Document Title

A9108 Data Sheet, sub1GHz Transceiver SOC

Revision History

Rev. No.	History	Issue Date	Remark
0.0	Initial issue.	May, 2012	Objective

Important Notice:

AMICCOM reserves the right to make changes to its products or to discontinue any integrated circuit product or service without notice. AMICCOM integrated circuit products are not designed, intended, authorized, or warranted to be suitable for use in life-support applications, devices or systems or other critical applications. Use of AMICCOM products in such applications is understood to be fully at the risk of the customer.

Table of Contents

1. General Description

A9108 is a high performance and low cost Sub1GHz ISM band system-on-chip (SOC) wireless transceiver. This device integrates high speed pipeline 8051 MCU, 16KBytes In-system programmable flash memory, 2KB SRAM, various powerful functions and excellent performance of Sub1GHz GFSK/FSK transceiver. A9108 has various operating modes, making it highly suited for systems where ultra-low power consumption is required. A9108 has a high resolution 24bit ADC for temperature senor; a 8bit ADC for RSSI and 4 channel 12bit ADC for general purpose. Three kinds of serial communication port (SPI, I2C and UART) can interact with other device(s).

A9108 is one of AMICCOM sub1GHz family. It integrates AMICCOM sub1GHz transceiver well and offers a low cost solution with advanced radio features such as high output power amplifier up to 20 dBm (433MHz band, excluding LPF and HPF) and low noise receiver (- 114 dBm @ 10Kbps, -110dBm @50Kbps). Therefore, A9108 is very suitable for long LOS (line-of-sight) applications without the need to add an external LNA or PA.

The on-chip data rate divider supports programmable on-air data rates from 2K to 250Kbps to satisfy different system requirements. For a battery powered system, A9108 supports fast PLL settling time (35 us), Xtal settling time (500 us) and on-chip Regulator settling time (450 us) to reduce average power consumption.

2. Typical Applications

- Wireless sensor networks
- Industrial monitoring and control
- Wireless alarm and security system
- Wireless ISM band data communication
- Remote control
- Home and building automation

3. Feature

- Small Package size (QFN6 X6, 48 pins).
- High performance pipeline complicated 8051
- Operation clock: 1, 1/2, 1/4, 1/8, 1/16, 1/32 of crystal oscillator.
- 16KB Flash memory, 2KB SARM
- UART, I^2C, SPI serial communication
- Three 16/8-bit counter/timers
- Two channel PWM
- Watchdog timer
- Sleep timer
- In-Circuit Debugger
- In-System programming/ In-Application programming
- 24 GPIO
- One channel 24bit ADC for thermal resistor
- Four channel 12bit SAR ADC
- One channel 8 bit ADC for RSSI and battery detect
- Programmable threshold of carrier detect.
- Frequency band: 315/433/470/868/915 MHz.
- FSK and GFSK modulation.
- Programmable data rate from 2Kbps to 250Kbps.
- RX Current consumption (AGC Off) 434MHz: 13.5 mA and 868MHz: 14 mA.
- TX Current consumption 433MHz: 30mA @ 10dBm, 70mA @ 17dBm.
- TX Current consumption 868MHz: 37mA @ 10dBm, 52mA @ 13dBm.
- Deep sleep current (1.2 uA)
- Low sleep current (5 uA)
- On chip regulator, support input voltage 2.0 ~ 3.6 V.
- High RX sensitivity 433.92MHz.
  - -117dBm at 2Kbps on-air data rate.
  - -114dBm at 10Kbps on-air data rate.
  - -110dBm at 50Kbps on-air data rate.
  - -107dBm at 100Kbps on-air data rate.
  - -106dBm at 150Kbps on-air data rate.
  - -103dBm at 250Kbps on-air data rate.
- Support low cost crystal 12.8 MHz /16 MHz).
- Support RTC clock 32.678KHz
- Fast PLL settling time (35 us).

- 9-bits Digital RSSI and Auto RSSI measurement
- Auto Calibrations.
- Auto FCS (CRC) and Filtering.
- On-chip full range VCO and Fractional-N PLL synthesizer.
- On-chip low power RC oscillator for WOR (Wake on RX) function.
- AFC (Auto Frequency Compensation) for frequency drift due to Xtal aging.
- Separated 64ytes FIFO for RX and TX.
- Built-in Battery Detect, Thermal Sensor and Crystal load capacitors.

4. Pin Configurations

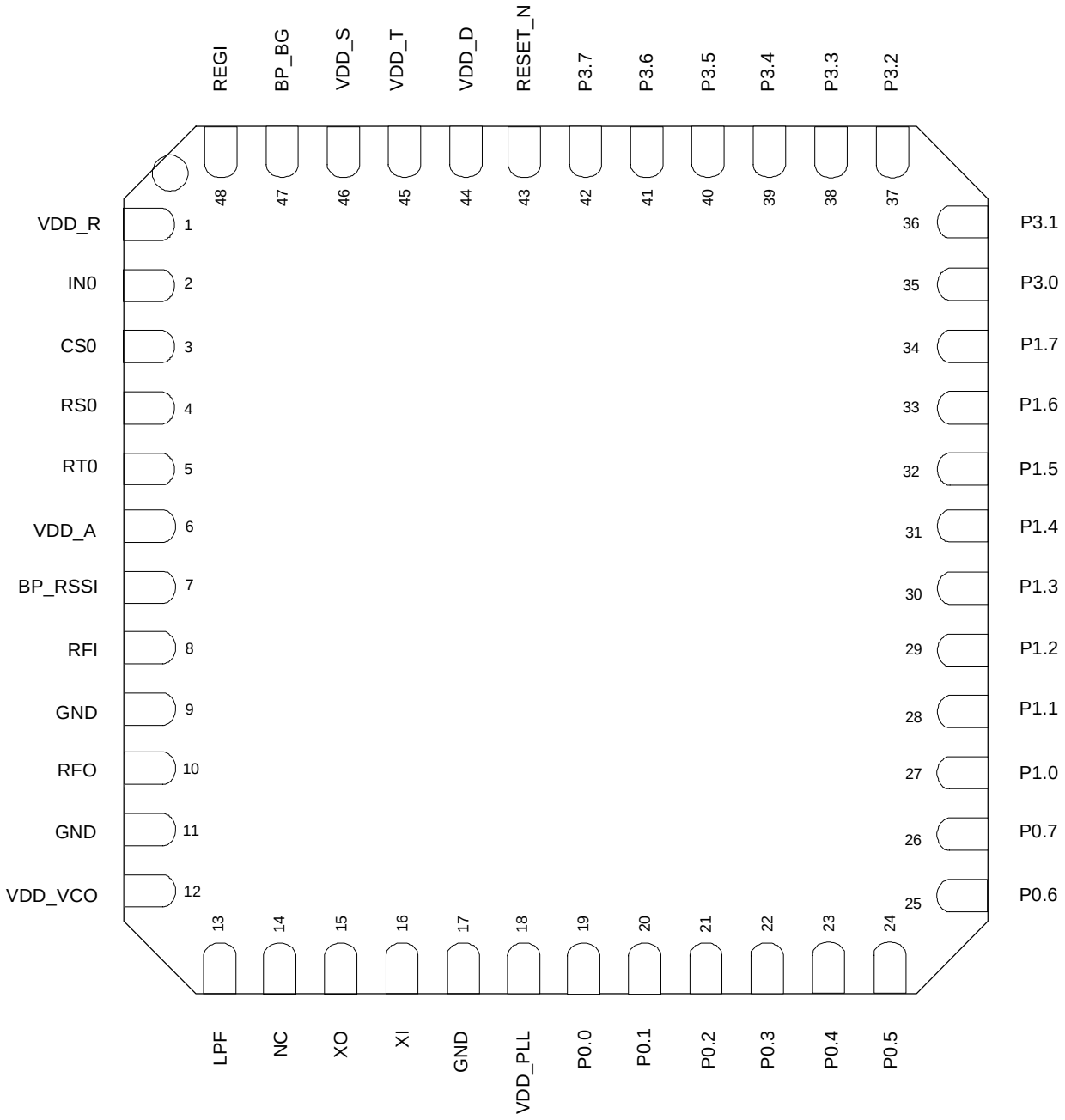


Fig 4-1. A9108 QFN 6x6 Package Top View

5. Pin Description (I: input; O: output, I/O: input or output)

Pin No.	Symbol	I/O	Function Description
1	VDD_R	O	Supply voltage output for external ADC reference. Connect to bypass capacitor.
2	IN0	I	Oscillator input pin.
3	CS0	I	Reference capacitor connection pin.
4	RS0	I	Reference resistor connection pin.
5	RT0	I	Resistor sensor connection pin for measurement.
6	VDD_A	O	Analog supply voltage output. Connect to bypass capacitor.
7	BP_RSSI	O	O: RSSI bypass. Connect to bypass capacitor.
8	RFI	I	RF input. Connect to matching circuit.
9	GND	G	Ground.
10	RFO	O	RF output. Connect to matching circuit. (recommend powered by VDD directly).
11	GND	G	Ground.
12	VDD_VCO	I	VCO supply voltage input.
13	CP	O	Charge-pump output. Connect to loop filter.
14	NC		
15	XO	O	Crystal oscillator output. Connect to tank capacitor.
16	XI	I	Crystal oscillator input. Connect to tank capacitor.
17	GND_PLL	O	PLL ground pin.
18	VDD_PLL	I	PLL supply voltage input driven by pin 11, VDD_D.
19	P0.0	DIO	SPI_SCLK
20	P0.1	DIO	SPI_MOSI
21	P0.2	DIO	SPI_MISO
22	P0.3	DIO	SPI_SSEL
23	P0.4	DIO	GPIO/ ICE mode
24	P0.5	DIO	I2C_SCL
25	P0.6	DIO	I2C_SDA
26	P0.7	DIO	INT2 /GIO1
27	P1.0	DIO	Timer2_T2
28	P1.1	DIO	Timer2_T2EX
29	P1.2	DIO	INT3 /GIO2
30	P1.3	DIO	INT4/ CKO
31	P1.4	DIO	TTAG_TTDIO
32	P1.5	DIO	TTAG_TTCK
33	P1.6	DIO	PWM0
34	P1.7	DIO	PWM1
35	P3.0	DIO	UART0_RX
36	P3.1	DIO	UART0_TX
37	P3.2	DIO	INT0/ADC0
38	P3.3	DIO	INT1/ADC1
39	P3.4	DIO	Timer0_T0/ADC2
40	P3.5	DIO	Timer1_T1/ADC3
41	P3.6	DIO	RTC_I
42	P3.7	DIO	RTC_O
43	RESETN	I	Reset input
44	VDD_D	O	Digital supply voltage output. Connect to bypass capacitor.
45	VDD_T	I	Timer supply voltage input. Connect to bypass capacitor.
46	VDD_S	I	Digital supply voltage output. Connect to bypass capacitor.
47	BP_BG	O	Band-gap bypass. Connect to bypass capacitor.
48	REGI	I	Regulator input. Connect to VDD supply.
	Back side plate	**G**	**Ground. Back side plate shall be well-solder to ground; otherwise, it will impact RF performance.**

6. Chip Block Diagram

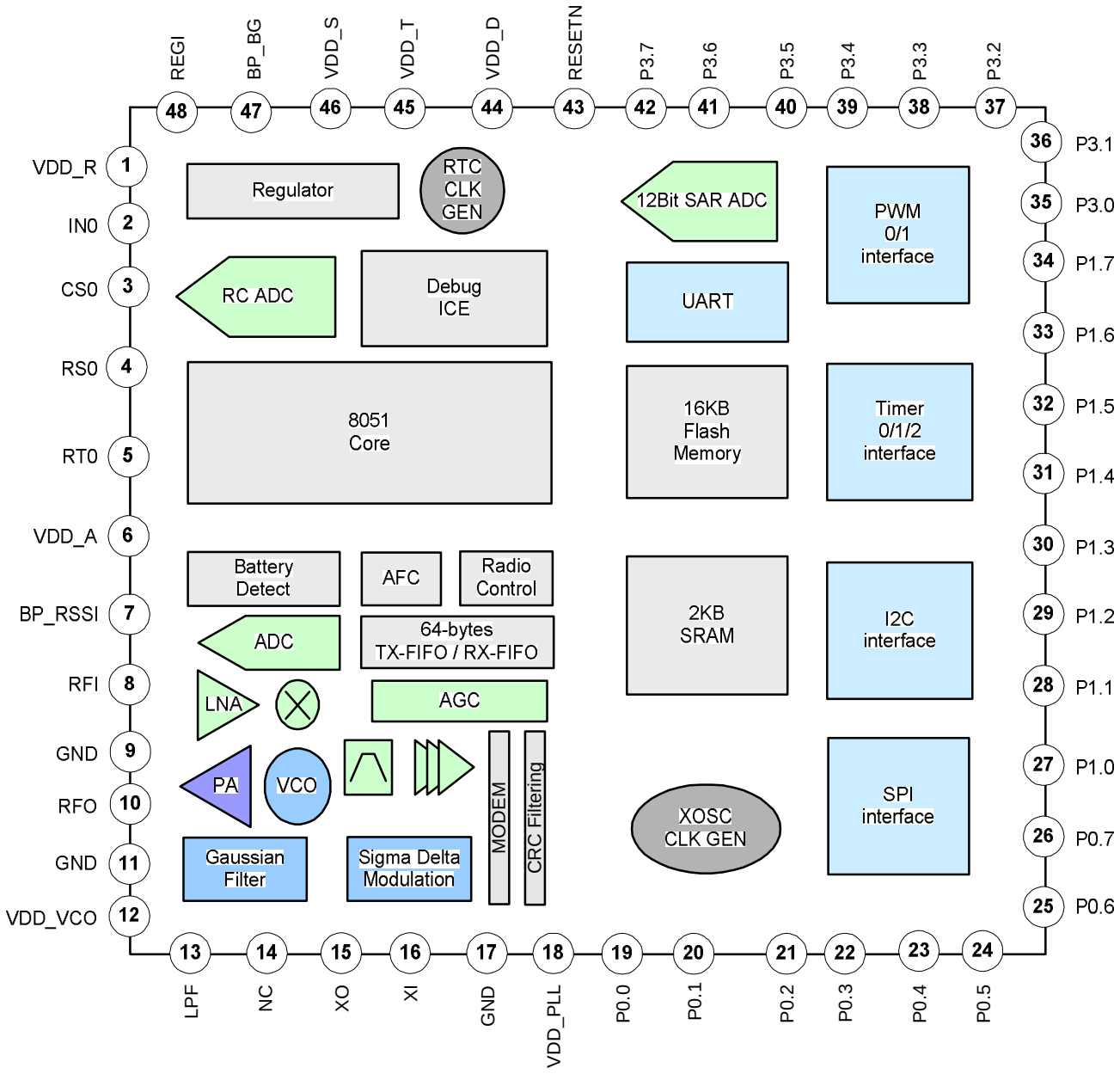


Fig 6-1. A9108 Block Diagram

7. Absolute Maximum Ratings

Parameter	With respect to	Rating	Unit
Supply voltage range (VDD)	GND	-0.3 ~ 3.6	V
Digital IO pins range	GND	-0.3 ~ VDD+0.3	V
Voltage on the analog pins range	GND	-0.3 ~ 2.1	V
Input RF level		14	dBm
Storage Temperature range		-55 ~ 125	°C
ESD Rating	HBM	± 2K	V
	MM	± 100	V

*Stresses above those listed under “Absolute Maximum Rating” may cause permanent damage to the device. These are stress ratings only; functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

*Device is ESD sensitive. Use appropriate ESD precautions. HBM (Human Body Mode) is tested under MIL-STD-883F Method 3015.7. MM (Machine Mode) is tested under JEDEC EIA/JESD22-A115-A.
*Device is Moisture Sensitivity Level III (MSL 3).

8. Electrical Specification

(Ta=25℃, VDD=3.3V, data rate= 250Kbps, F_{XTAL} =16MHz, On Chip Regulator = 1.8V, PN9 pattern, with matching network and low pass filter, unless otherwise noted.)

Parameter	Description	Min.	Typ.	Max.	Unit
General					
Operating Temperature		-40		85	°C
Supply Voltage		2.0	3.3	3.6	V
Current Consumption	Normal mode		TBD		
	PMM mode		TBD		
	Idle mode (No CPU clock)		TBD		
	Sleep Mode		5.5		uA
	Deep Sleep Mode		1.2		uA
Current Consumption 433MHz band	PLL mode		8.5		mA
	RX mode (AGC Off)		13.5		mA
	RX mode (AGC On)		14.5		mA
	TX -12dBm (TBG=0, TDC=0, PAC=0)		16		mA
	TX 1dBm (TBG=1, TDC=0, PAC=0)		20		mA
	TX 5dBm (TBG=2, TDC=0, PAC=0)		22		mA
	TX 10dBm (TBG=3, TDC=0, PAC=0)		30		mA
	TX 13dBm (TBG=4, TDC=0, PAC=0)		39		mA
	TX 15dBm (TBG=5, TDC=0, PAC=0)		48		mA
	TX 16dBm (TBG=6, TDC=0, PAC=0)		55		mA
	TX 17dBm (TBG=7, TDC=2, PAC=1)		70		mA
	TX 17.5dBm (TBG=7, TDC=3, PAC=3)		78		mA
Current Consumption 315MHz band	TX 20dBm (TBG=7, TDC=2, PAC=1) Without LPF and HPF.		70		mA
Current Consumption 868MHz band	PLL mode		8.5		mA
	RX mode (AGC Off)		14		mA
	RX mode (AGC On)		15.5		mA
	TX -16dBm (TBG=0, TDC=0, PAC=0)		16		mA
	TX -2dBm (TBG=3, TDC=0, PAC=0)		20		mA
	TX 2dBm (TBG=4, TDC=0, PAC=0)		23		mA
	TX 6dBm (TBG=5, TDC=0, PAC=0)		29		mA
	TX 10dBm (TBG=6, TDC=0, PAC=0)		37		mA
	TX 12dBm (TBG=7, TDC=0, PAC=0)		45		mA
	TX 13dBm (TBG=6, TDC=1, PAC=0)		52		mA
	TX 15dBm (TBG=7, TDC=1, PAC=0)		60		mA
	TX 16dBm (TBG=7, TDC=2, PAC=0)		70		mA
	TX 17dBm (TBG=7, TDC=3, PAC=0)		75		mA
	TX 18dBm (TBG=7, TDC=2, PAC=3)		93		mA
Phase Locked Loop					
X'TAL Settling Time [2]	Idle to standby, 49US type		0.5		ms
X'TAL frequency	General case		**12.8**/16		MHz
	Data rate = 250Kbps		16		MHz

	Data rate = 32.768K or 16.384Kbps		12.582912		MHz
	Data rate = 38.4Kbps		19.6608		MHz
X'TAL ESR				100	Ohm
X'TAL Capacitor Load (Cload)	Recommended		20		pF
433MHz PLL Phase noise (loop component: R1=820,C1=33nF,C2=2.2nF)	PN @100k offset		90		dBc/Hz
	PN @1M offset		110		dBc/Hz
	PN @10M offset		130		dBc/Hz
868MHz PLL Phase noise (loop component: R1=560,C1=47nF,C2=3.3nF)	PN @100k offset		85		dBc/Hz
	PN @1M offset		105		dBc/Hz
	PN @10M offset		125		dBc/Hz
PLL Settling Time @settle to 25kHz	Standby to PLL		35		us
Reference spur			80		dBc
Transmitter					
TX Power Range	433MHz (excluding LPF and HPF)	-12	13	20	dB
	868MHz (excluding LPF and HPF)	-16	12	20	dB
TX Settling Time	PLL to TX		30		µs
TX Spurious Emission 1. Pout = 12 dBm 2. With LPF and HPF	f < 1GHz (RBW =100kHz)			-36	dBm
	47MHz< f <74MHz 87.5MHz< f <118MHz 174MHz< f <230MHz 470MHz< f <862MHz (RBW =100kHz)			-54	dBm
	Above 1GHz (RBW = 1MHz)			-30	dBm
	2nd Harmonic			-30	dBm
	3rd Harmonic			-30	dBm
Receiver					
IF Frequency	50K Mode		100		KHz
	100K Mode		200		
	150K Mode		300		
	250K Mode		500		
IF Filter Bandwidth	50K Mode (10 ppm Xtal needed)		50		KHz
	100K Mode		100		
	150K Mode		150		
	250K Mode		250		
315MHz RX Sensitivity [3] @BER=0.1% high gain mode	50kbps (Fdev = 18.75KHz)		-111		dBm
	100kbps (Fdev = 37.5KHz)		-108		
	150kbps (Fdev = 56.25KHz)		-107		
	250kbps (Fdev = 93.75KHz),16MHz Xtal		-104		
480MHz RX Sensitivity [3] @BER=0.1% high gain mode	10kbps (FSK) (IFBW = 50KHz, Fdev = 25KHz)		-115		dBm
	10kbps (GFSK) (IFBW = 50KHz, Fdev = 25KHz)		-114		
433MHz RX Sensitivity [3] @BER=0.1% high gain mode	2kbps (IFBW = 50KHz, Fdev = 8KHz)		-117		dBm
	2kbps (IFBW = 100KHz, Fdev = 8KHz)		- 114		dBm
	10kbps (IFBW = 50KHz, Fdev = 18.75KHz)		-114		dBm
	10kbps (IFBW = 100KHz, Fdev = 37.5KHz)		-112		dBm
	50kbps (Fdev = 18.75KHz)		-110		dBm
	100kbps (Fdev = 37.5KHz)		-107		
	150kbps (Fdev = 56.25KHz)		-106		

	250kbps (Fdev = 93.75KHz) ,16MHz Xtal		-103		
868MHz RX Sensitivity [3] @BER=0.1% high gain mode	2kbps (IFBW = 50KHz, Fdev = 8KHz)		-114		dBm
	2kbps (IFBW = 100KHz, Fdev = 8KHz)		-109		dBm
	10kbps (IFBW = 50KHz, Fdev = 18.75KHz)		-110		dBm
	10kbps (IFBW = 100KHz, Fdev = 37.5KHz)		-111		dBm
	50kbps (Fdev = 18.75KHz)		-106		dBm
	100kbps (Fdev = 37.5KHz)		-103		
	150kbps (Fdev = 56.25KHz)		-102		
	250kbps (Fdev = 93.75KHz),16MHz Xtal		-100		
915MHz RX Sensitivity [3] @BER=0.1% high gain mode	50kbps (Fdev = 18.75KHz)		-105		dBm
	100kbps (Fdev = 37.5KHz)		-101		
	150kbps (Fdev = 56.25KHz)		-100		
	250kbps (Fdev = 93.75KHz),16MHz Xtal		-97		
Image rejection			20		dB
Interference (868.3MHz, 100Kbps)	Co-channel		-14		dB
	ACR1 (C/I_{ch1})		21		dB
	ACR2 (C/I_{ch2})		37		dB
	Offset ± 10MHz		50		dB
RX Spurious	25MHz ~ 1GHz			-57	dBm
	Above 1GHz			-47	dBm
RSSI Range	AGC on	-110		-30	dBm
Max Operation Input Power	@ RF input (BER = 0.1%)			10	dBm
RX Settling Time	PLL to RX		30		us
	Standby to RX		250		us
12Bit SAR ADC					
Input voltage range		0		1.8	V
External reference voltage			1.8		V
Input capacitor			TBD		pF
Bandwidth			TBD		KHz
EOB, effective number of bits			TBD		bit
SINAD, signal to noise and distortion			TBD		db
Conversion time		80			uS
Current consumption			0.4		mA
24Bit ADC					
Resistor for oscillation	CS0, CT0 > 740pF	1			Kohm
Oscillation frequency	Resistor for oscillation=1Kohm		TBD		KHz
	Resistor for oscillation=10Kohm		TBD		KHz
	Resistor for oscillation=100Kohm		TBD		KHz
Current consumption			TBD		mA
Regulator					
Regulator settling time	Pin 19 connected to 1nF		450		µs
Band-gap reference voltage			1.2		V
Regulator output voltage		1.8	**1.8**	2.1	V
Digital IO DC characteristics					
High Level Input Voltage (V_{IH})		0.8*VDD		VDD	V
Low Level Input Voltage (V_{IL})		0		0.2*VDD	V

High Level Output Voltage (V_{OH})	@I_{OH}= -0.5mA	VDD-0.4		VDD	V
Low Level Output Voltage (V_{OL})	@I_{OL}= 0.5mA	0		0.4	V

Note 1:

9. SFR & RFR(Radio Frequency Register)

A9108 contains standard 8051 SFRs(special function registers) and RFR (RF control registers). A9108's SFR location is almost the same as the standard 8052 SFR location. RFR is Radio Frequency Registers are located in XDATA spaces and located in 0x0800 ~ 0x08FF. For more detail information, please reference Section 9.2.

9.1 SFR Overview

Table 9.1 A9108 Special Function Registers (SFRs) table

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	EIP	OSCCON						
0xF0	B	I2CSADR	I2CSCR	I2CSBUF	I2CMSA	I2CMCR	I2CMBUF	I2CMTP
0xE8	EIE				SPCR	SPSR	SPDR	SSCR
0xE0	ACC	P3OE	P3PUN	P3WUN	SPCR1	SPSR1	SPDR1	SSCR1
0xD8	WDCON	P1OE	P1PUN	P1WUN				
0xD0	PSW	P0OE	P0PUN	P0WUN				
0xC8	T2CON	T2IF	RLDL	RLDH	TL2	TH2		DEVICR
0xC0	SCON1	SBUF1						
0xB8	IP	PCONE	RSFLAG	IOSEL				
0xB0	P3	PWM1CON	PWM1H	PWM1L				
0xA8	IE	PWM0CON	PWM0H	PWM0L				
0xA0	P2							
0x98	SOCN0	SBUF0	FLASHCTRL	FLASHMR				
0x90	P1	EIF					USBADDR	USBDATA
0x88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	DMAIR
0x80	P0	SP	DPL0	DPH0	DPL1	DPH1	DPS	PCON

: It means bit-addressable
: It means reserved.

Following are description of SFRs related to the operation of A9108 System Controller. Detailed descriptions of the remaining SFRs are including the sections of the datasheet associated with their corresponding system function. The arithmetic section of the processor performs extensive data manipulation and is comprised of the 8-bit arithmetic logic unit (ALU), an ACC(0xE0) register, B(0xF0) register and PSW(0xD0) register.

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
E0h ACC Reset	R/W	0	0	0	0	0	0	0	0

Accumulator A Register

The B register is used during multiply and divide operations. In other cases may be used as normal SFR.

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F0h B Reset	R/W	0	0	0	0	0	0	0	0

B Register

The ALU performs typical arithmetic operations as: addition, subtraction, multiplication, division and additional operations such as: increment, decrement, BCD-decimal-add-adjust and compare. Within logic unit are performance: AND, OR, Exclusive OR, complement and rotation. The Boolean processor performance the bit operations as: set, clear, complement, jump-if-not-set, jump-if-set-and-clear and move to/from carry.

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D0h PSW	R/W	CY	AC	F0	RS1	RS2	OV	F1	P
Reset		0	0	0	0	0	0	0	0

Program Status Word register

CY - Carry flag
AC - Auxiliary carry
F0 - General purpose flag 0
RS[1:0] - Register bank select bits

RS[1:0]	Function description
00	- Bank 0, data address 0x00-0x07
01	- Bank 1, data address 0x08-0x0F
10	- Bank 2, data address 0x10-0x17
11	- Bank 3, data address 0x18-0x1F

OV - Overflow flag
F1 - General purpose flag 1
P - Parity flag
The PSW contains several bits that reflect the current state of the CPU.

9.2 RFR Overview

Address / Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x800h RSTCTL	W	RESETN	FWPRN	FRPRN	ADC12RN	FIFORN	BFCRN	RCADCRN	WORRN
	R	--	FECF	CRCF	CER	XER	PLLER	TRSR	TRER
0x801h MODEC1	W	STRB7	STRB6	STRB5	STRB4	STRB3	STRB2	STRB1	STRB0
0x802h MODEC2	W/R	STRR	ARSSI	--	MSCD	WOR_EN	FMT	FMS	ADCM
0x803h CALC	W/R	WORS[3:0]				VCC	VBC	FBC	RSSCR
0x804h FIFO1	W	FEP[7:0]							
0x805h FIFO2	W	FPM[1:0]		TPSA[5:0]					
0x806h RCOSC1	W	WORDLY[7:0]							
0x807h RCOSC2	W	WORDLY[9:8]		WRDLY[5:0]					
0x808h RCOSC3	W	SPSS	--	RCOT2	RCOT1	RCOT0	TMRE	TSEL	TWOR
0x809h RCOSC4	W	--	--	--	WSEL1	WSEL0	MVS1	MVS0	ENCAL
	R	NUMLH11	NUMLH10	NUMLH9	NUMLH8	--	RCOC9	RCOC8	ENCAL
0x80Ah	W	MRCT9	MRCT8	--	--	--	--	MAN	MCALS

RCOSC5	R	NUMLH7	NUMLH6	NUMLH5	NUMLH4	NUMLH3	NUMLH2	NUMLH1	NUMLH0
0x80Bh RCOSC6	W	MRCT7	MRCT6	MRCT5	MRCT4	MRCT3	MRCT2	MRCT1	MRCT0
	R	RCOC7	RCOC6	RCOC5	RCOC4	RCOC3	RCOC2	RCOC1	RCOC0
0x80Ch CKO	W	PRS	CKOS[3:0]				CKOI	--	--
0x80Dh GPIO1	W	HWCKS	WRCKS	GIO1S[3:0]				GIO1I	--
0x80Eh GPIO2	W	MCNT[1:0]		GIO2S[3:0]				GIO2I	--
0x80Fh CLOCK	W/R	GRS	GRC[4:0]					CGS	XS
0x810h PLL1	W	CPS	CPC[1:0]		MDIV	RRC[3:0]			
0x811h PLL2	W	CKX2	MD[1:0]		VCS[1:0]		SDPW	NSDO	EDI
0x812h PLL3	W	IP[7:0]							
0x813h PLL4	W	FP[15:8]							
0x814h PLL5	W	FP[7:0]							
0x815h PLL6	W/R	AFC	MC[14:8]						
0x816h PLL7	W/R	MC[7:0]							
0x817h CHG1	W/R	IPL[7:0]							
0x818h CHG2	W/R	IPH[7:0]							
0x819h CHG3	W/R	FPH[3:0]				FPL[3:0]			
0x81Ah TX1	W	MCNTR	BT[1:0]		TME	GS	FDP[2:0]		
0x81Bh TX2	W	FD[7:0]							
0x81Ch DELAY1	W	DPRY[2:0]			TDLY[1:0]		PDLY[2:0]		
0x81Dh DELAY2	W	WSEL[2:0]			AGC_DLY[1:0]		RS_DLY[2:0]		
0x81Eh RX	W	--	AGCE	BW[1:0]		RXDI	DMG[1:0]		ULS
	R	ADCO[8]	--	--	--	--	--	--	--
0x81Fh ADCC	W	MXD	RADC	AVS[1:0]		MVS[1:0]		XADSR	CDM
	R	ADCO[7:0]							
0x820h RXAGC1	W	VRSEL	ERSSM	LGM[1:0]		MGM[1:0]		IGM[1:0]	
	R	--	--	LGC[1:0]		MGC[1:0]		IGC[1:0]	
0x821h RXAGC2	W	HDM	EXRSI	MS	MSCL[4:0]				
	R	RHM[7:0]							
0x822h RSSI	W	RTH[7:0]							
	R	ADC[7:0]							
0x823h AGCHT	W	IRTH[7:0]							
	R	RLM[7:0]							
0x824h AGCLT	W	IRTL[7:0]							
0x825h CODE1	W	MCS	WHTS	FECS	CRCS	--	--	PML[1:0]	

0x826h CODE2	W	ETH2	ETH1	ETH0	IDL1	IDL0	--	PMD1	PMD0
0x827h CODE3	W	MSCRC	WS6	WS5	WS4	WS3	WS2	WS1	WS0
0x828h IFC1	W	BGS	CRCDNP	CRCINV	MFBS	MFB3	MFB2	MFB1	MFB0
	R	--	--	--	FBCF	FB3	FB2	FB1	FB0
0x829h IFC2	W	--	TRT2	TRT1	TRT0	ASMV2	ASMV1	ASMV0	AMVS
	R	--	--	--	FCD4	FCD3	FCD2	FCD1	FCD0
0x82Ah VCOCC	W	SWT	--	--	VCOC3	VCOC2	VCOC1	VCOC0	MVCS
	R	--	--	--	VCCF	VCB3	VCB2	VCB1	VCB0
0x82Bh VCOBC1	W	--	--	--	VBS	MVBS	MVB2	MVB1	MVB0
	R	--	--	DVT1	DVT0	VBCF	VB2	VB1	VB0
0x82Ch VCOBC2	W	--	INTXC	VTL2	VTL1	VTL0	VTH2	VTH1	VTH0
0x82Dh PM	W	QDP	RGAGC1	RSAGC0	XCL4	XCL3	XCL2	XCL1	XCL0
0x82Eh RFI	W	RF23D1	RF23D0	PRRC1	PRRC0	PRIC1	PRIC0	RMP1	RMP0
0x82Fh XTST	W	QCLIM	--	RXCC	RXCP1	RXCP0	XCC	XCP1	XCP0
0x830h BD	W	CA1	CA0	RGV1	RGV0	BVT2	BVT1	BVT0	BDS
	R	--	--	--	VBD	--	--	--	--
0x831h TXT1	W	TXDI	PAC1	PAC0	TDC1	TDC0	TBG2	TBG1	TBG0
0x832h TXT2	W	QDS	RFT2	RFT1	RFT0	LODV1	LODV0	TXIB1	TXIB0
0x833h RXDEM1	W	CST	DMT	MLP1	MLP0	SLF2	SLF1	SLF0	DMOS
0x834h RXDEM2	W/R	DCL2	DCL1	DCL0	DCM1	DCM0	CSC2	CSC1	CSC0
0x835h RXDEM3	W	DCV7	DCV6	DCV5	DCV4	DCV3	DCV2	DCV1	DCV0
0x836h DRCK	W/R	--	SDR6	SDR5	SDR4	SDR3	SDR2	SDR1	SDR0
0x837h RTC	W	--	--	--	RTCOE	RTCI	RTC1	RTC0	RTCE
0x838h ID0	W/R	ID63	ID62	ID61	ID60	ID59	ID58	ID57	ID56
0x839h ID1	W/R	ID55	ID54	ID53	ID52	ID51	ID50	ID49	ID48
0x83Ah ID2	W/R	ID47	ID46	ID45	ID44	ID43	ID42	ID41	ID40
0x83Bh ID3	W/R	ID39	ID38	ID37	ID36	ID35	ID34	ID33	ID32
0x83Ch ID4	W/R	ID31	ID30	ID29	ID28	ID27	ID26	ID25	ID24
0x83Dh ID5	W/R	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
0x83Eh ID6	W/R	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
0x83Fh ID7	W/R	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
0x840h DID0	R	DID31	DID30	DID29	DID28	DID27	DID26	DID25	DID24
0x841h DID1	R	DID23	DID22	DID21	DID20	DID19	DID18	DID17	DID16

0x842h DID2	R	DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8
0x843h DID3	R	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0
0x844h RADCA0	W/R	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
0x845h RADCA1	W/R	RA15	RA14	RA13	RA12	RA11	RA10	RA9	RA8
0x846h RADCA2	W/R	RA23	RA22	RA21	RA20	RA19	RA18	RA17	RA16
0x847h RADCB0	W/R	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
0x848h RADCB1	W/R	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8
0x49h RADCB2	W/R	RB23	RB22	RB21	RB20	RB19	RB18	RB17	RB16
0x84Ah RADMOD	W/R	RACK2	RACK1	RACK0	MOD	CKS1	CKS0	HSC1	HSC0
0x84Bh RADWTC	W/R	WTC7	WTC6	WTC5	WTC4	WTC3	WTC2	WTC1	WTC0
0x84Ch RADCON	W/R	OVF_FLG	MVT[1]	MVT[0]	--	RCCS	RCEC	RCEN	RARUN
0x84Dh ADCCTL	W	--	CKS1	CKS0	MODE	MVS[2]	MVS[1]	MVS[0]	ADCE
	R	--	--	--	MODE	MVS[2]	MVS[1]	MVS[0]	ADCE
0x84Eh ADCAVG1	R	MVADC[11]	MVADC[10]	MVADC[9]	MVADC[8]	ADC[11]	ADC[10]	ADC[9]	ADC[8]
0x84Fh ADCAVG2	R	MVADC[7]	MVADC[6]	MVADC[5]	MVADC[4]	MVADC[3]	MVADC[2]	MVADC[1]	MVADC[0]
0x850h ADCAVG3	R	ADC[7]	ADC[6]	ADC[5]	ADC[4]	ADC[3]	ADC[2]	ADC[1]	ADC[0]
0x851h EXT1	W/R	--	--	CTR5	CTR4	CTR3	CTR2	CTR1	CTR0
0x852h EXT2	W/R	--	BOD	REGR	CGB4	CGB3	CGB2	CGB1	CGB0

9.2.0 RSTCTL (Address: 0800h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSTCTL	W	RESETN	FWPRN	FRPRN	--	FIFORN	BFCRN	--	--
	R	PWR	FECF	CRCF	CER	XER	PLLER	TRSR	TRER
Reset		0	0	0	0	0	0	0	0

RESETN: Software reset for baseband
FWPRN: Software reset for TX fifo pointer.
FRPRN: Software reset for RX fifo pointer.
FIFORN: Software reset for RX fifo.
BFCRN:

PWR: Power Status
[0]: Power off.
[1]: Power on.

CER:
[0]:
[1]:

XER: Crystal Status
[0]: Disable.
[1]: Enable.

PLLER: PLL Status.
[0]: Disable.
[1]: Enable.

TRSR: TRX Mode Status.
[0]: RX mode.
[1]: TX mode.

TRER: TRX Status.
[0]: Disable.
[1]: Enable.

9.2.1 MODEC1 (Address: 0801h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MODEC1	W	STRB7	STRB6	STRB5	STRB4	STRB3	STRB2	STRB1	STRB0
Reset		0	0	0	0	0	0	0	0

Strobe Command								Description
STRB 7	STRB 6	STRB 5	STRB 4	STRB 3	STRB 2	STRB 1	STRB 0	
1	0	0	0	0	x	x	x	Sleep mode
1	0	0	1	x	x	x	x	Idle mode
1	0	1	0	x	x	x	x	Standby mode
1	0	1	1	x	x	x	x	PLL mode
1	1	0	0	x	x	x	x	RX mode
1	1	0	1	x	x	x	x	TX mode

9.2.2 MODEC2 (Address: 0802h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MODEC2	W/R	STRR	ARSSI	--	MSCD	WOR_EN	FMT	FMS	ADCM
Reset		0	0	0	0	0	0	0	0

STRR: .

ARSSI: Auto RSSI measurement enable.
[0]: Disable.
[1]: Enable.

MSCD: .

WOR_EN: .

FMT: .

FMS: .

ADCM: .

9.2.3 CALC (Address: 0803h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CALC	W/R	WORS3	WORS2	WORS1	WORS0	VCC	VBC	FBC	RSSCR
Reset		0	0	0	0	0	0	0	0

WORS[3:0]: .

VCC: .

VBC: .

FBC: .

RSSCR: .

9.2.4 FIFO1 (Address: 0804h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FIFO1	W	FEP7	FEP6	FEP5	FEP4	FEP3	FEP2	FEP1	FEP0
Reset		0	0	0	0	0	0	0	0

FEP[7:0]: .

9.2.5 FIFO2 (Address: 0805h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FIFO2	W	FPM1	FPM0	TPSA5	TPSA4	TPSA3	TPSA2	TPSA1	TPSA0
Reset		0	0	0	0	0	0	0	0

FPM[1:0]: .

TPSA[5:0]: .

9.2.6 RCOSC1 (Address: 0806h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCOSC1	W	WORDLY7	WORDLY6	WORDLY5	WORDLY4	WORDLY3	WORDLY2	WORDLY1	WORDLY0
Reset		0	0	0	0	0	0	0	0

WORDLY[9:0]: .

9.2.7 RCOSC2 (Address: 0807h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCOSC2	W	WORDLY9	WORDLY8	WRDLY5	WRDLY4	WRDLY3	WRDLY2	WRDLY1	WRDLY0
Reset		0	0	0	0	0	0	0	0

WORDLY[9:0]: .

WRDLY[5:0]: .

9.2.8 RCOSC3 (Address: 0808h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCOSC3	W	SPSS	--	RCOT2	RCOT1	RCOT0	TMRE	TSEL	TWOR
Reset		0	0	0	0	0	0	0	0

SPSS: .

RCOT[2:0]: .

TMRE: .

TSEL: .

TWOR: .

9.2.* RCOSC4 (Address: 0809h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCOSC4	W	--	--	--	WSEL1	WSEL0	MVS1	MVS0	ENCAL

	R	NUMLH11	NUMLH10	NUMLH9	NUMLH8	--	RCOC9	RCOC8	ENCAL
Reset		0	0	0	0	0	0	0	0

ENCAL: Enable RC-OSC Calibration.

MVS[1:0]: Main clock divider.
[00]: MSCK
[01]: MSCK / 2
[10]: MSCK / 3
[11]: MSCK / 4

WSEL[1:0]: RC-OSC Calibration Source Clock Selection.
[00]: 16 MHz.
[01]: 8 MHz.
[10]: 4 MHz.
[11]: 2 MHz.

RCOC[9:0]: RC-OSC calibration value.

TGNUM[11:0]: RC-OSC calibration target number.

NUMLH[11:0]: RC-OSC calibration latch number.

9.2.10 RCOSC5 (Address: 080Ah)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCOSC5	W	MRCT9	MRCT8	--	--	--	--	MAN	MCALS
	R	NUMLH7	NUMLH6	NUMLH5	NUMLH4	NUMLH3	NUMLH2	NUMLH1	NUMLH0
Reset		0	0	0	0	0	0	0	0

MCALS: Enable Continuous RC-OSC Calibration.

MAN: Enable Manual RC-OSC Calibration.

MRCT[9:0]: Manual RC-OSC calibration value setting.

NUMLH[11:0]: RC-OSC calibration latch number.

9.2.11 RCOSC6 (Address: 080Bh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCOSC6	W	MRCT7	MRCT6	MRCT5	MRCT4	MRCT3	MRCT2	MRCT1	MRCT0
	R	RCOC7	RCOC6	RCOC5	RCOC4	RCOC3	RCOC2	RCOC1	RCOC0
Reset		0	0	0	0	0	0	0	0

MRCT[9:0]: Manual RC-OSC calibration value setting.

RCOC[9:0]: RC-OSC calibration value.

9.2.12 CKO (Address: 080Ch)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CKO	W	PRS	CKOS3	CKOS2	CKOS1	CKOS0	CKOI	--	--
Reset		0	0	0	0	0	0	0	0

PRS: .

CKOS[3:0]: .

CKOI: .

9.2.13 GPIO1 (Address: 080Dh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GPIO1	W	HWCKS	WRCKS	GIOS3	GIOS2	GIOS1	GIOS0	GIO1I	--
Reset		0	0	0	0	0	0	0	0

HWCKS: .

WRCKS: .

GIO1S[3:0]: .

GIO1I: .

9.2.14 GPIO2 (Address: 080Eh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GPIO2	W	MCNT1	MCNT0	GIO2S3	GIO2S2	GIO2S1	GIO2S0	GIO2I	--
Reset		0	0	0	0	0	0	0	0

MCNT[1:0]: .

GIO2S[3:0]: .

GIO2I: .

9.2.15 CLOCK (Address: 080Fh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLOCK	W/R	GRS	GRC4	GRC3	GRC2	GRC1	GRC0	CGS	XS
Reset		0	0	0	0	0	0	0	0

GRS: .

GRC[4:0]: .

CGS: .

XS: .

9.2.16 PLL1 (Address: 0810h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLL1	W	CPS	CPC1	CPC0	MDIV	RRC3	RRC2	RRC1	RRC0
Reset		0	0	0	0	0	0	0	0

CPS: .

CPC[1:0]: .

MDIV: .

RRC[3:0]: .

9.2.17 PLL2 (Address: 0811h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLL2	W	CKX2	MD1	MD0	VCS1	VCS0	SDPW	NSDO	EDI
Reset		0	0	0	0	0	0	0	0

9.2.18 PLL3 (Address: 0812h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLL3	W	IP7	IP6	IP5	IP4	IP3	IP2	IP1	IP0
Reset		0	0	0	0	0	0	0	0

9.2.19 PLL4 (Address: 0813h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLL4	W	FP15	FP14	FP13	FP12	FP11	FP10	FP9	FP8
Reset		0	0	0	0	0	0	0	0

9.2.20 PLL5 (Address: 0814h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLL5	W	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
Reset		0	0	0	0	0	0	0	0

9.2.21 PLL6 (Address: 0815h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLL6	W/R	AFC	MC14	MC13	MC12	MC11	MC10	MC9	MC8
Reset		0	0	0	0	0	0	0	0

9.2.22 PLL7 (Address: 0816h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PLL7	W/R	MC7	MC6	MC5	MC4	MC3	MC2	MC1	MC0
Reset		0	0	0	0	0	0	0	0

9.2.23 CHG1 (Address: 0817h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CHG1	W/R	IPL7	IPL6	IPL5	IPL4	IPL3	IPL2	IPL1	IPL0
Reset		0	0	0	0	0	0	0	0

9.2.24 CHG2 (Address: 0818h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CHG2	W/R	IPH7	IPH6	IPH5	IPH4	IPH3	IPH2	IPH1	IPH0
Reset		0	0	0	0	0	0	0	0

9.2.25 CHG3 (Address: 0819h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CHG3	W/R	FPH3	FPH2	FPH1	FPH0	FPL3	FPL2	FPL1	FPL0
Reset		0	0	0	0	0	0	0	0

9.2.26 TX1 (Address: 081Ah)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TX1	W	MCNTR	BT1	BT0	TME	GS	FDP2	FDP1	FDP0
Reset		0	0	0	0	0	0	0	0

9.2.27 TX2 (Address: 081Bh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TX2	W	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
Reset		0	0	0	0	0	0	0	0

9.2.28 TX2 (Address: 081Ch)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TX2	W	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
Reset		0	0	0	0	0	0	0	0

9.2.29 DELAY1 (Address: 081Dh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DELAY1	W	DPRY2	DPRY1	DPRY0	TDLY1	TDLY0	PDLY2	PDLY1	PDLY0
Reset		0	0	0	0	0	0	0	0

9.2.30 DELAY2 (Address: 081Eh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DELAY2	W	WSEL2	WSEL1	WSEL0	AGC_DLY1	AGC_DLY0	RS_DLY2	RS_DLY1	RS_DLY0
Reset		0	0	0	0	0	0	0	0

9.2.31 RX (Address: 081Fh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RX	W	--	AGCE	BW1	BW0	RXDI	DMG1	DMG0	ULS
	R	ADCO8	--	--	--	--	--	--	--
Reset		0	0	0	0	0	0	0	0

9.2.32 ADCC (Address: 0820h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCC	W	MXD	RADC	AVS1	AVS0	MVS1	MVS0	XADSR	CDM
	R	ADCO7	ADCO6	ADCO5	ADCO4	ADCO3	ADCO2	ADCO1	ADCO0
Reset		0	0	0	0	0	0	0	0

9.2.33 RXAGC1 (Address: 0821h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXAGC1	W	VRSEL	ERSSM	LGM1	LGM0	MGM1	MGM0	IGM1	IGM0
	R	--	--	LGC1	LGC0	MGC1	MGC0	IGC1	IGC0
Reset		0	0	0	0	0	0	0	0

9.2.34 RXAGC2 (Address: 0822h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXAGC2	W	HDM	EXRSI	MS	MSCL4	MSCL3	MSCL2	MSCL1	MSCL0
	R	RHM7	RHM6	RHM5	RHM4	RHM3	RHM2	RHM1	RHM0
Reset		0	0	0	0	0	0	0	0

9.2.35 RSSI (Address: 0823h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSSI	W	RTH7	RTH6	RTH5	RTH4	RTH3	RTH2	RTH1	RTH0
	R	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Reset		0	0	0	0	0	0	0	0

9.2.36 AGCHT (Address: 0824h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGCHT	W	IRTH7	IRTH6	IRTH5	IRTH4	IRTH3	IRTH2	IRHT1	IRTH0
	R	RML7	RLM6	RLM5	RLM4	RLM3	RLM2	RLM1	RLM0
Reset		0	0	0	0	0	0	0	0

9.2.37 AGCIT (Address: 0825h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AGCIT	W	IRTL7	IRTL6	IRTL5	IRTL4	IRTL3	IRTL2	IRTL1	IRTL0
Reset		0	0	0	0	0	0	0	0

9.2.38 CODE1 (Address: 0826h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CODE1	W	MCS	WHTS	FECS	CRCS	--	--	PML1	PML0
Reset		0	0	0	0	0	0	0	0

9.2.39 CODE2 (Address: 0826h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CODE2	W	ETH2	ETH1	ETH0	IDL1	IDL0	--	PMD1	PMD0
Reset		0	0	0	0	0	0	0	0

9.2.40 CODE3 (Address: 0827h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CODE3	W	MSCRC	WS6	WS5	WS4	WS3	WS2	WS1	WS0
Reset		0	0	0	0	0	0	0	0

9.2.41 IFC1 (Address: 0828h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IFC1	W	BGS	CRCDNP	CRCINV	MFBS	MFB3	MFB2	MFB1	MFB0
	R	--	--	--	FBCF	FB3	FB2	FB1	FB0
Reset		0	0	0	0	0	0	0	0

9.2.42 IFC2 (Address: 0829h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IFC2	W	--	TRT2	TRT1	TRT0	ASMV2	ASMV1	ASMV0	AMVS
	R	--	--	--	FCD4	FCD3	FCD2	FCD1	FCD0
Reset		0	0	0	0	0	0	0	0

9.2.43 VCOCC (Address: 082Ah)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VCOCC	W	SWT	--	--	VCOC3	VCCO2	VCOC1	VCOC0	MVCS
	R	--	--	--	VCCF	VCB3	VCB2	VCB1	VCB0
Reset		0	0	0	0	0	0	0	0

9.2.44 VCOBC1 (Address: 082Bh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VCOBC1	W	--	--	--	VBS	MVBS	MVB2	MVB1	MVB0
	R	--	--	DVT1	DVT0	VBCF	VB2	VB1	VB0
Reset		0	0	0	0	0	0	0	0

9.2.45 VCOBC2 (Address: 082Ch)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VCOBC2	W	--	INTXC	VTL2	VTL1	VTL0	VTH2	VTH1	VTH0
Reset		0	0	0	0	0	0	0	0

9.2.46 PM (Address: 082Dh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

PM	W	QDP	RGAGC1	RGAGC0	XCL4	XCL3	XCL2	XCL1	XCL0
Reset		0	0	0	0	0	0	0	0

9.2.47 RFI (Address: 082Eh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFI	W	RF23D1	RF23D0	PRRC1	PRRC0	PRIC1	PRIC0	RMP1	RMP0
Reset		0	0	0	0	0	0	0	0

9.2.48 XTST (Address: 082Fh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XTST	W	QCLIM	--	RXCC	RXCP1	RXCP0	XCC	XCP1	XCP0
Reset		0	0	0	0	0	0	0	0

9.2.49 BD (Address: 0830h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BD	W	CA1	CA0	RGV1	RGV0	BVT2	BVT1	BVT0	BDS
	R	--	--	--	VBD	--	--	--	--
Reset		0	0	0	0	0	0	0	0

9.2.50 TXT1 (Address: 0831h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXT1	W	TXDI	PAC1	PAC0	TDC1	TDC0	TBG2	TBG1	TBG0
Reset		0	0	0	0	0	0	0	0

9.2.51 TXT2 (Address: 0832h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TXT2	W	QDS	RFT2	RFT1	RFT0	LODV1	LODV0	TXIB1	TXIB0
Reset		0	0	0	0	0	0	0	0

9.2.52 RXDEM1 (Address: 0833h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXDEM1	W	CST	DMT	MLP1	MLP0	SLF2	SLF1	SLF0	DMOS
Reset		0	0	0	0	0	0	0	0

9.2.53 RXDEM2 (Address: 0834h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXDEM2	W/R	DCL2	DCL1	DCL0	DCM1	DCM0	CSC2	CSC1	CSC0
Reset		0	0	0	0	0	0	0	0

9.2.54 RXDEM3 (Address: 0835h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RXDEM3	W	DCV7	DCV6	DCV5	DCV4	DCV3	DCV2	DCV1	DCV0
Reset		0	0	0	0	0	0	0	0

9.2.55 DRCK (Address: 0836h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DRCK	W/R	--	SDR6	SDR5	SDR4	SDR3	SDR2	SDR1	SDR0
Reset		0	0	0	0	0	0	0	0

9.2.56 RTC (Address: 0837h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

DRCK	W	--	--	--	RTCOE	RTCI	RTC1	RTC0	RTCE
Reset		0	0	0	0	0	0	0	0

9.2.57 ID0 (Address: 0838h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ID0	W/R	ID63	ID62	ID61	ID60	ID59	ID58	ID57	ID56
Reset		0	0	0	0	0	0	0	0

9.2.58 ID1 (Address: 0839h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ID1	W/R	ID55	ID54	ID53	ID52	ID51	ID50	ID49	ID48
Reset		0	0	0	0	0	0	0	0

9.2.59 ID2 (Address: 083Ah)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ID2	W/R	ID47	ID46	ID45	ID44	ID43	ID42	ID41	ID40
Reset		0	0	0	0	0	0	0	0

9.2.60 ID3 (Address: 083Bh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ID3	W/R	ID39	ID38	ID37	ID36	ID35	ID34	ID33	ID32
Reset		0	0	0	0	0	0	0	0

9.2.61 ID4 (Address: 083Ch)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ID4	W/R	ID31	ID30	ID29	ID28	ID27	ID26	ID25	ID24
Reset		0	0	0	0	0	0	0	0

9.2.62 ID5 (Address: 083Dh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ID5	W/R	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16
Reset		0	0	0	0	0	0	0	0

9.2.63 ID6 (Address: 083Eh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ID6	W/R	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8
Reset		0	0	0	0	0	0	0	0

9.2.64 ID7 (Address: 083Fh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ID7	W/R	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
Reset		0	0	0	0	0	0	0	0

9.2.65 DID0 (Address: 0840h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DID0	R	DID31	DID30	DID29	DID28	DID27	DID26	DID25	DID24
Reset		0	0	0	0	0	0	0	0

9.2.66 DID1 (Address: 0841h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DID1	R	DID23	DID22	DID21	DID20	DID19	DID18	DID17	DID16

Reset		0	0	0	0	0	0	0	0

9.2.67 DID2 (Address: 0842h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DID2	R	DID15	DID14	DID13	DID12	DID11	DID10	DID9	DID8
Reset		0	0	0	0	0	0	0	0

9.2.68 DID3 (Address: 0843h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DID3	R	DID7	DID6	DID5	DID4	DID3	DID2	DID1	DID0
Reset		0	0	0	0	0	0	0	0

9.2.4 RADCA0 (Address: 0844h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RADCA0	W/R	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
Reset		0	0	0	0	0	0	0	0

RA[23:0]: 24-bits Counter A.

9.2.4 RADCA1 (Address: 0845h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RADCA1	W/R	RA15	RA14	RA13	RA12	RA11	RA10	RA9	RA8
Reset		0	0	0	0	0	0	0	0

RA[23:0]: 24-bits Counter A.

9.2.4 RADCA2 (Address: 0846h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RADCA2	W/R	RA23	RA22	RA21	RA20	RA19	RA18	RA17	RA16
Reset		0	0	0	0	0	0	0	0

RA[23:0]: 24-bits Counter A.

9.2.4 RADCB0 (Address: 0847h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RADCB0	W/R	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
Reset		0	0	0	0	0	0	0	0

RB[23:0]: 24-bits Counter B.

9.2.4 RADCB1 (Address: 0848h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RADCB1	W/R	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8
Reset		0	0	0	0	0	0	0	0

RB[23:0]: 24-bits Counter B.

9.2.4 RADCB2 (Address: 0849h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RADCB2	W/R	RB23	RB22	RB21	RB20	RB19	RB18	RB17	RB16
Reset		0	0	0	0	0	0	0	0

RB[23:0]: 24-bits Counter B.

9.2.4 RADMOD (Address: 084Ah)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RADMOD	W/R	RACK2	RACK1	RACK0	MOD	CKS1	CKS0	HSC1	HSC0
Reset		0	0	0	0	0	0	0	0

RACK[2:0]: Select base clock for RC-ADC
[000]: LSCLK (31.25k Hz)
[001]: LSCLKx2 (62.5k Hz)
[010]: HSCLK (define by HSC[1:0])
[011]: HSCLK / 2
[100]: HSCLK / 4
[101]: HSCLK / 8
[110]: Prohibited.
[111]: Prohibited.

MOD: Operation Mode select.
[0]: Generates an interrupt request by counter A overflow.
[1]: Generates an interrupt request by counter B overflow.

CKS[1:0]: Select Ring oscillator clock.
[00]: RCOSC clock 0.
[01]: RCOSC clock 1.
[10]: RCOSC clock 2.
[11]: Prohibited.

HSC[1:0]: HSCLK clock rate selection.
[00]: 4M Hz
[01]: 2M Hz
[10]: 1M Hz
[11]: 500k Hz

9.2.4 RADWTC (Address: 084Bh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RADWTC	W/R	WTC7	WTC6	WTC5	WTC4	WTC3	WTC2	WTC1	WTC0
Reset		0	0	0	0	0	0	0	0

WTC[7:0]: Delay time for Ring oscillator settling.
Overall delay time = WTC[7:0] cycles x (base clock time).

9.2.4 RADCON (Address: 084Ch)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RADCON	W/R	OVF_FLG	MVT[1]	MVT[0]	--	RCCS	RCEC	RCEN	RARUN
Reset		0	0	0	0	0	0	0	0

OVF_FLG: Overflow Flag
[0]: Overflow. (Read only)
[1]: Manual clear overflow flag (Write only)

MVT[1:0]: Select moving average times.
[00]: No moving average.
[01]: 8.
[10]: 16.
[11]: Prohibited.

RCCS: Resistor select.
[0]: Resistor 0.
[1]: Resistor 1.

RCEC: Ring oscillator enable condition.
[0]: Enable by RARUN
[1]: Enable by RCEN

RCEN: Ring oscillator enable.
[0]: Disable
[1]: Enable

RARUN: Start RC-ADC operation.
[0]: Disable
[1]: Enable
When OVF_FLG = 1, RARUN will auto clear to 0.

9.2.4 ADCCTL (Address: 084Dh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCCTL	W	--	CKS[1]	CKS[0]	MODE	MVS[2]	MVS[1]	MVS[0]	ADCE
	R	--	--	--	MODE	MVS[2]	MVS[1]	MVS[0]	ADCE
Reset		0	0	0	0	0	0	0	0

ADCE: 12-bit ADC Enable.
[0]: Disable.
[1]: Enable. (auto clear when MODE = 0)

MVS[3:0]: ADC average mode select.
[000]: No moving average
[001]: 2 times average mode.
[010]: 4 times average mode.
[011]: 8 times average mode.
[100]: 16 times average mode.
[101]: 32 times average mode.
[110]: 64 times average mode.
[111]: 128 times average mode.

MODE: ADC mode select.
[0]: Single mode
[1]: Continuous mode.

CKS[1:0]: ADC source clock select.
[00]: 4 MHz.
[01]: 2 MHz.
[10]: 1 MHz.
[11]: 500 KHz.

9.2.4 ADCAVG1 (Address: 084Eh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCAVG1	W	--	--	--	--	--	--	--	--
	R	MVADC[11]	MVADC[10]	MVADC[9]	MVADC[8]	ADC[11]	ADC[10]	ADC[9]	ADC[8]
Reset		0	0	0	0	0	0	0	0

ADC[11:0]: ADC value.
MVADC[11:0]: Moving average ADC value.

9.2.4 ADCAVG2 (Address: 084Fh)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCAVG2	W	--	--	--	--	--	--	--	--
	R	MVADC[7]	MVADC[6]	MVADC[5]	MVADC[4]	MVADC[3]	MVADC[2]	MVADC[1]	MVADC[0]
Reset		0	0	0	0	0	0	0	0

MVADC[11:0]: Moving average ADC value.

9.2.4 ADCAVG3 (Address: 0850h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCAVG3	W	--	--	--	--	--	--	--	--
	R	ADC[7]	ADC[6]	ADC[5]	ADC[4]	ADC[3]	ADC[2]	ADC[1]	ADC[0]
Reset		0	0	0	0	0	0	0	0

ADC[11:0]: ADC value.

10.SOC Architectural Overview

A9108 microcontroller is instruction set compatible with the industry standard 8051. Besides IEEE802.15.4 DSSS RF transceiver, A9108 integrates many features, three 8/16bit counters/timers, watchdog timer, RTC, UART, SPI interface, I^2C interface, 2 channels PWM, 4 channels ADC, battery detector and AES engine, The interrupt controller is extended to support 6 interrupt sources; watchdog timer, RTC, SPI, I^2C, ADC, RF and AES engine. A9108 includes TTAG (2-wire) debug circuitry that provides full time, real-time, in-circuit debugging.

10.1 Pipeline 8051 CPU

A9108 microcontroller has pipelined RSIC architecture 10 times faster compared to standard 8051 architecture. The pipeline 8051 is fully compatible with the MCS-51™ instruction set. User can use standard 8051 assemblers and compilers to develop software. The pipelined architecture 8051 has greatly increases its instruction throughput over the standard 8051 architecture. A9108 has a total of 110 instructions. The table below shows the total number of instructions that require each execution time. For more detail information of instruction, please refer Table 10.1.

Clock to Execute	1	2	3	4	5	6
Number of instructions	24	38	29	11	8	1

10.2 Memory Organization

The memory organization of A9108 is similar to the standard 8051. The memory organization is shown as figure 10.1

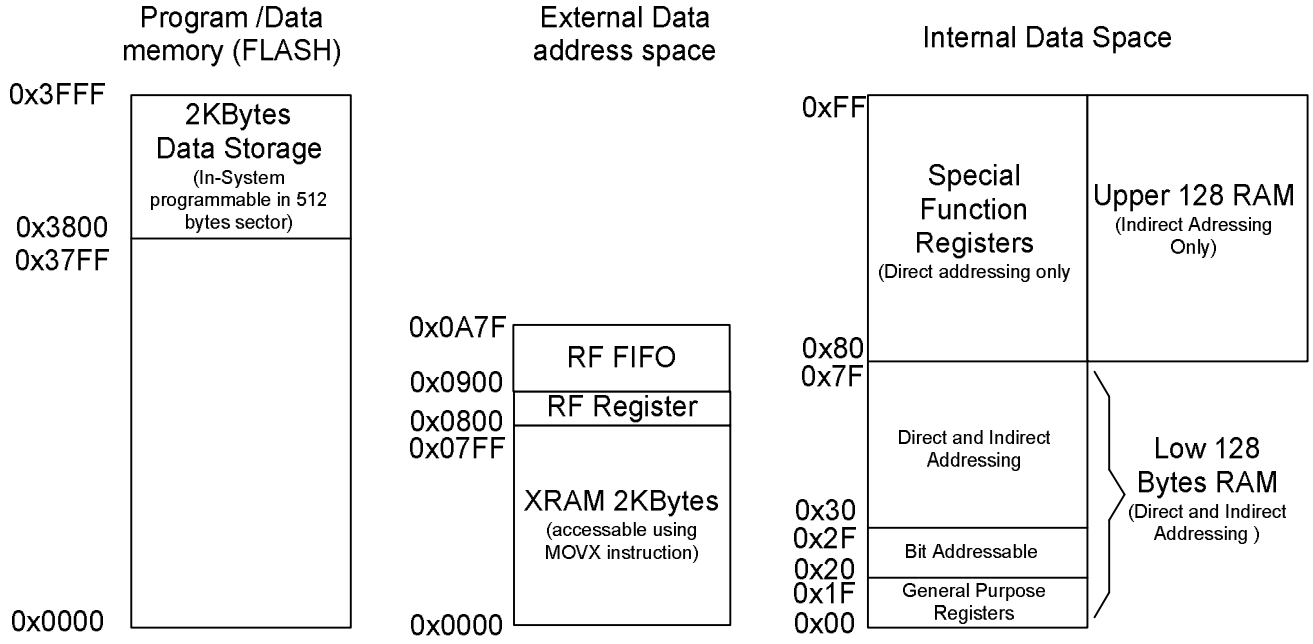


Figure 10.1 Memory Organization

10.2.1 Program memory

The standard 8051 core has 64KB program memory space. A9108 implement 32KB flash in two 16x 8Kb flash macro. The last 2KB program memory space (0x 7800 ~ 0x7FFF) supports IAP (In-Application Programming) function. The each block size in this area is 128Bytes. User has 16 blocks in 2KB program memory space to storage data. Program memory is normally assumed to be read-only. However, A9108 can write to program memory by IAP function call. Please reference xxxxx to write program memory.

10.2.2 Data memory

The A9108 includes 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode. The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFR) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 10.1 illustrates the data memory organization of the A9108.

10.2.3 General Purpose Registers

The lower 32 bytes of data memory, locations 0x00 through 0x1F, may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 9.4). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

10.2.4 Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination). The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte. For example, the instruction:
MOV C, 22.3h ;moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

10.2.5 Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 9.2 lists the SFRs implemented in the CIP-51 System Controller.
The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, SCON0, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the data sheet, as indicated in Table xxx, for a detailed description of each register.

10.2.6 Stack

A9108 has 8-bit stack point called SP (0x81) located in the internal RAM space. It is incremented before data is stored during PUSH and CALL execution and decremented after data is popped during POP, RET and RETI execution. In the other words it always points to the last valid stack byte. The SP is accessed as any other SFRS.

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
81h SP	R/W								
Reset		0	0	0	0	0	1	1	1

Stack pointer register

10.2.7 Data Pointer Register

A9108 are implemented dual data pointer registers, auto increment and auto decrement to speed up data block copying. DPTR0 and DPTR1 are located at four SFR addresses. Active DPTR register is selected by SEL bit (0x86.0). If SEL = 0 the DPTR0 is selected otherwise DPTR1.

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
82h	R/W								

DPL0									
Reset		0	0	0	0	0	0	0	0

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
83h DPH0	R/W								
Reset		0	0	0	0	0	0	0	0

Data Pointer Register DPTR0

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
84h DPL1	R/W								
Reset		0	0	0	0	0	0	0	0

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
85h DPH1	R/W								
Reset		0	0	0	0	0	0	0	0

Data Pointer 1 Register DPTR1

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
86h DPS	R/W	ID1	ID0	TSL	AU1	AU0	-	-	SEL
Reset		0	0	0	0	0	0	0	0

Data Pointers Select Register

ID[1:0] - Increment/decrement function select. See table below.
TSL - Toggle select enable. When set, this bit allows the following DPTR related instruction to toggle the SEL bit following execution of the instruction:

```
MOVC A, @A+DPTR
INC DPTR
MOVX @DPTR, A
MOVX A, @DPTR
MOV DPTR, #data16
```

When TSL=0, DPTR related instructions do not affect state of SEL bit.

AU -When set to '1' performs automatic increment(0)/ decrement(1) of selected DPTR according to IDx bits, after each MOVX @DPTR, MOVC @DPTR instructions
SEL - Select active data pointer – see table below
- - Unimplemented bit. Read as 0 or 1.

ID1	ID0	SEL=1	SEL=0
0	0	INC DPTR1	INC DPTR
0	1	INC DPTR1	DEC DPTR
1	0	DEC DPTR1	INC DPTR
1	1	DEC DPTR1	DEC DPTR

Table DPTR0, DPTR1 operations

Selected data pointer register in used in the following instructions:

```
MOVX @DPTR,A
MOVX A,@DPTR
MOVC A,A+DPTR
JMP @A+DPTR
```

```
INC DPTR
MOV DPTR,#data16
```

10.2.8 RF Registers, RF FIFO and AES FIFO

RF registers are RF radio control registers and located in 0x0800 ~ 0x08ff. Please refer the section 9.2 and the related function setting in the datasheet. A9108 has 384 Bytes FIFO located from 0x0900 to 0x0A7F. There are 128 bytes FIFO from 0x0900 ~ 0x097F for data transmitting. There are 128 bytes FIFO from 0x0980 ~ 0x09FF for data receiving. There are 128 bytes FIFO from 0x0A00 ~ 0x0A7F for AES/CCM* operation.

10.3 Instruction set

A9108 use a high performance, pipeline 8051 core and it is filly compatible with the standard MCS-51™ instruction set. Standard 8051 development tools can used to develop software for A9108. All A9108 instruction sets are the binary and functional equivalent of the MCS-51™. However, instruct timing is different with the standard 8051. All instruction timings are specified in the terms of clock cycles as shown in the table 10.1

Table 10.1 Instruction set sorted by alphabet

Mnemonic	Description	Code	Bytes	Cycles
ACALL addr11	Absolute subroutine call	0x11-0xF1	2	4
ADD A,#data	Add immediate data to accumulator	0x24	2	2
ADD A,@Ri	Add indirect RAM to accumulator	0x26-0x27	1	2
ADD A,direct	Add direct byte to accumulator	0x25	2	2
ADD A,Rn	Add register to accumulator	0x28-0x2F	1	1
ADDC A,#data	Add immediate data to A with carry flag	0x34	2	2
ADDC A,@Ri	Add indirect RAM to A with carry flag	0x36-0x37	1	2
ADDC A,direct	Add direct byte to A with carry flag	0x35	2	2
ADDC A,Rn	Add register to accumulator with carry flag	0x38-0x3F	1	1
AJMP addr11	Absolute jump	0x01-0xE1	2	3
ANL C,/bit	AND complement of direct bit to carry	0xB0	2	2
ANL A,#data	AND immediate data to accumulator	0x54	2	2
ANL A,@Ri	AND indirect RAM to accumulator	0x56-0x57	1	2
ANL A,direct	AND direct byte to accumulator	0x55	2	2
ANL A,Rn	AND register to accumulator	0x58-0x5F	1	1
ANL C,bit	AND direct bit to carry flag	0x82	2	2
ANL direct,#data	AND immediate data to direct byte	0x53	3	3
ANL direct,A	AND accumulator to direct byte	0x52	2	3
CJNE @Ri,#data	Compare immediate to ind. and jump if not equal	0xB6-0xB7	3	5
CJNE A,#datare	Compare immediate to A and jump if not equal	0xB4	3	4
CJNE A,directre	Compare direct byte to A and jump if not equal	0xB5	3	5
CJNE Rn,#datar	Compare immediate to reg. and jump if not equal	0xB8-0xBF	3	4
CLR A	Clear accumulator	0xE4	1	1
CLR bit	Clear direct bit	0xC2	2	3
CLR C	Clear carry flag	0xC3	1	1
CPL A	Complement accumulator	0xF4	1	1

CPL bit	Complement direct bit	0xB2	2	3
CPL C	Complement carry flag	0xB3	1	1
DA A	Decimal adjust accumulator	0xD4	1	3
DEC @Ri	Decrement indirect RAM	0x16-0x17	2	3
DEC A	Decrement accumulator	0x14	1	1
DEC direct	Decrement direct byte	0x15	1	3
DEC Rn	Decrement register	0x18-0x1F	1	2
DIV A,B	Divide A by B	0x84	1	6
DJNZ direct,rel	Decrement direct byte and jump if not zero	0xD5	3	5
DJNZ Rn,rel	Decrement register and jump if not zero	0xD8-0xDF	2	4
INC @Ri	Increment indirect RAM	0x06-0x07	1	3
INC A	Increment accumulator	0x04	1	1
INC direct	Increment directbyte	0x05	2	3
INC Rn	Increment register	0x08-0x0F	1	2
INC DPTR	Increment data pointer	0xA3	1	1
JB bit,rel	Jump if direct bit is set	0x20	3	5
JBC bit,directre	Jump if direct bit is set and clear bit	0x10	3	5
JC rel	Jump if carry flag is set	0x40	2	3
JMP@A+DPTR	Jump indirect relative to the DPTR	0x73	1	5
JNB bit,rel	Jumpifdirectbitisnotset	0x30	3	5
JNC	Jump if carry flag is not set	0x50	2	3
JNZ rel	Jump if accumulator is not zero	0x70	2	4
JZ rel	Jump if accumulator is zero	0x60	2	4
LCALL addr16	Long subroutine call	0x12	3	4
LJMP addr16	Long jump	0x02	3	4
MOV A,@Ri	Move indirect RAM to accumulator	0xE6-0xE7	1	2
MOV bit,C	Move carry flag to direct bit	0x92	2	3
MOV @Ri,#data	Move immediate data to indirect RAM	0x76-0x77	2	2
MOV @Ri,A	Move accumulator to indirect RAM	0xF6-0xF7	1	2
MOV @Ri,direct	Move direct byte to indirect RAM	0xA6-0xA7	2	3
MOV A,#data	Move immediate data to accumulator	0x74	2	2
MOV A,direct	Move direct byte to accumulator	0xE5	2	2
MOV A,Rn	Move register to accumulator	0xE8-0xEF	1	1
MOV C,bit	Move direct bit to carry flag	0xA2	2	2
MOV direct,#data	Move immediate data to direct byte	0x75	3	3
MOV direct,@Ri	Move indirect RAM to direct byte	0x86-0x87	2	3
MOV direct,A	Move accumulator to direct byte	0xF5	2	2
MOV direct,Rn	Move register to direct byte	0x88-0x8F	2	2

MOV direct1,direct2	Move direct byte to direct byte	0x85	3	3
MOV DPTR,#data16	Load 16-bit constant in to active DPTR	0x90	3	3
MOV Rn,#data	Move immediate data to register	0x78-0x7F	2	2
MOV Rn,A	Move accumulator to register	0xF8-0xFF	1	1
MOV Rn,direct	Move direct byte to register	0xA8-0xAF	2	3
MOVC A,@A+DPTR	Move code byte relative to DPTR to accumulator	0x93	1	5
MOVC A,@A+PC	Move code byte relative to PC to accumulator	0x83	1	4
MOVX @DPTR,A	Move A to external SRAM (16-bitaddress)	0xF0	1	1
MOVX @Ri,A	Move A to external RAM (8-bitaddress)	0xF2-0xF3	1	1*
MOVX A,@DPTR	Move external RAM (16-bitaddress) to A	0xE0	1	2*
MOVX A,@Ri	Move external RAM (8-bitaddress) to A	0xE2-0xE3	1	2*
MUL A,B	Multiply A and B	0xA4	1	2
NOP	No operation	0x00	1	1
ORL direct,A	OR accumulator to direct byte	0x42	2	3
ORL A,#data	OR immediate data to accumulator	0x44	2	2
ORL A,@Ri	OR indirect RAM to accumulator	0x46-0x47	1	2
ORL A,direct	OR direct byte to accumulator	0x45	2	2
ORL A,Rn	OR register to accumulator	0x48-0x4F	1	1
ORL C,/bit	OR complement of direct bit to carry	0xA0	2	2
ORL C,bit	OR direct bit to carry flag	0x72	2	2
ORL direct,#data	OR immediate data to direct byte	0x43	3	3
POP direct	Pop direct byte from internal ram stack	0xD0	2	2
PUSH direct	Push direct byte on to internal ram stack	0xC0	2	3
RET	Return from subroutine	0x22	1	4
RETI	Return from interrupt	0x32	1	4
RL A	Rotate accumulator left	0x23	1	1
RLC A	Rotate accumulator left through carry	0x33	1	1
RR A	Rotate accumulator right	0x03	1	1
RRC A	Rotate accumulator right through carry	0x13	1	1
SETB C	Set carry flag	0xD3	1	1
SETB bit	Set direct bit	0xD2	2	3
SJMP rel	Short jump (relative address)	0x80	2	3
SUBB A,@Ri	Subtract indirect RAM from A with borrow	0x96-0x97	1	2
SUBB A,direct	Subtract direct byte from A with borrow	0x95	2	2
SUBB A,#data	Subtract immediate data from A with borrow	0x94	2	2
SUBB A,Rn	Subtract register from A with borrow	0x98-0x9F	1	1
SWAP A	Swap nibbles within the accumulator	0xC4	1	1
XCH A,@Ri	Exchange indirect RAM with accumulator	0xC6-0xC7	1	3

XCH A,direct	Exchange direct byte with accumulator	0xC5	2	3
XCH A,Rn	Exchange register with accumulator	0xC8-0xCF	1	2
XCHD A,@Ri	Exchange low-order nibble indirect RAM with A	0xD6-0xD7	1	3
XRL direct,#data	ExclusiveOR immediate data to direct byte	0x63	3	3
XRL A,#data	ExclusiveOR immediate data to accumulator	0x64	2	2
XRL A,@Ri	ExclusiveOR indirect RAM to accumulator	0x66-0x67	1	2
XRL A,direct	ExclusiveOR direct byte to accumulator	0x65	2	2
XRL A,Rn	ExclusiveOR register to accumulator	0x68-0x6F	1	1
XRL direct,A	ExclusiveOR accumulator to direct byte	0x62	2	3

10.4 Interrupt handler

This section describes 8051 external interrupts and their functionality. For peripheral related interrupts, please refer to an appropriate peripheral section. The external interrupts symbol is shown in figure above. And the pins functionality is described in the following table. All pins are one directional. There are no three-state output pins and internal signals.

Name	ACTIVE	TYPE	DESCRIPTION
int0(P3.2)	low/falling	Input	External interrupt 0 line
int1(P3.3)	low/falling	Input	External interrupt 1 line
int2(P0.7)	low	Input	External interrupt 2 line
int3*(P1.2)	low	Input	External interrupt 3 line
int4*(P1.3)	low	Input	External interrupt 4 line
RF_int	failing		
Key_int	failing		

Table 10.2 External interrupts pins description

Note1：Number of external interrupt sources depends on core configuration. It can be adjusted upon request. The int0 & int1 sources are always available. Please check your configuration.
Note2：*pin functionality depends on compare / capture unit.

10.4.1 FUNCTIONALITY

All 8051 IP cores have implemented two levels interrupt priority control. Each external interrupt can be in high or low level priority group by setting or clearing a bit in the IP(0xB8), EIP(0xF8), and DEVICR(0xCF) registers. External interrupt pins are activated at low level or by a falling edge. Interrupt requests are sampled each system clock at the rising edge of CLK.

Interrupt flag	Function	Active level/edge	Flag resets	Vector[1]	Natural priority
IE0	Device pin INT0	Low/falling	Hardware	0x03	1
TF0	Internal, Timer 0	-	Hardware	0x0B	2
IE1	Device pin INT1	Low/falling	Hardware	0x13	3
TF1	Internal, Timer 1	-	Hardware	0x1B	4
TI0 & RI0	Interrupt, UART0	-	Software	0x23	5
TF2	Interrupt, Timer 2	-	Software	0x2B	6
TI1 & RI1	Interrupt, UART1	-	Software	0x33	7
INT2F	Device pin INT2	Low	Hardware	0x3B	8
INT3F	Device pin INT3	Low	Hardware	0x43	9
INT4F	Device pin INT4	Low	Hardware	0x4B	10
RFINT	Interrupt, RF	-	Software	0x53	11
KEYINT	Interrupt, Key	-	Software	0x5B	12
WDIF	Internal, Watchdog	-	Software	0x63	13
I2CMIF	Internal, I2C MASTER MODULE	-	Software	0x6B	14
I2CSIF SPIIF	Internal, DI2CS/ Internal, SPI	-	Software	0x73	15
MACIF	Internal DMAC	-	Hardware	0x7B	16
USBIF	Internal USB2	-	Hardware	0x83	17

Table10.3　8051 interrupts summary

1- This is a default location when IRQ_INTERVAL = 8, in other case is equal to (IRQ_INTERVAL* n) + 3, when n = (natural Priority - 1)

Each interrupt vector can be individually enabled or disabled by setting or clearing a corresponding bit in the IE(0xA8), EIE(0xE8), DEVICR(0xCF). The IE contains global interrupt system disable(0) / enable(1) bit called EA.

IE register (0xA8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A8h IE	R/W	EA	-	ET2	ES0	ET1	EX1	ET0	EX0
Reset		0	0	0	0	0	0	0	0

EA：Enable global interrupts
EX0：Enable INT0 interrupts
ET0：Enable Timer 0 interrupts
EX1：Enable INT1 interrupts
ET1：Enable Timer 1 interrupts
ES0：Enable UART0 interrupts
ET2：Enable Timer 2 interrupts

All of bits that generate interrupts can be set or cleared by software, with the same result as if they had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled by software. The exceptions of this rule are the request flags IE0 and IE1. If the external interrupts 0 or 1 are programmed to be level activated, IE0 and IE1 are controlled by the external source via pin INT0 and INT1, respectively. Thus, writing a one to these bits will not set the request flag IE0 and/or IE1. The same exception is related to INT2F, INT3F, INT4F, INT5F, and INT6F – external interrupts number 2, 3, 4, 5, 6.

IP register (0xB8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B8h IP	R/W	-	-	PT2	PS0	PT1	PX1	PT0	PX0
Reset		0	0	0	0	0	0	0	0

PX0：INT0 priority level control (at 1-high-level)
PT0：Timer 0 priority level control (at 1-high-level)
PX1：INT1 priority level control (at 1-high-level)
PT1：Timer 1 priority level control (at 1-high-level)
PS0：UART0 priority level control (at 1-high-level)
PT2：Timer 2 priority level control (at 1-high-level)

TCON register (0x88)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
88h TCON	R/W	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Reset		0	0	0	0	0	0	0	0

IT0：INT0 level (at 0) / edge (at 1) sensitivity
IT1：INT1 level (at 0) / edge (at 1) sensitivity
IE0：INT0 interrupt flag
　Cleared by hardware when processor branches to interrupt routine
IE1：INT1 interrupt flag
　Cleared by hardware when processor branches to interrupt routine
TF0：Timer 0 interrupt (overflow) flag

Cleared by hardware when processor branches to interrupt routine

TF1：Timer 1 interrupt (overflow) flag

Cleared by hardware when processor branches to interrupt routine

SCON0 register (0x98)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
98h SCON0	R/W	SM00	SM01	SM02	REN0	TB08	RB08	TI0	RI0
Reset		0	0	0	0	0	0	0	0

RI0：UART0 receiver interrupt flag

TI0：UART0 transmitter interrupt flag

SCON1 register (0xC0)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C0h SCON1	R/W	SM10	SM11	SM12	REN1	TB18	RB18	TI1	RI1
Reset		0	0	0	0	0	0	0	0

RI1：UART1 receiver interrupt flag

TI1：UART1 transmitter interrupt flag

EIE register (0xE8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
E8h EIE	R/W	EI2CS ESPI	EI2CM	EWDI	EKEYINT	ERFINT	EINT4	EINT3	EINT2
Reset		0	0	0	0	0	0	0	0

EINT2：Enable INT2 interrupts

EINT3：Enable INT3

EINT4：Enable INT4

ERFINT：Enable RF INT

EKEYINT：Enable KEY INT

EWDI：Enable Watchdog interrupts

EI2CM：Enable DI2CM interrupts

EI2CS：Enable DI2CS interrupts

ESPI：Enable DSPI interrupts

EIP register (0xF8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F8h EIP	R/W	PI2CS PSPI	PI2CM	PWDI	PKEYINT	PRFINT	PINT4	PINT3	PINT2
Reset		0	0	0	0	0	0	0	0

PINT2：INT2 priority level control (at 1-high-level)

PINT3：INT3/Compare 0 priority level control (at 1-high-level)

PINT4：INT4/Compare 1 priority level control (at 1-high-level)

PRFINT：RFINT priority level control (at 1-high-level)

PKEYINT：KEYINT priority level control (at 1-high-level)

PWDI：Watchdog priority level control (at 1-high-level)

PI2CM：DI2CM priority level control (at 1-high-level)

PI2CS：DI2CS priority level control (at 1-high-level)

PSPI：DSPI priority level control (at 1-high-level)

EIF register (0x91)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

91h EIF	R/W	I2CSF SPIF	I2CMF	-	INT6F	INT5F	INT4F	INT3F	INT2F
Reset		0	0	0	0	0	0	0	0

INT2F：INT2 interrupt flag

Should be cleared by external hardware when processor branches to interrupt routine. This bit is a copy of INT2 pin updated every CLK period. It cannot be set by software.

INT3F*：INT3/Compare 0 interrupt flag

Should be cleared by external hardware when processor branches to interrupt routine. This bit is a copy of INT3 pin updated every CLK period, else must be cleared by software writing 0x02 when Compare 0 enabled CCEN[1:0]=10 .It cannot be set by software.

INT4F*：INT4/Compare 1 interrupt flag

Should be cleared by external hardware when processor branches to interrupt routine. This bit is a copy of INT4 pin updated every CLK period, else must be cleared by software writing 0x04 when Compare 0 enabled CCEN[3:2]=10

It cannot be set by software.

RFINT：RFINT interrupt flag

Must be cleared by software writing 0x08 when controlled by INT5 pin, else must be cleared by software writing 0x08 when Compare2 is enabled CCEN[5:4]=10. It cannot be set by software.

KEYINT：KEYINT interrupt flag

Must be cleared by software writing 0x10 when controlled by INT6 pin, else must be cleared by software writing 0x10 when Compare3 is enabled CCEN[7:6]=10. It cannot be set by software.

I2CMIF：DI2CM interrupt flag. Must be cleared by software writing 0x40. It cannot be set by software

I2CSIF：DI2CS interrupt flag

SPIIF：DSPI interrupt flag

Software should determine the source of interrupt by checking both modules' interrupt related bits. Must be cleared by software writing 0x80. It cannot be set by software.

Note1: * flag can be set by Compare/Capture unit channel when enabled in CCEN register

Note2: A peripheral related bit is available if this peripheral device is included in the system. Can be modified upon request. Please check your configuration.

SPIIF：DSPI interrupt flag

Software should determine the source of interrupt by checking both modules' interrupt related bits. Must be cleared by software writing 0x80. It cannot be set by software.

Table 10.4 Power manager

	CPU speed	16MHz	Internal RC	RTC	RAM	Back to Normal	LVR	RF
Normal	16MHz	V	V	V	v	X	V	ALL
PMM	8/4/2/1 MHz IRC/RTC	V	V	V	v	Interrupt / mode switch	V	ALL
Idle (PM1)	X	X	V	V	v	H/W reset / wakeup key / Interrupt Key / Sleep timer	V	WOR/Sleep
Sleep (PM2)	X	X	X	X	v	H/W reset / wakeup key / Interrupt KEY	V*	Sleep
Deep Sleep (PM3)	X	X	X	X	?	Reset Key Reset	X	Sleep

Key interrupt

1. P0 / P1 ==> 1 個 wakeup bit, control 2 個 pin.
2. P3 ==> 1 個 wakeup bit, control 1 個 pin.

10.5 Reset Circuit

Reset Flag

POR Flag

LVS

LVR

TX flag
RX flag

RSFLAG (0xBA):

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BAh RSFLAG	R	-	-	-	-	-	BODF	RESETNF	PORF
Reset		0	0	0	0	0	0	0	0

PORF (power-on reset flag)
= 1: Occurred Power-on Reset
= 0: No Power-on Reset
RESETNF (resetn flag)
= 1: Occurred ResetN reset
= 0: No ResetN resetno resetn reset
BOD (Low voltage detect) flag
= 1: Occurred Low Voltage Reset
= 0: No Low Voltage reset

11. I/O Ports

A9108 has 24 Digital I/O Pins. There are separated to 3 Ports and each of the Port pin can be defined as general-purpose I/O (GPIO) or peripheral I/O signals connected to the timers, UART, I2C and SPI functions. Thus, each pin can also be used to wake A9108 up from sleep mode. User can select each pin function by setting register. Each port has itself port register like P0 (0x80), P1 (0x90) and P3 (0xB0) that are both byte addressable and bit addressable. When reading, the logic levels of the Port's input pins are returned. Each port has three registers to setting Pull-up (PU), Output-enable (OE) and Wake-up enable (WUE). As shown the bellow block diagram, Fig. 11.1. Unused I/O pins should have a defined level and not be left floating. One way to do this is to leave the pin unconnected and configure the pin as a general-purpose I/O input with pull-up resistor.

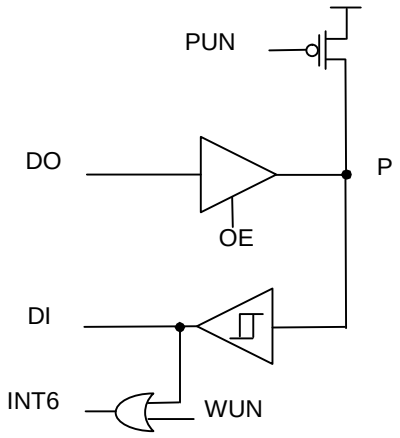


Figure11.1 Ports I/O block diagram

Table 11.1 OE and PUN setting and Output(P) and Input(DI)

OE	PUN	P	DI
0	0	1	1
0	1	HZ	INH
1	X	DO	DO

Table 11.2 WUN setting and INT6 source

WUN	INT6
0	DI
1	1

11.2 FUNCTIONALITY

It has three 8-bit full bi-directional ports, P0, P1 and P3. Each port bit can be individually accessed by bit addressable instructions.

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
80h P0	R/W								
Reset		0	0	0	0	0	0	0	0

Port 0 register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
90h P1	R/W								
Reset		0	0	0	0	0	0	0	0

Port 1 register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B0h P3	R/W								
Reset		0	0	0	0	0	0	0	0

Port 3 register

Read and write accesses to the I/O port are performed via their corresponding SFRs P0(0x80), P1(0x90), and P3(0xB0). Some port-reading instructions read the data register and others read the port's pin. The "Read-Modify-Write" instructions are directed to the data registers and are shown below. All the other instructions used to read a port exclusively read the port's pin.

Instruction	Function description
ANL	Logic AND
ORL	Logic OR
XRL	Logic eXclusive OR
JBC	Jump if bit is set and clear
CPL	Complement bit
INC, DEC	Increment, decrement byte
DJNZ	Decrement and jump if not zero
MOV Px.y, C	Move carry bit to y of port x
CLR Px.y	Clear bit y of port x
SETB Px.y	Set bit y of port x

Table11.2 Read-modify-write instructions

According the Table 11.1, all Port pins can be configured as Output, Input with pull-up resistor(around 100 Kohm) or Input. Please refer the following truth table to know every function setting. When OE=1, this pin is configured as Output. Otherwise OE =0, this pin is configured as Input. User can set PU =1 or 0 depending on application. When OE =1, PU=0 is recommended for saving power..

OE	PU	P	DI
1	X	DO	DO
0	1	Pull-up	P
0	0	HZ	Input

All Port pins can wake A9108 up when WUEN=1 and configured GPIO. All Port pins' WEU signals connect one AND gate to INT2. It means pin wake up function needs INT2 ISR to take care this interrupt.

WUEN	WUNDI
1	1
0	DI

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B2h P0PU	R/W								
Reset		0	0	0	0	0	0	0	0

Port 0 Pull Up Register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B3h P0OE	R/W								
Reset		0	0	0	0	0	0	0	0

Port 0 Output Enable Register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B4h P0WUE	R/W								
Reset		0	0	0	0	0	0	0	0

Port 0 Wake Up Enable Register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B5h P1PU	R/W								
Reset		0	0	0	0	0	0	0	0

Port 1 Pull Up Register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B6h P1OE	R/W								
Reset		0	0	0	0	0	0	0	0

Port 1 Output Enable Register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B7h P1WUE	R/W								
Reset		0	0	0	0	0	0	0	0

Port 1 Wake Up Enable Register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AAh P3PU	R/W								
Reset		0	0	0	0	0	0	0	0

Port 3Pull Up Register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ABh P3OE	R/W								
Reset		0	0	0	0	0	0	0	0

Port 3 Output Enable Register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ACh P3WUE	R/W								
Reset		0	0	0	0	0	0	0	0

Port 3 Wake Up Enable Register

IOSEL Register (0xBB)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BBh IOSEL	R/W	-	-	ADCIOS	RTCIOS	BBIOS	-	I2CIOS	URT0IOS
Reset		0	0	0	0	0	0	0	0

URT0IOS (UART0 I/O select)
= 1: The pad is selected for UART0 mode0 (open drain I/O)
= 0: The pad is normal I/O
I2CIOS (I2C I/O select)
= 1: The pad is selected for I2C (open drain I/O)
= 0: The pad is normal I/O
BBIOS (Base band I/O select)
= 1: Output
= 0: Input
RTCIOS (Real-time clock I/O select)
= 1: The pad is for RTC clock
= 0: The pad is normal I/O
ADCIOS (ADC I/O select)
= 1: The pad is for ADC analog input

= 0: The pad is normal I/O

12 Timer 0 & 1 &2

A9108 contains three 16-bit timer/counters, Timer 0, Timer 1 and Timer 2. Timer 0 and Timer 1 in the “timer mode“, timer registers are incremented every 4/12/CLK periods depends on CKCON (0x8E) setting, when appropriate timer is enabled. In the “counter mode” the timer registers are incremented every falling transition on theirs corresponding input pins: T0 or T1. The input pins are sampled every CLK period.

The Timer 2 is one of the most powerful peripheral units of the core. It can be used for all kinds of digital signal generation and event capturing like pulse generation, pulse width modulation, pulse width measuring etc.

12.1 Timer 0 & 1 PINS DESCRIPTION

The pins functionality is described in the following table. All pins are one directional.

PIN	ACTIVE	TYPE	DESCRIPTION
T0(P3.4)	Falling	Input	Timer 0 clock line
GATE0(P3.2)	High	Input	Timer 0 clock line gate control
T1(P3.5)	Falling	Input	Tiner 1 clock line
GATE1(P3.3)	High	Input	Timer 1clock line gate control

Table12.1 Timer 0, 1 pins description

12.2 Timer 0 & 1 FUNCTIONALITY

12.2.1 OVERVIEW

Timer 0 and Timer 1 are fully compatible with the standard 8051 timers. Each timer consists of two 8-bit registers TH0 (0x8C), TL0 (0x8A), TH1 (0x8D), TL1 (0x8B). Timers 0, 1 work in the same four modes. The modes are described below.

M1	M0	Mode	Function description
0	0	0	THx operates as 8-bit timer/counter with a divide by 32 prescaler served by lower 5-bit of TLx.
0	1	1	16-bit timer/counter. THx and TLx are cascaded.
1	0	2	TLx operates as 8-bit timer/counter with 8-bit auto-reload by THx.
1	1	3	TL0 is configured as 8-bit timer/counter controlled by the standard Timer 0 bits. TH0 is an 8-bit timer controlled by the Timer 1 controls bits. Timer 1 holds its count.

Table12.2 Timer 0 and 1 modes

12.2.2 Timer 0 & 1 Registers

TMOD register (0x89)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
89h TMOD	R/W	GATE1	CT	M1	M0	GATE0	CT	M1	M0
		Timer 1 control bits				Timer 0 control bits			
Reset		0	0	0	0	0	0	0	0

GATE：Gating control

=1, Timer x enabled while GATEx pin is high and TRx control bit is set.

=0, Timer x enabled while TRx control bit is set.

CT：Counter or timer select bit

=1, Counter mode, Timer x clock from Tx pin.

=0, Timer mode, internally clocked.

M[1：0]：Mode select bits

TCON register (0x88)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
88h TCON	R/W	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Reset		0	0	0	0	0	0	0	0

TR0：Timer 0 run control bit

=1, enabled.
=0, disabled.

TR1：Timer 1 run control bit
=1, enabled.
=0, disabled.

TF0：Timer 0 interrupt (overflow) flag.
Cleared by hardware when processor branches to interrupt routine.

TF1：Timer 1 interrupt (overflow) flag.
Cleared by hardware when processor branches to interrupt routine.

CKCON register (0x8E)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8Eh CKCON	R/W	-	-	-	T1M	T0M	MD2	MD1	MD0
Reset		0	0	0	0	0	0	0	0

T0M：This bit controls the division of the system clock that drives Timer 0.
=1, Timer 0 uses a divided-by-4 of the system clock frequency.
=0, Timer 0 uses a divided-by-12 of the system clock frequency.

T1M：This bit controls the division of the system clock that drives Timer 1.
=1, Timer 1 uses a divided-by-4 of the system clock frequency.
=0, Timer 1 uses a divided-by-12 of the system clock frequency.

IE register (0xA8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A8h IE	R/W	EA	-	ET2	ES0	ET1	EX1	ET0	EX0
Reset		0	0	0	0	0	0	0	0

EA：Enable global interrupts.
ET0：Enable Timer 0 interrupts.
ET1：Enable Timer 1 interrupts.

IP register (0xB8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B8h IP	R/W	-	-	PT2	PS0	PT1	PX1	PT0	PX0
Reset		0	0	0	0	0	0	0	0

PT0：Timer 0 priority level control (at 1-high level)
PT1：Timer 1 priority level control (at 1-high level)

Timer 0, 1 related bits that generate interrupts can be set or cleared by software, with the same result as if they had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled by software.

Interrupt flag	Function	Active level/edge	Flag resets	Vector	Natural priority
TF0	Internal, Timer 0	-	Hardware	0x0B	2
TF1	Internal, Timer 1	-	Hardware	0x1B	4

Table12.3　Timer 0, 1 interrupts

12.2.3 Timer 0 – Mode 0

In this mode, the Timer 0 register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s. Timer 0 interrupt flag TF0 is set. The counted input is enabled to the Timer 0 when TCON.4 = 1 and either TMOD.3 = 1 or GATE0 = 1. (Setting TMOD.3 = 1 allows the Timer 0 to be controlled by external input GATE0, to facilitate pulse width measurement). The 13-bit register consists of all 8-bit of TH0 and lower 5 bits of TL0.The upper 3 bits of TL0 are indeterminate and should be ignored.

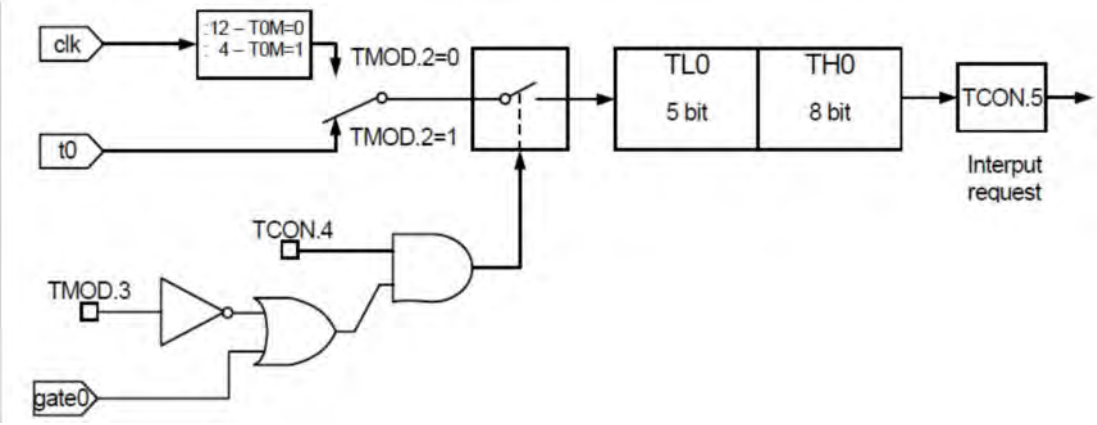


Figure12.1　Timer/Counter 0, Mode 0：13-Bit Timer/Counter

12.2.4 Timer 0 – Mode 1

Mode 1 is the same as Mode 0, except that the timer register is running with all 16 bits. Mode 1 is shown in figure below.

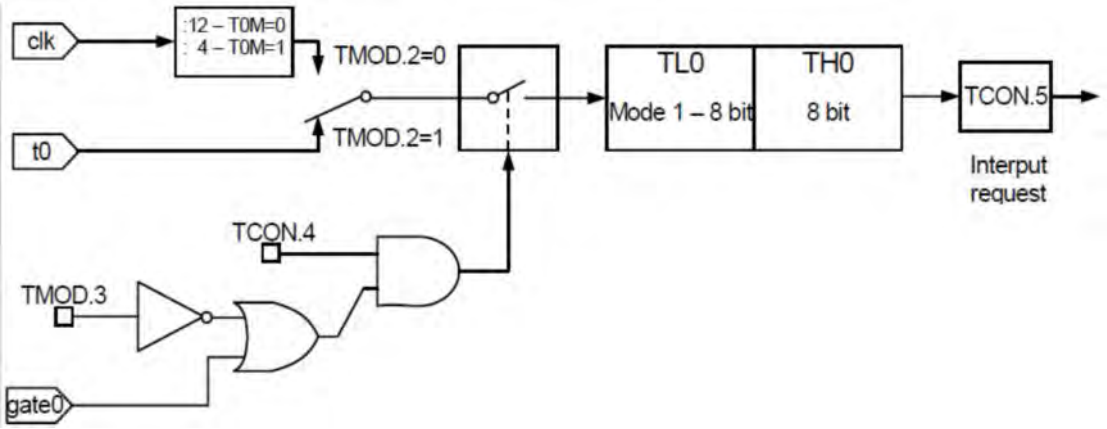


Figure12.2　Timer/Counter 0, Mode 1：16-Bit Timer/Counter

12.2.5 Timer 0 – Mode 2

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reloads, as shown in figure below. Overflow from TL0 not only sets TF0, but also reloads TL0 with the contents of TH0, which is loaded by software. The reload leaves TH0 unchanged.

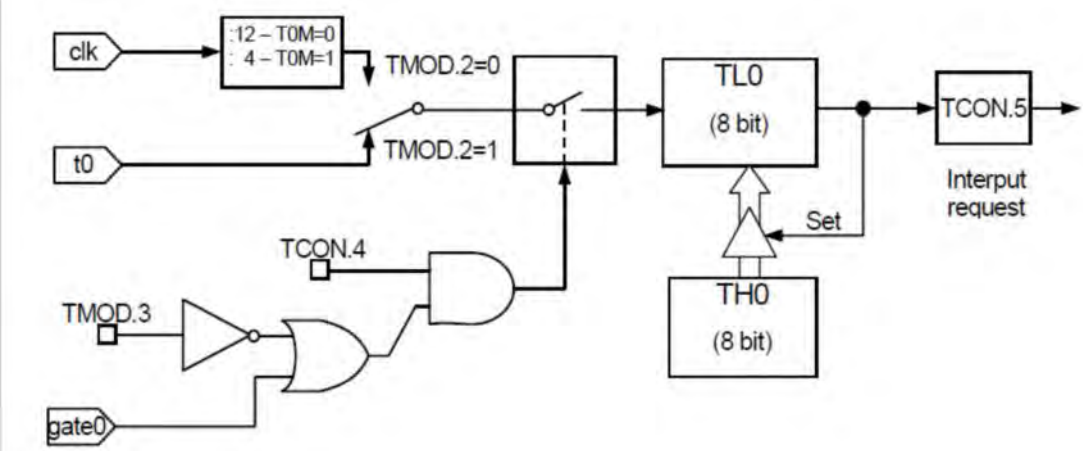


Figure12.3　Timer/Counter 0, Mode 2：8-Bit Timer/Counter with Auto-Reload

12.2.6 Timer 0 – Mode 3

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in figure below. TL0 uses the Timer 0 control bits：C/T, GATE, TR0, GATE0 and TF0. TH0 is locked into a timer function and use the TR1 and TF1 flag from Timer1 and controls Timer1 interrupt. Mode 3 is provided for applications requiring an extra 8-bit timer/counter. When Timer 0 is in Mode 3, Timer 1 can be turned off by switching it into its own Mode 3, or can still be used by the serial channel as a baud rate generator, or in any application where interrupt from Timer 1 is not required.

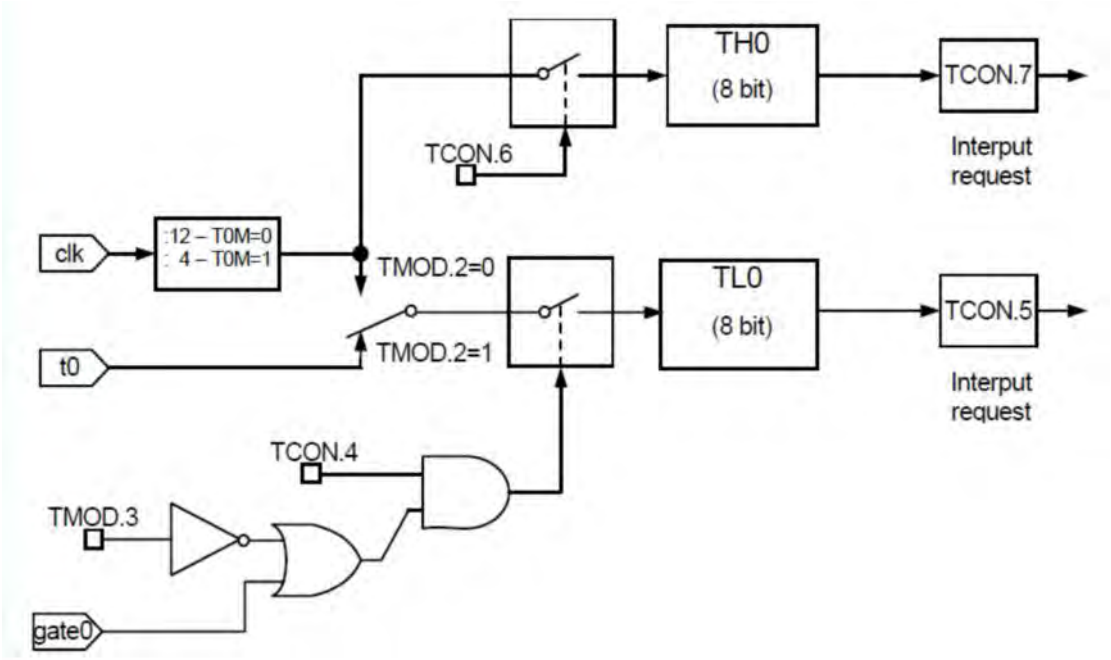


Figure12.4　Timer/Counter 0, Mode 3：Two 8-Bit Timers/Counters

12.2.7 Timer 1 – Mode 0

In this Mode, the Timer1 register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, Timer1 interrupt flag TF1 is set. The counted input is enabled to the Timer1 when TCON.6 = 1 and either TMOD.6 = 0 or GATE1 = 1. (Setting TMOD.7 = 1 allows the Timer1 to be controlled by external input GATE1, to facilitate pulse width measurements). The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored.

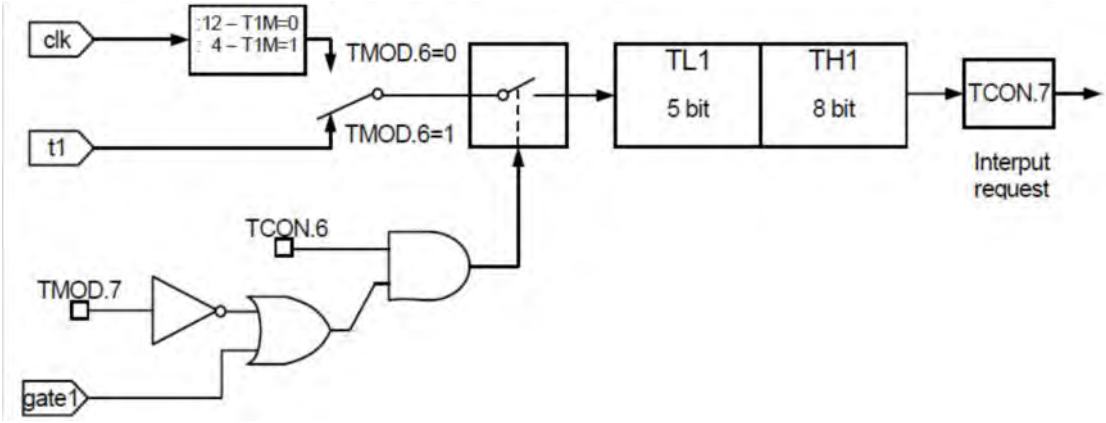


Figure12.5 Timer/Counter 1, Mode 0：13-Bit Timers/Counters

12.2.8 Timer 1 – Mode 1

Mode 1 is the same as Mode 0, except that timer register is running with all 16 bits. Mode 1 is shown in figure below.

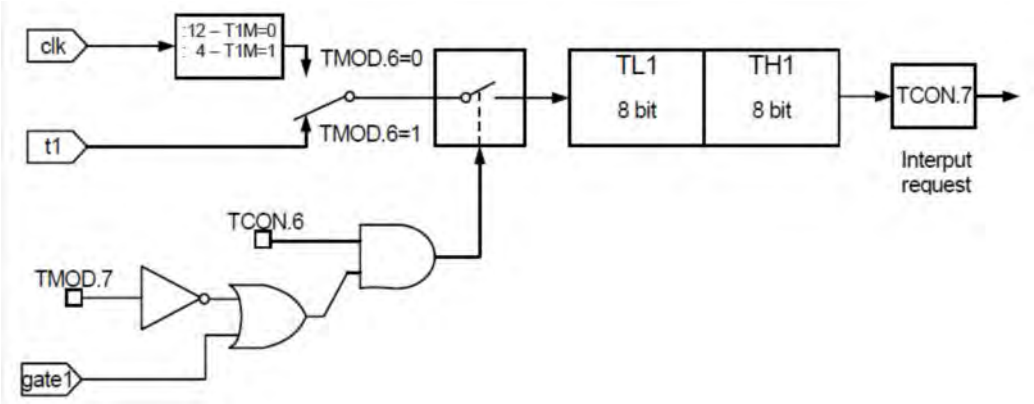


Figure12.6　Timer/Counter 1, Mode 0：16-Bit Timers/Counter

12.2.9 Timer 1 – Mode 2

Mode 2 configures the timer register as an 8-bit counter (TL1) with automatic reloads, as shown in figure below. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is loaded by software. The reload leaves TH1 unchanged.

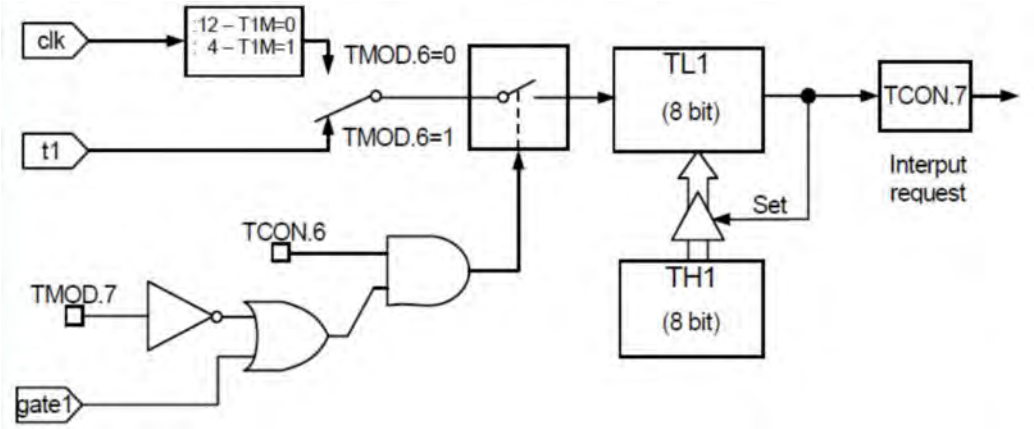


Figure12.7　Timer/Counter 1, Mode 2：8-Bit Timer/Counter with Auto-Reload

12.2.10 Timer 1 – Mode 3

Timer 1 in Mode 3 is held counting. The effect is the same as setting TR1=0.

12.3 Timer2 PINS DESCRIPTION

The Timer 2 pins functionality is described in the following table. All pins are one directional.

PIN	ACTIVE	TYPE	DESCRIPTION
t2(P1.0)	falling	INPUT	Timer 2 clock line
t2ex(P1.1)	high	INPUT	Timer 2 control

Table12.4 Compare/Capture pins description

12.4 Timer2 FUNCTIONALITY

12.4.1 OVERVIEW

Timer 2 is fully compatible with the standard 8052 Timer 2. It is up counter. Totally five SFRs control the Timer 2 operation: TH2/TL2(0xCD/0xCC) counter registers, RLDH/RLDL (0xCB/0xCA) capture registers and T2CON(0xC8) control register. Timer 2 works in the three modes selected by T2CON bits as shown in table below.

RCLK, TCLK	CPRL2	TR2	Function description
0	0	1	16-bit auto-reload mode. The Timer 2 overflow sets TF2 bit and the TH2,TL2 registers reloaded 16-bit value from RLDH, RLDL.
0	1	1	16-bit capture mode. The Timer 2 overflow sets TF2 bit. When the EXEN2 = 1, the TH2, TL2 register values are stored into RLDH, RLDL while falling edge is detected on T2EX pin.
1	X	1	Baud rate generator for the UART0 interface. It auto-reloads its counter with RLDH, RLDL values each overflows.
X	X	0	Timer 2 is off

Table12.5 Timer 2 modes

12.4.2 Timer 2 Registers

T2CON register (0xC8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C8h	R/W	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	CT2	CPRL2
APOL									
Reset		0	0	0	0	0	0	0	0

EXF2：Falling edge indicator on T2EX pin when EXEN = 1. Must be cleared by software.

RCLK：Receive clock enable

=1, UART0 receiver is clocked by Timer 2 overflow pulses

=0, UART0 receiver is clocked by Timer 2 overflow pulses

TCLK：Transmit clock enable

=1, UART0 transmitter is clocked by Timer 2 overflow pulses

=0, UART0 transmitter is clocked by Timer 2 overflow pulses

EXEN2：Enable T2EX pin functionality.

=1, Allows capture or reload as a result of T2EX pin falling edge.

=0, ignore T2EX events

TR2：Start / Stop Timer 2

=1, start

=0, stop

CT2：Timer / counter select

=1, external event counter. Clock source is T2 pin.

=0, timer 2 Internally clocked

CPRL2：Capture / Reload select

=1, T2EX pin falling edge causes capture to occur when EXEN2 = 1

=0, automatic reload occurs：on Timer 2 overflow or falling edge T2EX pin when EXEN2 = 1. When RCLK or TCLK is set this bit is ignored and automatic reload on Timer 2 overflow is forced.

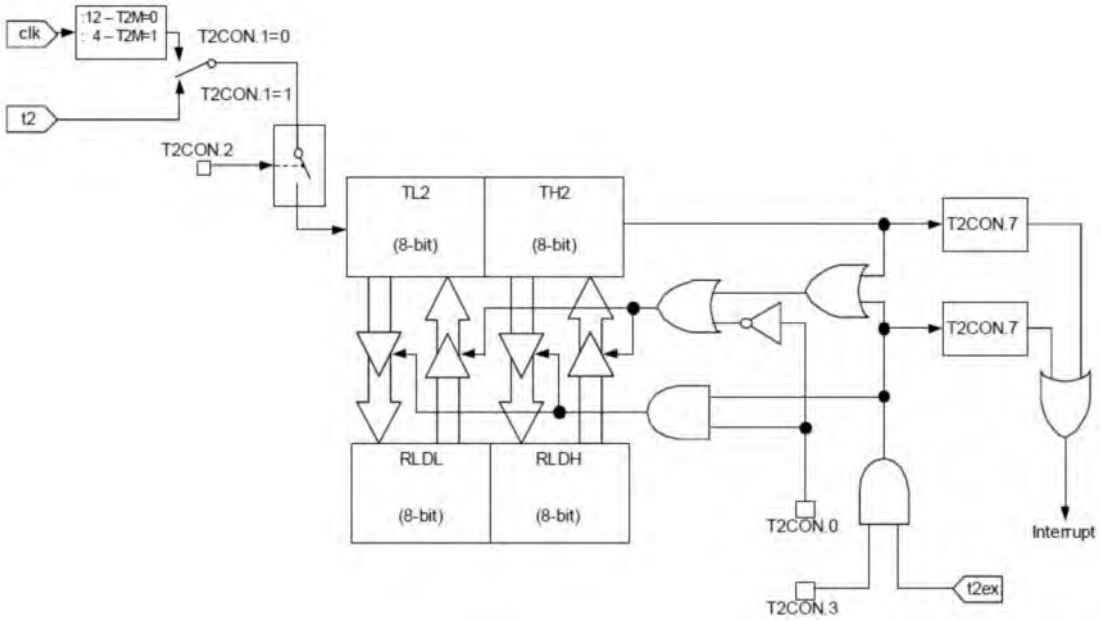


Figure12.9　Timer 2 block diagram in timer mode

CKCON register　(0x8E)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8Eh CKCON	R/W	-	-	-	T1M	T0M	MD2	MD1	MD0
Reset		0	0	0	0	0	0	0	0

T2M：This bit controls the division of the system clock that drives Timer 2. This bit has no effect when the timer is in baud rate generator mode.

=1, Timer 2 uses a divide-by-4 of the system clock frequency.

=0, Timer 2 uses a divide-by-12 of the system clock frequency.

Timer 2 interrupt related bits are shown below. An interrupt can be turned on/off by IE (0xA8) register, and set into high/low priority group by IP register.

IE register　(0xA8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A8h IE	R/W	EA	-	ET2	ES0	ET1	EX1	ET0	EX0
Reset		0	0	0	0	0	0	0	0

EA：Enable global interrupts.

ET2：Enable Timer 2 interrupts.

IP register　(0xB8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B8h IP	R/W	-	-	PT2	PS0	PT1	PX1	PT0	PX0
Reset		0	0	0	0	0	0	0	0

PT2：Timer 2 priority level control (at 1-high level)

-：Unimplemented bit. Read as 0 or 1.

T2CON register　(0xC8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
C8h T2CON Reset	R/W	TF2 0	EXF2 0	RCLK 0	TCLK 0	EXEN2 0	TR2 0	CT2 0	CPRL2 0

TF2：Timer 2 interrupt (overflow) flag. Must be cleared by software.
The flag will not be set when either RCLK or TCLK is set.

All Timer 2 related bits generate interrupts can be set or cleared by software, with the same result as if they had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled by software.

Interrupt flag	Function	Active level / edge	Flag resets	Vector	Natural priority
TF2	Interrnal, Timer2	-	Software	0x2B	6

Table12.6　Timer 2 interrupt

Interrupt is also generated at falling edge of T2EX pin, while EXEN2 bit is set. This interrupt doesn't set TF2 flag, but EXF2 only and also uses 0x2B vector. Please see picture below. Timer2 internal logic configured as baud-rate generator is shwon below.

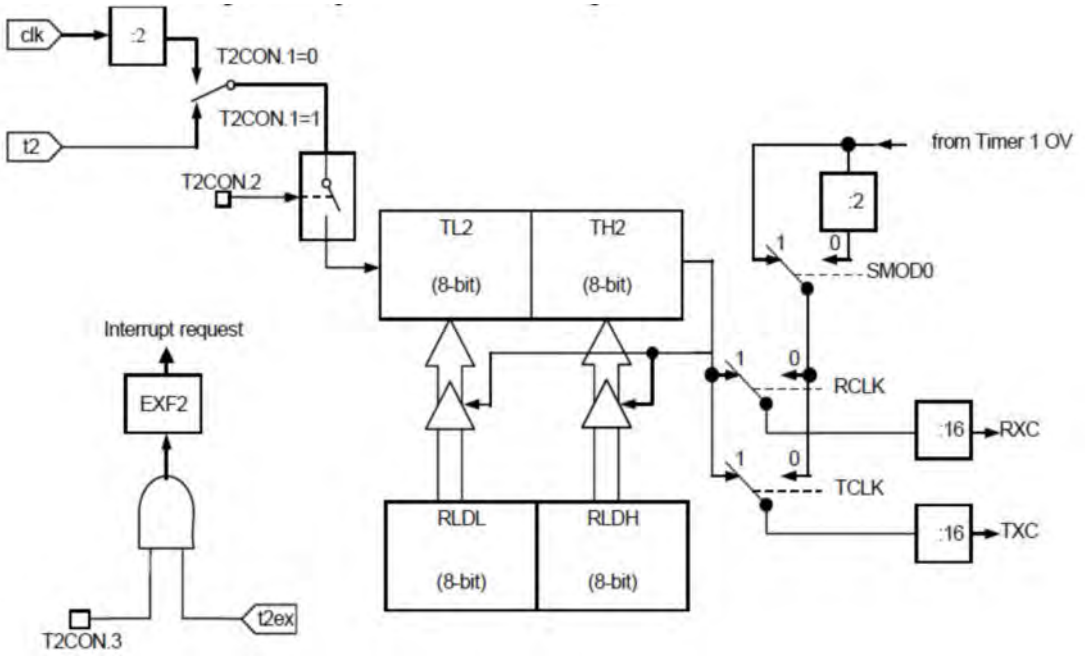


Figure12.9　Timer 2 block diagram as UART0 baud rate generator

Please note that SMOD0 bit is ignored by UART when clocked by Timer2. The RLCK/TCLK frequency is equal to：

$$xCLK = \frac{CLK}{2 \cdot (65536 - RLD)}$$

$$where\ xCLK = TCLK, RCLK$$

13. UART 0,1

UART0 is full duplex, meaning it can transmit and receive concurrently. It is receive double-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register. Writing to SBUF0 loads the transmit register, and reading SBUF0 reads a physically separate receive register. The serial port can operate in 4 modes: one synchronous and three asynchronous modes. Mode 2 and 3 has a special feature for multiprocessor communications. This feature is enabled by setting SM02 bit in SCON0 register. The master processor first sends out an address byte, which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM02 = 1, no slave will be interrupted by a data byte. An address byte will interrupt all slaves. The addressed slave will clear its SM02 bit and prepare to receive the data bytes that will be coming. The slaves that were not being addressed leave their SM02 set and ignoring the incoming data.

UART1 is also full duplex, meaning it can transmit and receive concurrently. It is receive double-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register. Writing to SBUF1 loads the transmit register, and reading SBUF1 reads a physically separate receive register. The serial port can operate in 4 modes: one synchronous and three asynchronous modes. Mode 2 and 3 has a special feature for multiprocessor communications. This feature is enabled by setting SM12 bit in SCON1 register. The master processor first sends out an address byte, which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM12 = 1, no slave will be interrupted by a data byte. An address byte will interrupt all slaves. The addressed slave will clear its SM12 bit and prepare to receive the data bytes that will be coming. The slaves that were not being addressed leave their SM12 set and ignoring the incoming data.

13.1 UART0/1 PINS DESCRIPTION

The UART0 pins functionality is described in the following table. All pins are one directional. There are no three-state output pins and internal signals.

PIN	ACTIVE	TYPE	DESCRIPTION
Rxd_0(P3.0)	-	Input / Output	Serial receiver I_0 / O_0
Txd_0(P3.1)	-	Output	Serial transmitter line 0

Table13.1 UART0 pins description

The UART1 pins functionality is described in the following table. All pins are one directional. There are no three-state output pins and internal signals.

PIN	ACTIVE	TYPE	DESCRIPTION
Rxd_1(P1.2)	-	Input / Output	Serial receiver I_1 / O_1
Txd_1(P1.3)	-	Output	Serial transmitter line 1

Table13.2 UART1 pins description

13.2 FUNCTIONALITY

The UART0 has the same functionality as a standard 8051 UART. The UART0 related registers are: SBUF0(0x99), SCON0(0x98), PCON(0x87), IE(0xA8) and IP(0xB8). The UART0 data buffer (SBUF0) consists of two separate registers: transmit and receive registers. A data writes into the SBUF0 sets this data in UART0 output register and starts a transmission. A data reads from SBUF0, reads data from the UART0 receive register.

SBUF0 register (0x99)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
99h SBUF0	R/W								
Reset		0	0	0	0	0	0	0	0

SB0[7:0]：UART0 buffer

SCON0 register (0x98)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
98h SCON0	R/W	SM00	SM01	SM02	REN0	TB08	RB08	TI0	RI0
Reset		0	0	0	0	0	0	0	0

SM02：Enable a multiprocessor communication feature

SM0[1:0]：Sets baud rate

SM00	SM01	Mode	Description	Baud Rate
0	0	0	Shift register	$F_{CLK}/12$, $F_{CLK}/4$
0	1	1	8-bit UART	Variable(16bit)
1	0	2	9-bit UART	$F_{CLK}/32$ or $F_{CLK}/64$
1	1	3	9-bit UART	Variable(16bit)

Timer 2 cannot be used as baud rate generator when Compare Capture unit is present in the system. The UART0 baud rates are presented in the table below.

Mode	Baud Rate
Mode 0	FCLK/12
Mode 1, 3	Timer 1 overflow rate – $T1_{ov}$
	SMOD0 = 0 $T1_{ov}/32$
	SMOD0 = 1 $T1_{ov}/16$
	Timer 2 overflow rate – $T2_{ov}$
	SMOD0 = x $T2_{ov}/16$
Mode 2	SMOD0 = 0 $F_{CLK}/64$
	SMOD0 = 1 $F_{CLK}/32$

The SMOD0 bit is located in PCON register.

REN0：If set, enable serial reception. Cleared by software to disable reception.

TB08：The 9th transmitted data bit in Modes 2 and 3. Set or cleared by the CPU, depending on the function it performs (parity check, multiprocessor communication etc.)

RB08：In Modes 2 and 3 it is the 9th data bit received. In Mode 1, if SM02 is 0, RB08 is the stop bit. In Mode 0 this bit is not used.

PCON register (0x87)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
87h PCON	R/W	SMOD0	SMOD1	-	PWE	-	SWB	STOP	PMM
Reset		0	0	0	0	0	0	0	0

SMOD0：UART0 double baud rate bit when clocked by Timer 1 only.

- INTERRUPTS

UART0 interrupt related bits are shown below. An interrupt can be turned on / off by IE register, and set into high / low priority group by IP register.

IE register (0xA8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A8h IE	R/W	EA	-	ET2	ES0	ET1	EX1	ET0	EX0
Reset		0	0	0	0	0	0	0	0

ES0：RI0 & TI0 interrupt enable flag

IP register (0xB8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

B8h IP	R/W	-	-	PT2	PS0	PT1	PX1	PT0	PX0
Reset		0	0	0	0	0	0	0	0

PS0：RI0 & TI0 interrupt priority flag

SCON0 register (0x98)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
98h SCON0	R/W	SM00	SM01	SM02	REN0	TB08	RB08	TI0	RI0
Reset		0	0	0	0	0	0	0	0

TI0：Transmit interrupt flag, set by hardware after completion of a serial transfer.
Must be cleared by software.
RI0：Receive interrupt flag, set by hardware after completion of a serial reception.
Must be cleared by software.

All of bits that generate interrupts can be set or cleared by software, with the same result as if they had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled by software.

Interrupt flag	Function	Active level / edge	Flag resets	Vector	Natural priority
TI0 & RI0	Internal, UART0	-	Software	0x23	5

Table13.3 UART0 interrupt

The UART1 has the same functionality as a standard 8051 UART. The UART1 related registers are: SBUF1(0xC1), SCON1(0xC0), PCON(0x87), IE(0xA8) and IP(0xB8). The UART1 data buffer (SBUF1) consists of two separate registers: transmit and receive registers. A data writes into the SBUF1 sets this data in UART1 output register and starts a transmission. A data reads from SBUF1, reads data from the UART1 receive register.

SBUF1 register (0xC1)

Reset	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0x00	SB1.7	SB1.6	SB1.5	SB1.4	SB1.3	SB1.2	SB1.1	SB1.0

SB1[7:0]：UART1 buffer

SCON1 register (0xC0)

Reset	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0x00	SM10	SM11	SM12	REN1	TB18	RB18	TI1	RI1

SM12：Enable a multiprocessor communication feature
SM1[1:0]：Sets baud rate

SM10	SM11	Mode	Description	Baud Rate
0	0	0	Shift register	$F_{CLK}/12$, $F_{CLK}/4$
0	1	1	8-bit UART	Variable(16bit)
1	0	2	9-bit UART	$F_{CLK}/32$ or $F_{CLK}/64$
1	1	3	9-bit UART	Variable(16bit)

Timer 2 cannot be used as baud rate generator when Compare Capture unit is present in the system. The UART1 baud rates are presented in the table below.

Mode	Baud Rate	
Mode 0	FCLK/12	
Mode 1, 3	Timer 1 overflow rate – $T1_{ov}$	
	SMOD1 = 0	$T1_{ov}/32$
	SMOD1 = 1	$T1_{ov}/16$
	Timer 2 overflow rate – $T2_{ov}$	
	SMOD1 = x	$T2_{ov}/16$
Mode 2	SMOD1 = 0	$F_{CLK}/64$
	SMOD1 = 1	$F_{CLK}/32$

The SMOD1 bit is located in PCON register.

REN1：If set, enable serial reception. Cleared by software to disable reception.

TB18：The 9th transmitted data bit in Modes 2 and 3. Set or cleared by the CPU, depending on the function it performs (parity check, multiprocessor communication etc.)
RB18：In Modes 2 and 3 it is the 9th data bit received. In Mode 1, if SM12 is 0, RB18 is the stop bit. In Mode 0 this bit is not used.

PCON register (0x87)

Reset	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0x00	SMOD0	SMOD1	-	PWE	-	SWB	STOP	PMM

SMOD1：UART1 double baud rate bit when clocked by Timer 1 only.

- INTERRUPTS

UART1 interrupt related bits are shown below. An interrupt can be turned on / off by IE register, and set into high / low priority group by IP register.

IE register (0xA8)

A8h IE Reset	R/W	EA 0	- 0	ET2 0	ES0 0	ET1 0	EX1 0	ET0 0	EX0 0

ES1：RI1 & TI1 interrupt enable flag

IP register (0xB8)

Reset	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0x00	-	PS1	PT2	PS0	PT1	PX1	PT0	PX0

PS1：RI1 & TI1 interrupt priority flag

SCON1 register (0xC0)

Reset	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
0x00	SM10	SM11	SM12	REN1	TB18	RB18	TI1	RI1

TI1：Transmit interrupt flag, set by hardware after completion of a serial transfer.
Must be cleared by software.
RI1：Receive interrupt flag, set by hardware after completion of a serial reception.
Must be cleared by software.

All of bits that generate interrupts can be set or cleared by software, with the same result as if they had been set or cleared by hardware. That is, interrupts can be generated or pending interrupts can be cancelled by software.

Interrupt flag	Function	Active level / edge	Flag resets	Vector	Natural priority
TI1 & RI1	Internal, UART1	-	Software	0x33	7

Table13.4 UART1 interrupt

13.3 OPERATING MODES

13.3.1 UART0 MODE 0, SYNCHRONOUS

Pin RXD0I serves as input and RXD0O as output. TXD0 output is a shift clock. The baud rate is fixed at 1/12 of the CLK clock frequency. Eight bits are transmitted with LSB first. Reception is initialized by setting the flags in SCON0 as follows: RI0=0 and REN0=1.

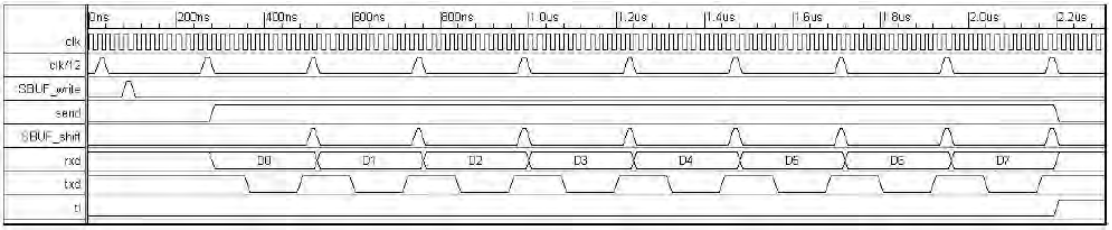


Figure13.3 UART0 transmission mode 0 timing diagram

13.3.2 UART0 MODE 1, 8-BIT UART, VARIABLE BAUD RATE, TIMER CLOCK SOURCE

Pin RXD0I serves as input, and TXD0 serves as serial output. 10 bits are transmitted: a start bit (always 0), 8 data bits (LSB first), and a stop bit (always 1). On receive, a start bit synchronizes the transmission, 8 data bits are available by reading SBUF0, and stop bit sets the flag RB08 in the SFR SCON0. The baud rate is variable and depends from Timer 1 or Timer 2 mode. To enable Timer 2 clocking set the TCLK, RCLK bits located in T2CON (0xC8) register. SMOD0 bit is ignored when UART is clocked by Timer2.

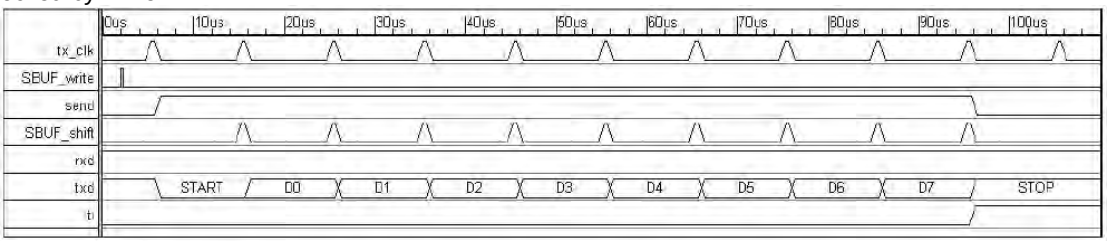


Figure13.4 UART0 transmission mode 1 timing diagram

13.3.3 UART0 MODE 2, 9-BIT UART, FIXED BAUD RATE

This mode is similar to Mode 1 with two differences. The baud rate is fixed at 1/32 or 1/64 of CLK clock frequency, and 11 bits are transmitted or received: a start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). The 9th bit can be used to control the parity of the UART0 interface: at transmission, bit TB08 in SCON0 is output as the 9th bit, and at receive, the 9th bit affects RB08 in SCON0.

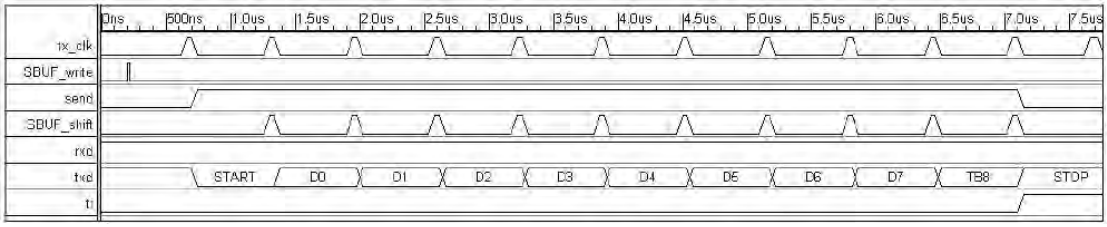


Figure13.5 UART0 transmission mode 2 timing diagram

13.3.4 UART0 MODE 3, 9-BIT UART, VARIABLE BAUD RATE, TIMER CLOCK SOURCE

The only difference between Mode 2 and Mode 3 is that the baud rate is a variable in Mode 3. When REN0=1 data receiving is enabled. The baud rate is variable and depends from Timer 1 or Timer 2 mode. To enable Timer 2 clocking set the TCLK, RCLK bits located in T2CON (0xC8) register. SMOD0 bit is ignored when UART is clocked by Timer2.

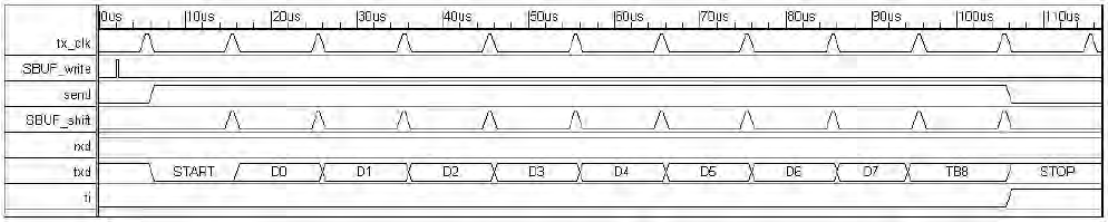


Figure13.6 UART0 transmission mode 3 timing diagram

A9108 supports different crystal frequency by programmable “Clock Register” (0Dh). Based on this, three important internal clocks F_{CGR} , F_{DR} and F_{SYCK} are generated.

(1) F_{XTAL}: Crystal frequency.
(2) F_{XREF}: Crystal Ref. Clock = F_{XTAL} * (DBL+1).
(3) F_{CGR}: Clock Generation Reference = 2MHz = F_{XREF} / (GRC+1), where F_{CGR} is used to generate 32M PLL.
(4) F_{MCLK}: Master Clock is either F_{XREF}: or 32M PLL, where F_{MCLK} is used to generate F_{SYCK}.
(5) F_{SYCK}: System Clock = 16MHz=F_{MCLK} / CSC= 32 * F_{IF}, where F_{IF} is recommended to set 500KHz.
(6) F_{DR}: Data Rate Clock = F_{IF} / (SDR+1).
(7) F_{FPD}: VCO Compared Clock = = F_{XREF} / (RRC+1).

13.3.5 UART1 MODE 0, SYNCHRONOUS

Pin RXD1I serves as input and RXD1O as output. TXD1 output is a shift clock. The baud rate is fixed at 1/12 of the CLK clock frequency. Eight bits are transmitted with LSB first. Reception is initialized by setting the flags in SCON1 as follows: RI1=0 and REN1=1.

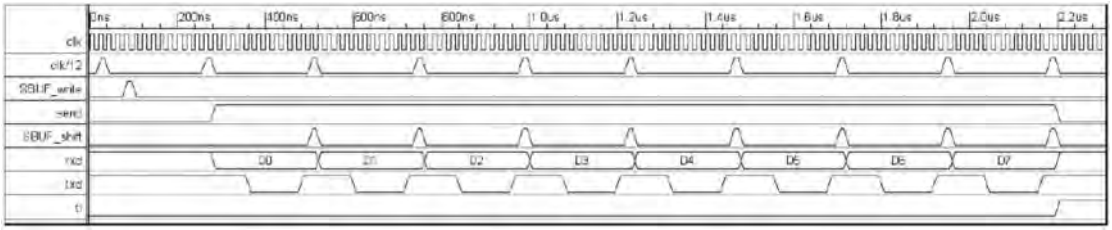


Figure13.7　UART1 transmission mode 0 timing diagram

13.3.6 UART1 MODE 1, 8-BIT UART, VARIABLE BAUD RATE, TIMER CLOCK SOURCE

Pin RXD1I serves as input, and TXD1 serves as serial output. 10 bits are transmitted: a start bit (always 0), 8 data bits (LSB first), and a stop bit (always 1). On receive, a start bit synchronizes the transmission, 8 data bits are available by reading SBUF1, and stop bit sets the flag RB18 in the SFR SCON1. The baud rate is variable and depends from Timer 1 or Timer 2 mode. To enable Timer 2 clocking set the TCLK, RCLK bits located in T2CON (0xC8) register. SMOD1 bit is ignored when UART is clocked by Timer2.

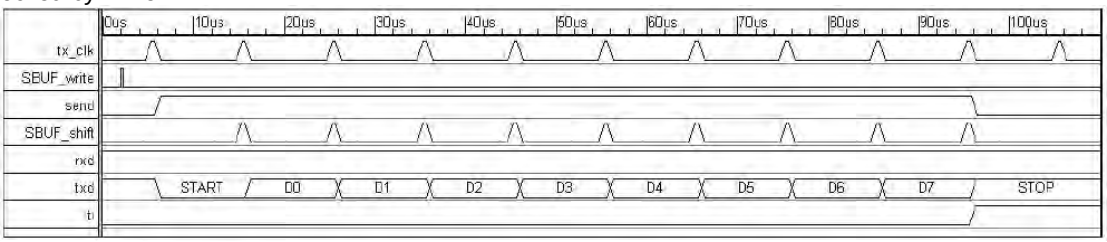


Figure13.8　UART1 transmission mode 1 timing diagram

13.3.7 UART1 MODE 2, 9-BIT UART, FIXED BAUD RATE

This mode is similar to Mode 1 with two differences. The baud rate is fixed at 1/32 or 1/64 of CLK clock frequency, and 11 bits are transmitted or received: a start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). The 9th bit can be used to control the parity of the UART1 interface: at transmission, bit TB18 in SCON1 is output as the 9th bit, and at receive, the 9th bit affects RB18 in SCON1.

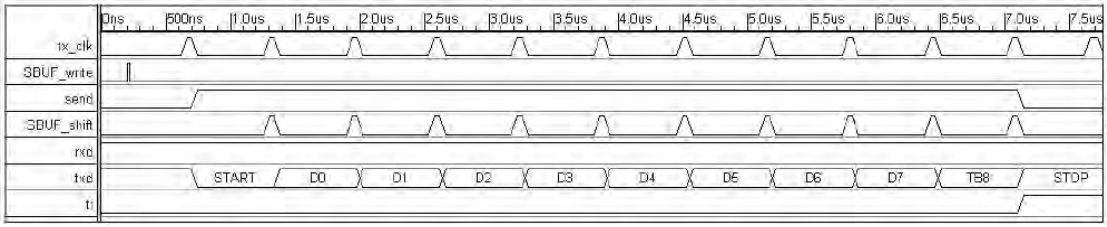


Figure13.9　UART1 transmission mode 2 timing diagram

13.3.8 UART1 MODE 3, 9-BIT UART, VARIABLE BAUD RATE, TIMER CLOCK SOURCE

The only difference between Mode 2 and Mode 3 is that the baud rate is a variable in Mode 3. When REN1=1 data receiving is enabled. The baud rate is variable and depends from Timer 1 or Timer 2 mode. To enable Timer 2 clocking set the TCLK, RCLK bits located in T2CON (0xC8) register. SMOD1 bit is ignored when UART is clocked by Timer2.

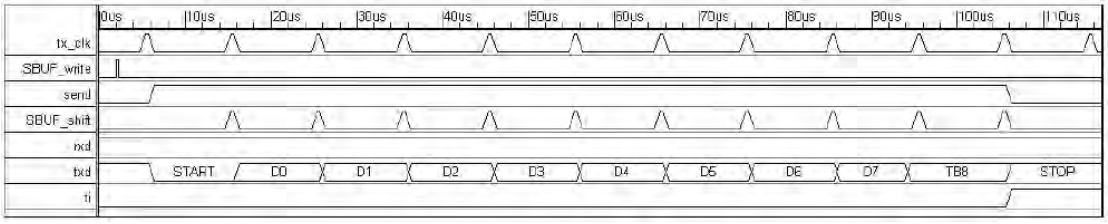


Figure13.10　UART1 transmission mode 3 timing diagram

14. IIC interface

A9108's I²C peripheral provides two-wire interface between the device and I²C -compatible device by the two-wire I²C serial bus. The I²C peripheral supports the following functions.

- Conforms to v2.1 of the I²C specification (published by Philips Semiconductor)
- Master transmitter / receiver
- Slave transmitter / receiver
- Flexible transmission speed modes: Standard (up to 100 Kb/s) and Fast (up to 400Kb/s)
- Multi-master systems supported
- Supports 7-bit addressing modes on the I²C bus
- Interrupt generation
- Allows operation from a wide range of input clock frequencies (build-in 8-bit timer)

PIN 23 and PIN 24 are I2C Interface in A9108. The alternate function is Port 0.5 and Port 0.6. User can set BBSEL (BBH) to set up the PIN function. Please refer the Chapter 11 for more detail information.

PIN	TYPE	DESCRIPTION
SCL(P0.5)	INPUT /OUTPUT	I²C clock input /output
SDA(P0.6)	INPUT/ OUTPUT	I²C data input /output

Table14.1 I2C interface pins description

14.1 Master mode I²C

The I²C master mode provides an interface between a microprocessor and an I²C bus. It can be programmed to operate with arbitration and clock synchronization to allow it to operate in multi-master systems. Master mode I²C supports transmission speeds up to 400Kb/s.

14.1.1 I²C REGISTERS

There are six registers used to interface to the host: the Control, Status, Slave Address, Transmitted Data, Received Data and Timer Period Register.

Register	Address
Slave address – I2CMSA	0xF4
Control – I2CMCR	0xF5
Transmitted data I2CBUF	0xF6
Timer period - I2CMTP	0xF7

Table14.3 I²C Registers for writing

Register	Address
Slave address – I2CMSA	0xF4
Status – I2CMSR	0xF5
Received data - I2CBUF	0xF6
Timer period - I2CMTP	0xF7

Table14.4 I²C Registers for reading

■ **I²C Master mode Timer Period Register**

To generate wide range of SCL frequencies the core have built-in 8-bit timer. Programming sequence must be done at least once after system reset. After reset, register have 0x01 value by default.

SCL_PERIOD = 2 x (1+TIMER_PRD) x (SCL_LP + 1) x CLK_PRD
For example： - CLK_PRD = 33,33ns (CLK_FRQ = 30MHz)； - TIMER_PRD = 3； - SCL_LP = 6；
SCL_PERIOD = 2 x (1 + 3) x (6 + 1) x 33,33ns = 3200ns = 2,666us
SCL_FREQUENCY = 1 / 2,666us = 375 KHz
SCL_PRD - SCL line period (I2C clock line)
TIMER PRD -Timer period register value (range 1 to 255)

SCL_LP	**- SCL_LOW_PERIOD constant value (range 2 to 15)**
CLK_PRD	**- System clock period (1/f_{clk})**

I2CMTP (0xE7)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
E7h I2CMTP	R/W	0	P.6	P.5	P.4	P.3	P.2	P.1	P.0
Reset		0	0	0	0	0	0	0	1

■ I²C CONTROL AND STATUS REGISTERS

The Control Register consists of eight bits: the RUN, START, STOP, ACK, HS, ADDR, SLRST and RSTB bit.

The RSTB bit performs reset of whole I²C controller and behaves identically as external reset provided by RST pin. Using this bit software application can reinitialize I²C mater module when some problem is encountered on I²C bus. In case when I²C Slave device blocks I²C bus, then SLRST bit should be set along with RUN bit (just after issuing the RSTB). SLRST bit causes that I²C master module generates 9 SCK clocks (no START is generated) to recover Slave device to known state and issues at the end STOP. This bit is automatically cleared by I2C MASTER MODULE, thus, it is always read as '0'. The BUSY bit should be checked to know when this transmission is ended.

The START bit will cause the generation of the START, or REPEATED START condition. The STOP bit determines if the cycle will stop at the end of the data cycle, or continue on to a burst. To generate a single send cycle, the Slave Address register is written with the desired address, the R/S bit is set to '0', and Control Register is written with HS=0, ACK=x, STOP=1, START=1, RUN=1 (binary xxx0x111 x-mean 0 or 1) to perform the operation and stop. When the operation is completed (or aborted due an error), the interrupt is generated. The data may be read from Received Data Register. When I2C MASTER MODULE core operates in Master receiver mode the ACK bit must be set normally to logic 1. This cause the I2C MASTER MODULE bus controller to send acknowledge automatically after each byte. This bit must be reset when the I2C MASTER MODULE bus controller requires no further data to be sent from slave transmitter.

The ADDR bit along with RUN bit cause the generation of the START condition and transmission of Slave Address. Next STOP can end transmission, or REPEATED START generates the START and ADDRRESS sequence once again. In both cases STOP can ends transmission. See I²C MASTER MODULE ACK Polling chapter for details.

I2CMCR (0xF5)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F5h I2CMCR	R/W	RSTB	SLRST	ADDR	HS	ACK	STOP	START	RUN
Reset		0	0	0	0	0	0	0	0

RSTB	SLRST	ADDR	HS	R/S	ACK	STOP	START	RUN	OPERATION
0	0	0	0	0	-	0	1	1	START condition followed by SEND (Master remains in Transmitter mode)
0	0	0	0	0	-	1	1	1	START condition followed by SEND and STOP condition
0	0	0	0	1	0	0	1	1	START condition followed by RECEIVE operation with **negative Acknowledge** (Master remains in Receiver mode)
0	0	0	0	1	0	1	1	1	START condition followed by RECEIVE and STOP condition
0	0	0	0	1	1	0	1	1	START condition followed by RECEIVE (Master remains in Receiver mode)
0	0	0	0	1	1	1	1	1	**forbidden sequence**
0	0	0	1	0	0	0	0	1	Master Code sending and switching to High-speed mode
1	0	0	-	-	-	-	-	-	I2CM module software reset
0	1	0	0	0	0	0	0	1	Reset slaves connected to I2C bus by generating 9 SCK clocks followed by STOP
0	0	1	0	0	0	0	0	1	START condition followed by Slave Address

Table14.5 Control bits combinations permitted in IDLE state *

RSTB	SLRST	ADDR	HS	R/S	ACK	STOP	START	RUN	OPERATION
0	0	0	0	-	-	0	0	1	SEND operation (Master remains in Transmitter mode)
0	0	0	0	-	-	1	0	0	STOP condition
0	0	0	0	-	-	1	0	1	SEND followed by STOP condition
0	0	0	0	0	-	0	1	1	Repeated START condition followed by SEND (Master remains in Transmitter mode)
0	0	0	0	0	-	1	1	1	Repeated START condition followed by SEND and STOP condition
0	0	0	0	1	0	0	1	1	Repeated START condition followed by RECEIVE operation with **negative Acknowledge** (Master remains in Receiver mode)
0	0	0	0	1	0	1	1	1	Repeated START condition followed by SEND and STOP condition
0	0	0	0	1	1	0	1	1	Repeated START condition followed by RECEIVE (Master remains in Receiver mode)
0	0	0	0	1	1	1	1	1	**forbidden sequence**
1	0	0	-	-	-	-	-	-	I2CM module software reset
0	0	1	0	0	-	0	1	1	Repeated START condition followed by Slave Address

Table14.6 Control bits combinations permitted in Master Transmitter mode

RSTB	SLRST	ADDR	HS	R/S	ACK	STOP	START	RUN	OPERATION
0	0	0	0	-	0	0	0	1	RECEIVE operation with **negative Acknowledge** (Master remains in Receiver mode)
0	0	0	0	-	-	1	0	0	STOP condition**
0	0	0	0	-	0	1	0	1	RECEIVE followed by STOP condition
0	0	0	0	-	1	0	0	1	RECEIVE operation (Master remains in Receiver mode)
0	0	0	0	-	1	1	0	1	**forbidden sequence**
0	0	0	0	1	0	0	1	1	Repeated START condition followed by RECEIVE operation with **negative Acknowledge** (Master remains in Receiver mode)
0	0	0	0	1	0	1	1	1	Repeated START condition followed by RECEIVE and STOP condition
0	0	0	0	1	1	0	1	1	Repeated START condition followed by RECEIVE (Master remains in Receiver mode)
0	0	0	0	0	-	0	1	1	Repeated START condition followed by SEND (Master remains in Transmitter mode)
0	0	0	0	0	-	1	1	1	Repeated START condition followed by SEND and STOP condition
1	0	0	-	-	-	-	-	-	I2CM module software reset

Table14.7 Control bits combinations permitted in Master Receiver mode

The status Register is consisted of six bits：the BUSY bit, the ERROR bit, the ADDR_ACK bit, the DATA_ACK bit, the ARB_LOST bit, and the IDLE bit.

I2CMSR (0xF5)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F5h I2CMSR	R/W	-	BUS_ BUSY	IDLE	ARB_ LOST	DATA_ ACK	ADDR_ ACK	ERROR	BUSY
Reset	0x20	0	0	1	0	0	0	0	0

IDLE：This bit indicates that I2C BUS controller is in the IDLE state。

BUSY：This bit indicates that I2C BUS controller receiving, or transmitting data on the bus, and other bits of Status register are no valid;
BUS_BUSY：This bit indicates that the Bus is Busy, and access is not possible. This bit is set / reset by START and STOP conditions;
ERROR：This bit indicates that due the last operation an error occurred: slave address wasn't acknowledged, transmitted data wasn't acknowledged, or I2C Bus controller lost the arbitration;
ADDR_ACK：This bit indicates that due the last operation slave address wasn't acknowledged;
ARB_LOST：This bit indicates that due the last operation I2C Bus controller lost the arbitration;

■ **SLAVE ADDRESS REGISTER**

The Slave address Register consists of eight bits：Seven address bits (A6-A0), and Receive/ not send bit R/S. The R/S bit determines if the next operation will be a Receive (high), or Send (low).

I2CMSA (0xF4)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F4h I2CMCA	R/W	A.6	A.5	A.4	A.3	A.2	A.1	A.0	R/S
Reset		0	0	0	0	0	0	0	0

■ **I²C Buffer – RECEIVER AND TRANSMITTER REGISTERS**

I2C module has two separated 1 byte buffer in receiver and transmitter and these are located in the same address (0xF6). The Transmitted Data Register consists of eight data bits which will be sent on the bus due the next Send, or Burst Send operation. The first send bit is D.7 (MSB).

I2CBUF (0xF6)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F6h I2CBUF	R/W	D.7	D.6	D.5	D.4	D.3	D.2	D.1	D.P
Reset		0	0	0	0	0	0	0	0

The Receiver Data Register consists of eight data bits which have been received on the bus due the last receive, or Burst Receive operation.
I2CBUF (0xF6)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F6h I2CBUF	R/W	D.7	D.6	D.5	D.4	D.3	D.2	D.1	D.P
Reset		0	0	0	0	0	0	0	0

14.2.4 I2C MASTER MODULE AVAILABLE SPEED MODES

Default transmission parameter/constant values are shown in sections below. SCL clock frequency can be changed by modification of timer period values as show in the table below.

■ **I2C MASTER MODULE STANDARD MODE**

Typical configuration values for Standard speed mode：

The following table gives an example parameters for standard I2C speed mode.

System clock	TIMER_PERIOD	Transmission speed
4 MHz	1 (01h)	100kb/s
6 MHz	2 (02h)	100kb/s
10 MHz	4 (04h)	100kb/s
16 MHz	7 (07h)	100kb/s
20 MHz	9 (09h)	100kb/s

Table14.8 I2C MASTER MODULE Timer period values for standard speed mode

■ **I2C MASTER MODULE FAST MODE**

Typical configuration values for Fast speed mode：

The following table gives example parameters for Fast I2C speed mode.

System clock	TIMER_PERIOD	Transmission speed
10 MHz	1 (01h)	250 Kb/s
16 MHz	1 (01h)	400 Kb/s
20 MHz	2 (02h)	333 Kb/s

Table14.8　I2C MASTER MODULE Timer period values for Fast speed mode

14.2.5 I2C MASTER MODULE AVAILABLE COMMAND SEQUENCES

■ **I2C MASTER MODULE SINGLE SEND**

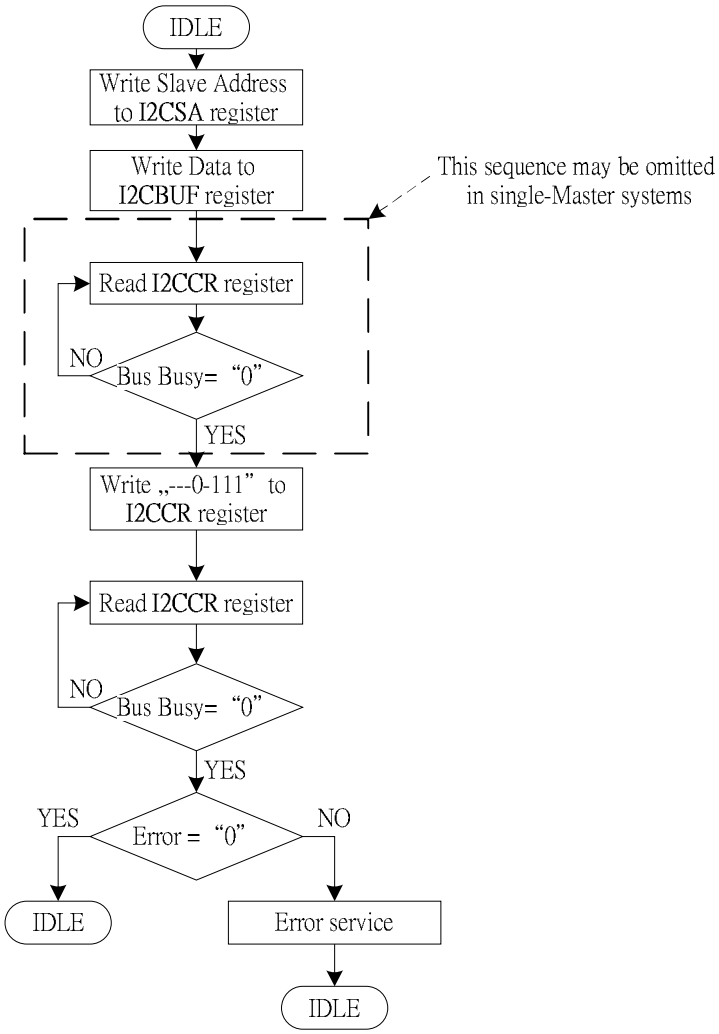


Figure14.4　I2C MASTER MODULE Single SEND flowchart

■ **I2C MASTER MODULE SINGLE RECEIVE**

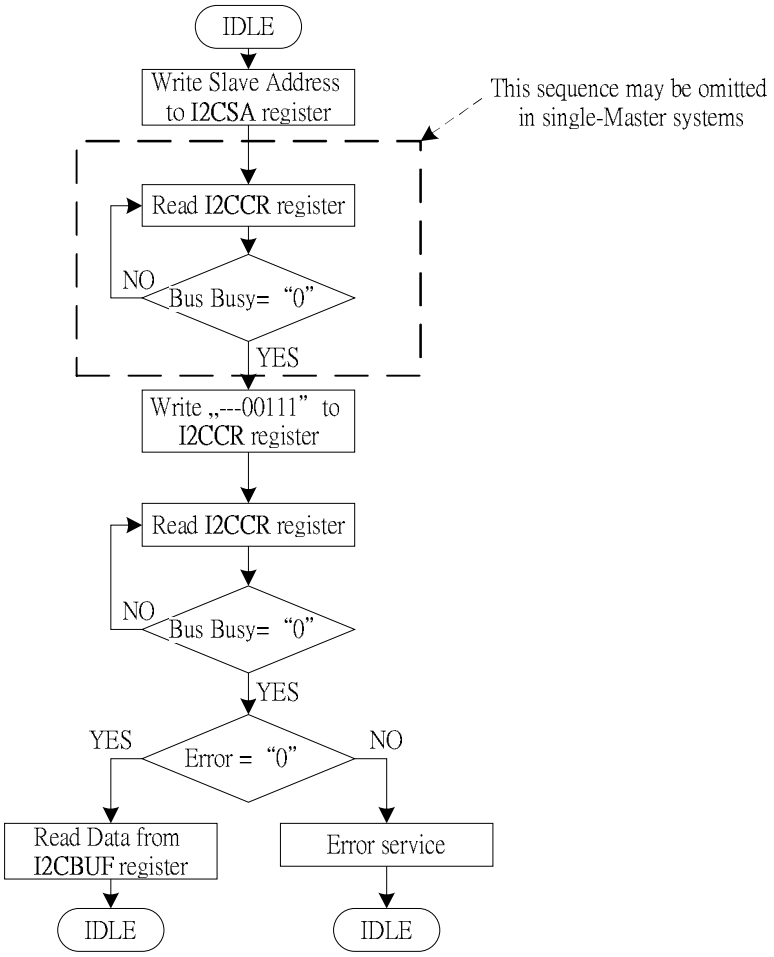


Figure14.5　Single RECEIVE flowchart

■ **I2C MASTER MODULE BURST SEND**

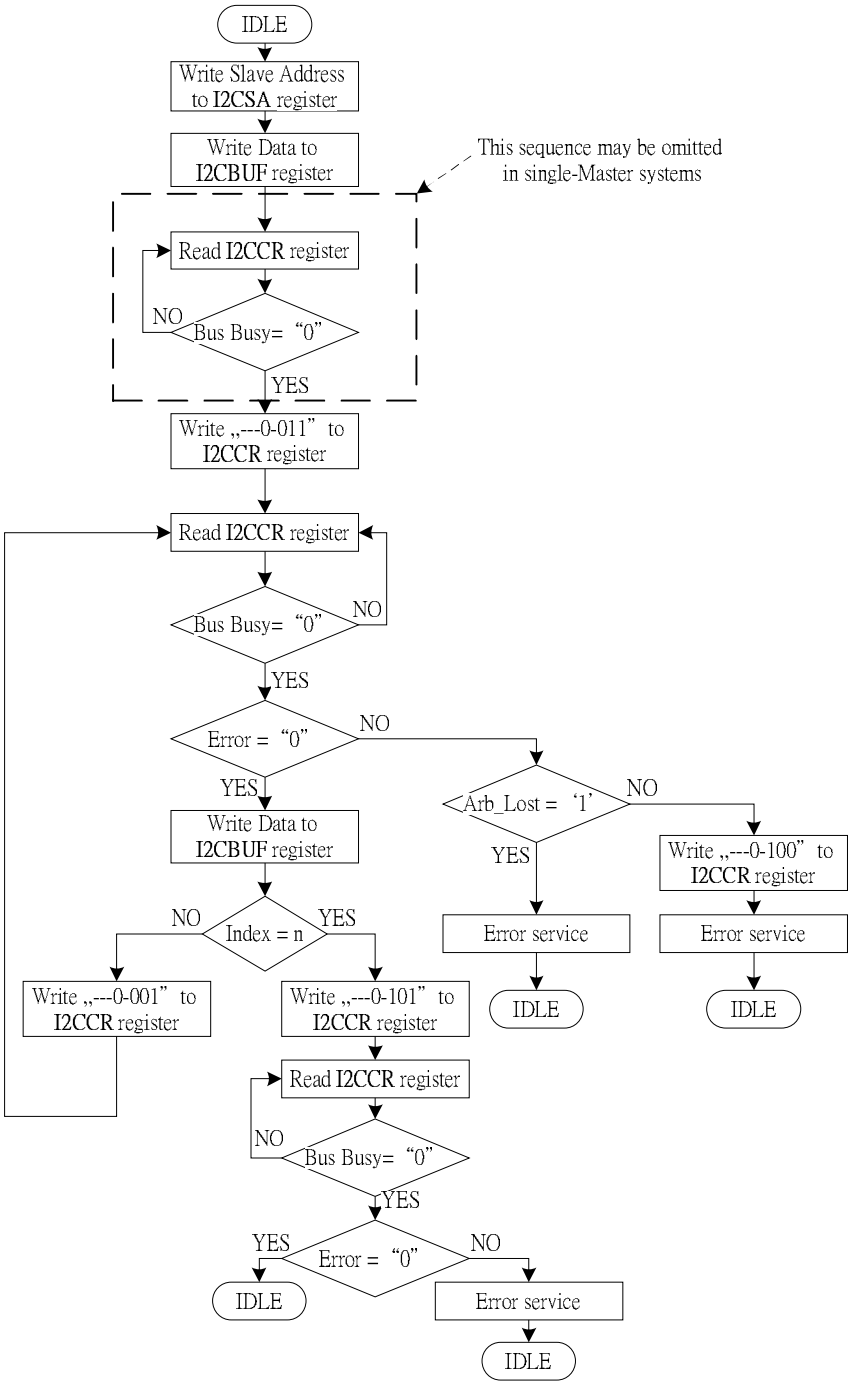


Figure14.6　I2C MASTER MODULE Sending n bytes flowchart

■ I2C MASTER MODULE BURST RECEIVE

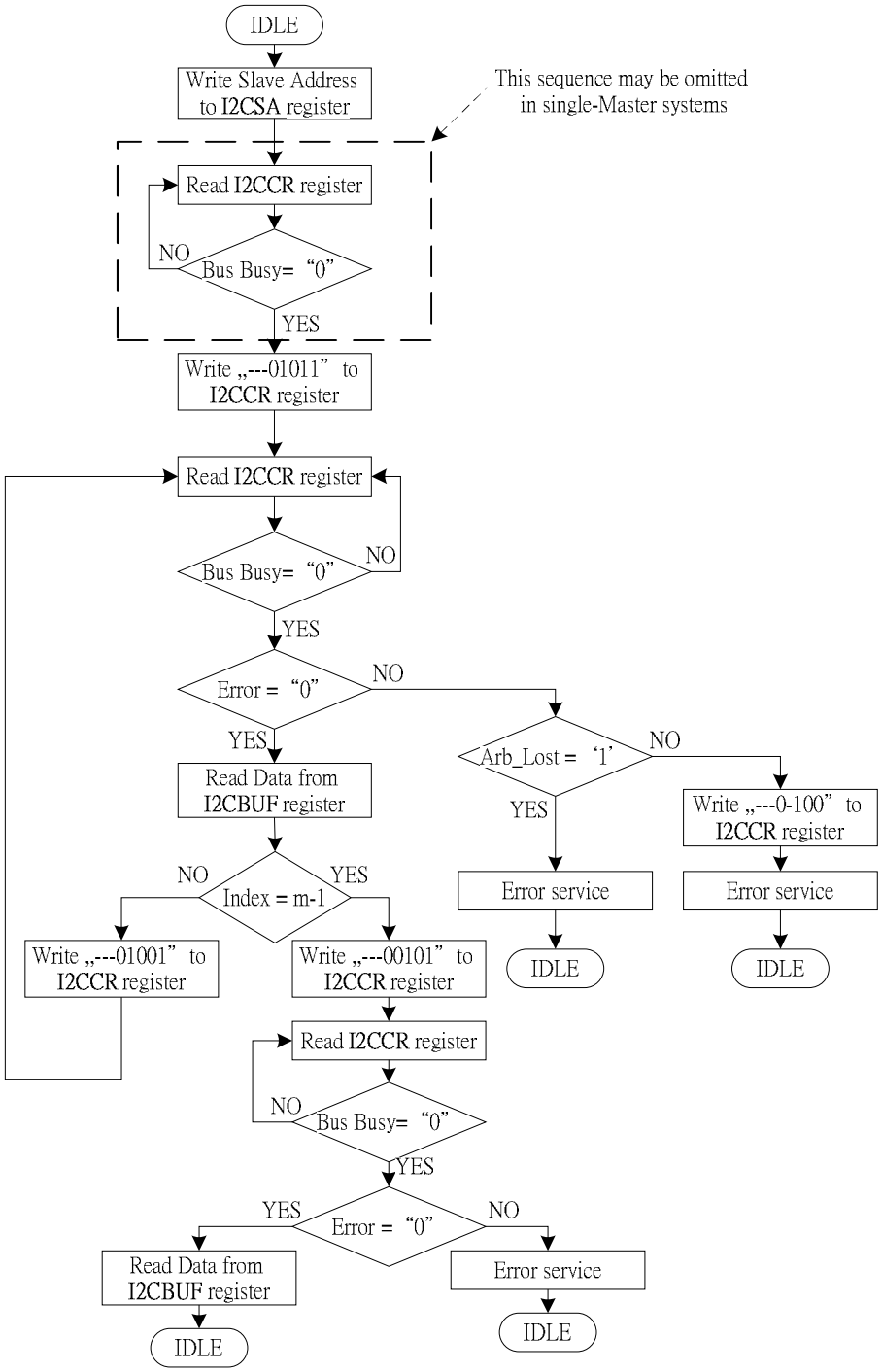


Figure14.7　I2C MASTER MODULE Receiving m bytes flowchart

■ **I2C MASTER MODULE BURST RECEIVE AFTER BURST SEND**

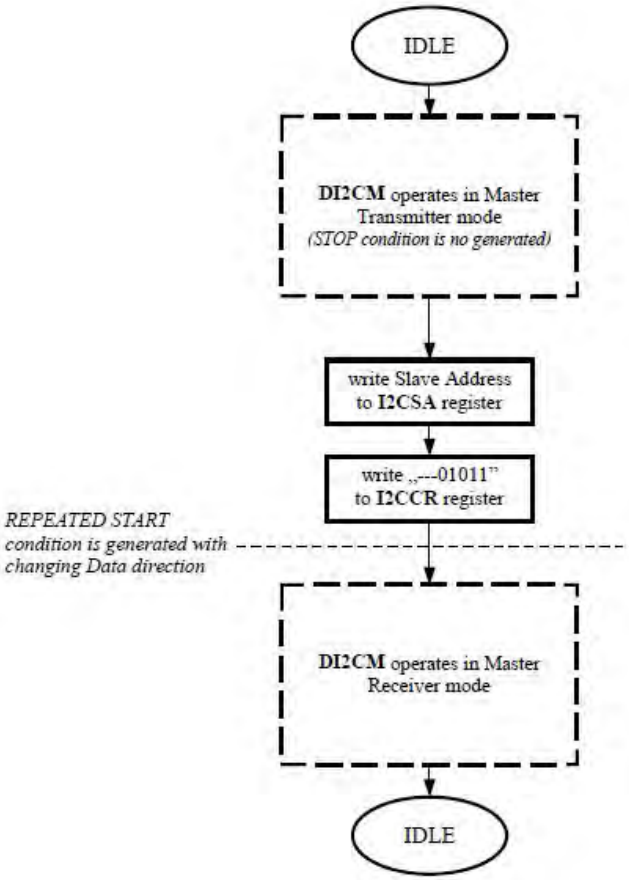


Figure14.8　I2C MASTER MODULE Sending n bytes then Repeated Start and Receiving m bytes flowchart

■ **I2C MASTER MODULE BURST SEND AFTER BURST RECEIVE**

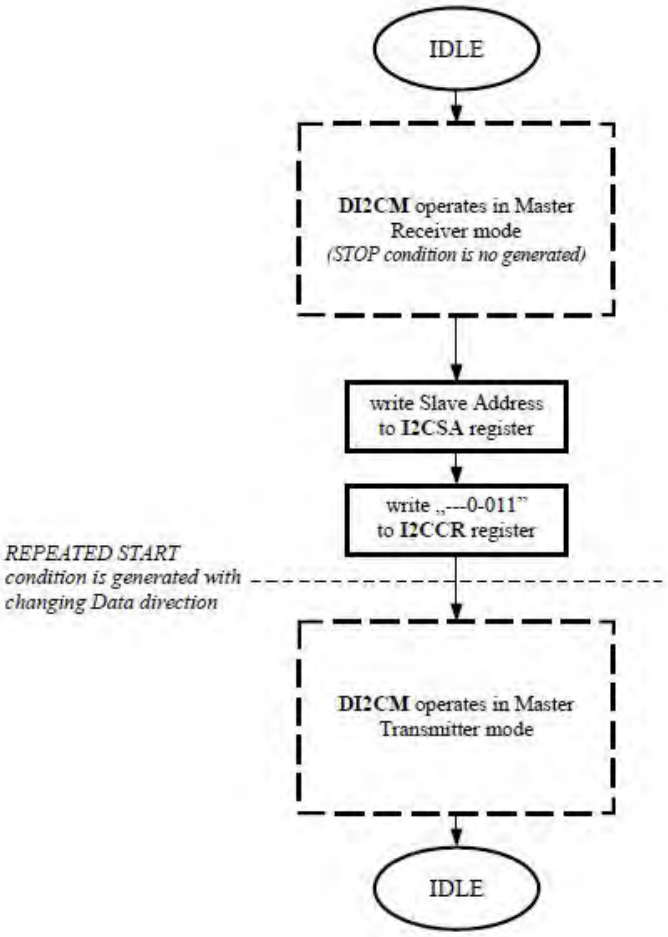


Figure14.9　I2C MASTER MODULE Receiving m bytes then Repeated Start and Sending n bytes flowchart

Figure14.10　I2C MASTER MODULE Single RECEIVE with 10-bit addressing flowchart

- **I2C MASTER MODULE ACK POLLING**

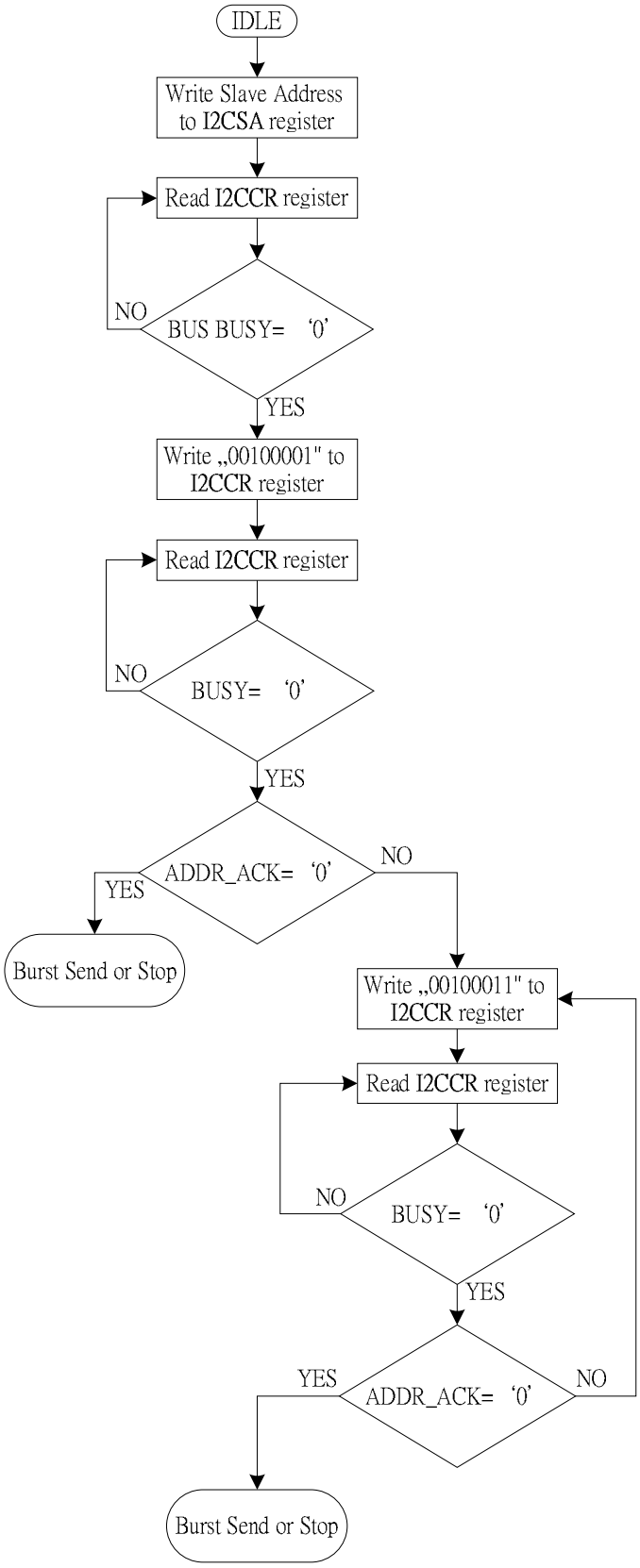


Figure14.11 I2C MASTER MODULE ACK Polling flowchart

14.3 I2C MASTER MODULE INTERRUPT GENERATION

I2C MASTER MODULE interrupt flag is automatically asserted when I2C transfer (send or receive a byte) is completed or transfer error has occurred. I2CMIF flag has to be cleared by software.

Interrupt flag	Function	Active level/edge	Flag resets	Vector	Natural priority
I2CMIF	Internal, I2C MASTER MODULE	-	Software	0x6B	14

Table14.11　I2C MASTER MODULE interrupt summary

I2C MASTER MODULE related interrupt bits have been summarized below. The IE (0xA8) contains global interrupt system disable (0) / enable (1) bit called EA.

EIE　(0xE8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
E8h EIE Reset	R/W	EI2CSESPI 0	EI2CM 0	EWDI 0	EINT6 0	EINT5 0	EINT4 0	EINT3 0	EINT2 0

EI2CM：Enable I2C MASTER MODULE interrupts

EIP　(0xF8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F8h EIP Reset	R/W	PI2CSPSPI 0	PI2CM 0	PWDI 0	PINT6 0	PINT5 0	PINT4 0	PINT3 0	PINT2 0

PI2CM：I2C MASTER MODULE priority level control (at 1-high-level)

EIF　(0x91)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
91h EIF Reset	R/W	I2CSFSPIF 0	I2CMF 0	- 0	INT6F 0	INT5F 0	INT4F 0	INT3F 0	INT2F 0

I2CMIF：I2C MASTER MODULE interrupt flag

Must be cleared by software writing logic '1'. Writing '0' does not change its content.

14.5 Slave mode I^2C

The I^2C module provides an interface between a microprocessor and I^2C bus. It can works as a slave receiver or transmitter depending on working mode determined by microprocessor/microcontroller. The core incorporates all features required by I^2C specification. The I^2C module supports all the transmission modes: Standard and Fast.

14.5.1 I2C MODULE INTERNAL REGISTERS

There are five registers used to interface to the target device：The Own Address, Control, Status, Transmitted Data and Received Data registers.

Register	Address
Own address – I2CSOA	0xF1
Control – I2CSCR	0xF2
Transmitted data – I2CSBUF	0xF3

Table14.12　I2C MODULE Registers for writing

Register	Address
Own address – I2CSOA	0xF1
Control – I2CSSR	0xF2
Received data – I2CSBUF	0xF3

Table14.13 I2C MODULE Registers for reading

■ **I2CSOA – OWN ADDRESS REGISTER**

The Own Address Register consists of seven address bits which identify I²C module core on I²C Bus. This register can be read and written at the address 0xF1.

I2CSOA (0xF1)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F1h I2CSOA	R/W	-	A.6	A.5	A.4	A.3	A.2	A.1	A0
Reset		0	0	0	0	0	0	0	0

■ **I2CSCR – CONTROL AND STATUS REGISTERS**

The Control Register consists of the bits：The RSTB and DA bit. The RSTB bit performs reset of whole I²C controller and behaves identically as external reset provided by RST pin. Using this bit software application can reinitialize I²C module when some problem is encountered on I²C bus. The DA bit enables (‘1’) and disable (‘0’) the I²C module device operation. DA is set immediately to ‘1’ when CPU write DA=1. This register can be only written at address 0xF2. Reading this address puts status register on data bus – see below.

I2CSCR (0xF2)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F2h I2CSCR	R/W	RSTB	DA	-	-	RECFINCLR	SENDFINCLR	-	-
Reset		0	0	0	0	0	0	0	0

DA：Device Active – enable or disable the I²C module device operation;
RSTB：Reset of whole I²C controller by writing ‘1’ to this bit. It behaves identically as RST pin
RECFINCLR：Writing ‘1’ to this bit clears RECFIN bit from the I2C MODULE status register.
SENDFINCLR：Writing ‘1’ to this bit clears SENDFIN bit from the I2C MODULE status register.

The Status Register consists of five bits: the DA, BUSACTIVE, RECFIN, SENDFIN bit, RREQ bit, TREQ bit. The receive finished RECFIN bit indicates that Master I2C controller has finished transmitting of data during single or burst receive operations. It also causes generation of interrupt on IRQ pin. The send finished SENDFIN bit indicates that Master I2C controller has finished receiving of data during single or burst send operations. It also causes generation of interrupt on IRQ pin. The Receive Request RREQ bit indicates that I²C module device has received data byte from I2C master. I²C module host device (usually CPU) should read one data byte from the Received Data register I2CSBUF. The Transmit Request TREQ bit indicates that I2C MODULE device is addressed as Slave Transmitter and I²C module host device (usually CPU) should write one data byte into the Transmitted Data register I2CSBUF. The BUSACTIVE ‘1’ signalizes that any transmission (send, receive or own address detection) is in progress. BUSACTIVE is cleared (‘0’) automatically by I²C module in case when there is no any transmission. This is read only bit.

The DA bit should be polled (read) when CPU wrote DA=0. The DA bit is not immediately cleared when any I2C transmission (send, receive or own address detection) is in progress. When current transmission has completed then this bit is cleared to ‘0’ and I²C module become inactive.

I2CSSR (0xF2)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F2h I2CSSR	R/W		DA	-	BUSACTIVE	RECFIN	SENDFIN	TREQ	RREQ
Reset		0	0	0	0	0	0	0	0

DA：Device Active – enable (‘1’) or disable (‘0’) the I2C MODULE device operation;
BUSACTIVE：Bus ACTIVE – ‘1’ signalizes that any transmission: send, receive or own address detection is in progress;
RREQ：Indicates that I²C module device has received data byte from I²C master;
It is automatically cleared by read of I2CSBUF.

TREQ：Indicates that I²C module device is addressed as transmitter and requires data byte from host device;
It is automatically cleared by write data I2CSBUF.
RECFIN：Indicates that Master I2C controller has ended transmit operation. It means that no more RREQ will be set during this single or burst I²C module receive operation. It is cleared by writing '1' to the RECFINCLR bit in the I²C module control register.
SENDFIN：Indicates that Master I2C controller has ended receive operation. It means that no more TREQ will be set during this single or burst I²C module send operation. It is cleared by writing '1' to the SENDFINCLR bit in the I2C control register.
NOTE：All bits are active at HIGH level ('1').

■ **I2CSBUF – RECEIVER AND TRANSMITTER REGISTERS**

The Transmitter Data Register consists of eight Data bits which will be sent on the bus due the next Send operation. The first send bit is the D.7(MSB).
I2CSBUF (0xF3)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F3h I2CSBUF	R/W	D.7	D.6	D.5	D.4	D.3	D.2	D.1	D.0
Reset		0	0	0	0	0	0	0	0

The Receiver Data Register consists of eight data bits which have been received on the bus due the last Receive operation.
I2CSBUF (0xF3)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F3h I2CSBUF	R/W	D.7	D.6	D.5	D.4	D.3	D.2	D.1	D.0
Reset		0	0	0	0	0	0	0	0

14.7 AVAILABLE I2C MODULE TRANSMISSION MODES

This chapter describes all available transmission modes of the I²C module core. Default I2C own address for all presented waveforms is 0x39 ("0111001").

14.7.1 I²C module SINGLE RECEIVE

The figure below shows a set of sequences during Single data Receive by I2C MODULE. Single receive sequences：

- ✧ Start condition
- ✧ I²C module is addressed by I2C Master as receiver
- ✧ Address is acknowledged by I²C module
- ✧ Data is received by I²C module
- ✧ Data is acknowledged by I²C module
- ✧ Stop condition

14.7.2 I²C module SINGLE SEND

The figure below shows a set of sequences during Single data Send by I2C MODULE. Single send sequences：

- ✧ Start condition
- ✧ I²C module is addressed by I2C Master as transmitter
- ✧ Address is acknowledged by I²C module
- ✧ Data is transmitted by I²C module
- ✧ Data is not acknowledged by I2C Master
- ✧ Stop condition

14.7.3 I²C module BURST RECEIVE

The figure below shows a set of sequences during Burst data Receive by I²C module. Burst receive sequences：

- ✧ Start condition
- ✧ I²C module is addressed by I2C Master as receiver
- ✧ Address is acknowledged by I²C module
- ✧ (1)Data is received by I²C module

- ✧ (2)Data is acknowledged by I^{2}C module
- ✧ STOP condition

Sequences (1) and (2) are repeated until Stop condition occurs.

14.7.4 I^{2}C module BURST SEND

The figure below shows a set of sequences during Burst Data Send by I^{2}C module. Burst send sequences：

- ✧ Start condition
- ✧ I^{2}C module is addressed by I2C Master as transmitter
- ✧ Address is acknowledged by I^{2}C module
- ✧ (1)Data is transmitted by I^{2}C module
- ✧ (2)Data is acknowledged by I2C Master
- ✧ (3)Last data is not acknowledged by I2C Master
- ✧ Stop condition

Sequences (1) and (2) are repeated until last transmitted data is not acknowledged (3) by I2C Master.

14.7.5 AVAILABLE I²C module COMMAND SEQUENCES FLOWCHART

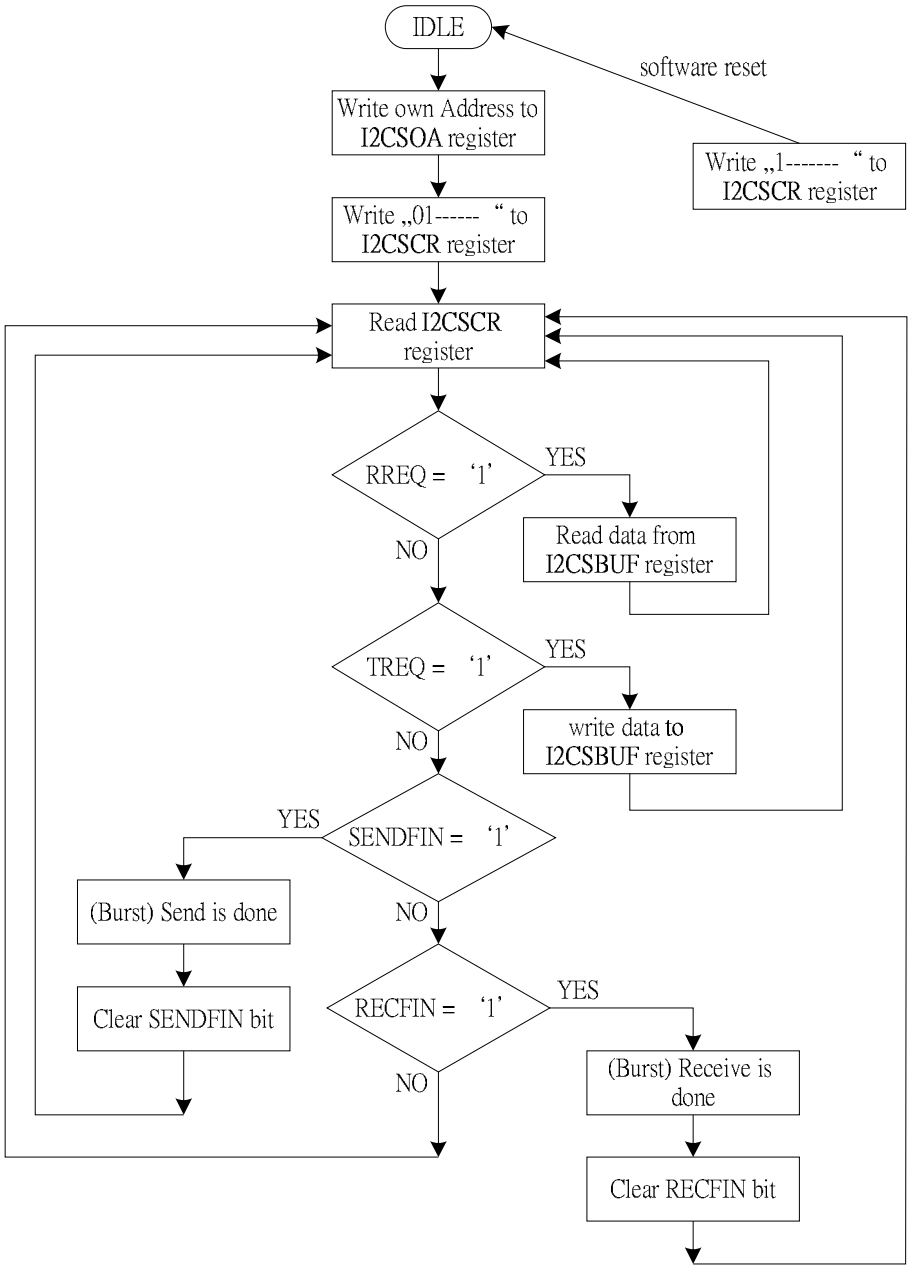


Figure14.20 Available I2C MODULE command sequences flowchart

14.8 I2C MODULE INTERRUPT GENERATION

I2C MODULE interrupt flag is automatically asserted when I2C transfer (send or receive a byte) is completed or transfer error has occurred. I2CSIF flag has to be cleared by software.

Interrupt flag	Function	Active level/edge	Flag resets	Vector	Natural priority
I2CSIF	Internal, DI2CS	-	Software	0x73	15

Table14.16 I2C MODULE interrupt summary

I2C MODULE related interrupt bits have been summarized below. The IE (0xA8) contains global interrupt system disable (0) / enable (1) bit called EA.

EIE (0xE8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
E8h EIE	R/W	EI2CS ESPI	EI2CM	EWDI	EINT6	EINT5	EINT4	EINT3	EINT2
Reset		0	0	0	0	0	0	0	0

EI2CS：Enable I2C MODULE interrupts

EIP (0xF8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F8h EIP	R/W	PI2CS PSPI	PI2CM	PWDI	PINT6	PINT5	PINT4	PINT3	PINT2
Reset		0	0	0	0	0	0	0	0

PI2CS：I2C MODULE priority level control (at 1-high-level)

EIF (0x91)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
91h EIF	R/W	I2CSF SPIF	I2CMF	-	INT6F	INT5F	INT4F	INT3F	INT2F
Reset		0	0	0	0	0	0	0	0

I2CSIF：I2C MODULE interrupt flag

Software should determine the source of interrupt by check both modules' interrupt related bits. Must be cleared by software writing 0x80. It cannot be set by software.

-：Unimplemented bit. Read as 0 or 1.

15. SPI interface

The SPI is a fully configurable SPI master/slave device, which allows user to configure polarity and phase of serial clock signal SCK.

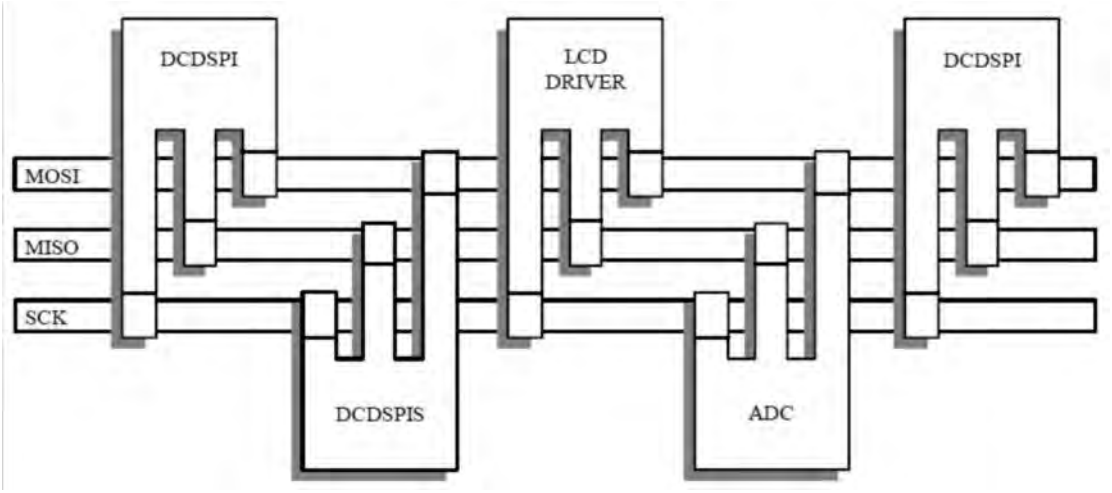
The SPI allows the microcontroller to communicate with serial peripheral devices. It is also capable of inter-processor communications in a multi-master system. A serial clock line (SCK) synchronizes shifting and sampling of the information on the two independent serial data lines. SPI data are simultaneously transmitted and received.

The SPI is a technology independent design that can be implemented in a variety of process technologies.

The SPI system is flexible enough to interface directly with numerous standard product peripherals from several manufacturers. The system can be configured as a master or a slave device. Data rates as high as System clock divided by four (CLK/4). Clock control logic allows a selection of clock polarity and a choice of two fundamentally different clocking protocols to accommodate most available synchronous serial peripheral devices. When the SPI is configured as a master, software selects one of four different bit rates for the serial clock.

The SPI automatically drive selected by SSCR (Slave Select Control Register) slave select outputs (SS7O – SS0O), and address SPI slave device to exchange serially shifted data.

Error-detection logic is included to support inter-processor communications. A write-collision detector indicates when an attempt is made to write data to the serial shift register while a transfer is in progress. A multiple-master mode-fault detector automatically disables SPI output drivers if more than one SPI devices simultaneously attempts to be become bus master.



15.1 KEY FEATURES

All features listed below are included in the current version of SPI core.

- SPI Master
  - Full duplex synchronous serial data transfer
  - Master operation
  - Multi-master system supported
  - Up to 8 SPI slaves can be addressed
  - System error detection
  - Interrupt generation
  - Supports speeds up to 1/4 up to system clock
  - Bit rates generated 1/4, 1/8, 1/32, 1/64, 1/128, 1/512 of system clock
  - Four transfer formats supported
  - Simple interface allows easy connection to microcontrollers
- SPI Slave
  - Full duplex synchronous serial data transfer
  - Slave operation
  - System error detection
  - Interrupt generation
  - Supports speeds up to 1/4 of system clock

- Simple interface allows easy connection to microcontrollers
- Four transfer formats supported

- Fully synthesizable, static synchronous design with no internal tri-states

15.2 SPI PINS DESCRIPTION

PIN	TYPE	ACTIVE	DESCRIPTION
Scki_Scko(P0.0)	INPUT / OUTPUT	-	SPI clock input / output
Miso(P0.1)	INPUT / OUTPUT	-	Master serial data input / Slave serial data output
simo(P0.2)	INPUT / OUTPUT	-	Slave serial data input / Master serial data output
ss(P0.3)	INPUT	low	Slave select
ss7o – ss0o(P0.4)	OUTPUT	low	Slave select output

Table15.1　SPI pins description

15.3 SPI HARDWARE DESCRIPTION

15.3.1 BLOCK DIAGRAM

When an SPI transfer occurs, an 8-bit character is shifted out on data pin while a different 8-bit character is simultaneously shifted in a second data pin. Another way to view this transfer is that an 8-bit shift register in the master and another 8-bit shift register in the slave are connected as a circular 16-bit shift register. When a transfer occurs, this distributed shift register is shifted eight bit positions; thus, the characters in the master and slave are effectively exchanged.

The central element in the SPI system is the block containing the shift register and the read data buffer. The system is single buffered in the transmit direction and double buffered in the receive direction. This fact means new data for transmission cannot be written to the shifter until the previous transaction is complete; however, received data is transferred into a parallel read data buffer so the shifter is free to accept a second serial character. As long as the first character is read out of the read data buffer before the next serial character is ready to be transferred, no overrun condition will occur.

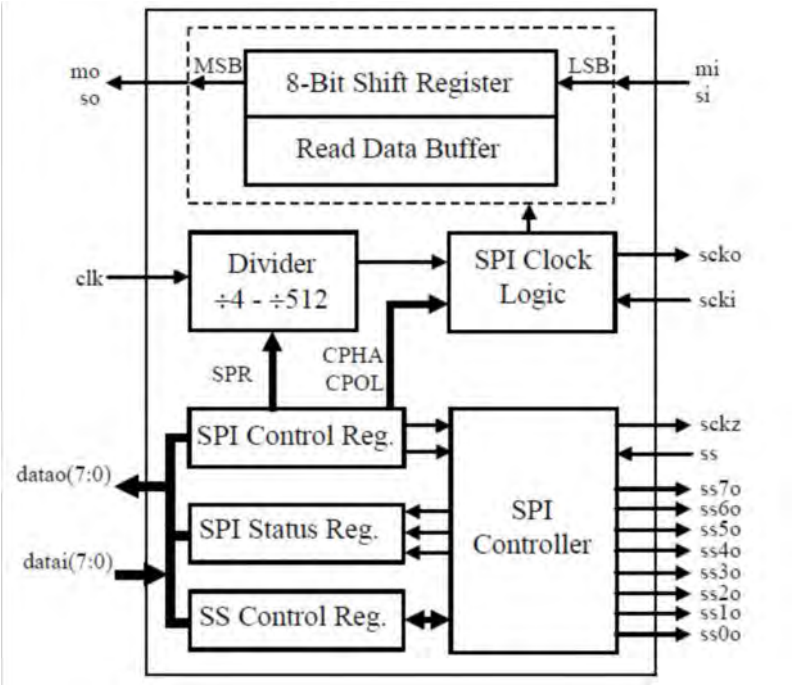


Figure 15.2　SPI Block Diagram

The eight pins are associated with the SPI: the SS, clock pins SCKI, SCKO and SCKEN, master pins MI and MO and slave pins SOEN, SI and SO.

The SS input pin in a master mode is used to detect mode-fault errors. A low on this pin indicates that some other device in a multi-master system has become a master and trying to select the SPI MODULE as a slave. The SS input pin in a slave mode is used to enable transfer.

The SCKI pin is used when the SPI is configured as a slave. The input clock from a master synchronizes data transfer between a master and the slave devices. The slave device ignore the SCKI signal unless the SS (slave select) pin is active low.

The SCKO and SCKEN pins are used as the SPI clock signal reference in a master mode. When the master initiates a transfer eight clock cycles is automatically generated on the SCKO pin.

When the SPI is configured as a slave the SI pin is the slave input data line, and the SO is the slave output data line.

When the SPI is configured as a master, the MI pin is the master input data line, and the MO is the master output data line.

15.3.2 INTERNAL REGISTERS

- **SPI Control Register**

The control register may be read or written at any time, is used to configure the SPI System.

SPCR (0xEC)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ECh EIE	R/W	SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0
Reset		0	0	0	0	0	1	0	0

SPIE：SPI interrupt enable
= 0, interrupts are disabled, polling mode is used
= 1, interrupts are enabled

SPE：SPI system enable
= 0, system is off
= 1, system is on

MSTR：Master/Slave mode select
= 0, slave
= 1, master

CPOL：Clock polarity select
= 0, high level; SCK idle low
= 1, low level; SCK idle high

CPHA：Clock phase.. Select one of two different transfer formats

SPR[2:0]：SPI clock rate select bits. See the table below

SPR2	SPR1	SPR0	System clock divided by
0	0	0	4
0	0	1	8
0	1	0	16
0	1	1	32
1	0	0	64
1	0	1	128
1	1	0	256
1	1	1	512

- **Slave Select Control Register**

The control register may be read or written at any time. It is used to configure which slave select output should be driven while SPI master transfer. Contents of SSCR register is automatically assigned on SS7O-SS0O pins when SPI master transmission starts.

SSCR (0xEF)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EFh SSCR	R/W	SS7	SS6	SS5	SS4	SS3	SS2	SS1	SS0
Reset		1	1	1	1	1	1	1	1

SS7 – SS0
= 0, Pin SSxO assigned while Master Transfer
= 1, Pin SSxO is forced to logic 1

- **SPI Status Register**

SPSR (0xED)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EDh EIE	R/W	SPIF	WCOL	-	MODF	-	-	-	SSCEN
Reset		0	0	0	0	0	1	0	0

SPIF：SPI interrupt request. The flag is automatically set to one at the end of an SPI transfer.
WCOL：Write collision error status flag. The flag is automatically set if the SPDR is written while a transfer is in process.
MODF：SPI mode-fault error status flag
This flag is set if SS pin goes to active low while the SPI is configured as a master (MSTR = 1)
SSCEN：
= 1, auto SS assertions enabled
= 0, auto SS assertions disabled – SSO always shows contents of SSCR

SPI status register (SPSR) contains flags indicating the completion of transfer or occurrence of system errors. All flags are set automatically when the corresponding event occur and cleared by software sequence. SPIF and WCOL are automatically cleared by reading SPSR followed by an access of the SPDR. MODF flag is cleared by reading SPSR with MODF set followed by a write to SPCR.

The SSCSEN bit is a enable bit of automatic Slave Select Outputs assertion. When SSCEN is set ('1') then during master transmission the SSXO lines are automatically loaded with contents of SSCR register before each byte transfer, and deasserted when byte is transferred. When SSCEN bit is cleared the SSXO lines always shows contents of the SSCR register, regardless of the transmission is in progress or SPI MODULE is in IDLE state.

- Receiver and Transmitter Registers

The Transmitted Data Register consists of eight data bits, which will be sending on the bus due the next Send operation. The first send bit is the D.7 (MSB).

SPDR (0xEE)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EEh SPDR	R/W	D.7	D.6	D.5	D.4	D.3	D.2	D.1	D.0
Reset		0	0	0	0	0	1	0	0

The Received Data Register consists of eight data bits, which were received on the bus due the last Receive operation.

SPDR (0xEE)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EEh SPDR	R/W	D.7	D.6	D.5	D.4	D.3	D.2	D.1	D.0
Reset		0	0	0	0	0	1	0	0

15.4 MASTER OPERATIONS

When the SPI MODULE core is configured as a SPI master, the transfer is initiated by write to the SPDR register. When the new byte is written to the SPDR register, SPI MODULE begins transfer on the nearest BAUD timer overflow. The serial clock SCK is generated by the SPI MODULE. In master mode the SPI MODULE activates the SCKEN to enable the SCK output driver.

The SPI MODULE in master mode can select one of the eight SPI slave devices, through the SSxO lines. The SSxO lines – Slave Select output lines are loaded with contents of the SSCR register (0x03). The SSCEN bit from the SPSR register select between automatic SSxO lines control and software control. When set the automatic Slave Select outputs assertion is enabled. With SSCEN bit set in master mode the SSXO lines are automatically loaded with contents of SSCR register before each byte transfer, and deasserted when byte is transferred. When SSCEN bit is cleared the SSXO lines are controlled by the software, and always shows contents of the SSCR register, regardless of the transmission is in progress or the SPI MODULE is in IDLE state.

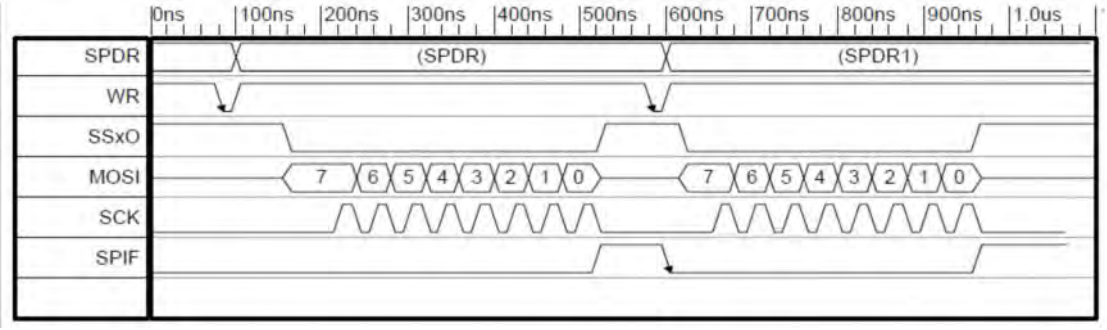


Figure15.3　Automatic slave select lines assertion

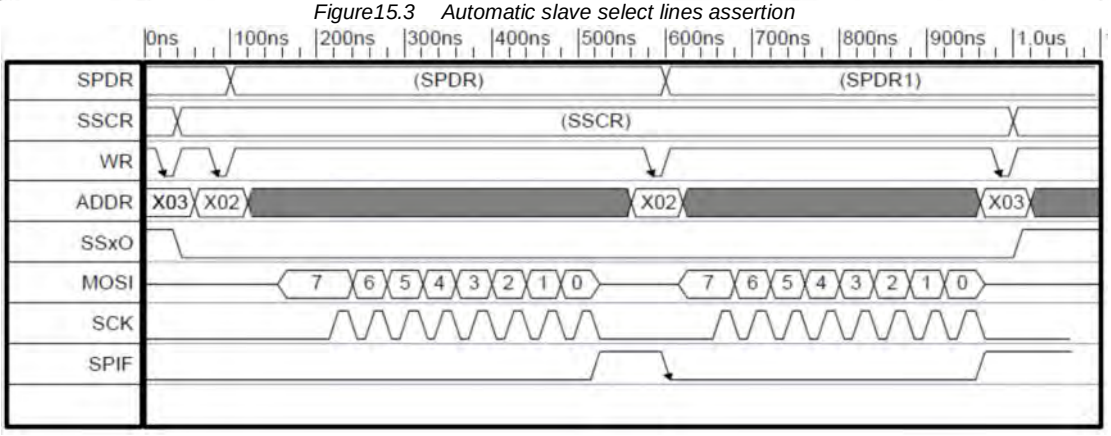


Figure15.4　Software controlled SSxO lines

15.4.1 MASTER MODE ERRORS

In master mode two system errors can be detected by the SPI MODULE. The first type of error arises in multiple-master system when more than one SPI device simultaneously tries to be a master. This error is called a Mode Fault. The second error type, a Write Collision, indicates that CPU tried to write the SPDR register while transfer was in progress.

◆ **MODE FAULT ERROR**

Mode fault error occur when the SPI MODULE is configured as a master and some other SPI master device will select this device as if it were a slave. If a Mode Fault Error occur：

- ✧ The MSTR bit is forced to zero to reconfigure the SPI MODULE as a slave.
- ✧ The SPE bit is forced to zero to disable the SPI MODULE system
- ✧ The MODF status flag is set and an interrupt request is generated

The MODF flag is cleared by reading SPSR with MODF set followed by a write to SPCR

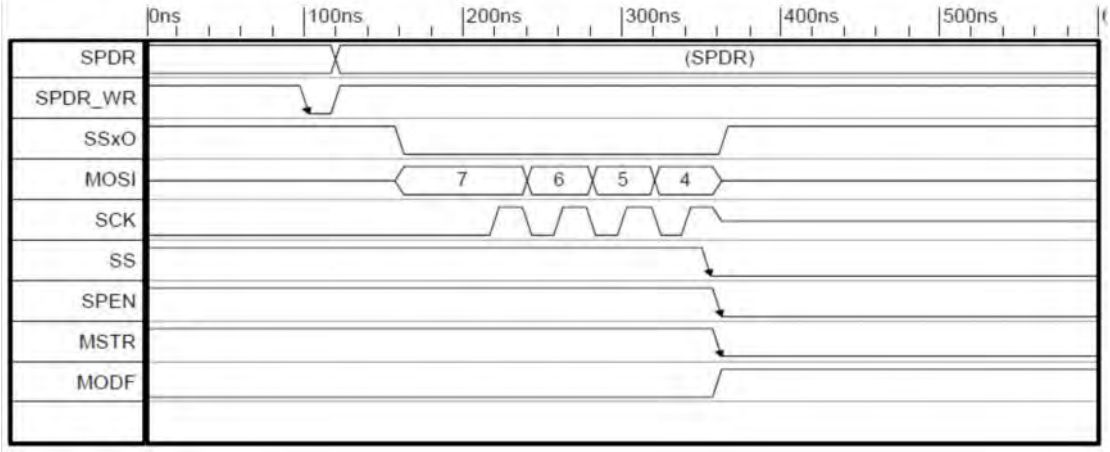


Figure15.5　Mode Fault Error generation

◆ **WRITE-COLLISION ERROR**

A write collision occurs if the SPI MODULE data register is written while a transfer is in progress. The transfer continues undisturbed, and the write data that caused the error is not written to the shifter. The Write Collision is indicated by the WCOL flag in SPSR (3) register.

The WCOL flag is set automatically by hardware, when the WCOL error condition occurs. To clear the WCOL bit, user should execute the following sequence:

- ✧ Read contents of the SPSR register
- ✧ Perform access to the SPDR register (read or write)

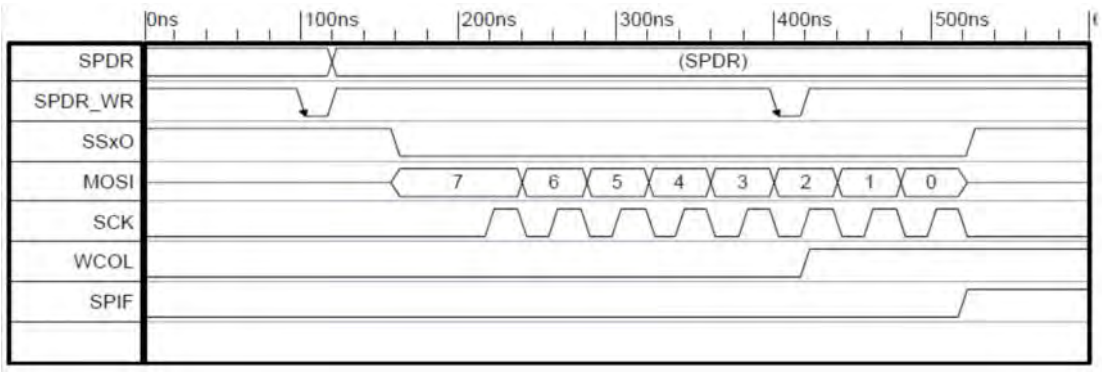


Figure15.6　Write Collision Error in SPI Master mode

15.5 SLAVE OPERATIONS

When configured as SPI Slave the SPI MODULE transfer is initiated by external SPI master module by assertion of the SPI MODULE Slave Select input, and generation of the SCK serial clock.

Before transfer starts, the SPI master has to assert the Slave Select line to determine which SPI slave will be used to exchange data. The SS is asserted (cleared = 0), the clock signal connected to the SXCK line will cause the SPI MODULE slave to shift into receiver shift register contents of the MOSI line, and drives the MISO line with contents of the Transmitter Shift register. When all eight bits are shifted in/out the SPI MODULE generates the Interrupt request by setting the IRQ output.

In SPI MODULE slave mode only one transfer error is possible – Write Collision Error.

15.5.1 SLAVE MODE ERRORS

In slave mode, only the Write Collision Error can be detected by the SPI MODULE.

The Write Collision Error occurs when the SPDR register write is performed while the SPI MODULE transfer is in progress.

In SLAVE mode when the CPHA is cleared, the write collision error may occur as long as the SS Slave Select line is driven low, even if all bits are already transferred. This is because there is not clearly specified the transfer beginning, and SS driven low after full byte transfer may indicate beginning of the next byte transfer.

◆ **WRITE-COLLISION ERROR**

A write collision occurs if the SPI MODULE data register is written while a transfer is in progress. The transfer continues undisturbed, and the write data that caused the error is not written to the shifter. The Write Collision is indicated by the WCOL flag in SPSR (3) register.

The WCOL flag is set automatically by hardware, when the WCOL error condition occurs. To clear the WCLO bit, user should execute the following sequence:

- ✧ Read contents of the SPSR register
- ✧ Perform access to the SPDR register (read or write)

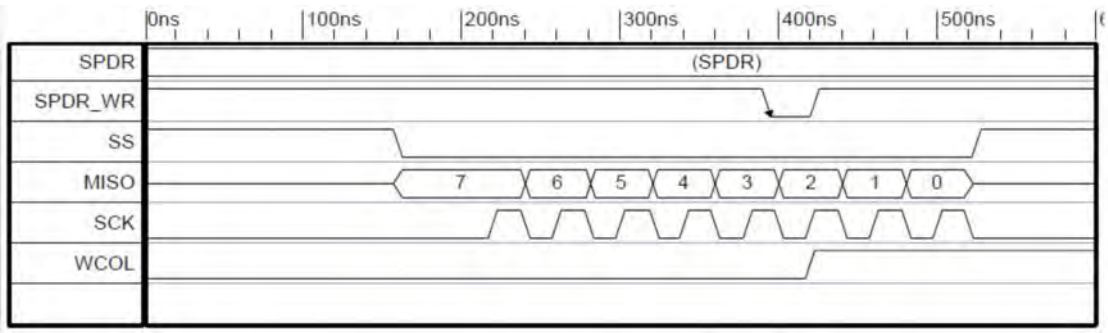


Figure15.7 Write Collision Error – SPI Slave mode – SPDR write during transfer

Figure below shows the WCOL generation, in case that the CPHA is cleared. As it is shown the WCOL generation is cause by any S{DR register write with SS line cleared. It is done even if the SPI master didn't generate the serial clock SCK. This is because there is not clearly specified the transfer beginning, and SS driven low after full byte transfer may indicate beginning of the next byte transfer.

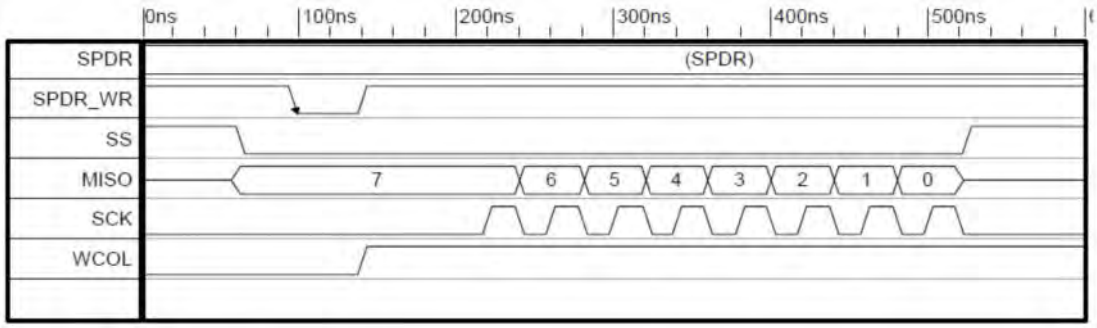


Figure15.8 WCOL Error-SPI Slave mode-SPDR write when CPHA = 0 and SS = 0

15.6 CLOCK CONTROL LOGIC

15.6.1 SPI CLOCK PHASE AND POLARITY CONTROLS

Software can select any of four combinations of serial clock (SCK) phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock and has no significant effect on the transfer format. The clock phase (CPHA) control bit selects one of two fundamentally different transfer formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transfers to allow a master device to communicate with peripheral slaves having different requirements. The flexibility of the SPI system on the SPI MODULE allows direct interface to almost any existing synchronous serial peripheral.

15.6.2 SPI MODULE TRANSFER FORMATS

During an SPI transfer, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). A serial clock line synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows individual selection of a slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. On a master SPI device, the slave select line can optionally be used to indicate a multiple-master bus contention.

15.6.3 CPHA EQUALS ZERO TRANSFER FORMAT

Figure below shows a timing diagram of an SPI transfer where CPHA is 0. Two waveforms are shown for SCK: one for CPOL equals 0 and another for CPOL equals 1. The diagram may be interpreted as a master or slave timing diagram since the SCK, master in/slave out (MISO), and master out/slave in (MOSI) pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The SS line is the slave select input to the slave; the SS pin of the master is not shown but is assumed to be inactive. The SS pin of the master must be high. This timing diagram functionally depicts how a transfer takes place; it should not be used as a replacement for data-sheet parametric information.

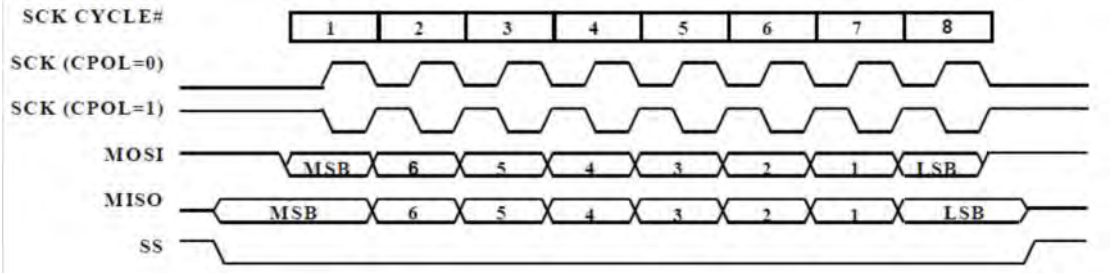


Figure15.9 CPHA Equals Zero SPI Transfer Format

When CPHA = 0, the SS line must be deasserted and reasserted between each successive serial byte. Also, if the slave writes data to the SPI data register (SPDR) while SS is active low, a write-collision error results. When CPHA = 1, the SS line may remain active low between successive transfers (can be tied low at all times). This format is sometimes preferred in systems having a single fixed master and a single slave driving the MISO data line.

15.6.4 CPHA EQUALS ONE TRANSFER FORMAT

Figure below is a timing diagram of an SPI transfer where CPHA = 1. Two waveforms are shown for SCK: one for CPOL = 0 and another for CPOL = 1. The diagram may be interpreted as a master or slave timing diagram since the SCK, MISO, and MOSI pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The SS line is the slave select input to the slave; the SS pin of the master is not shown but is assumed to be inactive. The SS pin of the master must be high or must be reconfigured as a general-purpose output not affecting the SPI.

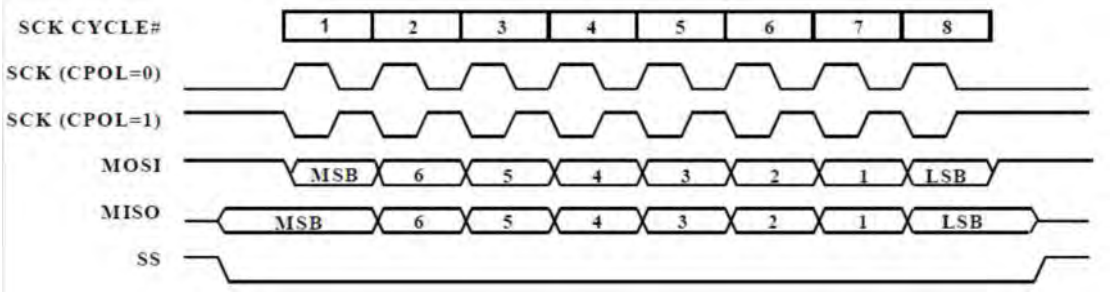


Figure15.10 CPHA Equals One SPI Transfer Format

15.7 SPI DATA TRANSFER

15.7.1 TRANSFER BEGINNING PERIOD (INITIATION DELAY)

All SPI transfers are started and controlled by a master SPI device. As a slave, the SPI MODULE considers a transfer to begin with the first SCK edge or the falling edge of SS, depending on the CPHA format selected. When CPHA = 0, the falling edge of SS indicates the beginning of a transfer. When CPHA = 1, the first edge on the SCK indicates the start of the transfer. In either CPHA format, a transfer can be aborted by taking the SS line high, which causes the SPI slave logic and bit counters to be reset. The SCK rate selected has no effect on slave operations since the clock from the master is controlling transfers.

When the SPI is configured as a master, transfers are started by a software write to the SPDR.

15.7.2 TRANSFER ENDING PERIOD

An SPI transfer is technically complete when the SPIF flag is set, but, depending on the configuration of the SPI system, there may be additional tasks. Because the SPI bit rate does not affect timing of the ending period, only the fastest rate is considered in discussions of the ending period. When the SPI is configured as a master, SPIF is set at the end of the eighth SCK cycle. When CPHA equals 1, SCK is inactive for the last half of the eighth SCK cycle.

When the SPI is operating as a slave, the ending period is different because the SCK line can be asynchronous to the MCU clocks of the slave and because the slave does not have access to as much information about SCK cycles as the master. For example, when CPHA = 1, where the last SCK edge occurs in the middle of the eighth SCK cycle, the slave has no way of knowing when the end of the last SCK cycle is. For these reasons, the slave considers the transfer complete after the last bit of serial data has been sampled, which corresponds to the middle of the eighth SCK cycle.

The SPIF flag is set at the end of a transfer, but the slave is not permitted to write new data to the SPDR while the SS line is still low.

15.8 TIMING DIAGRAMS

15.8.1 MASTER TRANSMISSION

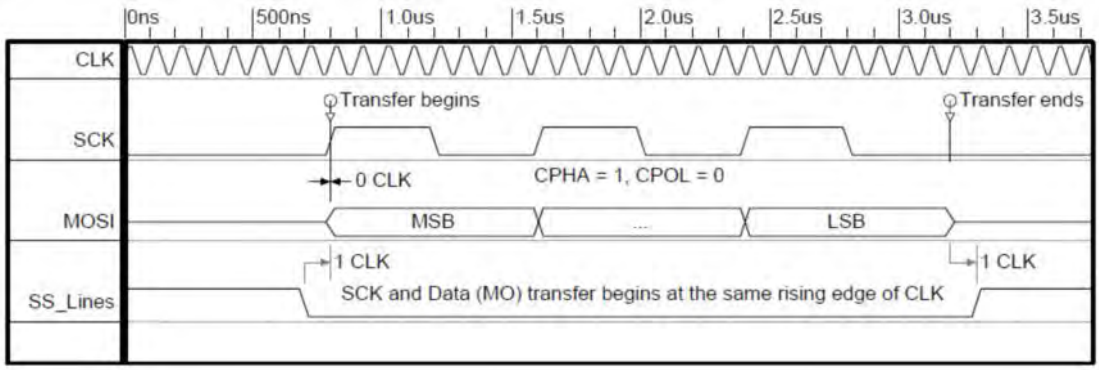


Figure15.11 Master mode timing diagram

15.8.2 SLAVE TRANSMISSION

At a beginning of transfer in Slave mode, the data on serial output (MISO) appears on first rising edge after falling edge on Slave Select (SS) line. Next bits of serial data are driving into MISO line on first rising edge of CLK after SKC active edge (in this case rising edge of SCK).

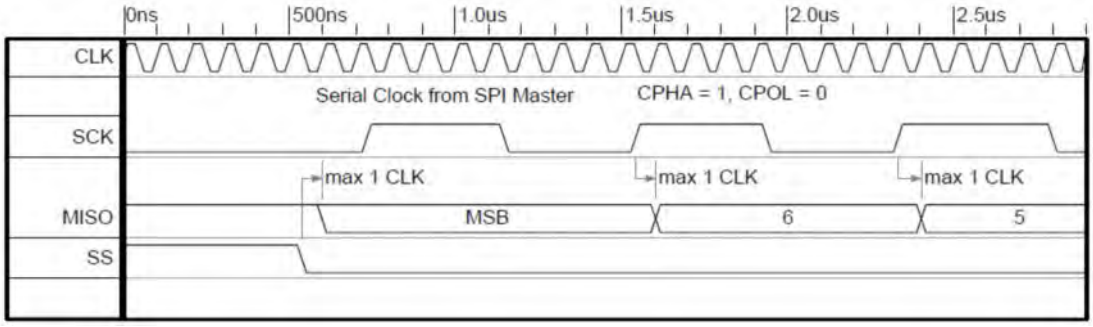


Figure15.12 Slave mode timing diagram

15.9 SPI MODULE INTERRUPT GENERATION

When interrupt is enabled (SPIE bit in SPCR=1), SPI interrupt flag is automatically asserted when SPI transfer is completed or transfer error has occurred. SPIIF flag has to be cleared by software.

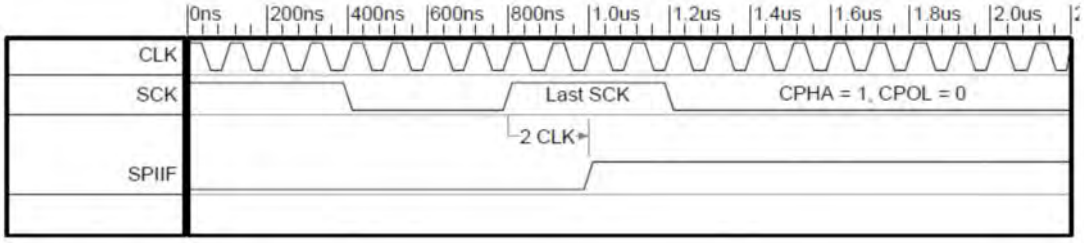


Figure15.13 Interrupt generation

Table15.2 SPI interrupt summary

Interrupt flag	Function	Active level/edge	Flag resets	Vector	Natural priority
SPIIF	Internal, SPI	-	Software	0x73	15

SPI related interrupt bits have been summarized below. The IE (0xA8) contains global interrupt system disable (0) / enable (1) bit called EA.

EIE (0xE8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
E8h EIE	R/W	EI2CS ESPI	EI2CM	EWDI	EINT6	EINT5	EINT4	EINT3	EINT2
Reset		0	0	0	0	0	0	0	0

ESPI：Enable SPI Interrupts

EIP (0xF8)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F8h EIP	R/W	PI2CS PSPI	PI2CM	PWDI	PINT6	PINT5	PINT4	PINT3	PINT2
Reset		0	0	0	0	0	0	0	0

PSPI：SPI priority level control (at 1-high-level)

EIF　(0x91)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
91h EIF	R/W	I2CSF SPIF	I2CMF	-	INT6F	INT5F	INT4F	INT3F	INT2F
Reset		0	0	0	0	0	0	0	0

SPIIF：SPI interrupt flag
Must be cleared by software

16. PWM

A9108 has two channels Pulse width modulator (PWM) output. Every channel PWM has an 8-bit counter with comparator, a control register (PWMxCON) and two setting registers (PWMxH and PWMxL). User can select clock source by setting PWMxCON. Enable PWM output and function by setting PWMxEN = 1; otherwise disable PWM output and function by setting PWMxEN =0. When user set PWMxEN=0, it output LOW single and reload the PWMxL to itself. When the counter is enabled and matches the content of PWMxH, its output is asserted HIGH; when the counter is overflow, its output is asserted LOW and reload PWMxL to itself. The pulse frequency and the duty cycle for 8-bit PWM is given by the below equation

$$\text{Pulse frequency} = \text{System clock} / 2^{pwxclk+1} / (255\text{-PWMxL})$$
$$\text{Duty cycle} = (255\text{-PWMxH}) / 255\text{-PWMxL})$$

Noted: PWMxH must be larger then PWMxL. Otherwise, PWM output always is LOW.

16.1 PWM FUNCTIONALITY

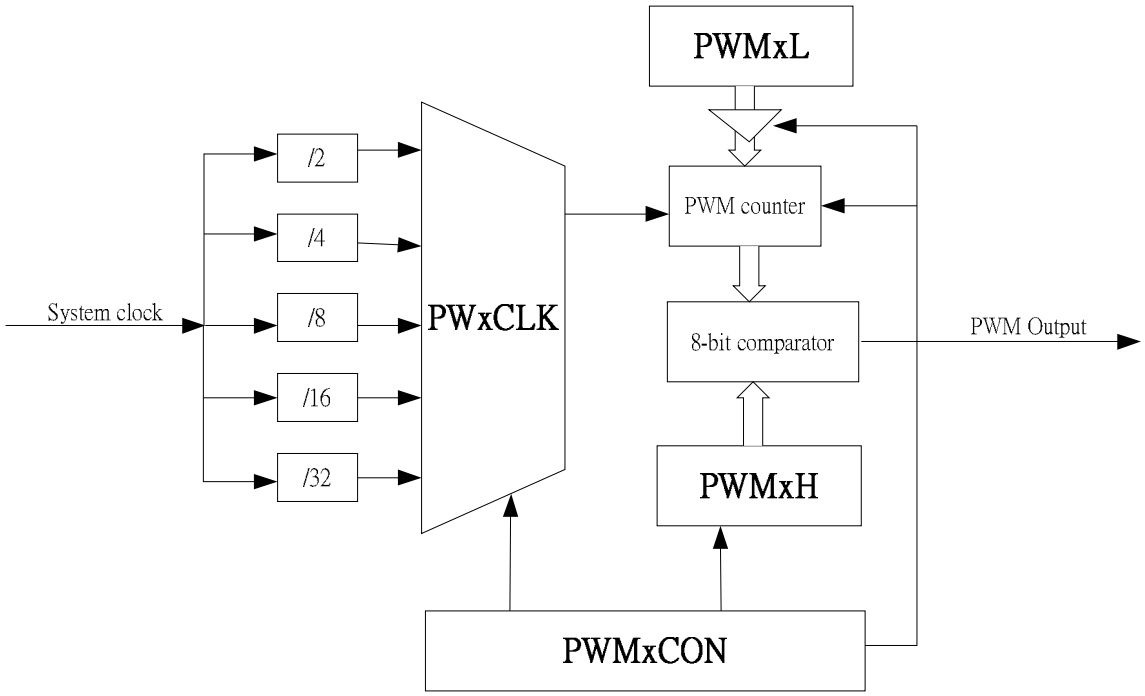


Figure16.1 PWM Block Digram

The PWM pins functionality is described in the following table. All pins are one directional.

PIN	ACTIVE	TYPE	DESCRIPTION
PWM0(P3.6)		OUTPUT	PWM 0 output
PWM1(P3.7)		OUTPUT	PWM 1 output

Table16.1 PWM PIN define

16.1.1 PWM Registers

PWM0/1 is new design from AMICCOM. They can output pulse width modulation. User adjusts to duty cycle by setting PWMxH. PWM counter is up counter. PWM counter is not access directly by MCU. User can set or reset PWM counter by setting PWMxCON. When PWMxEN =1, PWM counter start to count. When PWMxEN=0, PWM counter stop counting and reload PWMxL to itself. PWxCLK is clock divider. It divide system clock to 2,4,8,16 and 32 by setting PWxCLK.

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A9h PWM0CON	R/W	PWM0EN	-	-	-	-	PW0CLK2	PW0CLK1	PW0CLK0

Reset		0	0	0	0	0	0	0	0

PWM0CON: PWM channel 0 control register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AAh PWM0H Reset	R/W	0	0	0	0	0	0	0	0

PWM0H: PWM channel 0 output HIGH register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ABh PWM0L Reset	R/W	0	0	0	0	0	0	0	0

PWM0L: PWM channel 0 frequency setting register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B0h PWM1CON Reset	R/W	PWM1EN 0	- 0	- 0	- 0	- 0	PW1CLK2 0	PW1CLK1 0	PW1CLK0 0

PWM1CON: PWM channel 1 control register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B1h PWM1H Reset	R/W	0	0	0	0	0	0	0	0

PWM1H: PWM channel 1 output HIGH register

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B2h PWM1L Reset	R/W	0	0	0	0	0	0	0	0

PWM1L: PWM channel 1 frequency setting register

17. ADC (Analog to Digital Converter)

A9108 has three built-in ADC. One is 8-bits ADC that do RSSI measurement as well as carrier detection function. THe ADC clock (F_{ADC}) is 4MHz. The ADC converting time is 20 x ADC clock periods. Another is 4 channel 12bit SAR ADC for general purpose use to measure the external analog signal. The other is 24bit ADC for measure thermal resister.

Bit		Mode	
XADS	RSS	Standby	RX
0	1	None	RSSI / Carrier detect

Table 17.1 Setting of ADC function

Relative Control Register

Mode Control Register (Address: 0802h)

Bit	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	R	DDPC	ARSSI	AIF	CD	WWSE	FMT	FMS	ADCM
	W	DDPC	ARSSI	AIF	DFCD	WWSE	FMT	FMS	ADCM
Reset		0	0	0	0	0	0	0	0

ADC Register (Address: 0821h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSSI Threshold	R	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
	W	RTH7	RTH6	RTH5	RTH4	RTH3	RTH2	RTH1	RTH0
Reset		1	0	0	1	0	0	0	1

ADC Control Register (Address: 0822h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADC Control	W	RSM1	RSM0	ERSS	FSARS	--	XADS	RSS	CDM
Reset		0	1	0	1	--	0	1	1

17.1 RSSI Measurement

A9108 supports 8-bits digital RSSI to detect RF signal strength. RSSI value is stored in ADC [7:0] (1Dh). Fig 17.1 shows a typical plot of RSSI reading as a function of input power. This curve is base on the current gain setting of A9108 reference code. A9108 automatically averages 8-times ADC conversion a RSSI measurement until A9108 exits RX mode. Therefore, each RSSI measuring time is (8 x 20 x F_{ADC}). Be aware RSSI accuracy is about ± 6dBm.

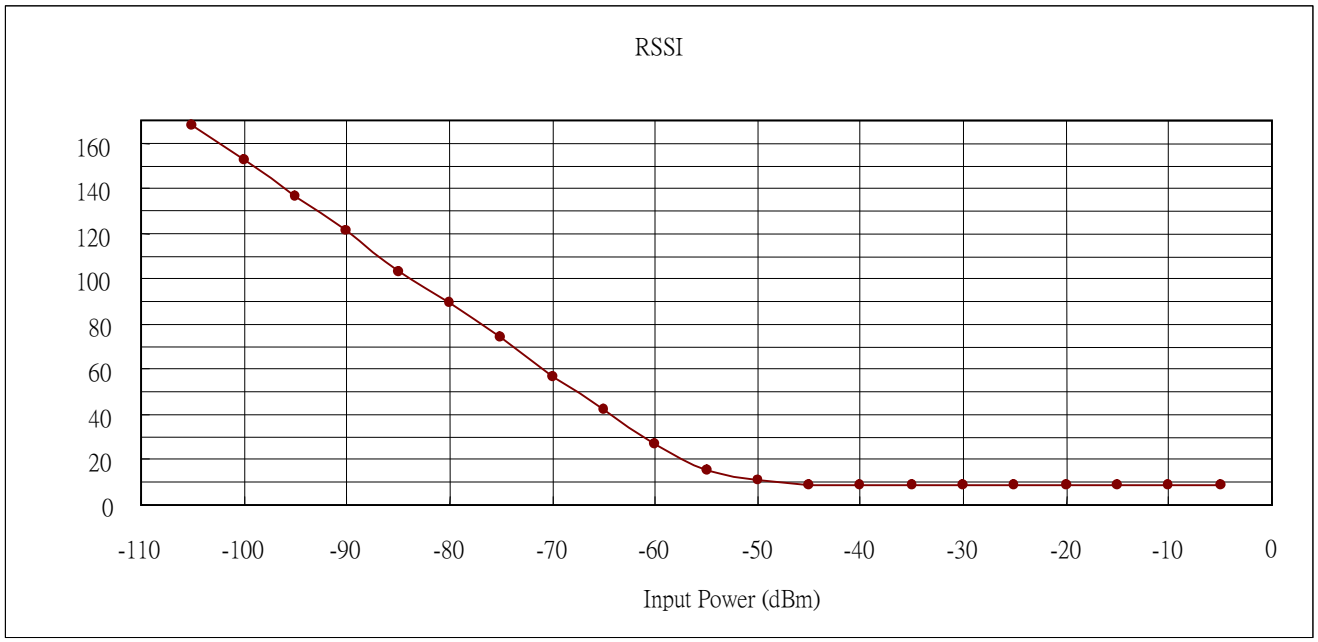
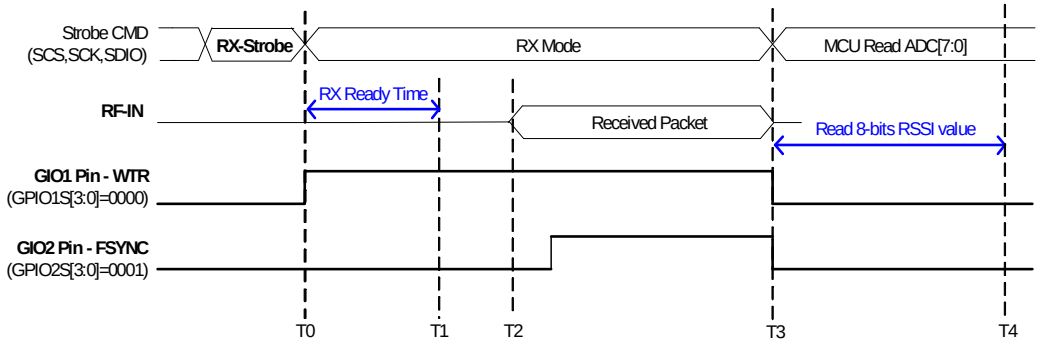


Figure 17.1 Typical RSSI characteristic.

Auto RSSI measurement for TX Power:

1. Set wanted F_{RXLO} (Refer to chapter 14).
2. Set RSS= 1 (1Eh), FSARS= 0 (1Eh, 4MHz ADC clock).
3. Enable ARSSI= 1 (01h).
4. Send RX Strobe command.
5. In RX mode, 8-times average a RSSI measurement periodically.
6. Exit RX mode, user can read digital RSSI value from ADC [7:0] (1Dh) for TX power.

In step 6, if A9108 is set in direct mode, MCU shall let A9108 exit RX mode within 40 us to prevent RSSI inaccuracy.



T0-T1: Settling Time
T2-T3: Receiving Packet
T3 : Exit RX mode automatically in FIFO mode
T3-T4: MCU read RSSI value @ ADC [7:0]

Figure 17.2 RSSI Measurement of TX Power.

Auto RSSI measurement for Background Power:

1. Set wanted F_{RXLO} (Refer to chapter 14).
2. Set RSS= 1 (1Eh), FSARS= 0 (1Eh, 4MHz ADC clock).
3. Enable ARSSI= 1 (01h).

4. Send RX Strobe command.
5. MCU delays min. 140us.
6. Read digital RSSI value from ADC [7:0] (1Dh) to get background power.
7. Send other Strobe command to let A9108 exit RX mode.

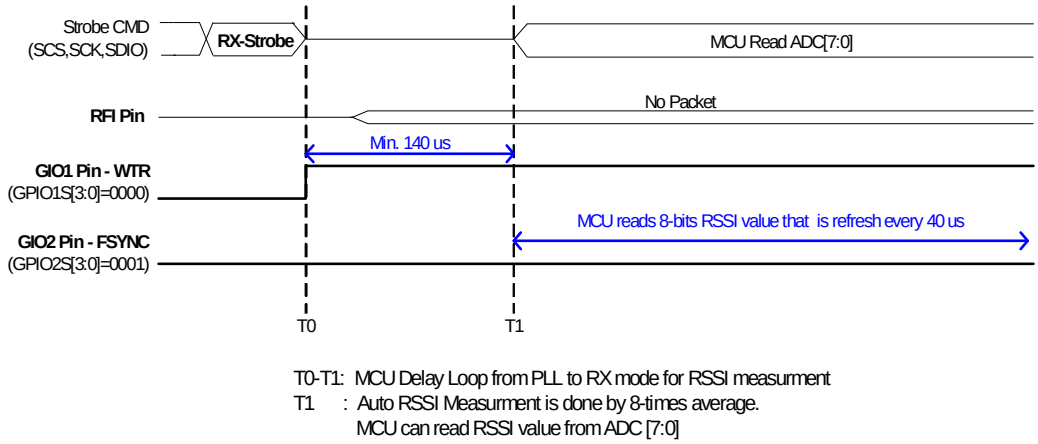


Figure 17.3 RSSI Measurement of Background Power.

17.2 Carrier Detect

Base on RSSI measurement, user can extend its application to do carrier detect (CD). In Carrier Detect mode, RSSI is refresh every 5 us without 8-times average. If RSSI level is below threshold level (RTH), CD is output high to GIO1 or GIO2 pin to inform MCU that current channel is busy.

Below is a reference procedure:

1. Set CDTH (0821h) for absolute RSSI threshold level (ex. RTH = 80d).
2. Set GIO2S = [0010] (080Eh) for Carrier Detect to GIO2 pin.
   (2-1) Set wanted F_{RXLO} (Refer to chapter 14).
   (2-2) Set RSM= [11] (0822h, CDM =0 and hysteresis =6, or CDM =1 and hysteresis =12).
   (2-3) Enable ARSSI= 1 (01h).
   (2-4) Send RX Strobe command.
   (2-5) MCU enables a timer delay (min. 100 us).
3. MCU checks GIO2 pin.
   (3-1) If ADC $\geqq$ CDTH, GIO2 = 0.
   (3-2) If ADC $\leqq$ CDTH-CDM, GIO2 = 1.
   (3-3) If ADC locates in hysteresis zone, GIO2 = previouse state.
4. Exit RX mode.

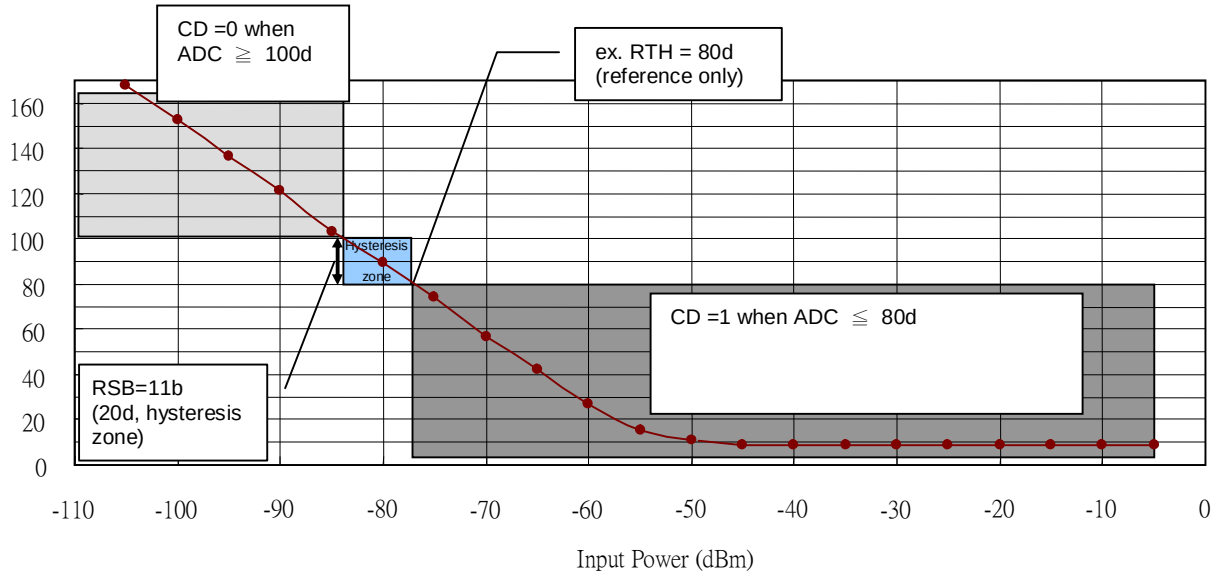


Figure 17.4 Carrier Detect Zone, a reference setting only.

17.3 Battery Detect

A9108 has a built-in battery detector to check supply voltage (REGI pin). The detecting range is 2.0V ~ 2.7V in 8 levels.

Relative Control Register

Battery detect Register (Address: 082Ch)

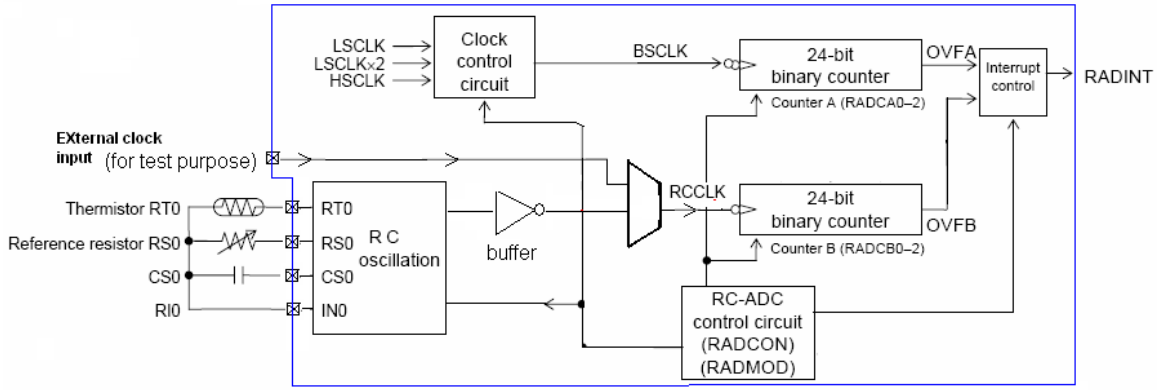
Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Battery detect	R	--	--	--	--	--	--	--	BDF
	W	ATP1	ATP0	QDS	BLE	BDS1	BDS0	BGS	BDE
Reset		0	0	0	--	0	1	1	0

BDS[1:0]: Battery detection threshold.
[00]: 2.0V. **[01]:** 2.2V. [**10]:** 2.4V. **[11]:** 2.6V.
When REGI < Threshold, BDF= low.
When REGI > Threshold, BDF= high.

Below is the procedure to detect low voltage input (ex. below 2.1V):

1. Set A9108 in standby or PLL mode.
2. Set BDS[1:0] (082Ch) = [001] and enable BDE (082Ch) = 1.
3. After 5 us, BDE is auto clear.
4. MCU reads BDF (082Ch).
   If REGI pin > 2.1V,
   BDF = 1 (battery high). Else, BDF = 0 (battery low).

18. 24bit ADC


LSCLK
LSCLK×2
HSCLK
Clock control circuit
BSCLK
24-bit binary counter
Counter A (RADCA0–2)
OVFA
Interrupt control
RADINT
External clock input (for test purpose)
Thermistor RT0
Reference resistor RS0
CS0
RI0
RT0
RS0
CS0
IN0
R C oscillation
buffer
RCCLK
24-bit binary counter
Counter B (RADCB0–2)
OVFB
RC-ADC control circuit (RADCON) (RADMOD)

19 Power Management

Low power operation is enabled through different power modes setting. A9108 has various operating mode are referred as normal mode, low power mode and ultra low power mode, power down and Stop mode. Table 19.1 shows the impact of different power modes on systems operation.

Table 19.1 Power mode

	CPU speed	16MHz	Internal RC	RTC	Back to Normal	LVR	RF
Normal	16MHz	V	V	V	X	V	ALL
Low Power	8/4/2/1 MHz IRC/RTC	V	V	V	Interrupt	V	ALL
Ultra low power	X	X	V	V	Interrupt Key/ Sleep timer	V	WOR/Sleep
Power down	X	X	X	X	Interrupt KEY	V*	Sleep
Stop	X	X	X	X	Reset Key	X	Sleep

Table 10.4 Power manager

	CPU speed	16MHz	Internal RC	RTC	RAM	Back to Normal	LVR	RF
Normal	16MHz	V	V	V	v	X	V	ALL
PMM	8/4/2/1 MHz IRC/RTC	V	V	V	v	Interrupt / mode switch	V	ALL
Idle (PM1)	X	X	V	V	v	H/W reset / wakeup key / Interrupt Key / Sleep timer	V	WOR/Sleep
Sleep (PM2)	X	X	X	X	v	H/W reset / wakeup key / Interrupt KEY	V*	Sleep
Deep Sleep (PM3)	X	X	X	X	?	Reset Key Reset	X	Sleep

There are two register to setting power manager. One is power manager control (PCON, 0x

20 A9108 RF

A9108 integrate Sub1GHz FSK/GFSK Sigma-delta modulation transceiver and use Strobe control register (0800h) to control RF state. There are 6 Strobe commands to control internal state machine for RF operations. These modes include Sleep mode, Idle mode, Standby mode, PLL mode, RX mode and TX mode. There are two 64Bytes FIFO for data transmitting and receiving. Sleep timer is used for WOR (Wake On Rx) and time-slotted mode operation.

20.1 Strobe Command

Strobe Control Register (Address: 0800h)

Bit	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	R					FPEN	RFSTATE2	RFSTATE1	RFSTATE0
	W	Strobe3	Strobe2	Strobe1	Strobe0			--	--
Write Reset Value		0	0	0	0	0	0	0	0

Use strobe command control RF state.
Strobe[3:0] is strobe command register.
Strobe[3:0] = 4'b1000: Sleep mode.
Strobe[3:0] = 4'b1001: Idle mode.
Strobe[3:0] = 4'b1010: Standby .
Strobe[3:0] = 4'b1011: PLL mode.
Strobe[3:0] = 4'b1100: RX mode
Strobe[3:0] = 4'b1101: TX mode

RFSTATE[2:0] is RF state flag.
RFSTATE[2:0] = 3'b000: Sleep mode.
RFSTATE[2:0] = 3'b001: Idle mode.
RFSTATE[2:0] = 3'b010: standby mode.
RFSTATE[2:0] = 3'b011: PLL mode.
RFSTATE[2:0] = 3'b100: RX mode
RFSTATE[2:0] = 3'b101: TX mode

20.1.1 Strobe Command - Sleep Mode

Refer to Strobe Control Register, user can write 0x80 to Strobe Control Register directly to set RF into Sleep mode.

20.1.2 Strobe Command - Idle Mode

Refer to Strobe Control Register, user can write 0x90 to Strobe Control Register directly to set RF into Idle mode.

20.1.3 Strobe Command - Standby Mode

Refer to Strobe Control Register, user can write 0xA0 to Strobe Control Register directly to set RF into Standby mode.

20.1.4 Strobe Command - PLL Mode

Refer to Strobe Control Register, user can write 0xB0 to Strobe Control Register directly to set RF into PLL mode.

20.1.5 Strobe Command - RX Mode

Refer to Strobe Control Register, user can write 0xC0 to Strobe Control Register directly to set RF into RX mode.

20.1.6 Strobe Command - TX Mode

Refer to Strobe Control Register, user can write 0xD0 to Strobe Control Register directly to set RF into TX mode.

20.2 RF Reset Command

In addition to power on reset (POR), A9108 could issue software reset (80h)to RF by setting Mode Register (0801h). A9108 generates an internal signal "RESETN" to initial RF circuit. After reset command, RF state is in standby mode and re-calibration is necessary.

20.3 FIFO Accessing Command

Before TX delivery, user only needs to write wanted data into TX FIFO in advance. Similarly, user can read RX FIFO once payload data is received. It is easy to delivery data to air. Below is the procedure of writing TX FIFO.

Step1: Send (n+1) bytes TX data in sequence by Data Byte 0, 1, 2 to n.

Step2: Send TX Strobe command for transmitting.

There are similar steps to read RX FIFO.

Step1: Send RX Strobe command for receiving data.
Step2: Read RX data from RX FIFO in sequence by Data Byte 0, 1, 2 to n.

21. Flash memory controller

1.2. SFR RELATED REGISTERS

FLASH memory is controlled using PCON(0x87)'s PWE bit, FLASHCTRL(0x9A) and FLSHTMR(0x9B). An SFR register named FLASHCTRL (0x9A) is used to control communication between CPU and flash. FLSHCTRL(0x9A) is consisted of 6 bits used to control all FLASH related operations. Lower five bits of FLSHTMR(0x9B) named FREQ[4:0] determine real CLK frequency with 1 MHz step resolution. FREQ[4:0] after reset is set to 20 MHz by default, provides optimal timing for flash macro.

FLSHCTRL (0x9A)

7	6	5	4	3	2	1	0	Reset
-	-	CTRL.5	CTRL.4	CTRL.3	CTRL.2	CTRL.1	CTRL.0	0x00

Figure 3. FLSHCTRL register

FLSHTMR (0x9B)

7	6	5	4	3	2	1	0	Reset
0	0	0	FREQ.4	FREQ.3	FREQ.2	FREQ.1	FREQ.0	0x00

Figure 4. FLSHTMR register

FREQ[4:0]	Frequency MHz
0x00	-
0x01	1
0x02	2
...	...
0x14	20

Table 3. FREQ intervals

Setting higher clock frequency is not supported since given flash has limited its clock frequency up to 20 MHz by Tkp read cycle time. FLASHCTRL register is write protected by TA enable procedure listed below:

```
  CLR  EA ;disable interrupt system
MOV TA, #0xAA
MOV TA, #0x55
MOV FLASHCTRL,#<value> ; Any direct addressing instruction writing FLASHCTRL register.
  SETB  EA  ;Enable interrupt system
```

When using DoCD debugger to control FLASH chip then TA enable procedure can be skipped, because DoCD debugger has direct access to FLASHCTRL. Reading from FLASHCTRL register is not protected. Setting of CERASE and SERASE bits at the same time is impossible. Such combination ("11") has no effect on both bits.

The Program Write Enable (PWE) bit, located in PCON register, is used to enable/disable PRGROMWR and PRGRRAMWR pin activity during MOVX instructions.

PCON (0x87)

7	6	5	4	3	2	1	0	Reset
SMOD0	SMOD1	-	PWE	-	SWB	STOP	PMM	0x00

Figure 4. PCON register – PWE bit

When PWE bit is set to logic 1, the MOVX @DPTR,A instruction writes data located in accumulator register into Program Memory addressed by DPTR register (active DPX:DPH:DPL). The MOVX @Rx,A instruction writes data located in accumulator register into program memory addressed by MXAX (bits 23:16), P2 register (bits 15:8) and Rx register (bits 7:0). Program Memory can be read by MOVC only regardless of PWE bit.

1.4.1. CHIP ERASE OPERATION

Chip erase operation is enabled by setting CTRL[5:0]=0x04 of FLSHCTRL register according to CPU TA enable procedure or direct write by DoCD. PCON.PWE bit must be set too, then first MOVX/DoCD instruction writing to program memory space at address belong to certain FLASH macro begins sector erase operation. During erase operation CPU is halted by asserting FLASHBUSY pin. When FLASH macro has been erased, FLASHBUSY pin is deactivated and FNOP is automatically written. CPU executes next instruction. CMT FLASH macro is blank and ready for new programming. To erase another FLASH macro the whole procedure needs to be repeated with changed MOVX/DoCD address pointing to certain FLASH macro.

Preprograming of whole FLASH macro is executed automatically without any interaction with user, before real chip erase. It extends lifecycle of CMT FLASH macro.

1.4.2. SECTOR ERASE OPERATION

The 16kB CMT FLASH macro has 128 sectors (128B each) which can be erased separately. Sector erase operation is enabled by setting CTRL[5:0]=0x02 of FLSHCTRL register according to CPU TA enable procedure or direct write by DoCD. PCON.PWE bit must be set too, then first MOVX/DoCD instruction writing to program memory space at selected sector address begins sector erase operation. During sector erase operation CPU is halted by asserting FLASHBUSY pin. When sector has been erased FLASHBUSY pin is deactivated and FNOP is automatically written. CPU executes next instruction. Selected CMT FLASH macro sector(s) is blank and ready for new programming. To erase another sectors whole procedure needs to be repeated.

Preprograming of whole sector is executed automatically without any interaction with user, before real sector erase. It extends lifecycle of CMT FLASH macro.

1.4.3. PROGRAM OPERATION

Word program operation is enabled by setting CTRL[5:0]=0x01 of FLSHCTRL register according to CPU TA enable procedure or direct write by DoCD. PCON.PWE bit must be set too, then each write to program memory space by MOVX/DoCD instruction addressing odd byte begins word program operation. During program operation CPU is halted by asserting FLASHBUSY pin. When word has been programmed FLASHBUSY pin is deactivated. CPU/DoCD executes next instruction which can be (i) programming of next memory word or (ii) CTRL[5:0]=0x00 according to CPU TA enable procedure or direct FNOP write by DoCD. Number of programmed bytes must be always even number (2,4,6,…). For example to program byte at address 0x003, first must be written byte at address 0x002 then second MOVX/DoCD instruction write at address 0x003 begins physical write to CMT FLASH macro. When number of programmed bytes is not even then it must be filled with extra neutral byte – for CMT FLASH macro it is 0x00. The neutral byte doesn't program any bit in a FLASH macro.

22 In Circuit Emulator (ICE)

A9108 support In Circuit Emulator on chip. It is a real-time hardware debugger as a non-intrusive system. It doesn't need to occupy any hardware resource such as the UART and Timer. User develops firmware complete producing code without any modification using ICE. It helps user to track down hidden bugs within the application running with microcontroller. The ICE with Hardware USB dongle provides a powerful SOC development tool with silicon using 2-wire protocol. The ICE fully supports Keil uVision2/3/4 interface to hardware debuggers. It allows Keil software user to work with uvision2/3/4. For more detail information, please reference Application note.

22.2 PIN define

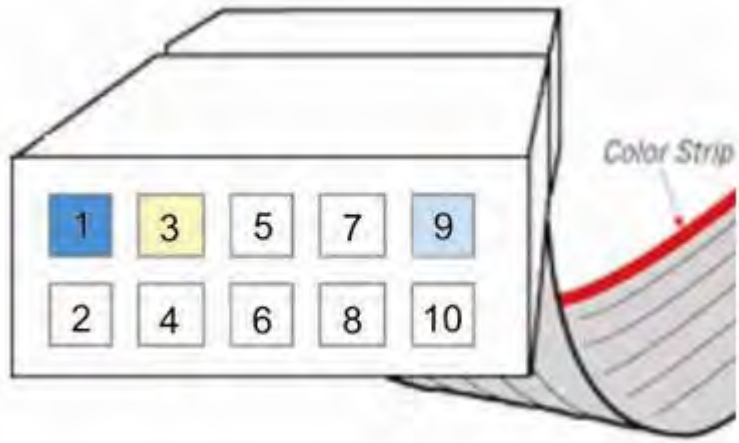


Fig 22.1 The USB connectors

Pin	Signal name	Description	Pin	Signal name	Description
1	ttck	Clock signal (in)	2	GND	Signal Ground
3	ttdio	Data (io)	4	VCCIO	Used to VCCIO detection
5	NU	Do not use	6	NU	Do not use or connect
7	NU	Do not use	8	NU	Do not use or connect
9	rsto	Reset output (od)	10	GND	Signal Ground

Fig22.2 The Pin define within USB connector

Note: RSTO pin is open drain (od) type active low. It forces logic zero to issue reset. When RSTO is inactive its output is floating, and should be connected to global system reset with pull-up resistor. This pin can be left unconnected.

There are 10 pin in the ICE connectors. 2-wire ICE only use 2 pins (PIN1 and PIN3). The PIN9 is optional and it can connects reset signal. PIN2 and PIN10 are GND pin. PIN4 is VCCIO pin. The recommended circuit shows as the below figure. (Fig21.3). There is a resister (100 ohm) between A8510 and pin connected the connector.

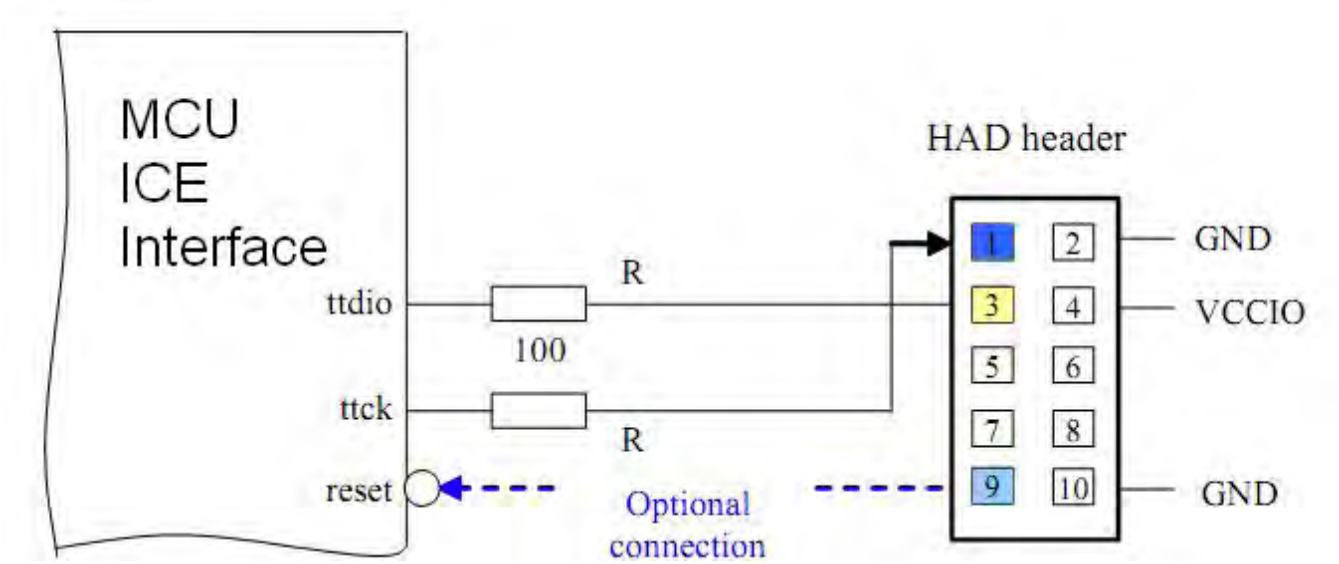


Fig 22.3 The connections between A9108 and USB connectors

22.2 ICE Key feature

The ICE supports source level debugging, 2 hardware breakpoint, auto refresh of all register and In system programming (ISP). User can use ICE to download firmware by Keil software or AMICCOM tool.

23. Application circuit

Below are AMICCOM's ref. design module, MD8501, circuit example and its PCB layout.

24. Abbreviations

ADC	Analog to Digital Converter
AIF	Auto IF
FC	Frequency Compensation
AGC	Automatic Gain Control
BER	Bit Error Rate
BW	Bandwidth
CD	Carrier Detect
CHSP	Channel Step
CRC	Cyclic Redundancy Check
DC	Direct Current
FEC	Forward Error Correction
FIFO	First in First out
FSK	Frequency Shift Keying
ID	Identifier
ICE	In Circuit Emulator
IF	Intermediate Frequency
ISM	Industrial, Scientific and Medical
LO	Local Oscillator
MCU	Micro Controller Unit
PFD	Phase Frequency Detector for PLL
PLL	Phase Lock Loop
POR	Power on Reset
PWM	Pulse width modulation
RX	Receiver
RXLO	Receiver Local Oscillator
RSSI	Received Signal Strength Indicator
SPI	Serial to Parallel Interface
SYCK	System Clock for digital circuit
TX	Transmitter
TXRF	Transmitter Radio Frequency
VCO	Voltage Controlled Oscillator
XOSC	Crystal Oscillator
XREF	Crystal Reference frequency
XTAL	Crystal

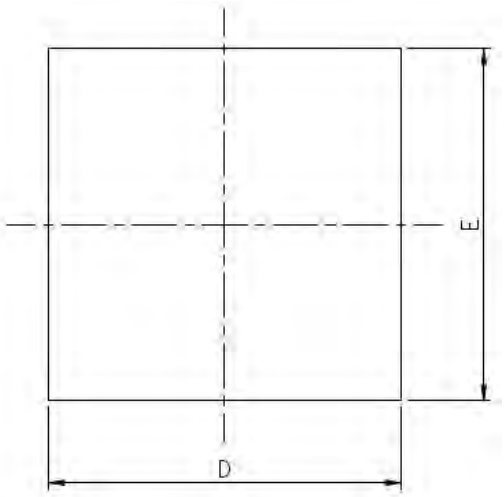
25. Ordering Information

Part No.	Package	Units Per Reel / Tray
A91X08AQCI/Q	QFN48L, Pb Free, Tape & Reel, -40℃～85℃	3K
A91X08AQCI	QFN48L, Pb Free, Tray, -40℃～85℃	490EA
A91X08AH	Die form, -40℃～85℃	100EA

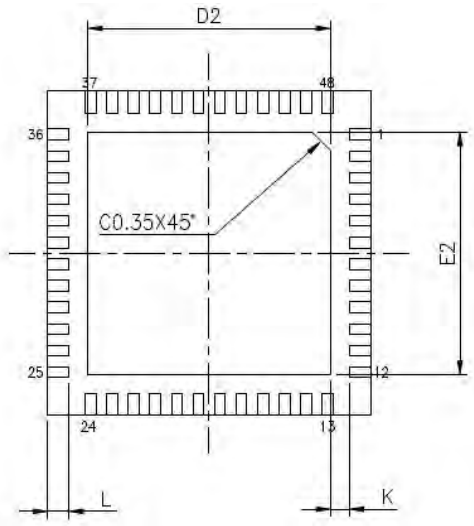
26. Package Information

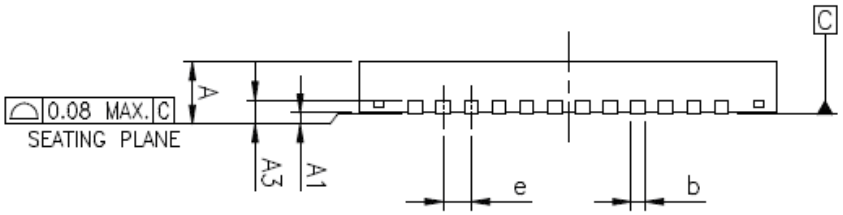
QFN 48L (6 X 6 X 0.75mm) Outline Dimensions

Top View



Bottom View

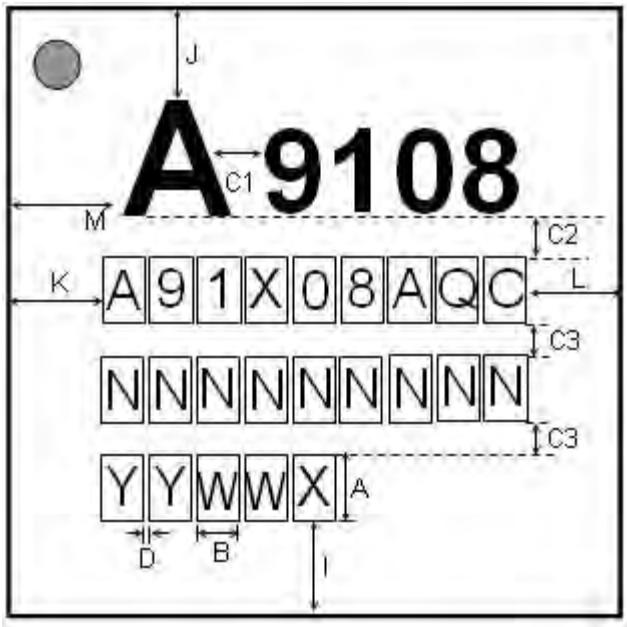




Symbol	Dimensions in inches			Dimensions in mm		
	Min	Nom	Max	Min	Nom	Max
A	0.028	0.030	0.031	0.7	0.75	0.8
A_1	0	0.001	0.002	0.00	0.02	0.05
A_3		0.009 REF.			0.23REF.	
b	0.006	0.008	0.010	0.15	0.2	0.25
D		0.240			6.1BSC	
D_2	0.146	0.177	0.179	3.70	4.50	4.55
E		0.240			6.1BSC	
E_2	0.146	0.177	0.179	3.70	4.50	4.55
e		0.016BSC			0.4BSC	
L	0.013	0.0160.019		0.32	0.4	0.48
k		0.008			0.2	

27. Top Marking Information

- Part No. : **A91X08AQCI**
- Pin Count : **48**
- Package Type : **QFN**
- Dimension : **6*6 mm**
- Mark Method : **Laser Mark**
- Character Type : **Arial**



❖ CHARACTER SIZE : (Unit in mm)

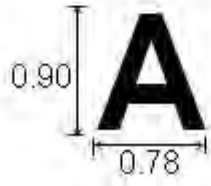
A : 0.65
B : 0.45
C1 : 0.3　C2 : 0.4　C3 : 0.3
D : 0.03
M : 1.5

YYWW : DATECODE

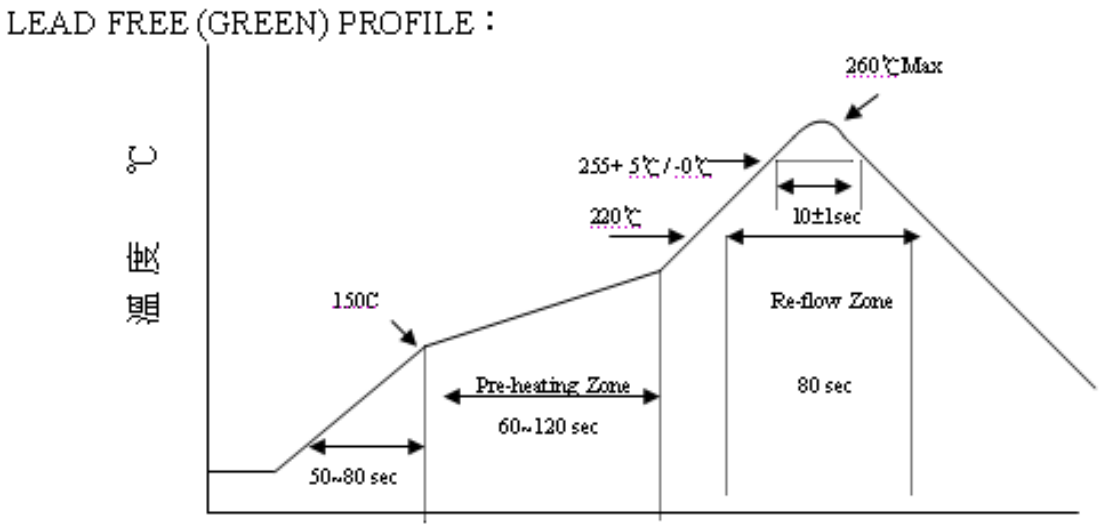
X : PKG HOUSE ID

NNNNNNNNN : LOT NO.
(max. 9 characters)

I=J
K=L



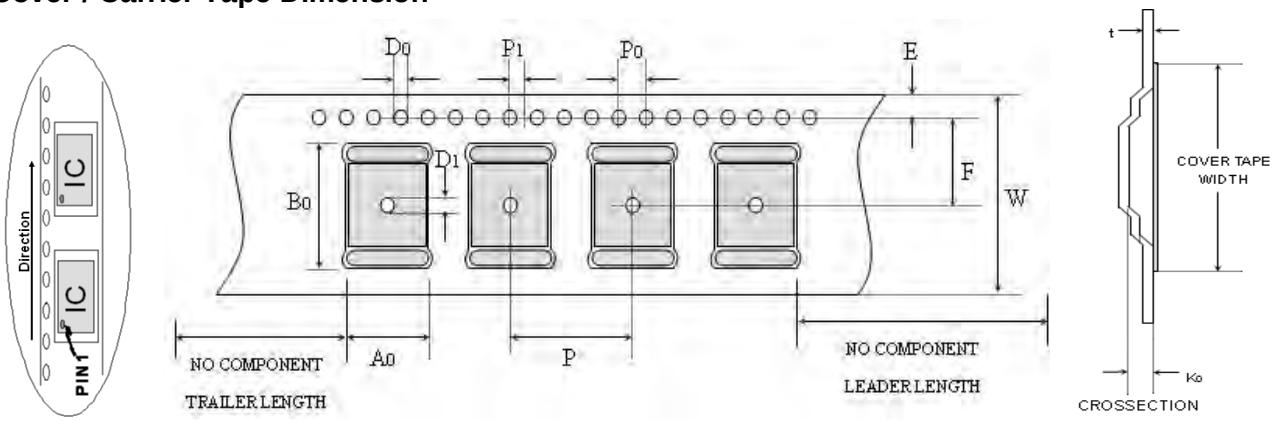
28. Reflow Profile



Actual Measurement Graph

29. Tape Reel Information

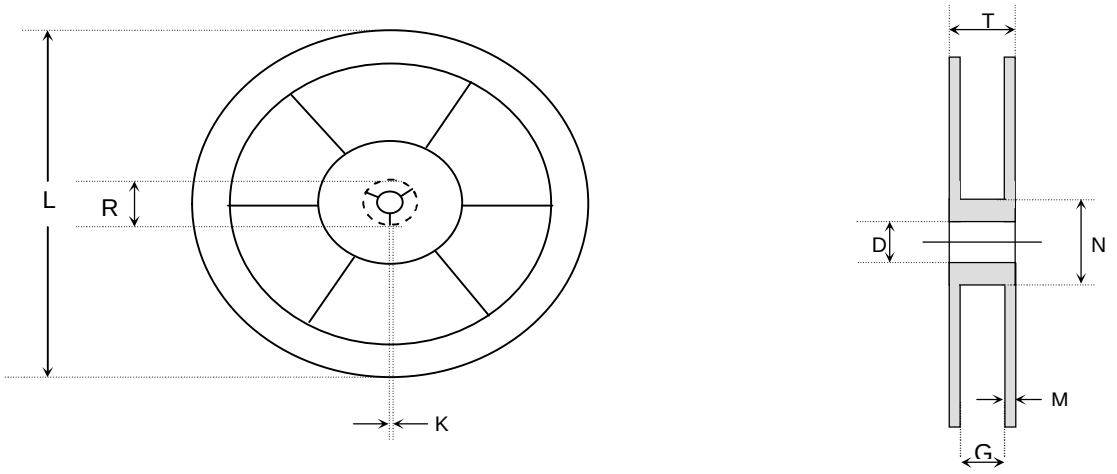
Cover / Carrier Tape Dimension



Unit: mm

TYPE	P	A0	B0	P0	P1	D0	D1	E	F	W	K0	t	Cover tape width
QFN3*3	8±0.1	3.25±0.1	3.25±0.1	4±0.2	2±0.1	1.5±0.1	1.5	1.75±0.1	5.5±0.05	12±0.3	1.25±0.1	0.3±0.05	9.3±0.1
QFN 4*4	8±0.1	4.35±0.1	4.35±0.1	4±0.2	2±0.1	1.5±0.1	1.5	1.75±0.1	5.5±0.05	12±0.3	1.25±0.1	0.3±0.05	9.3±0.1
QFN 5*5	8±0.1	5.25±0.1	5.25±0.1	4±0.2	2±0.1	1.5±0.1	1.5	1.75±0.1	5.5±0.05	12±0.3	1.25±0.1	0.3±0.05	9.3±0.1
SSOP	12±0.1	8.2±1	8.8±1.5	4.0±0.1	2.0±0.1	1.5±0.1	1.5±0.1	1.75±0.1	7.5±0.1	16±0.1	2.1±0.4	0.3±0.05	13.3±0.1

REEL DIMENSIONS



Unit: mm

TYPE	G	N	M	D	K	L	R
QFN	12.9±0.5	102 REF±2.0	2.3±0.2	13.15±0.35	2.0±0.5	330±3.0	19.6±2.9
SSOP	16.3±1	102 REF±2.0	2.3±0.2	13.15±0.35	2.0±0.5	330±3.0	19.6±2.9

30. Product Status

Data Sheet Identification	Product Status	Definition
Objective	Planned or Under Development	This data sheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	Engineering Samples and First Production	This data sheet contains preliminary data, and supplementary data will be published at a later date. AMICCOM reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
No Identification	Noted Full Production	This data sheet contains the final specifications. AMICCOM reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Obsolete	Not In Production	This data sheet contains specifications on a product that has been discontinued by AMICCOM. The data sheet is printed for reference information only.

RF ICs AMICCOM

Headquarter
A3, 1F, No.1, Li-Hsin 1st Rd., Hsinchu Science Park,
Hsinchu, Taiwan 30078
Tel: 886-3-5785818

Shenzhen Office
Rm., 2003, DongFeng Building, No. 2010,
Shennan Zhonglu Rd., Futian Dist., Shenzhen, China
Post code: 518031

Web Site
http://www.amiccom.com.tw