increase direction control

0 = DPTR increase

1 = DPTR decrease

DPAID: DPTR auto increment enable

0 = auto increment disable

1 = auto increment enable

DPTSL: DPSEL toggle enable

0 = DPSEL toggle disable

1 = DPSEL toggle enable

DPSEL: DPTR Select

0 = active DPTR0

1 = active DPTR1

The data pointers (DPTR0 and DPTR1) are used to assign a memory address for the MOVX instructions. This address can point to a MOVX RAM location. Two pointers are useful when moving data from one memory area to another. The user can select the active pointer through a dedicated SFR bit (DPSEL: DPCON.0), or can activate an automatic toggling feature for altering the pointer selection (DPTSL: DPCON.2). An additional feature, if selected, provides automatic incrementing or decrementing of the current DPTR.

Data pointer increment/decrement bits DPID0 (DPCON.5) and DPID1 (DPCON.4) define how the INC DPTR instruction functions in relation to the active DPTR.

The AX107X offers a programmable option that allows any instructions related to data pointer to toggle the DPSEL bit automatically. This option is enabled by setting the toggle-select-enable bit (DPTSL) to logic 1.

Once enabled, the DPSEL bit is automatically toggled after the execution of one of the following 5 DPTR related instructions:

MOVC A, @A+DPTR

MOVX A, @DPTR

MOVX @DPTR, A

INC DPTR

MOV DPTR, #data16

The AX107X also offers a programmable option that automatically increases (or decreases) the contents of the selected data pointer by 1 after the execution of a DPTR-related instruction. The actual function (increment or decrement) is dependent on the setting of the DPAID bits. This option is enabled by setting the automatic increment/decrement enable (DPAID: DPCON.3) to a logic 1 and is affected by one of the following 3 DPTR-related instructions.

DPTR-related instructions are:

MOVC A, @A+DPTR

MOVX A, @DPTR

MOVX @DPTR, A

Register 3-2 SP – Stack Pointer Low Byte

Position	7	6	5	4	3	2	1	0
Name	SP							
Default	0	0	0	0	0	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register 3-3 SPH – Stack Pointer High Byte

Position	7	6	5	4	3	2	1	0
Name	SPH							
Default	0	0	0	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

In a standard 8051, there is only an 8-bit stack pointer (SP). It can only use the internal 256 byte data memory as stack memory. To increase the stack space for more complex application, AX107X supports a 16-bit extend stack pointer, it can use both internal data RAM and the 15K byte on-chip SRAM as stack memory. There are 2 registers for stack control.

Register 3-4 PSW – Processor Status Word

Position	7	6	5	4	3	2	1	0
Name	CY	AC	F1	RS1	RS0	OV	F0	P
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CY: Carry Flag

AC: Auxiliary carry flag

F1: General purpose flag available for user

RS1, **RS0**: Register bank select

00 = bank0

01 = bank1

10 = bank2

11 = bank3

OV: Overflow flag

F0: General purpose flag available for user

P: Odd parity check of ACC

0 = There are even number of '1' bits in ACC

1 = There are odd number of '1' bits in ACC

Register 3-5 SPMODE – Special mode

Position	7	6	5	4	3	2	1	0
Name	SINT	XOSCOSTA	PWRUP	MIC_FCT_EN	DACRAMCEM	DECRAMCEM	IRAMCEM	IROMCEM
Default	0	0	1'b1	1'b1	1	1	1	1
Access	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W

SINT: Software interrupt pending

0 = no software interrupt

1 = software interrupt

XOSCOSTA: XOSCO state

1 = Normal mode

0 = Reserved

PWRUP: System power up flag

0 = CPU writes 0 to PWRUP.

1 = System power up or CPU writes 1 to PWRUP.

MIC_FCT_EN: MIC function enable

0 = Disable MIC function part

1 = Enable MIC function part

DACRAMCEM: DAC RAM CE mode control

0 = Always stay at 0

1 = Normal

DECRAMCEM: DECRAM CE mode control

0 = Always stay at 0

1 = Normal

IRAMCEM: IRAM CE mode control

0 = Always stay at 0

1 = Normal

IROMCEM: IROM CE mode control

0 = Always stay at 0

1 = Normal

Register 3-6 MEMCON – Memory Mapping Config

Position	7	6	5	4	3	2	1	0
Name	OTPDIV2	OTPEPT	MROMEPT	CC3	AUTO_EN	CC2	CC1	CC0
Default	0	0	1	0	0	0	0	0
Access	R/W	WO	WO	R/W	R/W	R/W	R/W	R/W

OTPDIV2: OTP clock divide 2 enable

0 = otp clock = system clock

1 = otp clock = system clock/2

OTPEPT: OTP Low 8K address encrypt enable

0 = OTP Low 2k-8K (7k space)address encrypt disable

1 = OTP Low 2k-8K (7k space)address encrypt enable

NOTE: This bit can be set, but can't be clear for protecting user program in OTP low 2k-8K address.

That is when this bit is "1", user can't access OTP low 8K address when program run at IRAM,MROM,DECRAM and high 8K OTP. only can be access When program running in low 8k OTP. when PC at 0x0000-0x1fff,cpu can not access OTP.

OTP low 1K mandatory encrypt by hardware, **can't be access anyway.**

MROMEPT:MROM encrypt enable

0 = access to MROM successful on whole MROM area

1 = access to MROM always mapping to lower 4K MROM area.

CC3: IROM3 mapping

0 = 4KB IROM3 disable, CC0 control IROM1、IROM2 map to address 0x2000~0x2ffff

1 = 4KB IROM3 map to address 0x2000~0x2fff

AUTO_EN: CODE 0xe0 and 0x f0 select

0 = Code fetch 0xe0f0 from address 0x000~0x1fff ,when CC2 =1

1 = Code fetch 0xf0f0 from address 0x0000~0x1fff ,when CC2=1

CC2: IRAM and memory copy mapping

0 = 5376B IRAM,IRAM2 map to address 0x0000~0x14ff

1 = 8KB Memory copy space map to address 0x000~0x1fff

CC1: OTP and DECRAM mapping

0 = OTP map to address 0x4000~0x7fff

1 = DECRAM map to address 0x4000~0x7fff

CC0: IROM1 and IROM2 mapping，when CC3==0

0 = IROM1 map to address 0x2000~0x3fff

1 = IROM2 map to address 0x2000~0x3fff

Register 3-7 IE0 – Interrupt Enable 0

Position	7	6	5	4	3	2	1	0
Name	EA	IE06	IE05	IE04	IE03	IE02	IE01	IE00
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EA: Global interrupt enable

0 = Disable

1 = Enable

IE06: USB SOF interrupt enable

0 = Disable

1 = Enable

IE05: sbc_uart_ovf enable

0 = Disable

1 = Enable

IE04: MP3 decoder/huffman interrupt enable

0 = Disable

1 = Enable

IE03: Timer2 interrupt enable

0 = Disable

1 = Enable

IE02: Timer1/ IR receiver /3in1_adc interrupt enable

0 = Disable

1 = Enable

IE01: Timer0 interrupt enable

0 = Disable

1 = Enable

IE00: SINT interrupt enable

0 = Disable

1 = Enable

Register 3-8 IE1 – Interrupt Enable 1

Position	7	6	5	4	3	2	1	0
Name	IE17	IE16	IE15	IE14	IE13	IE12	IE11	IE10
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IE17: reserved

IE16: RTCC/UART/LVD/sbc_uart_normal interrupt enable

0 = Disable

1 = Enable

IE15: WDT interrupt enable

0 = Disable

1 = Enable

IE14: Timer 3/TK interrupt enable

0 = Disable

1 = Enable

IE13: SPI interrupt enable

0 = Disable

1 = Enable

IE12: Port interrupt enable

0 = Disable

1 = Enable

IE11: SDC interrupt enable

0 = Disable

1 = Enable

IE10: USB control interrupt enable

0 = Disable

1 = Enable

Register 3-9 IP0 – Interrupt Priority 0

Position	7	6	5	4	3	2	1	0
Name	IP07	IP06	IP05	IP04	IP03	IP02	IP01	IP00
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IP07: reserved

IP06: USB SOF interrupt priority

0 = Disable

1 = Enable

IP05: sbc_uart_ovf interrupt priority

0 = Disable

1 = Enable

IP04: MP3 decoder/huffman interrupt priority

0 = Disable

1 = Enable

IP03: Timer2 interrupt priority

0 = Disable

1 = Enable

IP02: Timer1/ IR /3in1_adc receiver interrupt priority

0 = Disable

1 = Enable

IP01: Timer0 interrupt priority

0 = Disable

1 = Enable

IP00: SINT interrupt priority

0 = Disable

1 = Enable

Register 3-10 IP1 – Interrupt Priority 1

Position	7	6	5	4	3	2	1	0
Name	IP17	IP16	IP15	IP14	IP13	IP12	IP11	IP10
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IP17: reserved

IP16: RTCC/UART/LVD/sbc_uart_normal interrupt priority

0 = Disable

1 = Enable

IP15: WDT interrupt priority

0 = Disable

1 = Enable

IP14: Timer 3/TK interrupt priority

0 = Disable

1 = Enable

IP13: SPI interrupt priority

0 = Disable

1 = Enable

IP12: Port interrupt priority

0 = Disable

1 = Enable

IP11: SDC interrupt priority

0 = Disable

1 = Enable

IP10: USB control interrupt priority

0 = Disable

1 = Enable

4 Reset Generation

4.1 Power-on Reset（POR）

AX107X provides an on-chip Power-On-Reset (POR) circuit to detect power-on and to reset internal logic before VDD reaches the pre-determined POR threshold voltage. Under VDD=1.8V, the POR threshold voltage is set to be about 1.2V-1.3V.

Sometimes, when the VDD is power-off and quickly power-on again, there might be cases that the POR will work improperly and internal reset might not be generated. For this reason, AX107X POR circuit incorporates an internal self-reset module to discharge PORB output during power-off to ensure each power cycle will work properly.

However, it is also highly recommended user should have a long time between power-off and next power-on to ensure proper start-up. The time depends on actual system board environment and how much decoupling capacitors between power and ground. User has to take into account this effect during board level design.

Figure 4-1 illustrates the power-on and reset signals waveform during proper power-on. Internally, there is T_{POR} and T_{RC} time for both the POR circuit and the internal counter. T_{POR} is the time for the POR circuit to stay at zero voltage until it reaches V_{POR} and the time varies for different VDD rise-up time. It can be assumed to be about 2/3 of the VDD rise-up time. T_{RC} is the time for internal counter to count 4ms using internal RC-oscillator when the counter sees a high logic from PORB signal. As a result, the overall internal reset time is the sum of T_{POR} and T_{RC}. Such a long time is required to ensure the Power is stable for system use. It also ensures all internal logics are properly reset.

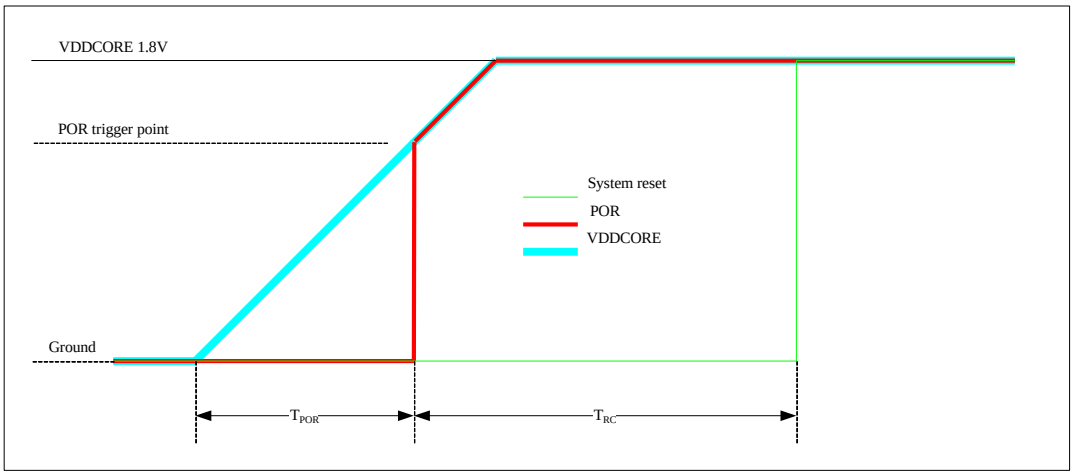


Figure 4-1 Power on reset

4.2 System Reset

All reset signals are OR'ed together inside the device to generate an overall system reset to reset the chip. Once reset, the program memory address is reset to 8000h, which is the start address of the Normal Mode.

Internal reset 6 sources include:

- Watchdog reset

- RTCC reset
- POR reset
- LVD reset
- ISD_CMD_RST
- TK_RST

External reset 2 sources include:

- Port wakeup reset
- IRTCC POR reset

4.3 Clock System

4.3.1 Clock Control

AX107X embeds X32KOSC internal oscillator circuits. External crystal is needed to generate a clock source. One internal PLL can generate 48MHz from the crystal clock source. One internal RC oscillator is also embedded.

To make sure the USB module operate properly, the USB clock must set to be 48MHz. In this case, system clock can be 48 MHz or 24M Hz.

Register 4-1 PCON0 – Power control 0

Position	7	6	5	4	3	2	1	0
Name	DRAMCEN	IRAMCEN	IROMCEN	IRAM2CEN	OTP_CLK_EN	IDLE	HOLD	SLEEP
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DRAMCEN: DECRAM clock enable

0 = Enable

1 = Disable

IRAMCEN: IRAM clock enable

0 = Enable

1 = Disable

IROMCEN: IROM clock enable

0 = Enable

1 = Disable

IRAM2CEN: IRAM2 clock enable

0 = Enable

1 = Disable

OTP_CLK_EN: OTP clock enable

0 = Enable

1 = Disable

IDLE: IDLE mode

0 = Disable

1 = Enable IDLE mode

HOLD: HOLD mode

0 = Disable

1 = Enable HOLD mode

SLEEP: SLEEP mode

0 = Disable

1 = Enable SLEEP mode

Register 4-2 PCON1 – Power control 1

Position	7	6	5	4	3	2	1	0
Name	DACCEN	MP3CEN	AUADCCEN	USBCEN	SDCEN	SPICEN	IRCEN	TMRCEN
Default	1	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DACCEN: DAC clock enable

0 = Enable

1 = Disable

MP3CEN: MP3 decoder clock enable

0 = Enable

1 = Disable

AUADCCEN: Audio ADC clock enable

0 = Enable

1 = Disable

USBCEN: USB clock enable

0 = Enable

1 = Disable

SDCCEN: SDC clock enable

0 = Enable

1 = Disable

SPICEN: SPI clock enable

0 = Enable

1 = Disable

IRCEN: IR reciever clock enables

0 = Enable

1 = Disable

TMRCEN: Timer clock enable

0 = Enable

1 = Disable

Register 4-3 PCON2 – Power control 2

Position	7	6	5	4	3	2	1	0
Name	TK_CLK_EN_	RC2MSM		UARTCEN	RTCCEN	WDTCEN	LVDCEN	ADCCEN
Default	0	0		0	0	0	0	0
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W

TK_CLK_EN_ : touch key module clock enable

0 = Enable

1 = Disable

RC2MSM: 2M RC sleep mode

0 = 2M RC is not disabled automatically by SLEEP mode

1 = 2M RC is disabled automatically by SLEEP mode

UARTCEN: UART clock enable

0 = Enable

1 = Disable

RTCCEN: RTCC clock enable

0 = Enable

1 = Disable

WDTCEN: WDT clock enable

0 = Enable

1 = Disable

LVDCEN: LVD clock enable

0 = Enable

1 = Disable

ADCCEN: ADC clock enable

0 = Enable

1 = Disable

Register 4-4 CLKCON – Clock control

Position	7	6	5	4	3	2	1	0
Name	X32KEN	X12MEN	RC12MEN	RC2MEN	XOSCSEL	HF_XOSCSEL	SCSEL	
Default	0	0	0	1	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

X32KEN: X32KOSC enable

0 = Disable 32K XOSC

1 = Enable 32K XOSC

X12MEN: 12M XOSC enable(The same function as X32KEN in AX107X)

0 = Disable 12M OSC

1 = Enable 12M OSC

RC12MEN: Internal 10M/14M RC enables

RC_CON[3]: 0->10M,1->14M

0 = Disable

1 = Enable

RC2MEN: Internal 2M RC enables

0 = Disable

1 = Enable

XOSCSEL: select XOSC(32K or 4/12/24M) as one of master clock source.union with RC clock

0 = disable

1 =enable

HF_XOSCSEL: External OSC select

1 = Select 12MHz OSC

0 = Select 32 KHz OSC

note: this bit is also used for pll clock reference select.

SCSEL: System clock select

00 = the mux out of RC clock and XOSC

01 = the 2M internal RC clock

10 = PLL(including OSCX2)

11 =the 10/14M internal RC clock

Register 4-5 CLKCON1 – Clock control 1

Position	7	6	5	4	3	2	1	0
Name	ATCLKSEL		-	DECDIV	SYSDIV		PLLDIVSEL	
Default	0	0	-	0	0	0	0	0
Access	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W

ATCLKSEL: Audio clock select

00 = invert of XOSC

01 = XOSC

10 = invert of PLL divide 2

11 = PLL divide 2

DECDIV: Decoder clock divide 2 from system clock

0 = Disable

1 = Enable

SYSDIV: System clock divide from clock source

00 = Divide 1

01 = Divide 2

10 = Divide 4

11 = Divide 8

PLLDIVSEL: PLL output divide select

00 = Divide 1

01 = Divide 2

10 = Divide 3

11 = Divide 4

Register 4-6 CLKCON2 – Clock control 2

Position	7	6	5	4	3	2	1	0
Name	RTCCS	WDTCSEL	IRCLKSEL[1:0]		DECDIV_SET	RCTM1EN	RCDIVSEL[1:0]	
Default	1'b0	1'b0	1'b0	1'b0	1'b0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

RTCCS: RTCC clock source select

0 = Select XOSC

1 = Select RC32K

WDTCSEL: WDT clock section

0 = RC32K

1 = XOSC

IRCLKSEL[1:0]: IR clock select

00 = PLL 1 MHz

01 = RC 1 MHz. not accurate

10 = XOSC 32KHz

11 =reserved

DECDIV_SET: decram clock 48M select bit.

0 =decram clock ,system clock working at the same.

1 =system clock work divide from decram clock.(decram clock is 48M,system clock divide set by CLKCON1[3:2]).

RCTM1EN: Enable RC clock to time1 inc source

0 = disable

1 = enable

RCDIVSEL[1:0] : RC to system clock select

00 = dive 8 namely system clock is about 512 KHz

01 = dive 4 namely system clock is about 1 MHz

10 = dive 2 namely system clock is about 2 MHz

11 = dive 0 namely system clock is about 4 MHz

Register 4-7 CLKCON3 –-CLOCK Configuration3

Position	7	6	5	4	3	2	1	0
Name	clkout_sel		XOSC_MODE_SEL	RC10MSEL	PLL_RC_SEL	PLLOSEL	PLLTUNEEN	PLLEN
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CLKOUT_SEL:

x1 = output rtc_clk by P16/P15/P06/P32/P33/P34/P35（higher priority）

1x = output sys_clk by P16/P15/P06/P32/P33/P34/P35 (lower priority)

XOSC_MODE_SEL: xosc one/two pin mode select

0 = two pin

1 = one pin

RC10MSEL:

0 = RC output 1M/512K from 2M RC

1 =RC output 1M/512K from 10/14M RC

PLL_RC_SEL: pll rc clk select

0 = select XOSC

1 = select rc10/14 MHz

PLLOSEL: PLL source select

0 = PLL come from internal pll block

1 = PLL come from internal clock X2 output

PLLTUNEEN: pll tune enable

0 = diable

1 = enable

PLLEN: PLL enable

0 = Disable

1 = Enable

Register 4-8 CLKCON4 – clock Configuration4

Position	7	6	5	4	3	2	1	0
Name	FRAC_DR_UPDATE	TUNE_WR	Reserved	Reserved	Reserved	Reserved	pllcfg	
Default	0	0	0	0	0	0	11	
Access	WO	WO	R/W	R/W	R/W	R/W	R/W	

FRAC_DR_UPDATE:

When write 1 to this register,update Frac Dr value to Frac module,and this operation must after of pllen be setted..

TUNE_WR: pll tune wr

0 = nothing

1 = trigger pll tune

pllcfg:

change gbw for different clock input

00 =10/14M internal RC or 12M/24M XOSC

01 =4M XOSC

1X = 32KHz XOSC

4.3.2 Clock Gating

AX107X provides comprehensive clock gating options for eliminating power-wasting activities. As shown in *Figure 4-2* system clock supplies clock signal to different clock domains. Every clock can be gated. It allows the user to shut

down the clock signal when the function is not needed.

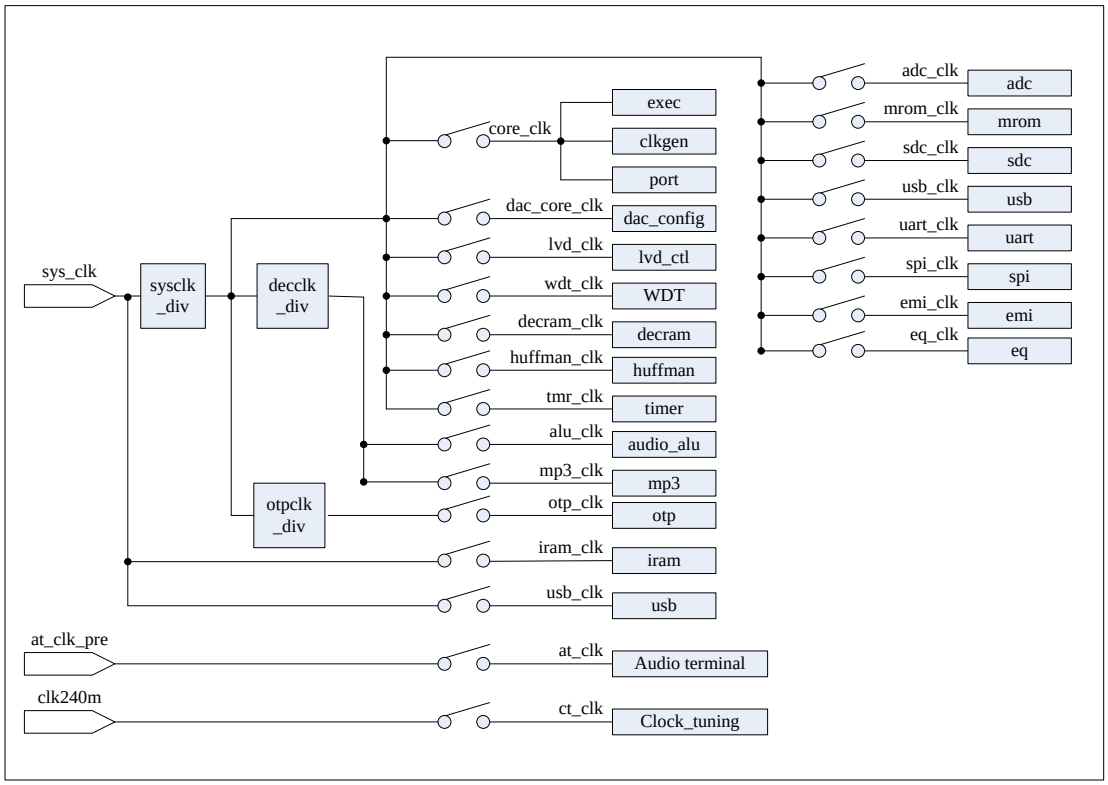


Figure 4-2 Sub-clock domains of system clock

4.3.3 Phase Lock Loop (PLL)

AX107X provides one on-chip Phase Locked Loop (PLL 48M) clock generators. The PLL has reference clock from external 32KHz,4M,12M,24M crystal,or internal 10M/14M RC oscillator to provide a stable reference clock and the reference clock is multiplied to provide the final system clock.

Register 4-9 PLLINTL – PLL integer low

Position	7	6	5	4	3	2	1	0
Name	INTL							
Default	0	0	1	0	0	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register 4-10 PLLFRACH – PLL fraction high

Position	7	6	5	4	3	2	1	0
Name	FRACH							
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register 4-11 PLLFRACL – PLL fraction low

Position	7	6	5	4	3	2	1	0
Name	FRACL							

Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

FRAC = fraction*65535;

Register 4-12 DRSIG_INTH clock tuning upper limit integer high part

Position	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	DISTE	CT_PATCH	SOFONLY	DRSIG_INTH		
Default	0	0	0	0	0	0	0	0
Access	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W

DRSIG_INTH: limit clock tuning upper range,integer part higher 3bit

SOFONLY: clock tuning refer to only SOF package

1 = enable

0 = disable

CT_PATCH: control behavior of clock tuning

1 = clock tuning stop work after first tuning finish.and shut down clock tuning module at same time.but new dr value have be update to cpu register.

0 = clock tuning module work all time during USB transfer

DISTE:

1 = disablet adjust range

0 = enable adjust

Register 4-13 DRSIG_INTL clock tuning limit integer low part

Position	7	6	5	4	3	2	1	0
Name	DRSIG_INTL							
Default	0	0	0	0	0	0	0	0
Access	R/W	R./W	R/W	R/W	R/W	R/W	R/W	R/W

Register 4-14 DRSIG_FRACH clock tuning limit frac high part

Position	7	6	5	4	3	2	1	0
Name	DRSIG_FRACH							
Default	0	0	0	0	0	0	0	0
Access	R/W	R./W	R/W	R/W	R/W	R/W	R/W	R/W

Register 4-15 DRSIG_FRACL clock tuning limit frac low part

Position	7	6	5	4	3	2	1	0
Name	DRSIG_FRACL							
Default	0	0	0	0	0	0	0	0
Access	R/W	R./W	R/W	R/W	R/W	R/W	R/W	R/W

DRSIG register limit clock tuning work range by DR±DRSIG

Slect XOSC or RC as input for PLL,

Configure PLL frequency：

1. PLL frequency division

PLL input reference clock is f0 (from RC or OSC), its internal dividing clock is 60M, frequency dividing ratio is 60M/f0.

Clock divide ratio consist of integer and decimal, the default value of integer part is 1831(default reference clock is 32.768k), the default value of decimal part is 0(only integral frequency division this time).

If f0=32.768k, frequency dividing ratio is 1831, decimal fraction part set 0.

If f0=12M, frequency dividing ratio is 5, decimal fraction part set 0.

If f0=4M, frequency dividing ratio is 15, decimal fraction part set 0.

2. If frequency dividing ratio is 58.a, then integer set 58, decimal fraction is a*65535.

Configure PLL input：

Select XOSC32K as PLL input：

```
// set pll clock
//PLLINTH = 1831 >> 8;            // 60M/32.768K//integer part is optimized
PLLINTL = 1831 & 0xff;
PLLFRACH = 0;
PLLFRACL = 0;
// select xosc 32k as pll reference clock
AIPCON5 &= ~BIT(2);          // disable pll_test_ref_sel
CLKCON3 &= ~BIT(3);          // select xosc as pll clock
CLKCON    &= ~BIT(2);        // HFXOSCSEL select xosc32k    as xosc
rtc_wdat    = 0;
rtc_wdat |= BIT(2) | BIT(7);          // enable F32KEN, X32KEN(OSC_EN_RTC)
irtcc_wr_n_byte(WRITE_CFG, 1, 0, &rtc_wdat);
CLKCON4 |= BIT(7);        // enable pll_en_v
CLKCON3 |= BIT(0);        // enable pll_en
DELAY1MS();
```

Select XOSC12M as PLL input：

```
// set pll clock
// PLLINTH = 5 >> 8;              // 60M/12M
PLLINTL = 5 & 0xff;
PLLFRACH = 0;
PLLFRACL = 0;
// select xosc 12m as pll reference clock
AIPCON5 &= ~BIT(2);          // disable pll_test_ref_sel
CLKCON3 &= ~BIT(3);          // select xosc as pll clock
CLKCON    |=    BIT(2);      // HFXOSCSEL select xosc12m    as xosc
rtc_wdat    = 0;
rtc_wdat |= BIT(6);              // enable X12MEN(OSC_EN_RTC)
irtcc_wr_n_byte(WRITE_CFG, 1, 0, &rtc_wdat);
CLKCON4 |= BIT(7);           // enable pll_en_v
```

```
CLKCON3 |= BIT(0);          // enable pll_en
DELAY1MS();
```

Select RC10M as PLL input：

```
// set pll clock
// PLLINTH = 6 >> 8;        // 60M/10M
PLLINTL = 6 & 0xff;
PLLFRACH = 0;
PLLFRACL = 0;
// select xosc 12m as pll reference clock
RC_CON     &= ~BIT(3);      // select rc10m
CLKCON    |=  BIT(5);       // enable RCOS_EN
CLKCON3 |=  BIT(3);         // select rc as pll clock
AIPCON5 &= ~BIT(2);         // disable pll_test_ref_sel
CLKCON4 |= BIT(7);          // enable pll_en_v
CLKCON3 |= BIT(0);          // enable pll_en
DELAY1MS();
```

4.3.4 RC

Register 4-16 RC_CON –RC control

Position	7	6	5	4	3	2	1	0
Name	TRIM_CLKO_SEL	RC2M_SEL	TRIM_CLK_SEL		RC_TRIM	Reserved	LDO_SEL	
Default	0	0	0	0	0	0	0	0
Access	R/W		R/W	R/W	R/W	R/W	R/W	R/W

TRIM_CLKO_SEL: trim_clk_sel from trim_clk and rc_sys_clk

0 = trim_clk (high speed about 12M)

1 = rc_sys_clk (512K,1M,2M,4M)

RC2M_SEL:

0 =rc clock source come from trim_clk

1 =rc clock source come from 10M/14M RC

TRIM_CLK_SEL:

00 =trim clk is 10M/14M RC

01 =trim clk is XOSC

10 =trim clk is 12M PLL.

11 = trim clk tie to 0

RC_TRIM:

0 =10M/14M RC output 10M

1 =10M/14M RC Output 14M

RCLDO_SEL: RC LDO output value select bit

00 =output 1.80v (Default)

01 =output 1.70v

10 = output 1.90v

11 = reserved

5 LOW POWER MANAGEMENT

5.1 Power Saving Mode

AX107X device has the low power management mode that can help reducing power consumption when the device does not require intensive CPU resources and speed. There are three low power modes available: SLEEP mode, Hold mode and IDLE mode.

5.1.1 Sleep mode

SLEEP mode is an ultimate power reduction mode that will stop all the clock sources, and all the memory chip select signals are disabled to further reduce power consumption. However, before entering sleep mode, all peripherals should be disabled separately, especially those analog peripherals and memory, unless those peripherals will stop themselves if no clock source is applied to the peripherals.

Note: Before Entering SLEEP mode, the system clock is recommended to change back to oscillator clock as the system clock.

To enter SLEEP mode, user need to write a '1' to SLEEP register (Bit0 of PCON0).

During SLEEP mode, the device can be wake up by either external port wakeup reset or watchdog reset or RTCC reset or IR receiver reset.

After exit SLEEP mode by wakeup, the device will be reset.

SLEEP mode will DECRAM and IRAM and system clock automatically.

5.1.2 Hold mode

HOLD mode will stop the clock from entering to system. The system clock is gated with the HOLD mode control. Once enter HOLD mode, clock to the system logic halts. Therefore, there will be no clock switching entering the system logic so that power consumption is minimized due to no AC switching. However, the clock sources are not disabled and they are still running. This allows the clock to be resumed in real time without waiting for the PLL to lock again. Watchdog interrupt, RTCC interrupt, Port interrupt, IR receiver interrupt and all reset event will cause system to exit HOLD mode.

TO enter HOLD mode, user need to write a '1' to HOLD register (Bit1 of PCON0).

When wakeup from HOLD Mode by port or RTCC or IR receiver, if interrupt is enabled, AX107X enters corresponding interrupt service subroutine (ISR), else AX107X will execute the instruction following HOLD.

When wakeup from HOLD Mode by watchdog, if watchdog reset enable, AX107X will be reset, else if watchdog interrupt is enabled, AX107X will enter watchdog's ISR. Else AX107X will execute the instruction following HOLD.

5.1.3 Idle mode

IDLE mode will stop the clock from entering to the CPU. The CPU clock is gated with the IDLE mode control. Once enter IDLE mode, clock to the CPU logic halts. Therefore, there will be no clock switching entering the CPU logic so

CPU power consumption is minimized.

All interrupt sources will cause system to exit IDLE mode, which include all peripheral interrupt.

TO enter IDLE mode, user need to write a '1' to IDLE register (Bit2 of PCON0).

When exit IDLE mode, AX107X will enter interrupt service subroutine if EA is enable. If EA is disabled, the instruction next to IDLE will be executed.

5.2 LDO

AX107X provides two on-chip low drop-out regulators (LDO) to convert from 5V to 3.3V and 3.3V to 1.8V to for internal core power use. It is there to provide high power supply noise rejection and also to minimize power consumption. LDO is always enabled.

To provide a more stable and reliable power source to internal core logic, it is recommended to add frequency compensation through external component. *Figure 5-1* shows the connection.

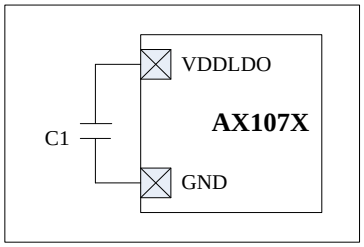


Figure 5-1 Frequency compensation through external component

Note:

- The recommended value for C1 is 10uF.
- C1 should be placed closely to the chip.

6 PORT

6.1 Overview

The general purpose input/output port (GPIO) provides 35 dedicated general purpose one-bit contacts that can be individually configured as either inputs or outputs. Contacts configured as outputs reflect internal register values, and those configured as inputs can be detected by reading internal registers. All GPIOs are divided into 4 groups: Port0, Port1, Port2, Port3 and Port4.

AX107X provides five ports (Port 0/1/2/3/4) for user to develop applications. Inputs are all Schmitt triggered with about 400-500mV hysteresis level to filter input voltage fluctuations. Each port pin can be independently set as input or output. Most of the port pins are built-in slew-rate controlled to reduce output bouncing noise. There is one internal 10KΩ pull-up and one 10KΩ pull-down resistor selectable for each input port pin.

Port 0/1/2 has 8 single IOs each, but Port 3 has only 6 IOs(P30~P35), and Port 4 has only 6 IOs(P40~P45).

NOTE:

1. Before using P40 and P41, AIPCON7.0 must be set as "1".
2. Before using P42 and P43,AIPCON7.1 must be set as " 1".
3. Before using P45, PIE1.7 must be clear as "0".this bit is cleared by default.
4. P4.5 is set as analog input IO by default and its reference voltage is VDDHP.

Table 6-1 Pad Type

Type	Driving (mA)			Pull-up resistor (Kohm)			Pull-down resistor (Kohm)			Mode
A	8	3	-	10	-	-	10	-	-	
B	24	8	-	10	-	-	10	-	-	
C	8	3	-	10	0.5	-	10	0.5	-	
D	24	8	3	10	-	-	10	0.2	-	
E	-	-	-	10	-	-	10	-	-	

6.2 Function multiplexing

In order to provide more flexible port functions and to minimize pin counts, some of the ports are multiplexed with other peripherals or functions. *Table 6-2* illustrates the “Ports MUXed mapping”.

Table 6-2 Port Mapping

IO pins	Function 1	Function 2	Function 3	Driving (mA)
P00	MUTE	wkup	Tmr3pwm	A
P01	ADC1,TK5/IR	Wkup	tmr2cap	A
P02	ADC0	TMR0_INC/IR/WKUP	-	A
P03	ADC0	TMR1_INC/IR/WKUP	-	B
P04	-	TMR1PWM	-	B

IO pins	Function 1	Function 2	Function 3	Driving (mA)
P05	-	TMR0PWM	-	B
P06	WKUP	TMR0CAP/TMR1CAP	SDCLK	B
P07	-	TMR1CAP	-	B
P10	ADC/IR,TK0	IR/WKUP/tmr2pwm/spiclk	SBC_UART_TX	A
P11	TK1	TMR3_INC/wkup/spidio	SBC_UART_RX	A
P12	TK2	WKUP/SPICLK	SDDAT	A
P13	TK3	WKUP/TMR3CAP/TMR2CAP	SDCLK/SPIDIO	A
P14	TK4/ADC2	TMR2_INC/WKUP	SDCMD/SPIDO	A
P15	ADC3	WKUP	SPI0CLKB/SDDAT/ OSCOUT	A
P16	ADC4	WKUP	SPIDIO/SDCLK/ OSCOUT	A
P17	VPP	-	SPI0DI/SDCMD	A
P20	-	-	-	C
P21	-	-	-	C
P22	-	-	-	C
P23	-	-	-	C
P24	-	-	-	C
P25	-	-	-	C
P26	-	-	-	C
P27	LVD_EXT/ADC0/IR	WKEUP	SBC_UART_RX	C
P30	AUXR0	-	SDCMD	C
P31	AUXL0	LCD_BL	SDDAT	D
P32	AUXR1	-	-	A
P33	AUXL1	-	-	A
P34	AUXR2	-	UART_RX	A
P35	AUXL2	-	UART_TX	A
P40	ADC10	USBDM1	WKUP	A
P41	ADC11	USBDP1	WKUP	A
P42	ADC12	USBDM2	WKUP/SDCLK	A
P43	ADC13	USBDP2	WKUP/SDDAT	A
P44	-	-	SBC_UART_TX	D
P45	DACL/AUXR3	-	-	E

6.3 Port Interrupt and wakeup

AX107X supports Port Wakeup. The PWKEN registers (Wakeup Enable Register) allow up 17 IO to cause wakeup.

The PWKEN registers are set to 0Fh by default. Clearing bit0-3 in the PWKEN register enables wakeup on corresponding pin. The trigger condition on the selected pin can be either rising edge or falling edge. The WKED register (Wakeup Edge Select) selects the desired transition edge. Setting a bit in WKED register selects the falling edge of the corresponding pin. Resetting the bit selects the rising edge.

Once a valid transition occurs on the selected pin, the WKPND (PWKEN.7~PWKEN.4) register (Wakeup Pending

Register) latches the transition in the corresponding bit position. Logic '1' indicates the occurrence of the selected trigger edge on the corresponding Port pins. Upon reset, logic '0' is set to all bits of WKPND.

Note:

Port 0 Wakeup initialization, to avoid any false signaling to port, the followings would be a recommended procedure for Wakeup initialization:

- Configure the edge select of Port 0 pins on WKEDG register,
- Clear the corresponding bits on WKPND Wakeup Pending Register
- Clear the corresponding bits in the PWKEN registers to enable the wakeup on the corresponding port pins

Upon exiting the sleep down mode, the Multi-Input Wakeup logic causes full chip reset.

6.4 GPIO Special Function Register

6.4.1 Data Direction configuration

There are five port data direction registers: P0DIR, P1DIR, P2DIR, P3DIR and P4DIR. All port pins are defined as "output" when it is set as "0" and as "input" when it is set as "1".

Register 6-1 P0DIR-P0 direction Register

Position	7	6	5	4	3	2	1	0
Name	P0DIR							
Default	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P0xDIR: P0x direction control

0 = Output

1 = Input

Register 6-2 P1DIR-P1 direction Register

Position	7	6	5	4	3	2	1	0
Name	P1DIR							
Default	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1xDIR: P1x direction control

0 = Output

1 = Input

Register 6-3 P2DIR-P2 direction Register

Position	7	6	5	4	3	2	1	0
Name	P2DIR							
Default	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2xDIR: P2x direction control

0 = Output

1 = Input

Register 6-4 P3DIR-P3 direction Register

Position	7	6	5	4	3	2	1	0
Name	P3DIR							
Default	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3xDIR: P3x direction control

0 = Output

1 = Input

Register 6-5 P4DIR-P4 direction Register

Position	7	6	5	4	3	2	1	0
Name	Reserved	P4DIR						
Default	0	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P4xDIR: P4x direction control

0 = Output

1 = Input

6.4.2 Port Data configuration

There are five port data registers: P0, P1, P2,P3 and P4. The port data value is stored as “0” when Px register is set to “0” and as “1” when Px register is set to “1”.

Register 6-6 P0 – P0 data register

Position	7	6	5	4	3	2	1	0
Name	P0							
Default	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P0[x]: P0x data. Valid when P0x is used as GPIO

0 = P0x is in low state when read and output low at P0x when write

1 = P0x is in high state when read and output high at P0x when write

Register 6-7 P1 – P1 data register

Position	7	6	5	4	3	2	1	0
Name	P1							
Default	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1[x]: P1x data. Valid when P1x is used as GPIO

0 = P1x is in low state when read and output low at P1x when write

1 = P1x is in high state when read and output high at P1x when write

Register 6-8 P2 – P2 data register

Position	7	6	5	4	3	2	1	0
Name	P2							
Default	x	x	x	x	x	x	x	x
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

P2[x]: P2x data. Valid when P2x is used as GPIO

0 = P2x is in low state when read and output low at P2x when write

1 = P2x is in high state when read and output high at P2x when write

Register 6-9 P3 – P3 data register

Position	7	6	5	4	3	2	1	0
Name	P3							
Default	x	x	x	x	x	x	x	x
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

P3[x]: P3x data. Valid when P3x is used as GPIO

0 = P3x is in low state when read and output low at P3x when write

1 = P3x is in high state when read and output high at P3x when write

Register 6-10 P4 – P4 data register

Position	7	6	5	4	3	2	1	0
Name	Reserved	P4						
Default	x	x	x	x	x	x	x	x
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

P4[x]: P4x data. Valid when P4x is used as GPIO

0 = P4x is in low state when read and output low at P4x when write

1 = P4x is in high state when read and output high at P4x when write

6.4.3 Drive selection

There are 5 drive ability selection register for P0,P1,P2,P3,P40-P44..

Table 6-3 shows the Detail description.

P17 output drive always is 8mA.

Table 6-3 DSx register setting

Register	**Address**	**Set bit as "1"**	**Clear bit as "0"**	**Initial value**
DS0[2:0]	R/W	P0[2:0] drive ability select 8mA	P0[2:0] drive ability select 3mA	3'b000
DS0[7:3]	R/W	P0[7:3] drive ability select 24mA	P0[7:3] drive ability select 8mA	5'b0_0000
DS1[6:0]	R/W	P1[6:0] drive ability select 8mA	P1[6:0] drive ability select 3mA	00h
DS2[7:0]	R/W	P2[7:0] drive ability select 8mA	P2[7:0] drive ability select 3mA	00h
DS3[5:2],DS3[0]	R/W	P3[5:2],P3[0] drive ability select 8mA	P3[5:2],P3[0] drive ability select 3mA	1'b0
{DS3[6], DS3[1]}	R/W	0X = P3[1] drive ability select 3mA		2'b00

Register	Address	Set bit as "1"	Clear bit as "0"	Initial value
		10 = select 8mA 11= select 24mA		
DS4[3:0]	R/W	P4[3:0] drive ability select 8mA	P4[3:0] drive ability select 3mA	00h
{DS4[5], DS4[4]}	R/W	0X = P4[4] drive ability select 3mA 10 = select 8mA 11 = select 24mA		2'b00

6.4.4 Pull-up and Pull-down configuration

There are six data pull-up registers: PUP0, PUP1, PUP2,PUP2L,PUP3 and PUP4. The port pins will be pull-up disabled when PUPx or PUP2EN is set to "0" or when set to output, when pull-up enabled it is set to "1" and the pin must set as input.

There are six data pull-down registers: PDN0, PDN1,PDN2,PDN2L, PDN3 and PDN4. The port pin will be pull-down disabled when PDNx register is set to "0" or the pin is set as output, and pull-down enabled when it is set to "1" and the pin is set as input.

PUP2L is port2 10K/500R pull-up resister select bit, 0: select 10K,1 select 500R

PUP2 is port2 pullup enable bit,0 : disable ,1 : enable

PDN2L is port2 10 K/500R pull-down resister select bit, 0: select 10K,1 select 500R

PDN2 is port2 pulldown enable bit,0 : disable ,1 : enable

Register 6-11 PUP0 – P0 pull-up resistor control

Position	7	6	5	4	3	2	1	0
Name	PUP0							
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PUP0: P0 10KΩ pull-up resister control. Valid when P0 is used as input

0 = 10KΩ pull-up resistor disabled

1 = 10KΩ pull-up resistor enabled

Register 6-12 PUP1 – P1 pull-up resistor control

Position	7	6	5	4	3	2	1	0
Name	PUP1							
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PUP1[4:0], PUP1[6]: P1[4:0],P1[6] 10KΩ pull-up resister control. Valid when P1[4:0],P1[6] is used as input

0 = 10KΩ pull-up resistor disabled

1 = 10KΩ pull-up resistor enabled

PUP1[7],PUP1[5]: P1[7],P1[5] 10KΩ pull-up resister control. Valid when P1[7],P1[5]is used as input

0 = pull-up resistor disabled

1 = pull-up resistor enabled

Note: P17, P15 0.5KΩ pull-up resister control is in the register **AIPCON5**.

Register 6-13 PUP2L – P2 10K/500Ω pull-up resistor control

Position	7	6	5	4	3	2	1	0
Name	PUP2L							
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PUP2L: P2 10K/500R pull-up resister control. Valid when P2 is used as input

0 = 10KΩ pull-up

1 = 0.5KΩ pull-up

Register 6-14 PUP2 – P2 pull-up resistor control

Position	7	6	5	4	3	2	1	0
Name	PUP2							
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PUP2: P2 pull-up resister control. Valid when P2 is used as input

0 = pull-up resistor disabled

1 = pull-up resistor enabled

Register 6-15 PUP3 – P3 pull-up resistor control

Position	7	6	5	4	3	2	1	0
Name	PUP3							
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PUP3: P3 10KΩ pull-up resister control. Valid when P3 is used as input

0 = 10KΩ pull-up resistor disabled

1 = 10KΩ pull-up resistor enabled

Register 6-16 PUP4 – P4 pull-up resistor control

Position	7	6	5	4	3	2	1	0
Name	Reserved	PUP4						
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PUP4: P4 10KΩ pull-up resister control. Valid when P4 is used as input

0 = 10KΩ pull-up resistor disabled

1 = 10KΩ pull-up resistor enabled

Register 6-17 PDN0 – P0 10KΩ pull-down resistor control

Position	7	6	5	4	3	2	1	0
Name	PDN0							
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PDN0[x]: P0x 10KΩ pull-down resister control. Valid when P0x is used as input

0 = 10KΩ pull-down resistor disabled

1 = 10KΩ pull-down resistor enabled

Register 6-18 PDN1 – P1 10KΩ pull-down resistor control

Position	7	6	5	4	3	2	1	0
Name	PDN1							
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PDN1[x]: P1x 10KΩ pull-down resister control. Valid when P1x is used as input

0 = 10KΩ pull-down resistor disabled

1 = 10KΩ pull-down resistor enabled

Register 6-19 PDN2L – P2 10K/500Ω pull-down resistor control

Position	7	6	5	4	3	2	1	0
Name	PDN2L							
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PDN2L: P2 10K/500R pull-down resister control. Valid when P2 is used as input

0 = 10KΩ pull-up

1 = 0.5KΩ pull-up

Register 6-20 PDN2 – P2 10KΩ pull-down resistor control

Position	7	6	5	4	3	2	1	0
Name	PDN2							
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PDN2[x]: P2x 10KΩ pull-down resister control. Valid when P2x is used as input

0 = 10KΩ pull-down resistor disabled

1 = 10KΩ pull-down resistor enabled

Register 6-21 PDN3 – P3 10KΩ pull-down resistor control

Position	7	6	5	4	3	2	1	0
Name	Reserved		PDN3					
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PDN3[x]: P3x 10KΩ pull-down resister control. Valid when P3x is used as input

0 = 10KΩ pull-down resistor disabled

1 = 10KΩ pull-down resistor enabled

Register 6-22 PDN4 – P4 10KΩ pull-down resistor control

Position	7	6	5	4	3	2	1	0
Name	Reserved		PDN4					
Default	0	0	0	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

PDN2[x]: P4x 10KΩ pull-down resister control. Valid when P4x is used as input

0 = 10KΩ pull-down resistor disabled

1 = 10KΩ pull-down resistor enabled

Table 6-4 PDNx register setting

Register	Address	Set bit “x” of PxDIR as “1”	Clear bit “x” of PxDIR as “0”	Initial value
PDN0	R/W	Enable pull-down	Disable pull- down	00h
PDN1	R/W	Enable pull- down	Disable pull- down	00h
PDN2L	R/W	Low resister 500R pull-down select	10K pull- down select	00h
PDN2	R/W	Enable pull-down	Disable pull-down	00h
PDN3	R/W	Enable pull- down	Disable pull- down	00h
PDN4		Enable pull- down	Disable pull- down	00h

6.4.5 Digital input enable

There is 2 digital input enable register: PIE0 and PIE1. There are several I/O MUXed with analog module. I/O digital input and output must be disabled when Analog Module is enabled.

Register 6-23 PIE0 – Port digital input enable control

Position	7	6	5	4	3	2	1	0
Name	PIE07	PIE06	PIE05	PIE04	PIE03	PIE02	PIE01	PIE00
Default	1	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PIE07: P16 Digital Input Enable Bit (For ADKEY6)

0 = P16 digital Input Disabled

1 = P16 digital Input Enabled

PIE06: P15 Digital Input Enable Bit (For ADKEY5)

0 = P15 digital Input Disabled

1 = P15 digital Input Enabled

PIE05: P10 digital input enable bit (For ADC7 input)

0 = P10 Input Disabled

1 = P10 Input Enabled

PIE04: P14 digital input enable bit (For ADC4)

0 = P10 Input Disabled

1 = P10 Input Enabled

PIE03: P03 Digital Input Enable Bit (For ADC3)

0 = P03 Input Disabled

1 = P03 Input Enabled

PIE02: P02 Digital Input Enable Bit (For ADC2)

0 = P02 Input Disabled

1 = P02 Input Enabled

PIE01: P01 Digital Input Enable Bit (For ADC1)

0 = P01 Input Disabled

1 = P01 Input Enabled

PIE00: P27 Digital Input Enable Bit (For ADC0 or LVD external input)

0 = P27 Input Disabled

1 = P27 Input Enabled

Register 6-24 PIE1 – Port digital input enable control1

Position	7	6	5	4	3	2	1	0
Name	PIE17	PIE16	PIE15	PIE14	PIE13	PIE12	PIE11	PIE10
Default	0	1	1	1	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PIE17: P45 Digital Input Enable Bit (For DACL)

0 = P45 digital Input Disabled

1 = P45 digital Input Enabled

PIE16: P35 Digital Input Enable Bit (For AUXL2 or VCMBUF)

0 = P35 digital Input Disabled

1 = P35 digital Input Enabled

PIE15: P34 Digital Input Enable Bit (For AUXR2)

0 = P34 digital Input Disabled

1 = P34 digital Input Enabled

PIE14: P33 Digital Input Enable Bit (For AUXL1)

0 = P33 digital Input Disabled

1 = P33 digital Input Enabled

PIE13: P32 Digital Input Enable Bit (For AUXR1)

0 = P32 digital Input Disabled

1 = P32 digital Input Enabled

PIE12: P31 Digital Input Enable Bit (For AUXL0)

0 = P31 digital Input Disabled

1 = P31 digital Input Enabled

PIE11: P30 Digital Input Enable Bit (For AUXR0)

0 = P30 digital Input Disabled

1 = P30 digital Input Enabled

PIE10: P17 Digital Input Enable Bit (For VPP)

0 = P17 digital Input Disabled

1 = P17 digital Input Enabled

6.5 Peripherals MUXed with Ports

6.6 Read and Write Ports

Port 0 to Port 4 are memory-mapped into the Data Memory addressing space. They are respectively mapped into 80h, 90h, A0h ,B0h and C0H registers for ports P0, P1, P2 P3 and P4. Writing to a port data register sets the voltage levels of the corresponding port pins that have been configured to operate as outputs. Reading from a data register reads the voltage levels of the corresponding port pins.

As illustrated in Figure 8-1, there are major differences reading the port values when the port is set as input and output. When the port is set as output, the CPU will read the port value from Px register instead of the port pin value. When the port is set as input, the CPU will read the value from port pin directly instead of the port value from Px register. As a result, the user should be very careful when using Read-then-Write instructions to access the ports and change PxDIR before write the output value to Px when using port as output. For example:

Code assembler:

```
ANL P0DIR, #0FEH
MOV P0, #01h
```

Code C51:

```
P0DIR &= 0xFE;
P0 = 0x01;
```

The first instruction in this example configures P00 as output, and then the second instruction writes the Port 0 data register (P0), which controls the output levels of the Port 0 pins, P00 through P07. Figure 8-1 shows the internal hardware structure and configuration registers for each pin of Port 0~4.

6.7 Wakeup

Register 6-25 PWKEN – Port wakeup enable

Position	7	6	5	4	3	2	1	0
Name	WKPND3	WKPND2	WKPND1	WKPND0	PWKEN3	PWKEN2	PWKEN1	PWKEN0
Default	0	0	0	0	1	1	1	1
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

WKPND3

0 = No P03/P06/P13/P16/P42 wakeup event occurred

1 = P03/P06/P13/P16/P42 wakeup event occurred

WKPND2

0 = No USBDP2/USBDM2/P02/P10/P12/P14/P15/P27 wakeup event occurred

1 = USBDP1/USBDM1/USBDP2/USBDM2/P02/P10/P12/P14/P15/P27 wakeup event occurred

WKPND1

0 = No port01/USBDP1/USBDM1 wakeup event occurred

1 = Port01/USBDP1/USBDM1 wakeup event occurred

WKPND0

0 = No port00/3in1 IO wakeup event occurred

1 = Port00/3in1 IO wakeup event occurred

PWKEN3

0 = Enable P03/P06/P13/P16/P42 Wakeup

1 = Disable P03/P06/P13/P16/P42 Wakeup

NOTE: detail pin selection description, please reference to AIPCON6.5~ AIPCON6.3.

PWKEN2

0 = Enable USBDP2/USBDM2/P02/P10/P12/P14/P15/P27 Wakeup

1 = Disable USBDP2/USBDM2/P02/P10/P12/P14/P15/P27 Wakeup

NOTE: detail pin selection description, please reference to AIPCON6.5~ AIPCON6.3.

PWKEN1

0 = Enable Port01/USBDP1/USBDM1 Wakeup

1 = Disable Port01/USBDP1/USBDM1 Wakeup

PWKEN0

0 = Enable Port00/3in1 IO Wakeup

1 = Disable Port00/3in1 IO Wakeup

Note:

1. Enable Port Wakeup is a condition of Port wakeup events occurred.
2. To enable WKPNDx, set PWKENx to ‘0’.
3. To clear WKPNDx, write ‘0’ to WKPNDx. WKPNDx will be ‘0’ 2 clocks later after write ‘0’ to WKPNDx.
4. WKPNDx is cleared when PWKENx is ‘1’.

Register 6-26 PWKEDGE – Port wakeup Event select

Position	7	6	5	4	3	2	1	0
Name	Reserved				WKEDG3	WKEDG2	WKEDG1	WKEDG0
Default	0	0	0	0	x	x	x	X
Access	RO	RO	RO	RO	R/W	R/W	R/W	R/W

WKEDGx: Port Wake up Edge Select

0 = select rising edge as wake up event

1 = select falling edge as wake up event

7 TIMERS

7.1 Timer0

Timer0 is a 16-bit timer/counter with a 7-bit prescaler. It can be configured as timer, counter or PWM generator.

7.1.1 Timer0 Feature

- 16 bits counter
- 7bits pre-scaler
- Counter mode (clock source from system clock or TMR0)
- Capture mode (event source from CAP0)
- PWM mode (PWM signal output to PWM0)

7.1.2 Timer0 Special Function Registers

Register 7-1 TMR0CON – Timer0 control

Position	7	6	5	4	3	2	1	0
Name	T0PND	T0ES	T0M		T0IS	T0PSR		
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T0PND: Timer0 Pending Flag

0 = Not Pending

1 = Pending

T0ES: Timer0 Capture Mode Edge Select

0 = P06 Rising Edge

1 = P06 Falling Edge

T0M: Timer0 Mode

00 = Disable

01 = Timer Mode

10 = PWM Mode

11 = Capture Mode

T0IS: Timer0 Increase Source

0 = System Clock

1 = P0.2 rising edge

T0PSR: Timer0 Prescaler

000 = Divided by 1

001 = Divided by 2

010 = Divided by 4

011 = Divided by 8

100 = Divided by 16

101 = Divided by 32

110 = Divided by 64

111 = Divided by 128

Register 7-2 TMR0CNTH/TMR0CNTL – Timer0 Counter

Position	7	6	5	4	3	2	1	0
Name	TMR0CNTH/TMR0CNTL							
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Timer0 will increase in proper condition while it is enabled. It overflows when TMR0CNT = TMR0PR, TMR0CNT will be clear to 0x0000 when overflow occurs, and the interrupt flag will be set '1' by hardware.

Register 7-3 TMR0PRH/TMR0PRL – Timer0 Period

Position	7	6	5	4	3	2	1	0
Name	TMR0PRH/TMR0PRL							
Default	1	1	1	1	1	1	1	1
Access	WO	WO	WO	WO	WO	WO	WO	WO

Note: The overflow period of the timer is: Tinc-source * T0PSR * (T0PR + 1).

Register 7-4 TMR0PWMH/TMR0PWML– Timer0 PWM duty

Position	7	6	5	4	3	2	1	0
Name	TMR0PWMH/TMR0PWML							
Default	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TMR0PWM is reserved in timer/counter mode. In PWM mode, it is used as duty cycle setting. In capture mode, the value of TMR0CNT will be captured to TMR0PWM when selected event occurs.

7.2 Timer1

Timer1 is a 16-bit timer/counter with a 7-bit prescaler. It can be configured as timer, counter or PWM generator.

7.2.1 Timer1 Feature

- 16bits counter
- 7bits pre-scaler
- Counter mode (clock source from system clock or TMR1)
- Capture mode (event source from CAP1)
- PWM mode (PWM signal output to PWM1)

7.2.2 Timer1 Special Function Registers

Register 7-5 TMR1CON0 – Timer1 control 0

Position	7	6	5	4	3	2	1	0
Name	T1ES		T1M		T1CPSEL		T1IS	
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T1ES: Timer1 Capture Edge Select

00 = capture input pin Rising Edge

01 = capture input pin Falling Edge

1X= capture input pin Rising Edge and Falling Edge

T1M: Timer1 Mode Select

00 = Timer1 Disable

01 = Timer Mode

10 = PWM Mode

11 = Capture Mode

T1CPSEL: Timer1 capture input pin select

00 = Capture P07

01 = Capture RC_32K(2MHz div 64) clock. See the figure 6-2 for more info.

10 = Capture P06

11 = Capture IR waveform

T1IS = Timer1 Increase Source

00 = P0.3 Rising Edge or RC clock Rising Edge(clkcon2.2 = 1 select RC)

01 = P0.3 Falling Edge or RC clock Falling Edge(clkcon2.2 = 1 select RC)

10 =3 2K OSC

11 = System Clock

Register 7-6 TMR1CON1 – Timer1 control 1

Position	7	6	5	4	3	2	1	0
Name	T1TPND	T1CPND	T1TIE	T1CIE	-	T1PSR		
Default	0	0	0	0	-	0	0	0
Access	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

T1TPND: Timer1 Over Flow Pending Bit

0 = Not Pending

1 = Pending

T1CPND: Timer1 Capture mode Pending Bit

0 = Not Pending

1 = Pending

T1TIE: Timer1 Over Flow Interrupt Enable Bit

0 = Interrupt Disable

1 = Interrupt Enable

T1CIE: Timer1 Capture mode Interrupt Enable Bit

0 = Disable

1 = Enable

T1PSR: Timer1 Prescaler

000 = divide by 1

001 = divide by 2

010 = divide by 4

011 = divide by 8

100 = divide by 16

101 = divide by 32

110 = divide by 64

111 = divide by 128

Register 7-7 TMR1CNTH/TMR1CNTL – Timer1 Counter

Position	7	6	5	4	3	2	1	0
Name	TMR1CNTH/TMR1CNTL							
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Timer1 will increase in proper condition while it is enable, it overflows when TMR1CNT = TMR1PR, TMR1CNT will be cleared to 0x0000 when overflow, and the interrupt flag will be set '1' by hardware.

Register 7-8 TMR1PRH/TMR1PRL – Timer1 Period

Position	7	6	5	4	3	2	1	0
Name	TMR1PRH/TMR1PRL							
Default	1	1	1	1	1	1	1	1
Access	WO	WO	WO	WO	WO	WO	WO	WO

The overflow period of the timer is: Tinc-source * T1PSR * (T1PR + 1).

Register 7-9TMR1PWMH/TMR1PWML – Timer1 PWM duty

Position	7	6	5	4	3	2	1	0
Name	TMR1PWMH/TMR1PWML							
Default	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TMR1PWM is reserved in timer/counter mode. In PWM mode, it is used as duty cycle setting. In capture mode, the value of TMR1CNT will be captured to TMR1PWM when selected event occurs.

7.3 Timer2

Timer2 is a 16-bit timer/counter with a 7-bit prescaler. It can be configured as timer, counter or PWM generator.

7.3.1 Timer2 Feature

- 16bits counter
- 7bits pre-scaler
- Counter mode (clock source from system clock or TMR2)
- Capture mode (event source from CAP2)
- PWM mode (PWM signal output to PWM2)

7.3.2 Timer2 Special Function Registers

Register 7-10 TMR2CON0 – Timer2 control 0

0	7	6	5	4	3	2	1	
T2IS	T2ES		T2M		T2CPSEL		T2IS	
Default	0	0	0	0	0	—	0	0
Access	R/W	R/W	R/W	R/W	R/W	—	R/W	R/W

T2ES: Timer2 Capture Edge Select

00 = capture input pin Rising Edge

01 = capture input pin Falling Edge

1X= capture input pin Rising Edge and Falling Edge

T2M: Timer2 Mode Select

00 = Timer2 Disable

01 = Timer Mode

10 = PWM Mode

11 = Capture Mode

T2CPSEL: Timer2 capture input pin select

0 = Capture P01

1 = Capture P13

T2IS: Timer2 Increase Source

00 = P1.4 Rising Edge

01 = P1.4 Falling Edge

10 = 32K OSC

11 = System Clock

Register 7-11 TMR2CON1 – Timer2 control 1

Position	7	6	5	4	3	2	1	0
Name	T2TPND	T2CPND	T2TIE	T2CIE	-	T2PSR		

Default	0	0	0	0	-	0	0	0
Access	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W

T2TPND: Timer2 Over Flow Pending Bit

0 = Not Pending

1 = Pending

T2CPND: Timer2 Capture mode Pending Bit

0 = Not Pending

1 = Pending

T2TIE: Timer2 Over Flow Interrupt Enable Bit

0 = Interrupt Disable

1 = Interrupt Enable

T2CIE: Timer2 Capture mode Interrupt Enable Bit

0 = Disable

1 = Enable

T2PSR: Timer2 Prescaler

000 = divide by 1

001 = divide by 2

010 = divide by 4

011 = divide by 8

100 = divide by 16

101 = divide by 32

110 = divide by 64

111 = divide by 128

Register 7-12 TMR2CNTH/TMR2CNTL – Timer2 Counter

Position	7	6	5	4	3	2	1	0
Name	TMR2CNTH/TMR2CNTL							
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Timer2 will increase in proper condition while it is enable, it overflows when TMR2CNT = TMR2PR, TMR2CNT will be clear to 0x0000 when overflow, and the interrupt flag will be set '1' by hardware.

Register 7-13TMR2PRH/TMR2PRL – Timer2 Period

Position	7	6	5	4	3	2	1	0
Name	TMR2PRH/TMR2PRL							
Default	1	1	1	1	1	1	1	1
Access	WO	WO	WO	WO	WO	WO	WO	WO

The overflow period of the timer is: Tinc-source * T2PSR * (T2PR + 1).

Register 7-14TMR2PWMH/TMR2PWML – Timer2 PWM duty

Position	7	6	5	4	3	2	1	0
Name	TMR2PWMH/TMR2PWML							

Default	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TMR2PWM is reserved in timer/counter mode. In PWM mode, it is used as duty cycle setting. In capture mode, the value of TMR2CNT will be captured to TMR2PWM when selected event occurs.

7.4 Timer3

Timer3 is a 16-bit timer/counter with a 7-bit prescaler. It can be configured as timer, counter or PWM generator.

7.4.1 Timer Feature

- 16 bits counter
- 7bits pre-scaler
- Counter mode (clock source from system clock or TMR3)
- Capture mode (event source from CAP3)
- PWM mode (PWM signal output to PWM3)

7.4.2 Timer3 SFR

Register 7-15 TMR3CON – Timer3 control

Position	7	6	5	4	3	2	1	0
Name	T3PND	T3ES	T3M		T3IS	T3PSR		
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

T3PND: Timer3 Pending Flag

0 = Not Pending

1 = Pending

T3ES: Timer3 Capture Mode Edge Select

0 = P13 Rising Edge

1 = P13 Falling Edge

T3M: Timer3 Mode

00 = Disable

01 = Timer Mode

10 = PWM Mode

11 = Capture Mode

T3IS: Timer3 Increase Source

0 = System Clock

1 = P11 rising edge

T3PSR: Timer3 Prescaler

000 = Divided by 1

001 = Divided by 2

010 = Divided by 4

011 = Divided by 8

100 = Divided by 16

101 = Divided by 32

110 = Divided by 64

111 = Divided by 128

Register 7-16 TMR3CNTH/TMR3CNTL – Timer3 Counter

Position	7	6	5	4	3	2	1	0
Name	TMR3CNTH/TMR3CNTL							
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: Timer3 will increase in proper condition while it is enabled. It overflows when TMR3CNT = TMR3PR, TMR3CNT will be clear to 0x0000 when overflow occurs, and the interrupt flag will be set '1' by hardware.

Register 7-17 TMR3PRH/TMR3PRL – Timer3 Period

Position	7	6	5	4	3	2	1	0
Name	TMR3PRH/TMR3PRL							
Default	1	1	1	1	1	1	1	1
Access	WO	WO	WO	WO	WO	WO	WO	WO

Note: The overflow period of the timer is: Tinc-source * T3PSR * (T3PR + 1).

Register 7-18 TMR3PWMH/TMR3PWML – Timer3 PWM duty

Position	7	6	5	4	3	2	1	0
Name	TMR3PWMH/TMR3PWML							
Default	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: TMR3PWM is reserved in timer/counter mode. In PWM mode, it is used as duty cycle setting. In capture mode, the value of TMR3CNT will be captured to TMR3PWM when selected event occurs.

7.5 Watchdog Timer (WDT)

The Watchdog Timer (WDT) logic consists of a 20bit Watchdog Timer. The Watchdog Timer is clocked by internal RC oscillator running at 32KHz. When device resets, the WDT is disabled and user should enable the WDT if it is needed.

In the default configuration, WDT overflows in 2 ms. The application program needs to write a '1' into WDTCON[5] at least once 2 s to prevent WDT time out. The lower 3 bits of the WDTCON register control the selection of overflow time period.

7.5.1 Watchdog Wake up

There are 2 modes for wake up operation: wake up without reset and wake up with reset. It determines by RSTEN bit (WDTCON[3]). When RSTEN sets to 0, the watchdog will generate a non-reset wake up after counter overflows. Only in HOLD Mode, non-reset wake up can wakeup AX107X and it will continue to execute next instruction. AX107X cannot be waken up by WDT without reset in SLEEP Mode. When RSTEN sets to 1, the watchdog will

generate a reset wakeup after counter overflows. Both in HOLD Mode and SLEEP Mode, watchdog reset can wake up the chip, and then, AX107X goes back to the initial state.

7.5.2 Watchdog Special Function Registers

Register 7-19 WDTCON – Watchdog control

Position	7	6	5	4	3	2	1	0
Name	WDTPD	WDTTO	CLRWDT	WDTEN	RSTEN	WDTPS		
Default	0	0	0	1	0	1	0	1
Access	RO	RO	WO	R/W	R/W	R/W	R/W	R/W

WDTPD:

0 = read '0' before sleep operation

1 = read '1' after sleep operation

WDTTO:

0 = Read '0' after clear Watchdog or Power up

1 = Read '1' after Watchdog time out

CLRWDT:

Write 1: Clear WDT counter，WDTPD and WDTTO will also be clear

Write 0: No action

WDTEN:

0 = Disables the Watchdog timer

1 = Enables the Watchdog timer

RSTEN:

0 = Disables the Watchdog reset

1 = Enables the Watchdog reset

WDTPS: WDT time out period setting

000 = 2ms

001 = 8ms

010 = 32ms

011 = 128ms

100 = 512ms

101 = 2048ms

110 = 8192ms

111 = 32768ms

7.6 Independent Power Real Time Clock Counter (IRTCC)

7.6.1 IRTCC Controller

IRTCC control can generate two interrupts: Second interrupt and Alarm interrupt.

IRTCC second interrupt can be enabled by writing 1 to IRTIE bit. When IRTCC works and IRTIE = 1, IRTCC second

interrupt will be generated every 1 second by setting IRTPND to 1. IRTPND can be cleared by software by writing 0 to IRTPND bit.

IRTCC alarm interrupt can be enabled by writing 1 to IRTALIE bit. When IRTCC works and IRTALIE = 1, IRTCC alarm interrupt will be generated when the current time is equal to the pre-set time by setting IRTALPND to 1. IRTALPND can be cleared by software by writing 0 to IRTALPND bit.

IRTCC is divided to two parts; one part is IRTCC control. The power of IRTCC control is VDDCORE. Another part is IRTCC. The part of IRTCC is VDDRTC. The communication between two parts is use like SPI protocol.

7.6.2 IRTCC Timer

IRTCC timer can be power independently. It can work even other logic in AX107X power off.

There is 6-bit valid address for the 64-byte user RAM. So the upper 2-bit of address in the writing RTC_RAM or reading RTC_RAM command are ignored. After one byte write/read, the internal address can increase automatically, this characteristic provide a burst mode to write/read the RAM. If the internal addresses increase greater than 63, it will roll back to 0.

7.6.3 Communication with IRTCC Timer

Special commands and corresponding parameters are used to communicate with IRTCC timer internal control or status registers and SRAM.

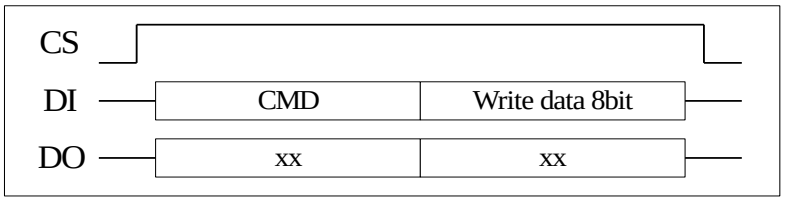
Table 7-1 IRTCC components communication commands

IRTCC component	Component type	Operation	Command Code	Command Parameters
Write_CFG(RTCCON)	A	Write	0x55	One byte
Read_CFG(RTCCON)	A	Read	0x54	One byte
Write_ CFG3(RTCC3)	A	Write	0x59	One byte
Write_RTC	B	Write	0xF0	Four byte
Read_RTC	B	Read	0xE0	Four byte
Write_ALM	B	Write	0x53	Four byte
Read_ALM	B	Read	0x52	Four byte
Write_RAM	C	Write	0x57	One byte address and N byte data
Read_RAM	C	Read	0x56	One byte address and N byte data
Write_PWR(PWRCON)	A	Read	0x5a	One byte
Write_WKO(WKOCON)	A	Write	0x5b	One byte
Read_WKO(WKOCON)	A	Read	0xa1	One byte
Write_VCL(VOLTAGE)	A	Read	0xa2	One byte
Read_VCL(VOLTAGE)	A	Write	0xa3	One byte

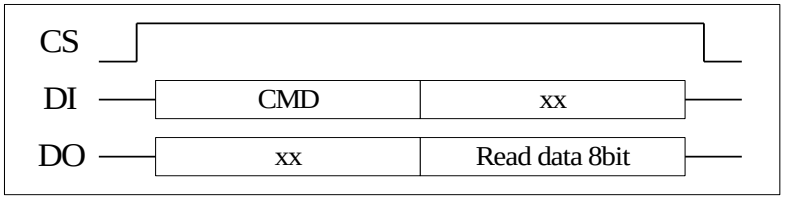
Communication operations:

1, Read or write A type components

Write:

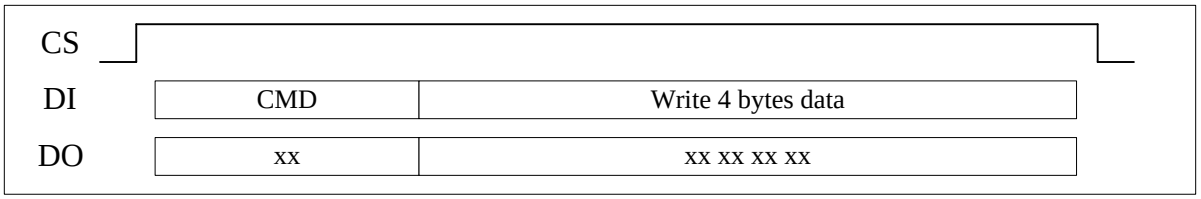


Read:

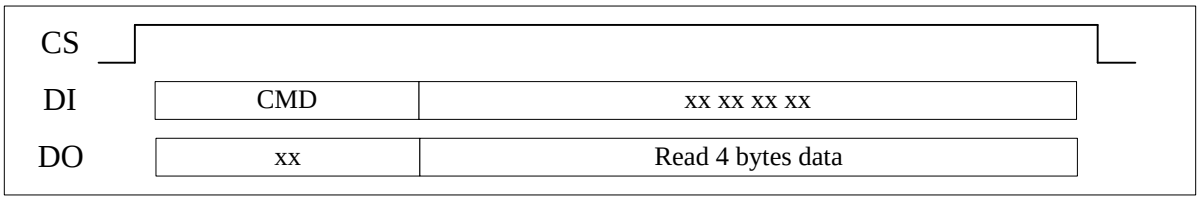


2, Read or write B type components

Write:

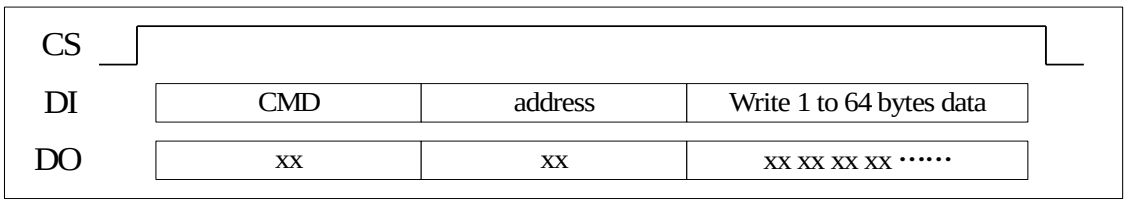


Read:

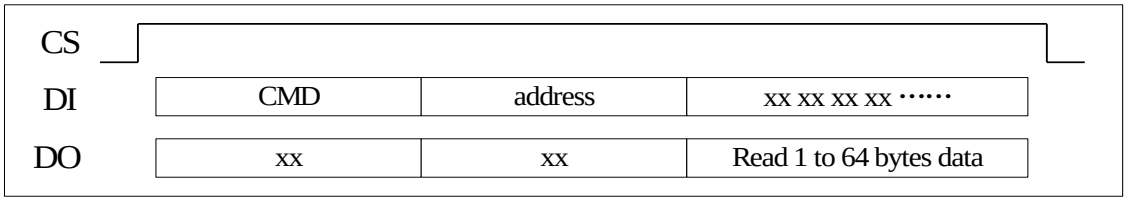


3, Read or write C type components

Write:



Read:



7.6.4 IRTCC Special Function Registers

Register 7-20 IRTCON – IRTCC control

Position	7	6	5	4	3	2	1	0
Name	IRTCSTEN	Reserved	IRTALPND	IRTALIE	IRTPND	IRTIE	DONE	EN
Default	0	0	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRTCSTEN: IRTCC sleep wake up enable

0 = Disable

1 = Enable

IRTALPND: IRTCC alarm pending

0 = No pending (Write 0 to clear pending)

1 = Pending

IRTALIE: IRTCC alarm interrupt enable

0 = Disable

1 = Enable

IRTALIE must be '1' if IRTCC alarm is used to wake up system.

IRTPND: IRTCC second pending

0 = No pending (Write 0 to clear pending)

1 = Pending

IRTALIE: IRTCC alarm second enable

0 = Disable

1 = Enable

IRTIE must be '1' if IRTCC second is used to wake up system.

DONE: Communication done flag

0 = done

1 = not done

EN: IRTCC communications enable

0 = Disable

1 = Enable

Register 7-21 IRTCDAT – RTCC communication data

Position	7	6	5	4	3	2	1	0
Name	IRTCDAT							
Default	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Write to IRTCDAT will start IRTCC communication and set DONE flag to 1.

Read IRTCDAT will return IRTCC data.

Register 7-22 SECCNT –IRTCC timer counter

Position	7	6	5	4	3	2	1	0

Name	SECCNT7	SECCNT6	SECCNT5	SECCNT4	SECCNT3	SECCNT2	SECCNT1	SECCNT0
Default	x	x	x	x	x	x	x	x
Access	WO	WO	WO	WO	WO	WO	WO	WO

RTCC second counter

Register 7-23IRTCON1 – RTCC control1

Position	7	6	5	4	3	2	1	0
Name	-	-	-	-	RTC_POR	IRTC_POR_EN	TIMER	TIMERIE
Default	-	-	-	-		0	0	0
Access	-	-	-	-	R/O	R/W	R/W	R/W

RTC_POR : RTCC POR bit

0 = RTCC POR be 0

1 = RTCC POR be 1

NOTE: only design specification can be known.

IRTC_POR_EN: IRTCC POR reset system clock enable

0 = Disable

1 = Enable

TIMER: Timer pending

0 = No pending (Write 0 to clear pending)

1 = When SECCNT equal to internal counter

TIMERIE: Timer pending interrupt enable

0 = Disable

1 = Enable

Register 7-24 RANDOM – random center regent

Position	7	6	5	4	3	2	1	0
Name	RANDOM[7:0]							
Default	-	-	-	-	-	-	-	-
Access	RO	RO	RO	RO	RO	RO	RO	RO

RANDOM : random center of 32k without default value

7.6.5 IRTCC components description

IRTCC timer can be power independently. It can work even other logic in TIGER is power off.

In IRTCC timer, there are one 8-bit configure register, one 32-bit real time counter, one 32-bit alarm register and 64-byte user RAM. All of these can be access (read or write) by several command sets through the IRTCC control.

There is 6-bit valid address for the 64-byte user RAM. So the upper 2-bit of address in the Write_RAM or Read_RAM command are ignored. After one byte write/read, the internal address can increase automatically, this characteristic provide a burst mode to write/read the RAM. If the internal address increase greater than 63, it will roll back to 0.

Register 7-25 RTCCON - RTCC control

Position	7	6	5	4	3	2	1	0

Name	32K_EN	12M_EN	XMODE	PDFLAG	F1HZEN	F32KHZEN	EX32KSEL	WKO32KOUT
Default	0	0	0	1	1	1	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

32K_EN : xosc 32k enable

0 = Disable

1 = Enable

12M_EN : xosc 12m enable

0 = Disable

1 = Enable

XMODE: xosc 1pin choose

0 =2 pin choose

1 =1pin choose

PDFLAG: RTCC timer power down flag

0 = RTCC timer is working

1 = RTCC timer is just power on

F1HZEN: 1Hz signal output enable

0 = Disable

1 = Enable

F32KHZEN: 32KHz signal output enable

0 = Disable

1 = Enable

EX32KSEL: RTCC timer clock source select

0 = RTCC timer works with IRTOSC 32KHz

1 = RTCC timer works with XOSC 32K.

WKO32KOUT: WKO output RTC analog 32K XOSC

0 = Disable

1 = Enable

Register 7-26 RTCC3 - RTCC configure register3

Position	7	6	5	4	3	2	1	0
Name	-	-	-	XOSC_SHP_RTC	RFB_SEL_RTC	PDNEN	DRSEL	
Default	-	-	-	0	0	1	0	0
Access	-	-	-	WO	WO	WO	WO	WO

XOSC_SHP_RTC: XOSC shaping method select

0 = comparator

1 = Schmitt trigger

RFB_SEL_RTC: 32K XOSC feedback resistor select

0 = disable

1 = enable

PDNEN: pull down enable

0 = disable

1 = enable

DRSEL: IRTCC OSC drive select

Register 7-27 PWRCON - Power control register

Position	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	-	LDO1P8EN
Default	-	-	-	-	-	-	-	0
Access	-	-	-	-	-	-	-	WO

LDO1P8EN: VDDCORE 1.8V and 3.3V LDO enable bit

0 = Enabled

1 = Disabled

Register 7-28 WKOCON - WKO control register

Position	7	6	5	4	3	2	1	0
Name	WKPIN_STA	FLTEN	ALMOE	WKOEN	WKOUTEN	WKOINEN	ALMEN	ALMOT
Default	0	0	0	0	0	0	0	X
Access	R/W	W/R	W/R	W/R	W/R	W/R	R/W	R/O

WKPIN_STA: Wake up pin output state

0 = wake up pin output 0

1 = wake up pin output 1

FITEN: WKO 1ms filter enable

0 = disable

1 = enable

ALMOE: Alarm output enable at WKO PIN output enable

0 = Disable

1 = Enable

WKOEN: WKO PIN enable

0 = Disable

1 = Enabled

WKOUTEN: WKO PIN output enable bit

0 = Disable

1 = Enabled

WKOINEN: WKO PIN input enable bit

0 = Disable

1 = Enabled

ALMEN: Alarm function enable

0 = Disable

1 = Enable

ALMOT: Alarm match flag.

0 = No alarm match happen

1 = Alarm match

This flag is set to ‘1’ by hardware when alarm register match real timer counter. It can be clear to ‘0’ of ALMEN is set to ‘0’ or ‘Write_ALM’ is detected.

Register 7-29 VCL VOLTAGE configure register

Position	7	6	5	4	3	2	1	0
Name	SC_RTC[3]	SC_RTC[2]	SC_RTC[1]	SC_RTC[0]	Reserved	HVDS	HVDEN	LVDEN
Default	1	0	1	0	0	0	0	0
Access	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R

SC_RTC [3:2]: OSCO capacitance select

SC_RTC [1:0]: OSCI capacitance select

HVDS: HVD level select

0 = 4.0V

1 = 4.2V

HVDEN: HVD enable bit

0 = Disabled

1 = Enabled

LVDEN: LVD enable bit

0 = Disabled

1 = Enabled

In IRTCC timer, there is one 32-bit real time counter. The unit of this counter is per second. If display the time on LCD, you should convert to second, minute, hour, date and so on. When use “Write_RTC” command to config this counter, the fisrt byte is config the highest counter, and the forth byte is config the lowest counter. When use “Read_RTC” command to read this counter, the first byte output is the highest counter, and the forth byte output is the lowest counter.

In IRTCC timer, there is one 32-bit alarm register. The unit of this counter is per second. If display the time on LCD, you should convert to second, minute, hour, date and so on. When use “Write_ALM” command to config this counter, the fisrt byte is config the highest counter, and the forth byte is config the lowest counter. When use “Read_ALM” command to read this counter, the first byte output is the highest counter, and the forth byte output is the lowest counter.

7.6.6 IRTCC Operating Guide

```
;---------------------------------
;       Write RTC Config
Write_Cfg:
        ORL     IRTCON, #(1<<0)         ;RTC enable
        MOV     A, #55H
        CALL    Send_Dat
        MOV     A, #0CCH
        CALL    Send_Dat
        ANL     IRTCON, #~(1<<0)        ;RTC Disable
        RET
;---------------------------------
```

```
;     Read Config
Read_Cfg:
            ORL       IRTCON, #(1<<0)          ;RTC enable
            MOV       A, #54H
            CALL      Send_Dat
            MOV       A, #00H
            CALL      Send_Dat
            ANL       IRTCON, #~(1<<0)         ;RTC Disable
            RET
;--------------------------------
;       Write_RTC
Write_RTC:
            ORL       IRTCON, #(1<<0)          ;RTC enable

            MOV       A, #0F0H
            CALL      Send_Dat

            MOV       A, #98H
            CALL      Send_Dat
            MOV       A, #76H
            CALL      Send_Dat
            MOV       A, #54H
            CALL      Send_Dat
            MOV       A, #32H
            CALL      Send_Dat

            ANL       IRTCON, #~(1<<0)         ;RTC Disable
            RET
;--------------------------------
;       Read_RTC
Read_RTC:
            ORL       IRTCON, #(1<<0)          ;RTC enable

            MOV       A, #0E0H
            CALL      Send_Dat

            MOV       A, #00H
            CALL      Send_Dat
            MOV       A, RTCDAT
```

```
        MOV     A, #00H
        CALL    Send_Dat
        MOV     A, RTCDAT
        MOV     A, #00H
        CALL    Send_Dat
        MOV     A, RTCDAT
        MOV     A, #00H
        CALL    Send_Dat
        MOV     A, RTCDAT

        ANL     IRTCON, #~(1<<0)        ;RTC Disable
        RET
;--------------------------------
;       Write RTC Alam
Write_Alam:
        ORL     IRTCON, #(1<<0)         ;RTC enable

        MOV     A, #53H
        CALL    Send_Dat

        MOV     A, #12H
        CALL    Send_Dat
        MOV     A, #34H
        CALL    Send_Dat
        MOV     A, #56H
        CALL    Send_Dat
        MOV     A, #78H
        CALL    Send_Dat

        ANL     IRTCON, #~(1<<0)        ;RTC Disable
        RET
;--------------------------------
;       Read RTC Alam
Read_Alam:
        ORL     IRTCON, #(1<<0)         ;RTC enable

        MOV     A, #52H
        CALL    Send_Dat
```

```
        MOV     A, #00H
        CALL    Send_Dat
        MOV     A, RTCDAT
        MOV     A, #00H
        CALL    Send_Dat
        MOV     A, RTCDAT
        MOV     A, #00H
        CALL    Send_Dat
        MOV     A, RTCDAT
        MOV     A, #00H
        CALL    Send_Dat
        MOV     A, RTCDAT

        ANL     IRTCON, #~(1<<0)        ;RTC Disable
        RET
;--------------------------------
;       Write RTC RAM
Write_Ram:
        ORL     IRTCON, #(1<<0)         ;RTC enable

        MOV     A, #57H
        CALL    Send_Dat
        MOV     A, #00H                 ;Ram Address
        CALL    Send_Dat

        MOV     R0, #64
    Write_Ram_Loop:
        MOV     A, #55H
        CALL    Send_Dat
        DJNZ    R0, Write_Ram_Loop

        ANL     IRTCON, #~(1<<0)        ;RTC Disable
        RET
;--------------------------------
;       Read RTC RAM
Read_Ram:
        ORL     IRTCON, #(1<<0)         ;RTC enable

        MOV     A, #56H
```

```
            CALL    Send_Dat
            MOV     A, #00H             ;Ram Address
            CALL    Send_Dat

            MOV     R0, #64
        Read_Ram_Loop:
            MOV     A, #00H
            CALL    Send_Dat
            MOV     A, RTCDAT
            DJNZ    R0, Read_Ram_Loop
            ANL     IRTCON, #~(1<<0)    ;RTC Disable
            RET
;       Write VCL
Write_Vcl:
            ORL     IRTCON, #(1<<0)     ;RTC enable
            MOV     A, #0A2H
            CALL    Send_Dat
            MOV     A, #0A7H
            CALL    Send_Dat
            ANL     IRTCON, #~(1<<0)    ;RTC Disable
            RET
;--------------------------------
;     Read VCL
Read_Vcl:
            ORL     IRTCON, #(1<<0)     ;RTC enable
            MOV     A, #0A3H
            CALL    Send_Dat
            MOV     A, #00H
            CALL    Send_Dat
            ANL     IRTCON, #~(1<<0)    ;RTC Disable
            RET

;       Write WKO
Write_Wko:
            ORL     IRTCON, #(1<<0)     ;RTC enable
            MOV     A, #5BH
            CALL    Send_Dat
            MOV     A, #0A7H
            CALL    Send_Dat
```

```
        ANL     IRTCON, #~(1<<0)        ;RTC Disable
        RET
;---------------------------------
;   Read WKO
Read_ Wko:
        ORL     IRTCON, #(1<<0)         ;RTC enable
        MOV     A, #0A1H
        CALL    Send_Dat
        MOV     A, #25H
        CALL    Send_Dat
        ANL     IRTCON, #~(1<<0)        ;RTC Disable
        RET
;   Write PWR
Write_Pwr:
        ORL     IRTCON, #(1<<0)         ;RTC enable
        MOV     A, #5AH
        CALL    Send_Dat
        MOV     A, #003H
        CALL    Send_Dat
        ANL     IRTCON, #~(1<<0)        ;RTC Disable
        RET
;---------------------------------
;   Send Data
Send_Dat:
        MOV     RTCDAT, A
    Send_Dat_Loop:
        MOV     A, IRTCON
        JB      ACC.1, Send_Dat_Loop
        RET
;---------------------------------
```

8 UART

8.1 UART0

UART is a serial port capable of asynchronous transmission. The UART can function in full duplex mode. Receive data is buffered in a holding register. This allows the UART to start reception of a second incoming data byte before software has finished reading the previous data byte.

When PSEL = 0

- Receive pin (RX) – P11
- Transmit pin (TX) – P12

When PSEL = 1

- Receive pin (RX) – P34
- Transmit pin (TX) – P35

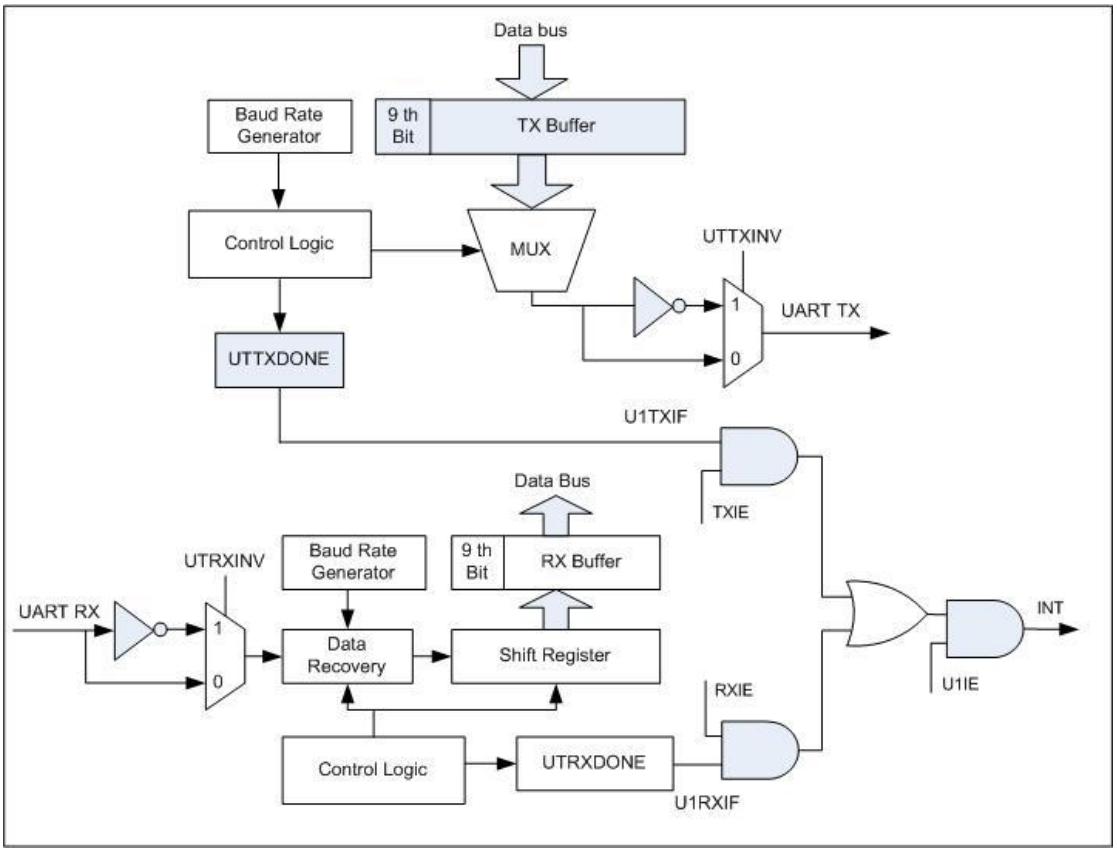


Figure 8-1 UART Block Diagram

8.1.1 Control registers

Register 8-1 UARTCON – UART control

Position	7	6	5	4	3	2	1	0

Name	UTSBS	UTTXNB	NBITEN	UTEN	UTTXINV	UTRXINV	TXIE	RXIE
Default	0	1	0	0	0	0	0	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

UTSBS: Stop Bit Select

0 = 1 bit as Stop Bit

1 = 2 bits as Stop Bit

UTTXNB: The ninth bit data of transmitter buffer. Write the ninth bit into this location that you want to transmit

NBITEN: Nine-BIT mode Enable Bit

0 = Eight-bit mode

1 = Nine-bit mode

UTEN: UART Enable Bit

0 = Disable UART module

1 = Enable UART module

UTTXINV: Transmit Invert Selection Bit

0 = Transmitter output without inverted

1 = Transmitter output inverted

UTRXINV: Receive Invert Selection Bit

0 = Receiver input without inverted

1 = Receiver input inverted

TXIE: Transmit Interrupt Enable

0 = Transmit interrupt disable

1 = Transmit interrupt enable

RXIE: Receive Interrupt Enable

0 = Receiver interrupt disable

1 = Receiver interrupt enable

Register 8-2 UARTSTA – UART status

Position	7	6	5	4	3	2	1	0
Name	UTRXNB	FEF	RXIF	TXIF	-	-	-	PSEL
Default	x	x	0	1	-	-	-	0
Access	R/W	R/W	R/W	RO	-	-	-	R/W

UTRXNB: The ninth bit data of receiver buffer

FEF: Frame Error Flag

0 = the stop bit is ‘1’ in the last received frame

1 = the stop bit is ‘0’ in the last received frame

RXIF: UART RX Interrupt Flag

0 = UART receive not done

1 = UART receive done

TXIF: UART TX Interrupt Flag

0 = UART transmit not done

1 = UART transmit done

Writing data to UTBUF will clear this flag.

PSEL: UART Port Select

0 = UART select

P11 as receive pin

P12 as transmit pin

1 = UART select

P34 as receive pin

P35 as transmit pin

Register 8-3 UARTBAUD – UART Baud Rate Low Byte

Position	7	6	5	4	3	2	1	0
Name	UARTBAUD							
Default	x	x	x	x	x	x	x	x
Access	WO	WO	WO	WO	WO	WO	WO	WO

Register 8-4 UARTBAUDH – UART Baud Rate High Byte

Position	7	6	5	4	3	2	1	0
Name	UARTDIV				UARTBAUDH			
Default	0	0	0	0	0	0	0	0
Access	WO	WO	WO	WO	WO	WO	WO	WO

UARTBAUD = {UARTBAUDH, UARTBAUD}

Baud Rate =Fsys clock / [(UARTDIV + 1)(UARTBAUD + 1)]

Register 8-5 UARTDATA – UART Data

Position	7	6	5	4	3	2	1	0
Name	UARTDATA							
Default	x	x	x	x	x	x	x	x
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Write this location will load the data to transmitter buffer. And read this location will read the data from the receiver buffer.

8.2 **UART1**

UART1 is a serial port capable of asynchronous transmission. The UART1 can function in normal and DMA full duplex mode. .Please see PMUXCON0 bit 6 description

- Receive pin (RX) – P27
- Transmit pin (TX) –P44

  Or

- Receive pin (RX) –P11
- Transmit pin (TX) –P10

8.2.1 Control registers

Register 8-6 UART1CON – UART1 control

Position	7	6	5	4	3	2	1	0
Name	UTSBS	UTTXNB	NBITEN	UTEN	TXIE	RXIE	OVERFLOWIE	DMASEL
Default	0	1	0	0	0	0	-	0
Access	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W

UTSBS: Stop Bit Select

0 = 1 bit as Stop Bit

1 = 2 bits as Stop Bit

UTTXNB: The ninth bit data of transmitter buffer. Write the ninth bit into this location that you want to transmit

NBITEN: Nine-BIT mode Enable Bit

0 = Eight-bit mode

1 = Nine-bit mode

UTEN: UART Enable Bit

0 = Disable UART module

1 = Enable UART module

TXIE: Transmit Interrupt Enable

0 = Transmit interrupt disable

1 = Transmit interrupt enable

RXIE: Receive Interrupt Enable

0 = Normal Receive interrupt disable or AUTO DMA mode Receive one word Interrupt disable

1 = Normal Receive interrupt enable or AUTO DMA mode Receive one word Interrupt enable

OVERFLOWIE: Receive DMA overflow interrupt enable

0 = overflow Interrupt disable

1 = overflow Interrupt enable

DMASEL:AUTO DMA choose

0 = AUTO DMA mode off

1= AUTO DMA mode on

Register 8-7 UAR1TSTA – UART1 status

Position	7	6	5	4	3	2	1	0
Name	UART_GIIE	PSEL	UTRXNB	RX_BYTE_HIGH	RXIF	TXIF	OVERFLOWIF	RXKICK
Default	0	0	x	0	0	1	0	0
Access	R/W	R/W	R/W	RO	R/W	R/W	R/W	WO

UART_GIE:UART Global Interrupt 15 Enable

0 = UART Global I Interrupt disable

1 = UART Global I Interrupt enable

UPSEL: UART Port Select

0 = UART select

P27 as receive pin

P44 as transmit pin

1 = UART select

P11 as receive pin

P10 as transmit pin

UTRXNB: The ninth bit data of receiver buffer

RX_BYTE_HIGH: receive data high byte(only for DMA)

0 = waiting receive data low byte

1 = waiting receive data high byte

RXIF: UART RX Interrupt Flag

0 = Normal Receive or AUTO DMA mode Receive one word not done

1 = Normal Receive or AUTO DMA mode Receive one word done

In normal mode , ,it become “1” every byte, but in DMA mode ,it become “1” every word.

TXIF: UART TX Interrupt Flag

0 = UART transmit not done

1 = UART transmit done

Writing data to UTBUF or Writing UARTDMATXCNT will clear this flag.

OVERFLOWIF: UART overflow Interrupt Flag

0 = UART overflow not done

1 = UART overflow done

RXKICK: UART DMA receive KICK start

0 = not KICK start

1 = KICK start

Register 8-8 UARTDIV – UART1 divide register

Position	7	6	5	4	3	2	1	0
Name	-	-	-	-	UARTDIV			
Default	-	-	-	-	0	0	0	0
Access	-	-	-	-	R/W	R/W	R/W	R/W

Register 8-9 UART1BAUD – UART1 Baud Rate register

Position	7	6	5	4	3	2	1	0
Name	UART1BAUD							
Default	x	x	x	x	x	x	x	x
Access	WO	WO	WO	WO	WO	WO	WO	WO

Baud Rate =Fsys clock / [(UARTDIV+1) (UART1BAUD + 1))

Register 8-10 UART1DATA – UART1 Data

Position	7	6	5	4	3	2	1	0
Name	UART1DATA							
Default	x	x	x	x	x	x	x	x

Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Write this location will load the data to transmitter buffer. And read this location will read the data from the receiver buffer.

Register 8-11 UARTDMATXCNT –UART1 DMA Transmit counter

Portion	7	6	5	4	3	2	1	0
Name	UARTDMATXCNT							
Default	x	x	x	x	x	x	X	x
Access	WO	WO	WO	WO	WO	WO	WO	WO

Nunit = UARTDMATXCNT + 1

Nbyte = Nunit * 2 = (UARTDMATXCNT + 1) * 2

Register 8-12 UARTDMATXPTR–UART1 DMA Transmit Start Pointer byte

Portion	7	6	5	4	3	2	1	0
Name	UARTDMATXPTR							
Default	x	x	x	X	x	x	X	x
Access	WO	WO	WO	WO	WO	WO	WO	WO

In order to get the correct DMA Start Pointer , you should write this register twice. First write the higher byte, then the low byte.

Register 8-13 UARTDMARXPTR–UART1 DMA receive Start Pointer byte

Portion	7	6	5	4	3	2	1	0
Name	UARTDMARXPTR							
Default	x	x	x	X	x	x	X	x
Access	WO	WO	WO	WO	WO	WO	WO	WO

In order to get the correct DMA Start Pointer , you should write this register twice. First write the higher byte, then the low byte.

Register 8-14 UART1MINUS–UART1 DMA receive data minus byte count by CPU

Portion	7	6	5	4	3	2	1	0
Name	UART1MINUS							
Default	x	x	x	X	x	x	X	x
Access	WO	WO	WO	WO	WO	WO	WO	WO

Nbyte = UART1MINUS+ 1'b1

Register 8-15 UART1POINT–UAR1T DMA point by CPU read

Portion	7	6	5	4	3	2	1	0
Name	UART1POINT							
Default	x	x	x	X	x	x	X	x
Access	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O

Register 8-16 UART1POINTH–UART DMA point by CPU read high byte

Portion	7	6	5	4	3	2	1	0
Name	UART1POINTH							

Default	x	x	x	X	x	x	X	x
Access	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O

Register 8-17 UART1LOOPCNT–UART1 DMA loop count

Position	7	6	5	4	3	2	1	0
Name				overlowcnt		dma_loop_cnt		
Default				0	0	0	0	0
Access	WO	WO	WO	WO	WO	WO	WO	WO

overlowcnt: less than bytes UART receive data ram size

00 = 4 bytes

01 = 8 bytes

10 = 16 bytes

11 = 32 bytes

dma_loop_cnt::UART receive data ram size

000 = 16 bytes

001 = 32 bytes

010 = 64 bytes

011 = 128 bytes

100 = 256 bytes

101 = 512 bytes

110 = 1K bytes

111 = forbidden

Register 8-18 UART1CNTH–UART1 DMA receive count high byte

Portion	7	6	5	4	3	2	1	0
Name	UART1CNTH							
Defeault	x	x	x	X	x	x	X	x
Access	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O

Register 8-19 UART1CNTL–UART1 DMA receive count low byte

Portion	7	6	5	4	3	2	1	0
Name	UART1CNTL							
Defeault	x	x	x	x	x	x	x	x
Access	R/O	R/O	R/O	R/O	R/O	R/O	R/O	R/O

8.3 Operation Guide

1）UART1 Normal mode Operation Flow:

1. Set IO in the correct direction.
2. Configure UARTDIV and UART1BAUD to choose sample rate and baud.

3 Enable UART1 module by setting UTEN to ‘1’

4. Set TXIE or RXIE ‘to 1’ if needed

5 write data to UART1DATA

6. Wait for PND to change to ‘1’, or wait for interrupt

7. Read received data from UART1DATA if needed

8. Go to Step 5 to start another process if needed or turn off UART1 by UTEN.

2）UART1 DMA Mode Operation Flow:

1. Set IO in the correct direction.
2. Configure UARTDIV and UART1BAUD to choose sample rate and baud.
3. Configure UART1CON Select DMA.
4. Write the start DMA address. for receive, Write data to UARTDMARXPTR
5. Enable UART module by setting UTEN to ‘1’.
6. kick-start a DMA receive process
7 Wait overflow or delay some time ,read UART1CNTH and UART1CNTL,read data by write UART1MINUS (UART1MINUS<{UART1CNTH,UART1CNTL}).
8. Write the start DMA address. for transmission, Write data to UARTDMATXPTR
9. Write data to UARTDMATXCNT to kick-start a DMA transmit process
10. Wait for PND to change to ‘1’, or wait for interrupt

9 DMA ARBITER

10 SPI

10.1 SPI0

SPI0 can serve as master or slave. It can operate in normal or DMA mode.

SPI0 can use either P0 or P1 as IO. SPI0CON.3 controls which group of IO is used by SPI0.

When AIPCON8[4] =0 (default)

When SPICON.3 = 0, it uses 2 pins for 2-wire mode: P15, P16

- Serial Data (SPI0DIDO) – P16
- Serial Clock (SPI0CLK) – P15

Or 3 pins for 3-wire mode: P17, P16, P15

- Serial Data Out (SPI0DO) – P16
- Serial Data In (SPI0DI) – P17
- Serial Clock (SPI0CLK) – P15

When SPI0CON.3 = 1, it uses 2 pins for 2-wire mode: P12 , P13

- Serial Data (SPI0DIDO) –P13
- Serial Clock (SPI0CLK) –P12

Or 3 pins for 3-wire mode: P12. P13 . P14

- Serial Data Out (SPI0DO) –P13
- Serial Data In (SPI0DI) –P14
- Serial Clock (SPI0CLK) –P12

When AIPCON8[4]=1,SPI can be used only in 2-wire mode, CLK and Data mapping to P10,P11

- Serial Data (SPI0DIDO) – P11
- Serial Clock (SPI0CLK) – P10

10.1.1 SPI0 registers

Register 10-1 SPI0CON – SPI0 control

Position	7	6	5	4	3	2	1	0
Name	SPI0PND	SPI0SM	SPI0RT	SPI0WS	SPI0PS	SPI0EDGE	SPI0IDST	SPI0EN
Default	1	0	0	0	0	0	0	0

Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SPI0PND: SPI0 Pending bit (read only, writing SPI0BUF will clear this bit)

0 = Transmission is not finish

1 = Transmission finish

SPI0SM: SPI0 mode selection

0 = Master mode

1 = Slave mode

SPI0RT: SPI0 RX/TX select bit in 2-wire mode or DMA mode

0 = TX

1 = RX

In 3-wire mode, SPI0 can both Transmit and receive at the same time. But when using DMA mode or 2-wire mode, just one direction (TX or RX) is allowed. Use this bit to select TX or RX.

SPI0WS: SPI0 2-wire mode/3-wire mode select bit

0 = 3-wire mode

1 = 2-wire mode

SPI0PS: SPI0 Port select

0 = Select Port1

1 = Select Port0

SPI0EDGE: SPI0 sampling edge select bit

When SPI0IDST = 0:

0 = sample at falling edge

1 = sample at rising edge

When SPI0IDST = 1:

0 = sample at rising edge

1 = sample at falling edge

SPI0IDST: SPI0 clock signal idle state

0 = Clock signal stay at 0 when idle

1 = Clock signal stay at 1 when idle

SPI0EN: SPI0 enable bit

0 = SPI0 disable

1 = SPI0 enable

Register 10-2 SPI0BAUD – SPI0 Baud Rate

Position	7	6	5	4	3	2	1	0
Name	SPI0BAUD							
Default	x	x	x	X	x	x	x	x
Access	WO	WO	WO	WO	WO	WO	WO	WO

Baud rate = Fsys / [2(SPI0BAUD+1)]

Register 10-3 SPI0BUF – SPI0 Data Buffer

Position	7	6	5	4	3	2	1	0

Name	SPI0BUF							
Default	x	x	x	X	x	x	x	X
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register 10-4 SPI0DMACNT – SPI0 DMA counter

Position	7	6	5	4	3	2	1	0
Name	SPI0DMACNT							
Default	x	x	x	X	x	x	x	x
Access	WO	WO	WO	WO	WO	WO	WO	WO

Nunit = SPI0DMACNT + 1

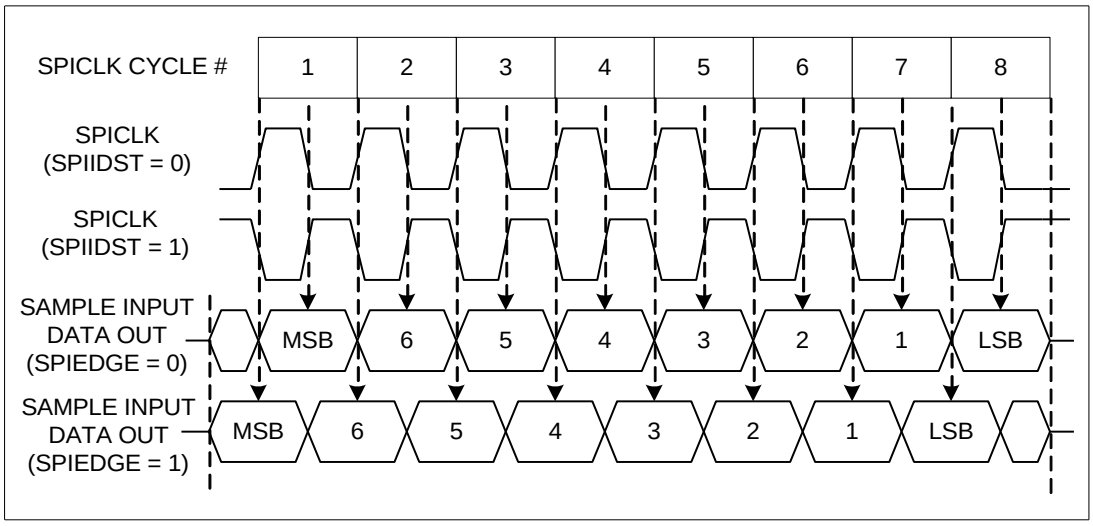
Nbyte = Nunit * 2 = (SPI0DMACNT + 1) * 2

Register 10-5 SPI0DMAPTRH– SPI0 DMA Start Pointer high byte

Position	7	6	5	4	3	2	1	0
Name	SPI0DMAPTRH							
Default	x	x	x	X	x	x	x	x
Access	WO	WO	WO	WO	WO	WO	WO	WO

Register 10-6 SPI0DMAPTRL– SPI0 DMA Start Pointer low byte

Position	7	6	5	4	3	2	1	0
Name	SPI0DMAPTRL							
Default	x	x	x	X	x	x	x	x
Access	WO	WO	WO	WO	WO	WO	WO	WO



10.1.2 Operation Guide

When SPI0CON1.1=0,

SPI0 Normal Mode Operation Flow:

1. Set P1.1 as output in transmission and as input in reception, set P1.0 as output in master mode and as input in slave mode, P1.2 is not use in 2-wire mode

2. Select SPI0RT in 2-wire mode if 2-wire mode is selected
3. Select master mode or slave mode
4. Configure clock frequency when master mode is selected in step 3
5. Select one of the four timing mode (refer to Figure 16-1)
6. Enable SPI0 module by setting SPI0EN ‘1’
7. Set SPI0IE ‘1’ if needed
8. Write data to SPI0BUF to kick-start the process
9. Wait for SPI0PND to change to ‘1’, or wait for interrupt
10. Read received data from SPI0BUF if needed
11. Go to Step 8 to start another process if needed or turn off SPI0 by clearing SPI0IE and SPI0EN

SPI0 DMA Mode Operation Flow:

1. Set P1.1 as output in transmission and as input in reception, set P1.0 as output in master mode and as input in slave mode, P1.2 is not used in 2-wire mode
2. Select SPI0RT for DMA direction
3. Select master mode or slave mode
4. Configure clock frequency when master mode is selected in step 3
5. Select one of the four timing modes (refer to Figure 16-1)
6. Enable SPI0 module by setting SPI0EN to ‘1’
7. Set SPI0IE ‘1’ if needed
8. Write the start address to SPI0DMASP
9. Write data to SPI0DMACNT to kick-start a DMA process
10. Wait for SPI0PND to change to ‘1’, or wait for interrupt
11. Go to Step 8 to start another DMA process if needed or turn off SPI0 by clearing SPI0IE and SPI0EN

11 SARADC

AX107X provides an eleven-channel moderate conversion speed and a moderate resolution 10-bit successive approximated register Analog to Digital Converter (SARADC) for users to develop applications in the following areas:

- Voice grade applications
- Audio applications requiring moderate performance
- Measurement requiring moderate performance and speed

SARADC conversion clock must be slower than 1 MHz.

11.1 Pins used

Table 11-1 pin mapping

ADC Channel	Function	Description
ADC10	P43	Normal ADC channel
ADC9	P42	Normal ADC channel
ADC8	P41	Normal ADC channel
ADC7	P40	Normal ADC channel
ADC6	LDO Band GAP	Battery voltage
ADC5	LDO in	Normal ADC channel
ADC4	P16	Normal ADC channel
ADC3	P15	Normal ADC channel
ADC2	P14	Normal ADC channel
ADC1	P01	Normal ADC channel
ADC0	Mix IO input	Normal ADC channel

11.2 SARADC registers

Register 11-1 ADCCON– SARADC control

Position	7	6	5	4	3	2	1	0
Name	ADCGO	EOC	TMREN	ADCTL	ADCEN	ADCSEL		
Default	0	0	x	x	0	0	0	0
Access	R/W	RO	R/W	R/W	R/W	R/W	R/W	R/W

ADCGO: ADC Conversion Start

when read:

0 = conversion finished

1 = conversion not finished

when write:

0 = N/A

1 = start conversion

EOC: Check if end of conversion

0 = finish

1 = not finish

TMREN: Timer Input Enable

0 = disable

1 = enable

ADCTL: Timer Source Select

0 = Timer0

1 = Timer1

ADCEN: ADC Module Enable

0 = Disable

1 = Enable

ADCSEL: ADC Channel Select

ADCS3==0，ADCSEL==000 ： Mix IO input

ADCS3==0，ADCSEL==001 ： P01

ADCS3==0，ADCSEL==010 ： P14

ADCS3==0，ADCSEL==011 ： P15

ADCS3==0，ADCSEL==100 ： P16

ADCS3==0，ADCSEL==101 ： LDO in

ADCS3==0，ADCSEL==110 ： LDO Band GAP

ADCS3==0，ADCSEL==111 ： P40

ADCS3==1，ADCSEL==000 ： P41

ADCS3==1，ADCSEL==001 ： P42

ADCS3==1，ADCSEL==010 ： P43

Register 11-2 ADCMODE– SARADC mode control

Position	7	6	5	4	3	2	1	0
Name	-	-	ADCS3	AUTOS	-	ADCSEL_SH		
Default	0	0	0	0	0	0	0	0
Access	RO	RO	R/W	R/W	R/W	R/W	R/W	R/W

ADCS3:ADC Channel Select 3

AUTOS: Auto channel switching mode

0 = Not switch

1 = Auto load ADCSEL_SH into ADCSEL after conversion finished. ADCS3 will be no change!

ADCSEL_SH: ADCSEL shadow

Register 11-3 ADCBAUD– SARADC baud rate control

Position	7	6	5	4	3	2	1	0
Name	-	-	ADCBAUD					

Default	-	-	x	x	x	x	x	x
Access	-	-	WO	WO	WO	WO	WO	WO

ADC conversion clock = system clock / (2 x (ADCBAUD + 1))

Register 11-4 ADCDATAL– SARADC Buffer low byte control

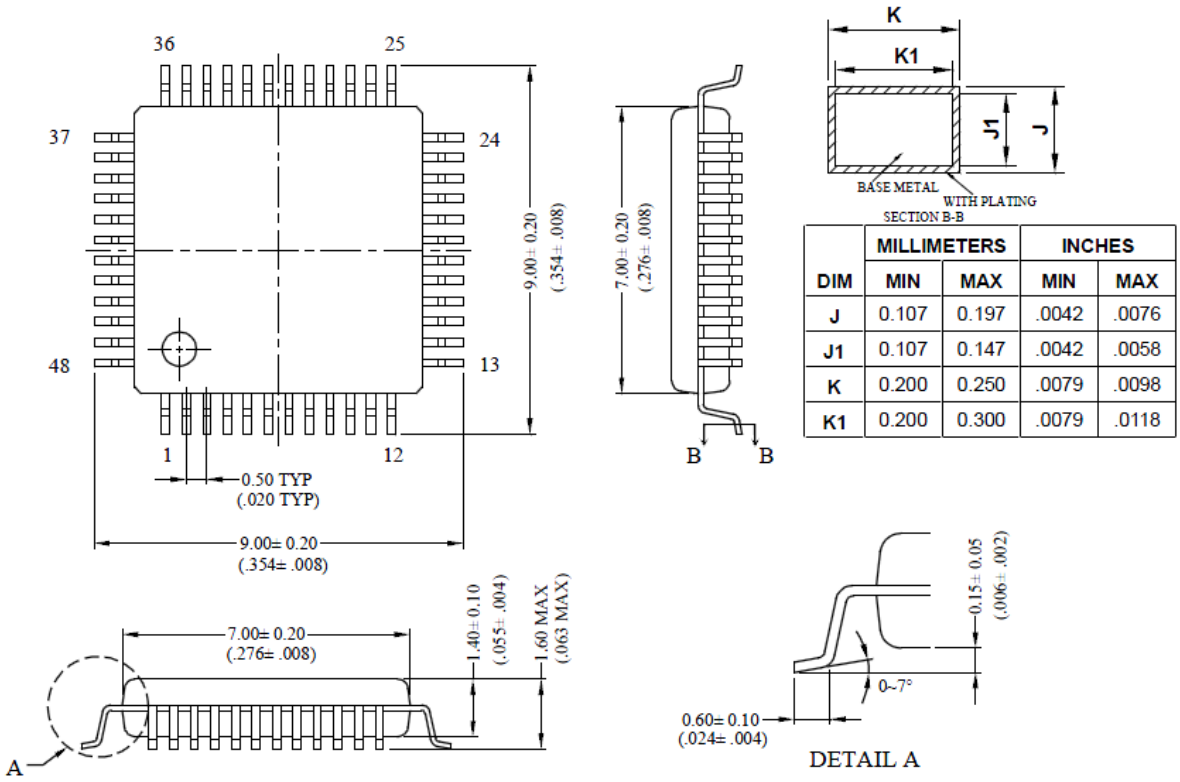
Position	7	6	5	4	3	2	1	0
Name	ADCDATAL		-	-	-	-	-	-
Default	x	x	-	-	-	-	-	-
Access	RO	RO	-	-	-	-	-	-

Register 11-5 ADCDATAH– SARADC Buffer high byte control

Position	7	6	5	4	3	2	1	0
Name	ADCDATAH							
Default	x	x	x	x	x	x	x	x
Access	RO	RO	RO	RO	RO	RO	RO	RO

12 Package Dimensions

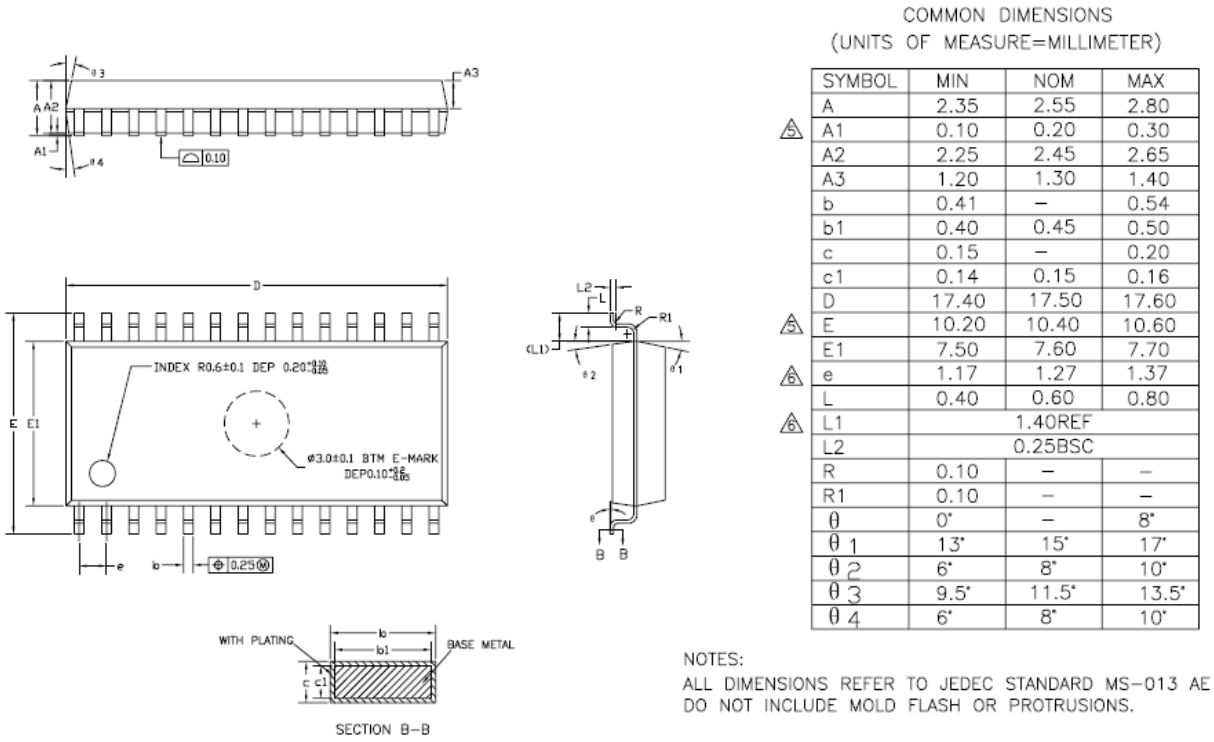
12.1 AX1070 LQFP48



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
J	0.107	0.197	.0042	.0076
J1	0.107	0.147	.0042	.0058
K	0.200	0.250	.0079	.0098
K1	0.200	0.300	.0079	.0118

Figure 12-1 AX1070 LQFP48 package dimensions

12.2 **AX1071 SOP28**



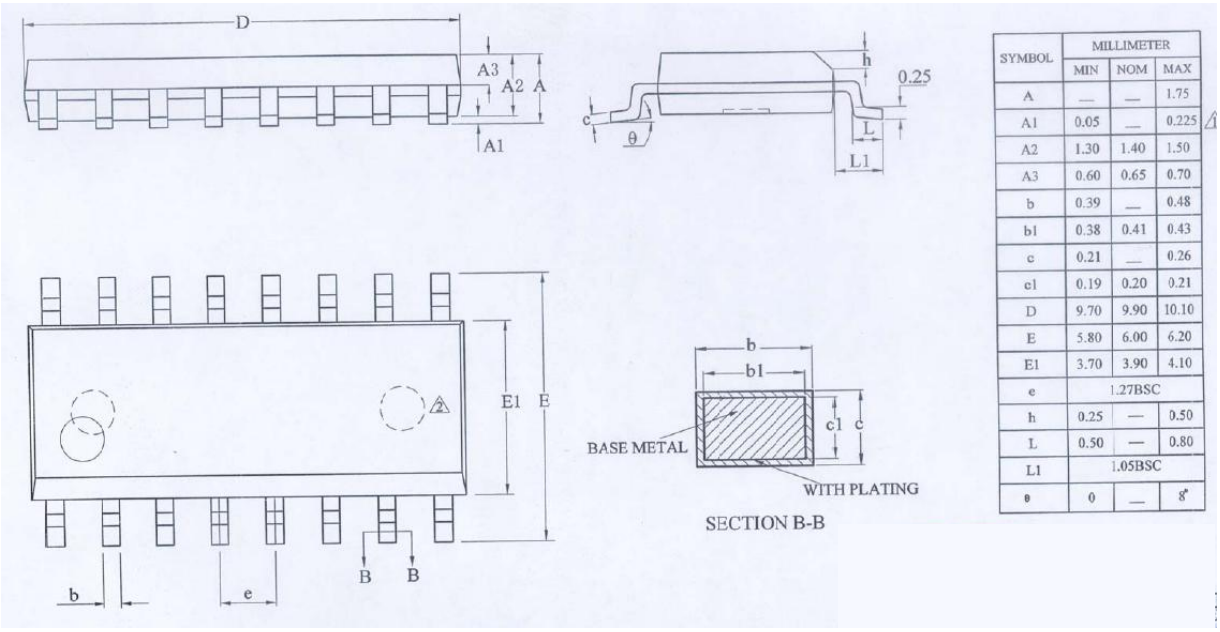
COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

	SYMBOL	MIN	NOM	MAX
	A	2.35	2.55	2.80
4	A1	0.10	0.20	0.30
	A2	2.25	2.45	2.65
	A3	1.20	1.30	1.40
	b	0.41	–	0.54
	b1	0.40	0.45	0.50
	c	0.15	–	0.20
	c1	0.14	0.15	0.16
	D	17.40	17.50	17.60
5	E	10.20	10.40	10.60
	E1	7.50	7.60	7.70
6	e	1.17	1.27	1.37
	L	0.40	0.60	0.80
6	L1	1.40REF		
	L2	0.25BSC		
	R	0.10	–	–
	R1	0.10	–	–
	θ	0°	–	8°
	θ1	13°	15°	17°
	θ2	6°	8°	10°
	θ3	9.5°	11.5°	13.5°
	θ4	6°	8°	10°

NOTES:
ALL DIMENSIONS REFER TO JEDEC STANDARD MS-013 AE
DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

Figure 12-2 AX1071 SOP28 package dimensions

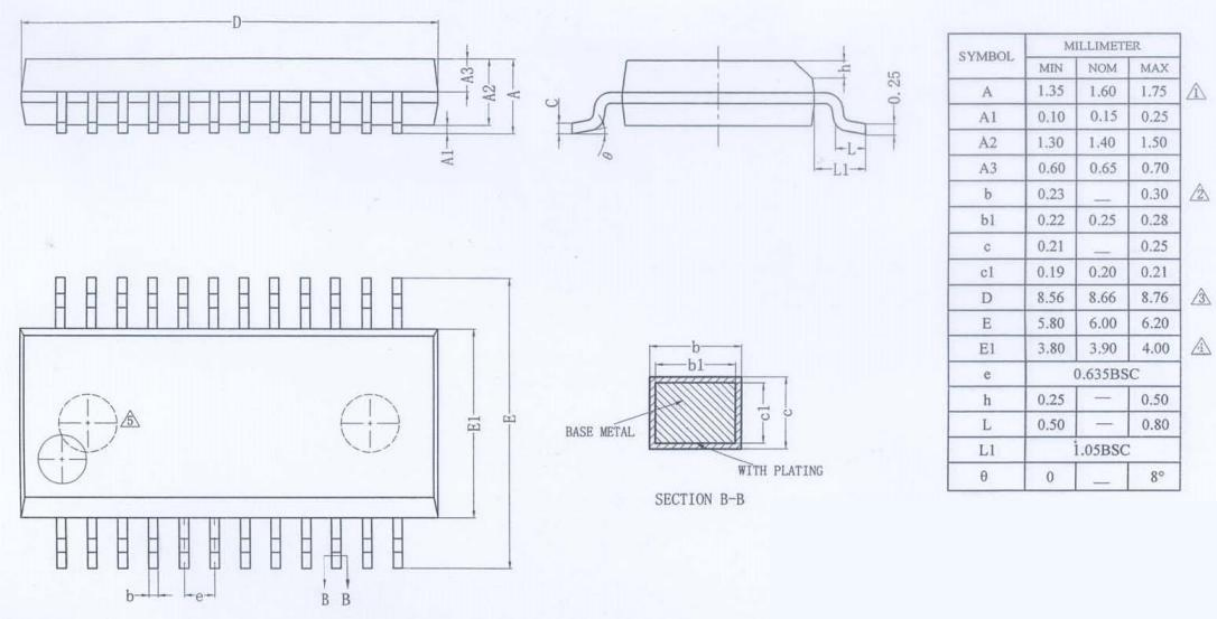
12.3 **AX1073/AX1073B SOP16**



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.05	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	—	0.48
b1	0.38	0.41	0.43
c	0.21	—	0.26
c1	0.19	0.20	0.21
D	9.70	9.90	10.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05BSC		
θ	0	—	8°

Figure 12-3 AX1073/AX1073B SOP16 package dimensions

12.4 **AX1076 SSOP24**



SYMBOL	MILLIMETER			
	MIN	NOM	MAX	
A	1.35	1.60	1.75	1
A1	0.10	0.15	0.25	
A2	1.30	1.40	1.50	
A3	0.60	0.65	0.70	
b	0.23	—	0.30	2
b1	0.22	0.25	0.28	
c	0.21	—	0.25	
c1	0.19	0.20	0.21	
D	8.56	8.66	8.76	3
E	5.80	6.00	6.20	
E1	3.80	3.90	4.00	4
e	0.635BSC			
h	0.25	—	0.50	
L	0.50	—	0.80	
L1	1.05BSC			
θ	0	—	8°	

Figure 12-4 AX1076 SSOP24 package dimensions

13 Characteristics

13.1 LDO Parameters

Sym	Characteristics	Min	Typ	Max	Unit	Conditions
Vin	LDO input voltage	3.2	5	5.5	V	
Vout1.8	1.8V output voltage	-	1.8	-	V	
Vout3.3	3.3V output voltage	-	3.3	-	V	
Iout1.8	1.8V output current	-	-	-	mA	
Iout3.3	3.3V output current	-	-	-	mA	

13.2 PLL Parameters

Sym	Characteristics	Min	Typ	Max	Unit	Conditions
FI	Frequency input	-	32	-	KHz	
FOUT1	Frequency output	-	48	-	MHz	
TLOCK	PLL locked time	-	-	-	us	

13.3 I/O Parameters

Symbol	Description	Min	Typ	Max	Units	Conditions
V_{IL}	Low-Level input voltage	-	-	30% * VDDIO	V	VDDIO = 3.3V
V_{IH}	High-level input voltage	70% * VDDIO	-	-	V	VDDIO = 3.3V
R_{PUP0}	Internal pull-up resister 0	2.64	3.3	3.96	KΩ	For PORT2
R_{PDN0}	Internal pull-down resister 0	2.64	3.3	3.96	KΩ	For PORT2
R_{PUP1}	Internal pull-up resister 1	8	10	12-	KΩ	For PORT0/1/3
R_{PDN1}	Internal pull-down resister 1	8	10	12	KΩ	For PORT0/1/3
I_{LEVEL0}	Level0 current driving	-	-	4	mA	For PORT0/2/3
I_{LEVEL1}	Level1 current driving	-	-	8	mA	For PORT1
I_{LEVEL2}	Level2 current driving	-	-	24	mA	For Port1.1

13.4 Audio DAC Parameters

Sym	Characteristics	Min	Typ	Max	Unit	Conditions
SNR		-	85	-	dB	
THD+N		-	-70	-	dB	No loading
PWRAB	ClassAB AMP power output	-	-	20	mW	16ohm, single channel
VPP	Maximum output voltage	-	-	2.4	V	

Appendix I Revision History

Date	Version	Comments	Revised by
2014-3-11	0.0.1	Initial version	Yuanxue
2014-3-12	0.0.2	Checked	Jinhong/zhuzhan
2014-3-13	1.0.0	Released	yuanxue
2014-5-16	1.0.1	Checked and modify	Jinhong/zhenxing
2014-5-16	1.1.0	Released	Yuanxue

In case of any questions or comments about this documentation, please feel free to contact AppoTech at documents@appotech.com .