Document Title

A8107M0 Data Sheet, Bluetooth Low Energy SoC

Revision History

Rev. No.	History	Issue Date	Remark
0.0	Initial issue.	Aug., 2016	Objective
0.1	Revise TFT LCD driver to TFT LCD controller	Sep., 2016	Preliminary
0.2	Update Features and Electrical Specification	Oct., 2016	Preliminary

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

A8107M0 is a high performance and low cost 2.4GHz FSK/GFSK system-on-chip (SOC) wireless transceiver. With on chip fraction-N synthesizer, it can support the application of data rate from 5Kbps to 2Mbps and frequency hopping system and it is designed for Bluetooth Low Energy (Bluetooth 4.0 Single MODE). It is a Bluetooth smart device. This device integrates high ARM-M0 MCU, 256K Bytes In-system programmable flash memory, 32KB SRAM, various powerful functions and excellent performance of a leading 2.4GHz FSK/GFSK RF transceiver. It can be operated with wide voltage from 2.0V ~ 3.6V. A8107M0 has various operating MODEs, making it highly suited for systems where ultra-low power consumption is required. A8107M0 has 256K bytes flash that supports AES128 engine and CCM. For low current consumption, A8107M0 is integrated with both LDO and DC-DC (buck) so that this device can be operated more efficient when VDD voltage range from 2.7V to 3.6V. User can configure one of them (LDO or DC-DC) as a powered source for device operations. The device has 2 package sizes: QFN5X5 40 pin package and QFN6x6 48pin package.

2. Typical Applications

- 2400 ~ 2483.5 MHz ISM frequency hopping system
- Smart remote controller
- Home and building automation
- Wireless keyboard and mouse
- Wireless toy and gaming
- Helicopter and airplane radio controller
- Bluetooth smart device

3. Features

- Package size (QFN5X5, 40 pins/ QFN6X6 48 pins).
- High performance ARM-M0 MCU
- Operation clock: 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64 of crystal oscillator.
- TFT DMA controller and 8080 output interface(I80)
- 256KB Flash memory with copy protection, 32KB SARM
- UART, I^2C, SPI serial communication
- Two 32-bit timers and one 32-bit dual MODE timer.
- Four Channel PWM
- Watchdog timer
- Two 16-bit Sleep timer
- In-Circuit Debugger
- In-System programming/ In-Application programming
- 23/31 GPIO for QFN40/48
- RX current consumption with MCU stop and DC-DC turn on: 6.4mA @BATH= 3.3V
- TX current consumption with MCU stop and DC-DC turn on: 9mA @ 5dBm, BATH=3.3V.
- Power saving MODE without sleep timer, no SRAM retention (1.3 uA)
- Power saving MODE with sleep timer, 16K SRAM retentio (2.1uA)
- Frequency band: 2400 – 2483MHz.
- FSK and GFSK modulation
- High sensitivity:
  - -97dBm at 500Kbps data rate
  - -94dm at 1Mbps data rate
  - -91dBm at 2Mbps data rate
- Programmable data rate 5K ~ 2Mbps.
- Fast settling time synthesizer for frequency hopping system.
- Built-in thermal sensor for monitoring relative temperature.
- Built-in one channel 8-bits ADC for external analog voltage. (0V ~ 0.9V).
- Built-in eight channels 12-bits ADC for general purpose analog input (0V ~ 1.8 V).
- Built-in Low Battery Detector.
- Support 16MHz crystal
- Easy to use.
  - Change frequency channel by one register setting.
  - 8-bits Digital RSSI for clear channel indication.
  - Auto RSSI measurement.
  - Auto WOR (wake up when receive RX packet).
  - Auto WOT (wake up to transmit TX packet).
  - Auto Calibrations.
  - Auto IF function.
  - Auto Frequency Compensation.
  - Auto CRC Check.

- Separated 256 bytes RX and TX FIFO.

4. Pin Configurations

4.1 QFN40 5x5

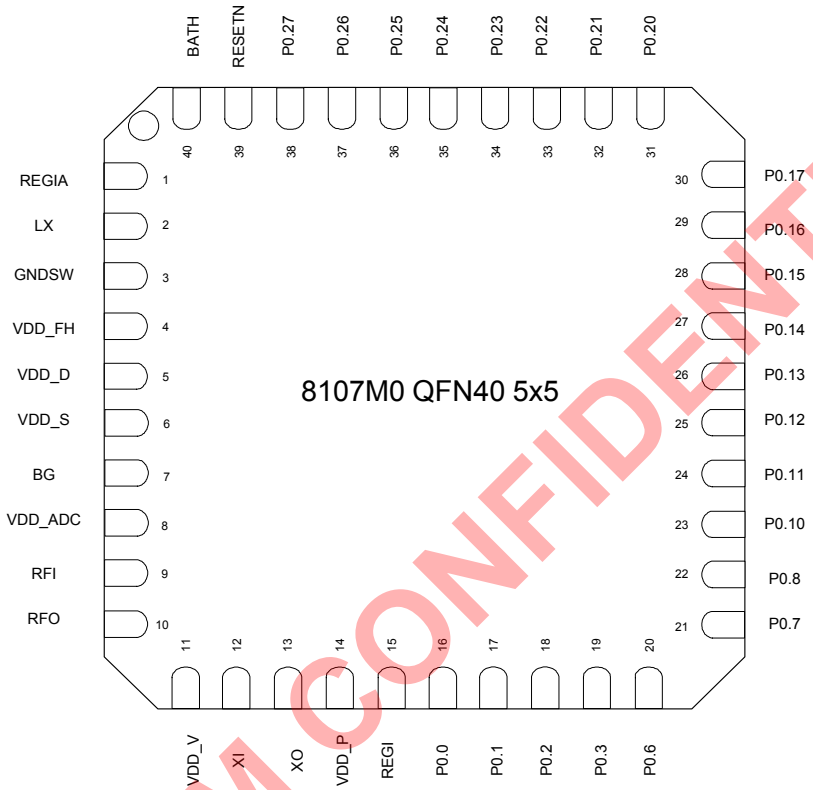


Fig 4-1. A8107M0 QFN40 5x5 Package Top View

4.2 QFN48 6x6

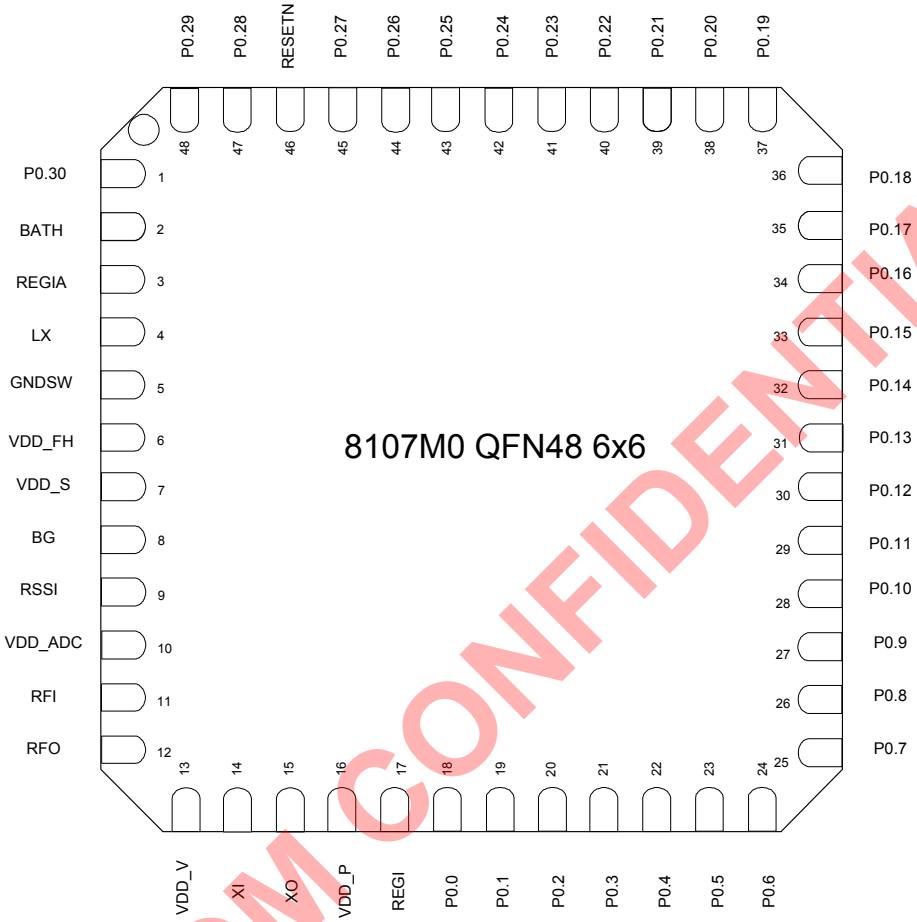


Fig 4-2. A8107M0 QFN48 6x6 Package Top View

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

5.1 A8107M0 QFN40 5x5 Package

Pin No.	Symbol	I/O	Function Description
1	REGIA	AO	DC-DC regulator output for buck
2	LX	AI	DC-DC Inductor connection pin
3	GNDSW	AI	Ground
4	VDD_FH	AO	Flash high voltage output
5	VDD_D	AO	VDD_D supply voltage output
6	VDD_S	AO	VDD_S supply voltage output
7	BG	AO	Band gap output
8	VDD_ADC	AO	VDD_ADC supply voltage output
9	RFI	AI	RF input
10	RFO	AO	RF output
11	VDD_V	AI	VCO supply voltage input
12	XI	AI	Crystal oscillator input
13	XO	AO	Crystal oscillator output
14	VDD_P	AO	PLL supply voltage output
15	REGI	AI	Regulator input
16	P0.0	DIO/AIO	SPI_CS
17	P0.1	DIO/AIO	SPI_MISO
18	P0.2	DIO/AIO	SPI_MOSI
19	P0.3	DIO/AIO	SPI_SCK
20	P0.6	DIO	SWDIOTMS
21	P0.7	DIO	SWCLKTCK
22	P0.8	DIO/AIO	timer0_EIN / ADC2 / BB_GIO1
23	P0.10	DIO/AIO	PWM2 / I^2C_SCL
24	P0.11	DIO/AIO	PWM3 / I^2C_SDA / LCD_TE
25	P0.12	DIO/AIO	ADC4 / ICE_MODE / LCD_nRD
26	P0.13	DIO/AIO	ADC5/BB_GIO1 / LCD_Data[2]
27	P0.14	DIO/AIO	ADC6/BB_GIO2 / LCD_Data[1]
28	P0.15	DIO/AIO	ADC7/BB_CKO / LCD_Data[0]
29	P0.16	DIO/AIO	UART0_RX / LCD_nCS
30	P0.17	DIO/AIO	UART0_TX / LCD_A0
31	P0.20	DIO/AIO	UART2_RX / PWM0 / LCD_nWR/ s_LCD_SCL
32	P0.21	DIO/AIO	UART2_TX / PWM1 / LCD_Data[7]
33	P0.22	DIO/AIO	RTCI
34	P0.23	DIO/AIO	RTCO
35	P0.24	DIO	General IO / LCD_Data[6]
36	P0.25	DIO	General IO / LCD_Data[5]
37	P0.26	DIO	General IO / LCD_Data(4)

38	P0.27	DIO	General IO / LCD_Data[3]
39	RESETN	AI	RESETN input
40	BATH	DIO/AIO	DC-DC converter voltage input for buck mode

5.2 A8107M0 QFN48 6x6 Package

Pin No.	Symbol	I/O	Function Description
1	P0.30	DIO	General IO
2	BATH	DIO/AIO	DC-DC converter voltage input for buck mode
3	REGIA	AO	DC-DC regulator output voltage
4	LX	AI	DC-DC Inductor connection pin
5	GNDSW	AI	Ground
6	VDD_FH	AO	Flash high voltage output
7	VDD_S	AO	VDD_S supply voltage output
8	BG	AO	Band gap output
9	RSSI	AO	RSSI Bypass
10	VDD_ADC	AO	VDD_ADC supply voltage output
11	RFI	AI	RF input
12	RFO	AO	RF output
13	VDD_V	AI	VCO supply voltage input
14	XI	AI	Crystal oscillator input
15	XO	AO	Crystal oscillator output
16	VDD_P	AO	PLL supply voltage output
17	REGI	AI	Regulator input
18	P0.0	DIO/AIO	SPI_CS
19	P0.1	DIO/AIO	SPI_MISO
20	P0.2	DIO/AIO	SPI_MOSI
21	P0.3	DIO/AIO	SPI_SCK
22	P0.4	DIO/AIO	I^2C_SCL
23	P0.5	DIO/AIO	I^2C_SDA
24	P0.6	DIO	SWDIOTMS
25	P0.7	DIO	SWCLKTCK
26	P0.8	DIO/AIO	Timer0_EIN / ADC2 / BB_GIO1
27	P0.9	DIO/AIO	Timer1_EIN / ADC3 / BB_GIO2
28	P0.10	DIO/AIO	PWM2 / I^2C_SCL
29	P0.11	DIO/AIO	PWM3 / I^2C_SDA / LCD_TE
30	P0.12	DIO/AIO	ADC4 / ICE_MODE / LCD_nRD
31	P0.13	DIO/AIO	ADC5/BB_GIO1 / LCD_Data[2]
32	P0.14	DIO/AIO	ADC6/BB_GIO2 / LCD_Data[1]
33	P0.15	DIO/AIO	ADC7/BB_CKO / LCD_Data[0] / s_LCD_SDA
34	P0.16	DIO/AIO	UART0_RX / LCD_nCS
35	P0.17	DIO/AIO	UART0_TX / LCD_A0
36	P0.18	DIO/AIO	UART1_RX / BB_GIO1

37	P0.19	DIO/AIO	UART1_TX / BB_GIO2
38	P0.20	DIO/AIO	UART2_RX / PWM0 / LCD_nWR/ s_LCD_SCL
39	P0.21	DIO/AIO	UART2_TX / PWM1 / LCD_Data[7]
40	P0.22	DIO/AIO	RTCI
41	P0.23	DIO/AIO	RTCO
42	P0.24	DIO	General IO / LCD_Data[6]
43	P0.25	DIO	General IO / LCD_Data[5]
44	P0.26	DIO	General IO / LCD_Data[4]
45	P0.27	DIO	General IO / LCD_Data[3]
46	RESETN	AI	RESETN input
47	P0.28	DIO	General IO
48	P0.29	DIO	General IO

6. Chip Block Diagram

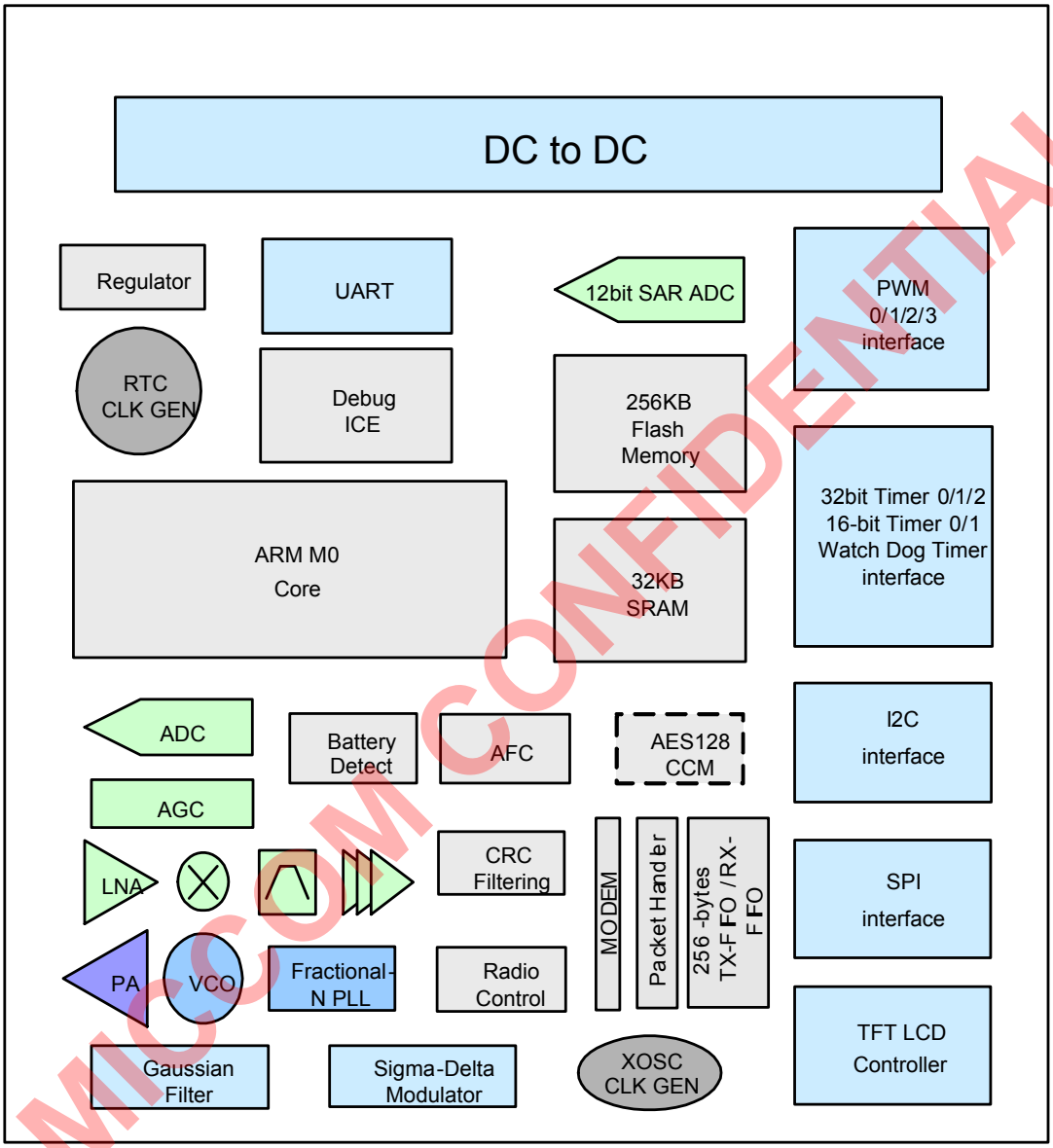


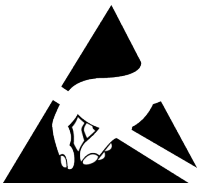
Fig6-1, A8107M0 Block Diagram

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℃, **BATH = 3.3V,** unless otherwise noted)

Parameter	Description	Min.	Typ.	Max.	Unit
General					
Operating Temperature		-40		85	℃
Supply Voltage (BATH)	BATH supply input	2.0		3.6	V
Current Consumption (MCU in stop MODE, and RF in sleep MODE)	PM1 with sleep timer		3.3		uA
	PM2 with sleep timer		3.3		uA
	PM3 with sleep timer, 16K SRAM retention		2.1		uA
	PM3 without sleep timer, 16K SRAM off		1.3		uA
Current Consumption (MCU in normal MODE) Without DC-DC MCU Clock @16MHz	Sleep MODE		3		mA
	Standby MODE		4.7		mA
	PLL MODE		9.1		mA
	RX MODE (AGC Off)		14.8		mA
	RX MODE (AGC On)		15.1		mA
	TX MODE (@5dBm output)		19.9		mA
Current Consumption (MCU in STOP MODE) Without DC-DC	Standby MODE		1		mA
	PLL MODE		5.5		mA
	RX MODE (AGC Off)		11.3		mA
	RX MODE (AGC On)		11.6		mA
	TX MODE (@5dBm output)		16.4		mA
Current Consumption (MCU in STOP MODE) With DC-DC	Standby MODE		0.8		mA
	PLL MODE		3.2		mA
	RX MODE (AGC Off)		6.4		mA
	RX MODE (AGC On)		6.7		mA
	TX MODE (@5dBm output)		9		mA
Synthesizer block					
Crystal settling time	Idle to standby (XTAL SMD2016)		0.6		ms
Crystal frequency			16		MHz
Crystal tolerance			±20		ppm
Crystal Load Capacitance			9		pF
Crystal ESR				80	ohm
PLL settling time	Standby to PLL		75		μS
Transmitter					
Carrier Frequency		2400		2483.5	MHz
Maximum Output Power			4		dBm
RF Power Control Range			20		dB
Out Band Spurious Emission[1]	30MHz~1GHz			-36	dBm
	1GHz~12.75GHz			-30	dBm
	1.8GHz~ 1.9GHz			-47	dBm
	5.15GHz~ 5.3GHz			-47	dBm
Frequency deviation	500Kbps		186K		Hz
	1M		250K		Hz
	2M		500K		Hz

Data rate		5K		2M	Bps
TX settling time	Standby to TX		120		μs
Receiver					
Receiver sensitivity @ BER = 0.1%	Data rate 2M (F_{IF} = 2MHz)		-91		dBm
	Data rate 1M (F_{IF} = 1MHz)		-94		dBm
	Data rate 500K (F_{IF} = 1MHz)		-97		dBm
IF frequency bandwidth			1200/2400		KHz
IF center frequency			1000/2000		KHz
Interference	Co-Channel (C/I_0)		11		dB
	1st Adjacent Channel (C/I_1)		2		dB
	2nd Adjacent Channel (C/I^2)		-18		dB
	3rd Adjacent Channel (C/I_3)		-28		dB
	Image (C/I_{IM})		-12		dB
Maximum Operating Input Power	@RF input (BER=0.1%)			0	dBm
RX Spurious Emission	30MHz~1GHz			-52	dBm
	1GHz~12.75GHz			-47	dBm
RSSI Range with AGC turn on	@RF input	-100		-10	dBm
RX settling time	Standby to RX		130		μS
12Bit SAR ADC					
Input voltage range		0		1.8	V
External reference voltage			1.8		V
Input capacitor			25		pF
Bandwidth			200		KHz
EOB, effective number of bits			10		bit
INL			+/- 2		LSB
DNL			+/-1		LSB
Conversion time		128		8	uS
Current consumption			0.4		mA
Regulator					
Regulator settling time			200		μs
Band-gap reference voltage			1.21		V
Regulator output voltage			1.21		V
Digital IO DC characteristics					
High Level Input Voltage (V_{IH})		0.8*BATH		BATH	V
Low Level Input Voltage (V_{IL})		0		0.2*BATH	V
High Level Output Voltage (V_{OH})	@I_{OH}= -0.5mA	BATH-0.4		BATH	V
Low Level Output Voltage (V_{OL})	@I_{OL}= 0.5mA	0		0.4	V
DC-DC Buck converter					
Input voltage range		2.0		3.6	V
Output voltage		1.5	1.6	1.8	V
Efficiency (with 100 ohm load @ 3.0V input.)	PWM MODE		89		%
Efficiency (with 100 ohm load @ 3.6V input.)	PWM MODE		86		%
Maximum load current				50	mA

9. Register List

A8107M0 contains Peripheral Register and RF Register.

9.1 Peripheral Register Overview

Base address	Peripheral	Description
0x40000000	Timer0	Timer0
0x40001000	Timer1	Timer1
0x40002000	Dual timer	Dual timer0,1
0x40004000	Uart0	Uart0
0x40005000	Uart1	Uart1
0x40006000	Uart2	Uart2
0x40008000	WDT	Watchdog
0x40010000	GPIO0	
0x40011000	GPIO1	
0x4001F000	SYSCTRL	
0x50000000	Power control	Power and clock control
0x50001000	Radio	RF configure
0x50002000	SPI	SPI master/slave
0x50003000	I^2C	I^2CMaster/I^2CSlave
0x50004000	PWM0	PWM0,1,2,3
0x50005000	RTC	Real time counter
0x50006000	Sleep timer	Sleep timer0,1
0x50007000	USB	USB
0x50008000	12 bit ADC	12 bit ADC0,1
0x50009000	RCADC	RCADC0,1
0x5000A000	LCD	LCD controller
0x5000B000	CODEC	Codec
0x5000C000	AES/CCM	AES/CCM
0x5000F000	IP Information	

9.2 Register Overview

Includes the following sections:

- Battery detect Registe
- Power control Register 1
- Power control Register 2
- Reset Register
- Strobe Register
- Status Register
- FIFO Register
- RSSI Register
- ADC Control Register I
- ADC Control Register II
- ADC Channel Register

Follow the instructions below to setup the register.

9.2.1 Battery detect Register (Address: 0x50000000h)

Name	Address	R/W	Description	Reset value
BD	PWR_BA+0x00	R/W	Battery detect register	32'h0000_0006

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
W	0	0	0	0	BVT2	BVT1	BVT0	BDS
R				BDF	BVT2	BVT1	BVT0	BDS

BDF: Battery detection flag.
[0]: Battery voltage less than threshold.
[1]: Battery voltage greater than threshold.

BVT [2:0]: Battery voltage detect threshold.
[000]: 1.875.
[001]: 1.95V.
[010]: 2.025V.
[011]: 2.1V.
[100]: 2.175V.
[101]: 2.25V.
[110]: 2.325V.
[111]: 2.4V.

BDS: Battery detect enable.
[0]: Disable.
[1]: Enable. It will be clear after battery detection done.

9.2.2 Power control Register 1 (Address: 0x50000008h)

Name	Address	R/W	Description	Reset value
PWRCTL1	PWR_BA+0x08	R/W	Power control register 1	32'h000A_D000

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

R							RGV[1]	RGV[0]

RGV [1:0]: VDD_D and VDD_A voltage setting in non-Sleep MODE. Recommend RGV = [11].
[00]: 1.35V.
[01]: 1.3V.
[10]: 1.25V.
[11]: 1.2V.

9.2.3 Power control Register 2 (Address: 0x5000000Ch)

Name	Address	R/W	Description	Reset value
PWRCTL2	PWR_BA+0x0C	R/W	Power control register 2	32'h0000_06D0

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
W	1	ENAV	QDSA	ENDV	QDSD	CEL	0	0
R								

ENAV: REGOA and REGOS connection.
QDSA: quick discharge select for REGOA.
ENDV: REGOA is connected to REGOD.
QDSD: quick discharge select for REGOD.
CEL: Digital voltage select in standby MODE. Recommend CEL = [0].
PM MODE: Low power operation select.

	MCU STOP
PM1(idle)	ENAV=1, QDSA=0, ENDV=1, QDSD=0
PM2(sleep)	ENAV=0, QDSA=1, ENDV=1, QDSD=0
PM3(deep sleep)	ENAV=0, QDSA=1, ENDV=0, QDSD=1

9.2.4 Reset Register (Address: 0x50001000h)

Name	Address	R/W	Description	Reset value
Reset	RF_BA+0x000	R/W	MODE reset register	32'h0000_0000

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
W	RESETN	FWPRN	FRPRN	0	BFCRN	0	0	0

R								

RESETN: Write to this register to issue reset command. (Write "1" to reset)

FWPRN: FIFO Write Point Software Reset. (Write "1" to reset)

FRPRN: FIFO Read Point Software Reset. (Write "1" to reset)

BFCRN: IF Filter Bank Calibration Software Reset. (Write "1" to reset)

9.2.5 Strobe Register (Address: 0x50001004h)

Name	Address	R/W	Description	Reset value
Strobe	RF_BA+0x004	R/W	Strobe command register	32'h0000_00A0

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
W	STRB7	STRB6	STRB5	STRB4	STRB3	STRB2	STRB1	STRB0
R								

STRB[7:0]: Strobe command register.
[80]: Sleep MODE.
[90]: Idle MODE.
[A0]: Standby MODE.
[B0]: PLL MODE.
[C0]: RX MODE.
[D0]: TX MODE.
Reverse for other settings.

9.2.6 Status Register (Address: 0x50001010h)

Name	Address	R/W	Description	Reset value
STATUS	RF_BA+0x010	R/W	Status Register	32'h0000_0000

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

R		FECF	CRCF	CER	XER	PLLER	TRSR	TRER

SYNC: SYNC flag.
[0]: SYNC pass.
[1]: SYNC error.

FECF: FEC flag.
[0]: FEC pass.
[1]: FEC error.

CRCF: CRC flag.
[0]: CRC pass.
[1]: CRC error.

CER: RF chip enable status.
[0]: RF chip is disabled.
[1]: RF chip is enabled.

XER: Internal crystal oscillator enabled status.
[0]: Crystal oscillator is disabled.
[1]: Crystal oscillator is enabled.

PLLER: PLL enabled status.
[0]: PLL is disabled.
[1]: PLL is enabled.

TRSR: TRX Status Register.
[0]: RX state.
[1]: TX state.
Serviceable if TRER=1 (TRX is enable).

TRER: TRX state enabled status.
[0]: TRX is disabled.
[1]: TRX is enabled.

9.2.7 FIFO Register (Address: 0x50001020h)

Name	Address	R/W	Description	Reset value
FIFO Control	RF_BA+0x020	R/W	FIFO Control Register	32'h4000_003F

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
W	FEP7	FEP6	FEP5	FEP4	FEP3	FEP2	FEP1	FEP0
R	FEP7	FEP6	FEP5	FEP4	FEP3	FEP2	FEP1	FEP0

FEP [7:0]: FIFO End Pointer for TX FIFO and Rx FIFO.

9.2.8 RSSI Register (Address: 0x50001244h)

Name	Address	R/W	Description	Reset value
RSSI Threshold	RF_BA+0x244	R/W	RSSI Threshold register	32'h0000_0000

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
W	RTH7	RTH6	RTH5	RTH4	RTH3	RTH2	RTH1	RTH0
R	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0

RTH [7:0]: Carrier detect threshold. Refer to chapter 19.1.2 for details

ADC [7:0]: ADC output value of temperature, RSSI or external voltage measurement.
ADC input voltage= 0.9 * ADC [7:0] / 256 V.

9.2.9 ADC Control Register I (Address: 0x50008000h)

Name	Address	R/W	Description	Reset value
ADCCTL	ADC12_BA+0x00	R/W	ADC Control register	32'h0000_0040

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	ADC12RN	0	0	0	0	0	0	0
R								
R/W	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
W	ADCIE	0	0	0	ADIVL	ADCYC	ENADC	DTMP
R								
R/W	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
W	BUFS	CKS1	CKS0	MODE	0	0	0	0
R	--	--	--	MODE	MVS2	MVS1	MVS0	ADCE

ADC12RN: 12-bits ADC Reset. (Write "1" to reset)
[1]: enable
[0]: disable.
ADCIE : 12-bits interrupt enable.
[0]: disable.
[1]: enable.
ADIVL: 12-bits sargen initial value select.
[1]: 0
[0]: 2048.
ADCYC: 12-bits sargen clock counter select.
[1]: 32.
[0]: 31.

ENADC: Enable ADC.
[1]: enable
[0]: disable.

DTMP: 12-bits temperature select. Refer to chapter 19 for details.
[1]: enable
[0]: disable.

BUFS: input buffer select for 12 bit ADC. Refer to chapter 19 for details.
[0]: disable.
[1]: enable.
CKS[1:0]: ADC clock selected.
[00]: 4 MHz
[01]: 2 MHz
[10]: 1 MHz
[11]: 500 kHz
MODE: ADC measurement MODE.
[0]: Single MODE.
[1]: Continuous MODE.
MVS [2:0]: ADC average times .
[000]: No Average.
[001]: Average 2 times.
[010]: Average 4 times.
[011]: Average 8 times.
[100]: Average 16 times.
[101]: Average 32 times.
[110]: Average 64 times.
[111]: Average 128 times.
ADCE: ADC measurement enable.
[1]: enable
[0]: disable.

9.2.10 ADC Control Register II (Address: 0x50008004h)

Name	Address	R/W	Description	Reset value
ADCAVG	ADC12_BA+0x04	R	ADC Value register	32'h0000_0000

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	0	0	0	0	0	0	0	0
R					MVADC11	MVADC10	MVADC9	MVADC8
R/W	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
W	0	0	0	0	0	0	0	0
R	MVADC7	MVADC6	MVADC5	MVADC4	MVADC3	MVADC2	MVADC1	MVADC0
R/W	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
W	0	0	0	0	0	0	0	0
R					ADC11	ADC10	ADC9	ADC8
R/W	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
W	0	0	0	0				0
R	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0

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

ADC [11:0]: ADC output value

9.2.11 ADC Channel Register (Address: 0x50008008h)

Name	Address	R/W	Description	Reset value

ADCIOS	ADC12_BA+0x08	W/R	ADC channel register	32'h0000_0000

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
W	0	0	0	0	0	0	0	0
R								
R/W	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
W	0	0	0	0	ADCCH3	ADCCH2	ADCCH1	ADCCH0
R					ADCCH3	ADCCH2	ADCCH1	ADCCH0

ADCCH[3:1] : ADC I/O select
[000]: Select P0.18 as ADC analog input.
[001]: Select P0.19 as ADC analog input.
[010]: Select P0.8 as ADC analog input.
[011]: Select P0.9 as ADC analog input.
[100]: Select P0.12 as ADC analog input.
[101]: Select P0.13 as ADC analog input.
[110]: Select P0.14 as ADC analog input.
[111]: Select P0.15 as ADC analog input.

ADCCH0: ADC input enable
[1]: Enable.
[0]: Disable.

10.SOC Architectural Overview

A8107M0 microcontroller is instruction set compatible with Cortex™-M0 profile processors. Besides FSK/GFSK RF transceiver, A8107M0 integrates many features, three 8/16bit counters/timers, watchdog timer, RTC, UART, SPI interface, I^2C interface, 4 channels PWM, 8 channels ADC and battery detector, The interrupt controller is extended to support 6 interrupt sources; watchdog timer, RTC, SPI, I^2C, ADC and RF. A8107M0 includes TTAG (2-wire) debug circuitry that provides full time, real-time, in-circuit debugging.

10.1 ARM Cortex-M0

The Cortex™-M0 processor is a configurable, multistage, 32-bit RISC processor which has an AMBA AHB-Lite interface and includes an NVIC component. It also has optional hardware debug functionality. The processor can execute Thumb code and is compatible with other Cortex™-M0 profile processors. The profile supports two MODEs - Thread MODE and Handler MODE. Handler MODE is entered as a result of an exception. An exception return can only be issued in Handler MODE. Thread MODE is entered on Reset and can be entered as a result of an exception return. The following figure shows the functional controller of the processor.

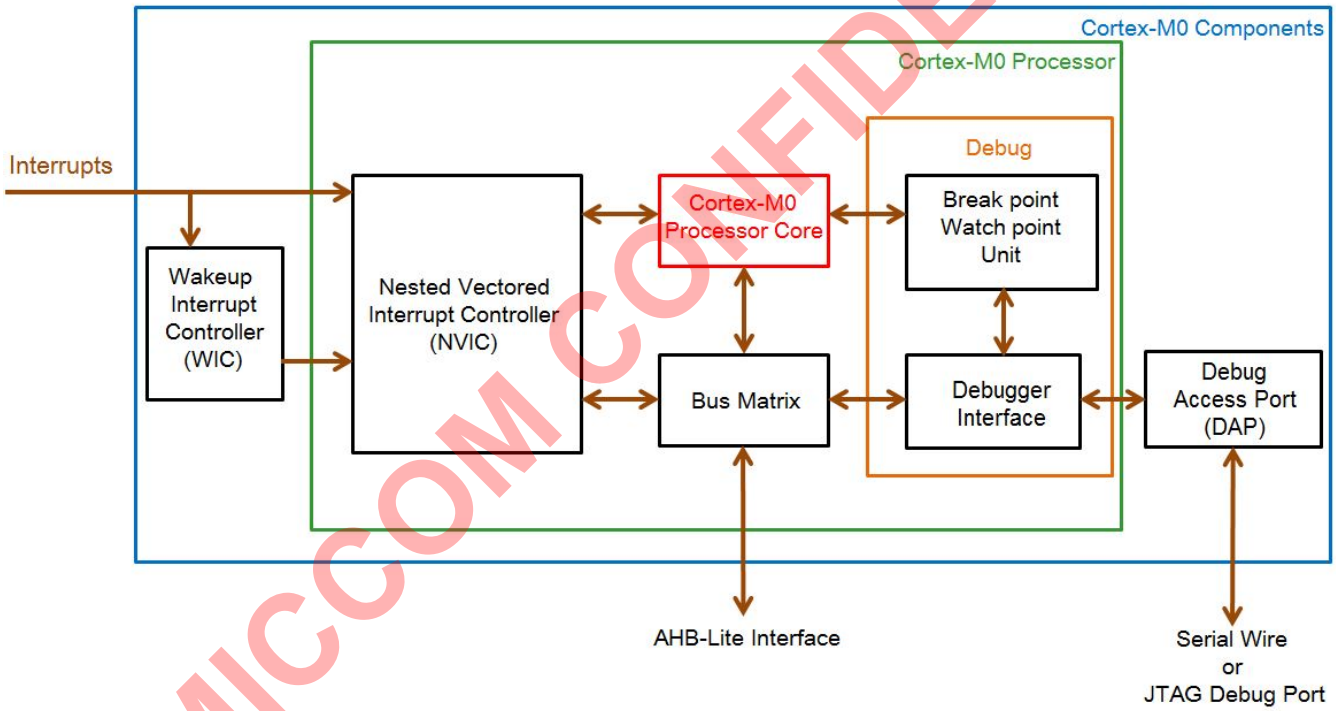


Figure 10.1 Core-M0 block diagram

10.1.1 Feature

- A low gate count processor
  - ARMv6-M Thumb® instruction set
  - Thumb-2 technology
  - ARMv6-M compliant 24-bit SysTick timer
  - A 32-bit hardware multiplier
  - System interface supported with little-endian data accesses
  - Ability to have deterministic, fixed-latency, interrupt handling
  - Load/store-multiples and multicycle-multiplies that can be abandoned and restarted to facilitate rapid interrupt handling
  - C Application Binary Interface compliant exception MODEl:
    This is the ARMv6-M, C Application Binary Interface (C-ABI) compliant exception MODEl that enables the use of

pure C functions as interrupt handlers
- Low power Idle MODE entry using the Wait For Interrupt (WFI), Wait For Event (WFE) instructions, or return from interrupt sleep-on-exit feature

- NVIC
  - 32 external interrupt inputs, each with four levels of priority
  - Dedicated Non-maskable Interrupt (NMI) input
  - Supports for both level-sensitive and pulse-sensitive interrupt lines
  - Supports Wake-up Interrupt Controller (WIC) and, providing Ultra-low Power Idle MODE
- Debug support
  - Four hardware breakpoints
  - Two watch points
  - Program Counter Sampling Register (PCSR) for non-intrusive code profiling
  - Single step and vector catch capabilities
- Bus interfaces
  - Single 32-bit AMBA-3 AHB-Lite system interface that provides simple integration to all system peripherals and memory
  - Single 32-bit slave port that supports the DAP (Debug Access Port)

10.2 Memory Organization

The memory organization is shown as figure 10.2

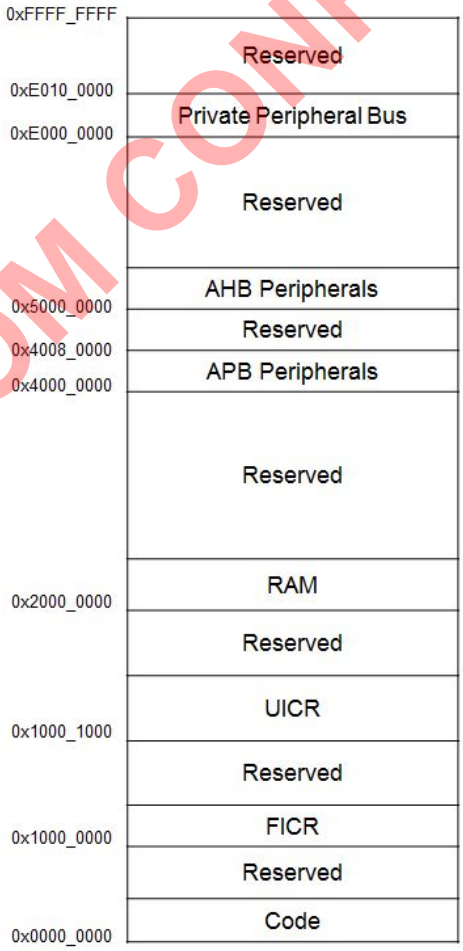


Figure 10.2 Memory Organization

10.3 Nested Vectored Interrupt Controller (NVIC)

The Cortex™-M0 CPU provides an interrupt controller as an integral part of the exception MODE, named as "Nested Vectored Interrupt Controller (NVIC)", which is closely coupled to the processor core and provides following features.

10.3.1 Feature

- Flexible interrupt management

In the Cortex-M0 processor, each external interrupt can be enabled or disabled and can have its pending status set or clear by software. It can also accept exception requests at signal level (interrupt request from a peripheral remain asserted until the interrupt service routine clears the interrupt request), as well as an exception request pulse (minimum 1 clock cycle). This allows the interrupt controller to be used with any interrupt source.

- Nested interrupt support

In the Cortex-M0 processor, each exception has a priority level. The priority level can be fixed or programmable. When an exception occurs, such as an external interrupt, the NVIC will compare the priority of this exception to the current level. If the new exception has a higher priority, the current running task will be suspended. Some of the registers will be stored on to the stack memory, and the processor will start executing the exception handler of the new exception. This process is called "preemption." When the higher priority exception handler is complete, it is terminated with an exception return operation and the processor automatically restores the registers from the stack and resumes the task that was running previously. This mechanism allows nesting of exception services without any software overhead.

- Vectored exception entry

When an exception occurs, the processor will need to locate the starting point of the corresponding exception handler. Traditionally, in ARM processors such as the ARM7TDMI, software usually handles this step. The Cortex-M0 automatically locates the starting point of the exception handler from a vector table in the memory.
As a result, the delay from the start of the exception to the execution of the exception handlers is reduced.

- Interrupt masking

The NVIC in the Cortex-M0 processor provides an interrupt masking feature via the PRIMASK special register. This can disable all exceptions except hard fault and NMI. This masking is useful for operations that should not be interrupted such as time critical control tasks or real-time multimedia codecs.

10.3.2 Exception Types and Interrupt Map

Each exception source in the Cortex-M0 processor has a unique exception number. The exception number for NMI is 2, and the exception numbers for the on-chip peripherals and external interrupt sources are from 16 to 47. The other exception numbers, from 1 to 15, are for system exceptions generated inside the processor, although some of the exception numbers in this range are not used. Each exception type also has an associated priority. The priority levels of some exceptions are fixed and some are programmable. Table 8.1 shows the exception types, exception numbers, and priority levels.

Exception Number	Exception Type	Priority	Interrupt Description
1	Reset	-3(Highest)	Reset
2	NMI	-2	Non maskable interrupt
3	Hard fault	-1	Fault handing exception
4-10	Reserved	--	--
11	SVC	Programmable	Supervisor call via SVC instruction
12-13	Reserved	--	--
14	PendSV	Programmable	Pendable request for system service
15	SysTick	Programmable	System tick timer
16-47	IRQ0~IRQ31	Programmable	IRQ

Figure 10.3 Exceptions Type

Exception Number	Interrupt Number Bit	Interrupt Name	Interrupt Description
16	0	--	-
17	1	UART0_INT	UART0 Tx/Rx/Overflow interrupt
18	2	SLPTMR0_INT	Sleep timer0 interrupt
19	3	RADIO_INT	RADIO interrupt
20	4	--	--
21	5	UART2_INT	UART2 Tx/Rx/Overflow interrupt
22	6	PORT0_COMB_INT	GPIO 0 combined interrupt for AHB GPIO and I/O port GPIO
23	7	--	--
24	8	TIMER0	Timer0 interrupt
25	9	TIMER1	Timer1 interrupt
26	10	Dual_Timer_INT	Dual Timer interrupt
27	11	MPU_LCD_INT	MPU_LCD interrupt
28	12	FSYNC	FSYNC interrupt
29	13	UART1_INT	UART1 Tx/Rx/Overflow interrupt
30	14	--	--
31	15	WUN_INT	REGWUN interrupt
32	16	SPI_INT	SPI interrupt
33	17	I^2C_INT	I^2CM/I^2CS interrupt
34	18	RTC_INT	Real Time Counter interrupt
35	19	--	--
36	20	AES_INT	AES/CCM interrupt
37	21	ADC_INT	12bits-ADC interrupt
38	22	--	--
39	23	--	--
40	24	SLPTMR0_INT	Sleep timer0 interrupt
41	25	--	--
42	26	--	--
43	27	--	--
44	28	--	--
45	29	--	--
46	30	--	--
47	31	EXT_INT	External signal interrupt from 0x50001300 bit[5]

Figure 10.4 Interrupt Map Vector Table

10.4 Reset source

Reset circuitry allows A8107M0 to be easily placed in a predefined default condition. LVD, Reset, POR, and Watchdog signal will reset A8107M0 when they happen.

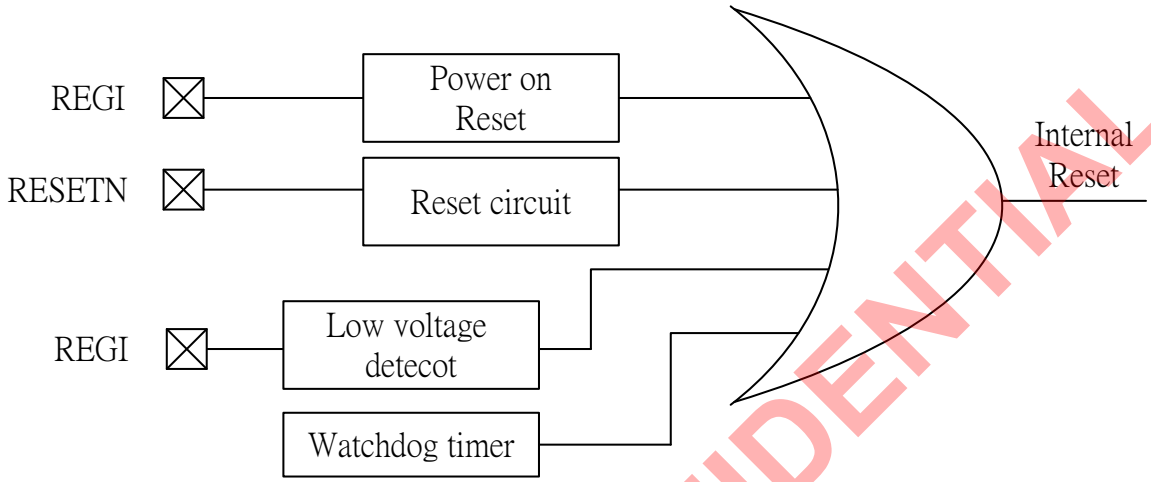


Figure 10.5 Reset source

Reset Flag : Please refer to Chap 9.2.3
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 reset
LVDF (Low voltage detect) flag
= 1: Occurred Low Voltage Reset
= 0: No Low Voltage reset

10.5 Clock source

A8107M0 has three clock source, crystal oscillator (pin 12,13/ Xi, XO), RTC crystal (pin 33,34/ P0.22, P0.23/ RTC_I, RTC_O) and internal RC oscillator. In the MCU part (digital peripherals), user choices the suitable clock source by power consumptions and performance. In the RF part, the clock source only comes from XO..

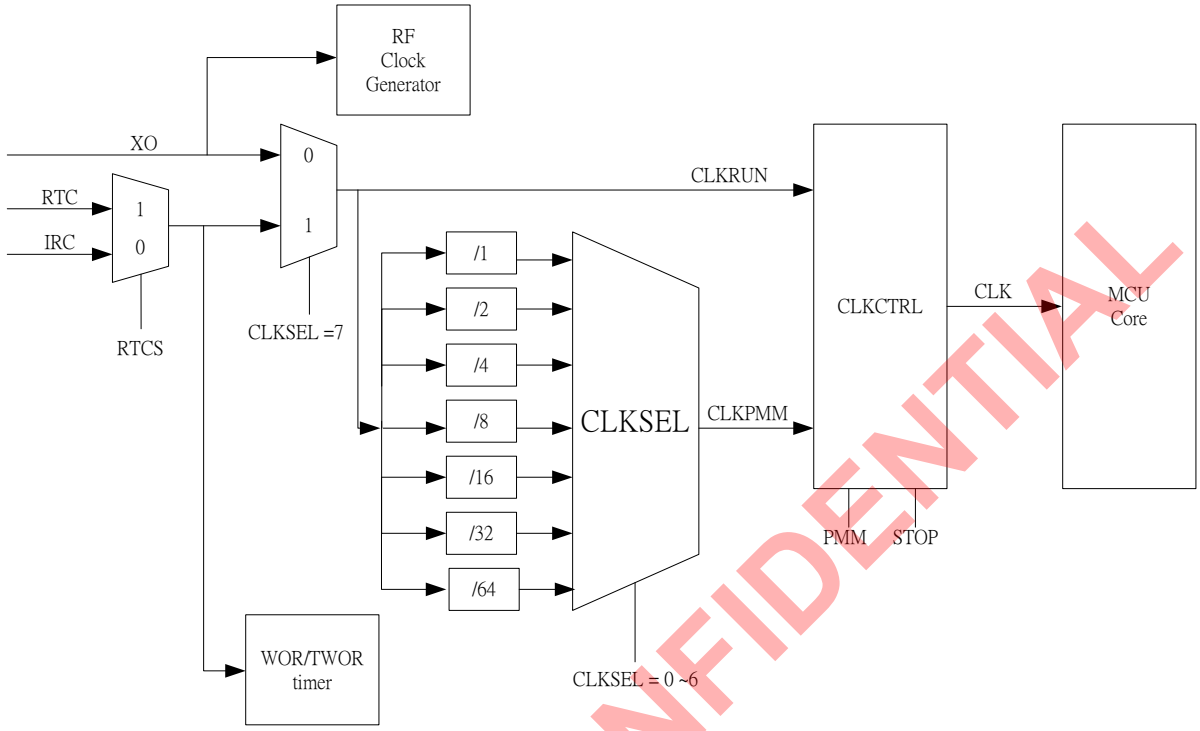


Figure 10.6 Whole chip clock

10.6 System Timer (SysTick)

The SysTick timer is a 24-bit down counter. It reloads automatically after reaching zero, and the reload value is programmable. When reaching zero, the timer can generate a SysTick exception (exception number 15). For the Cortex-M0 processor, a simple timer called the SysTick is included to generate this regular interrupt request.

10.6.1 Register Overview

Name	Address	R/W	Description	Reset value
SYST Base Address = 0xE000_E010				
SYST_CSR	SYST_BA+0x00	R/W	SysTick Control and Status Register	32'h0000_0000
SYST_RVR	SYST _BA+0x04	R/W	SysTick Reload Value Register	32'h0000_0000
SYST_CVR	SYST _BA+0x08	R/W	SysTick Current Value Register	32'h0000_0000
SYST_CALIB	SYST _BA+0x0C	R	SysTick Calibration Value Register	32'h0000_0000

10.6.2 SysTick Control and Status Register (Address: 0xE000E010h)

Name	Address	R/W	Description	Reset value
SYST_CSR	SYST_BA+0x00	R/W	SysTick Control and Status Register	32'h0000_0000

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	--							
R								
R/W	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
W	--							COUNTFLAG

R								
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W R	--							
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W R	--					CLKSOURCE	TICKINT	ENABLE

COUNTFLAG Returns 1 if timer counted to 0 since the last read of this register.
CLKSOURCE Selects the SysTick timer clock source
0 = external reference clock
1 = processor clock.
If your device does not implement a reference clock, this bit reads-as-one and ignores writes.
TICKINT Enables SysTick exception request:
0 = counting down to zero does not assert the SysTick exception request
1 = counting down to zero to asserts the SysTick exception request.
ENABLE Enables the counter:
0 = counter disabled
1 = counter enabled.

10.6.3 SysTick Reload Value Register (Address: 0xE000E014h)

Name	Address	R/W	Description	Reset value
SYST_RVR	SYST _BA+0x04	R/W	SysTick Reload Value Register	32'h0000_0000

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

RELOAD Value to load into the SYST_CVR when the counter is enabled and when it reaches 0, see Calculating the RELOAD value.

10.6.4 SysTick Current Value Register (Address: 0xE000E018h)

Name	Address	R/W	Description	Reset value
SYST_CVR	SYST _BA+0x08	R/W	SysTick Current Value Register	32'h0000_0000

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W R	--							
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W R	CURRENT[23:16]							
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W R	CURRENT[15:8]							

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	CURRENT[7:0]							
R								

CURRENT Reads return the current value of the SysTick counter.

10.6.5 SysTick Calibration Value Register (Address: 0xE000E01Ch)

Name	Address	R/W	Description	Reset value
SYST_CALIB	SYST _BA+0x0C	R	SysTick Calibration Value Register	32'h0000_0000

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	NOREF	SKEW	--					
R								
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W	TENMS[23:16]							
R								
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	TENMS[15:8]							
R								
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	TENMS[7:0]							
R								

NOREF Indicates whether the device provides a reference clock to the processor:
0 = reference clock provided
1 = reference clock provided.
If your device does not provide a reference clock, the SYST_CSR.CLKSOURCE bit reads-as-one and ignores writes.
SKEW Indicates whether the TENMS value is exact:
0 = TENMS value is exact
1 = TENMS value is inexact, or not given.
An inexact TENMS value can affect the suitability of SysTick as a software real time clock.
TENMS Reload value for 10ms (100Hz) timing, subject to system clock skew errors.
If the value reads as zero, the calibration value is not known.
If calibration information is not known, calculate the calibration value required from the frequency of the processor clock or external clock.

10.7 RTC (REAL TIMER CLOCK) TIMER WITH ALARM

10.7.1 REAL TIME CLOCK TIMER WITH ALARM FUNCTION

A8107M0 real time clock timer has 6 bits counter for seconds, 6 bits counter forminutes, 5 bits counter for hours and 3bits counter for weeks. This function enable if RTCEN =1 and the counters will update every one second. There is other counters for 10 milliseconds and it is separated with RTC counters. The clock source of RTC will be choice 32KHZ, 32.768KHz or A8107M0 internal RC clock.

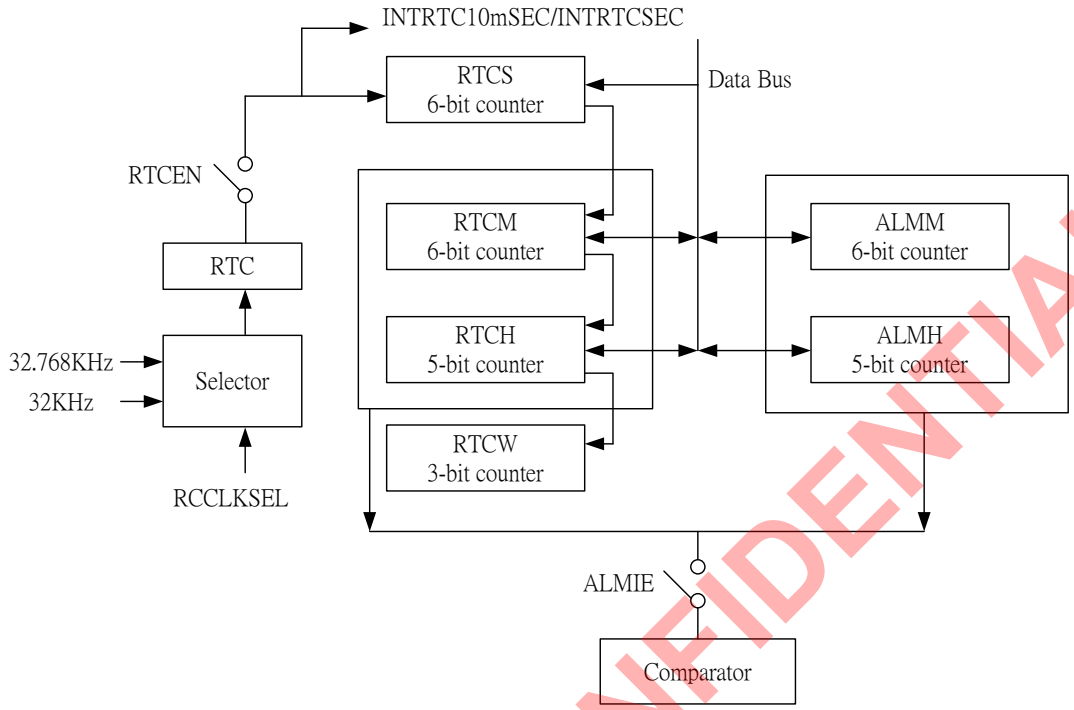


Figure 10.7 RTC architecture overview

10.7.2 REAL TIME CLOCK TIMER WITH ALARM OPERATION

A8107M0 real time clock timer is 6-bits alarm minute counter, 5-bits alarm hour counter. When enable Alarm (ALMIE=1), real time clock timer will compare with alarm register, an interrupt request signal, RTCINT, when the timer counter is equal to alarm register.

Real time clock timer and alarm register setting operation flow :

1. Enable real time clock (RTCON=1).
2. Set real time clock timer value(RTCH, RTCM, RTCS register).
3. Enable real time clock timer counting (RTCEN=1).
4. Set alarm time value (ALMH, ALMM register).
5. Enable alarm interrupt (ALMIE=1).

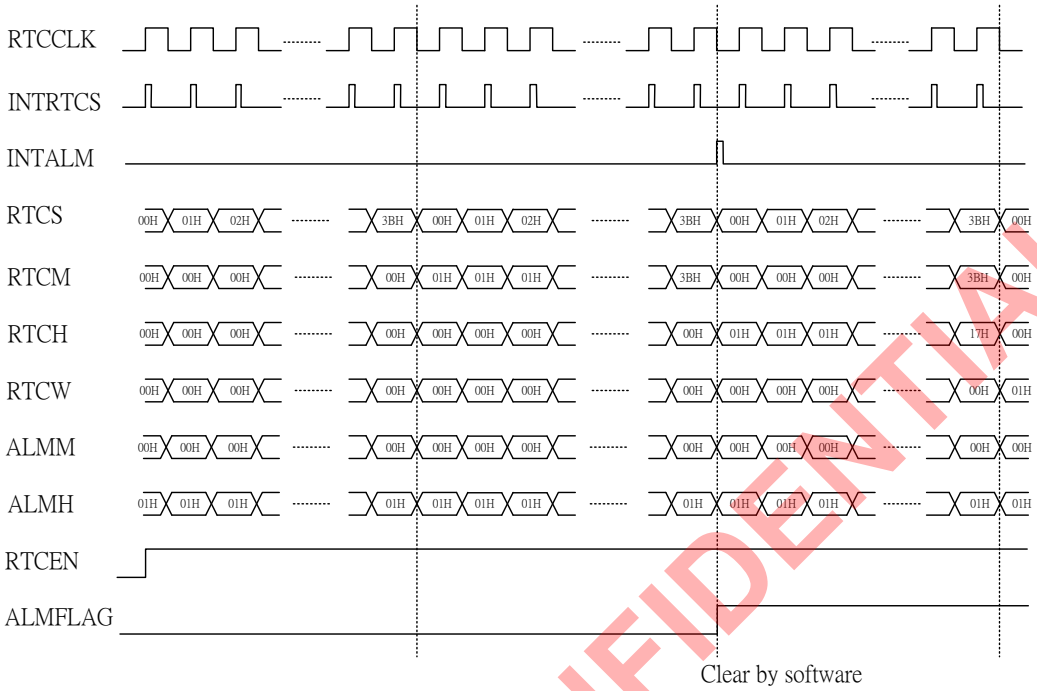


Figure 10.8 Real time clock timer with alarm Operation Timing

10.7.3 REAL TIME CLOCK TIMER WITH ALARM CONTROL REGISTER

Please refer. to chap 9.2.72 ~ 76.

11. I/O Ports

A8107M0 has 32pins digital I/O Pins 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, I^2C and SPI functions. Thus, each pin can also be used to wake up from sleep MODE. User can select each pin function by setting register. Each port has itself port register like P0 (0x40010000), 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 (PUN), Output-enable (OE) and Wake-up enable (WUN). 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.

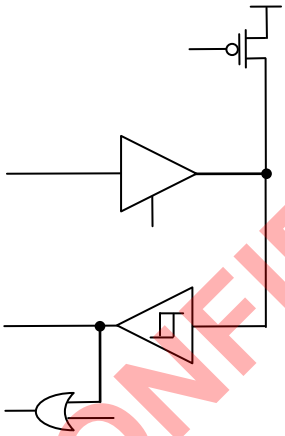


Figure11.1 Ports I/O block diagram

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

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

WUN	KEYINT
0	Enable
1	Disable

Table 11.2 WUN setting and KEYINT source

11.1 FUNCTIONALITY

It has 32-bit full bi-directional ports, P0. Each port bit can be individually accessed by bit addressable instructions.

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40010000h P0[31:24]	R/W	P0[31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40010000h P0[23:16]	R/W	P0[23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**

0x40010000h P0[15:8]	R/W	P0[15:8]							
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**
0x40010000h P0[7:0]	R/W	P0[7:0]							
RESET		0	0	0	0	0	0	0	0

Port 0 register

According the Table 11.1, all Port pins can be configured as Output, Input with the 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 PUN =1 or 0 depending on application. When OE =0, PUN=0 is recommended for saving power.

All Port pins can wake A8107m0 up when WUEN=0 and configured GPIO. All Port pins' WUN signals connect one OR gate to KEYINT. It means pin wake up function needs KEYINT ISR to take care this interrupt event.

Address name	**R/W**	**Bit 31**	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
0x40010010h P0OE[31:24]	R/W	P0OE[31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40010010h P0OE[23:16]	R/W	P0OE[23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40010010h P0OE[15:8]	R/W	P0OE[15:8]							
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**
0x40010010h P0OE[7:0]	R/W	P0OE[7:0]							
RESET		0	0	0	0	0	0	0	0

Port 0 Output Enable Register

Address name	**R/W**	**Bit 31**	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
0x40010008h P0PUN[31:24]	R/W	P0PUN[31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40010008h P0PUN[23:16]	R/W	P0PUN[23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40010008h P0PUN[15:8]	R/W	P0PUN[15:8]							
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**
0x40010008h P0PUN[7:0]	R/W	P0PUN[7:0]							
RESET		0	0	0	0	0	0	0	0

Port 0 Pull Up Register

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x4001000Ch P0WUN[31:24]	R/W	P0WUN[31:24]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x4001000Ch P0WUN[23:16]	R/W	P0WUN[23:16]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x4001000Ch P0WUN[15:8]	R/W	P0WUN[15:8]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0x4001000Ch P0WUN[7:0]	R/W	P0WUN[7:0]							
RESET		1	1	1	1	1	1	1	1

Port 0 Wake Up Enable Register

ADCCH Register (0x50008008)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x50008008h ADCCH	R/W	--	--	--	--	ADCCH3	ADCCH2	ADCCH1	ADCCH0
Reset		0	0	0	0	0	0	0	0

ADCCH[3:1] (ADC I/O select)
[000]: Select P0.18 as ADC analog input.
[001]: Select P0.19 as ADC analog input.
[010]: Select P0.8 as ADC analog input.
[011]: Select P0.9 as ADC analog input.
[100]: Select P0.12 as ADC analog input.
[101]: Select P0.13 as ADC analog input.
[110]: Select P0.14 as ADC analog input.
[111]: Select P0.15 as ADC analog input.
ADCCH0
[1]: Enable ADC analog input
[0]: Disable ADC analog input

11.2 Key interrupt

User can use P0 port as key input and meanwhile these key are clicked to event a key interrupt to wake up A8107M0 or enter key process flow. It is a helpful use to design a remote controller and low power consumption with power saving MODE setting.

The KEY interrupts can wake up A8107M0 back to normal MODE in PM1 and PM2. In PM3, Port 0.26~Port 0.29 and RESETN PIN will reset A8107M0 and A8107M0 need to initial all needed peripherals and take care key interrupt event.

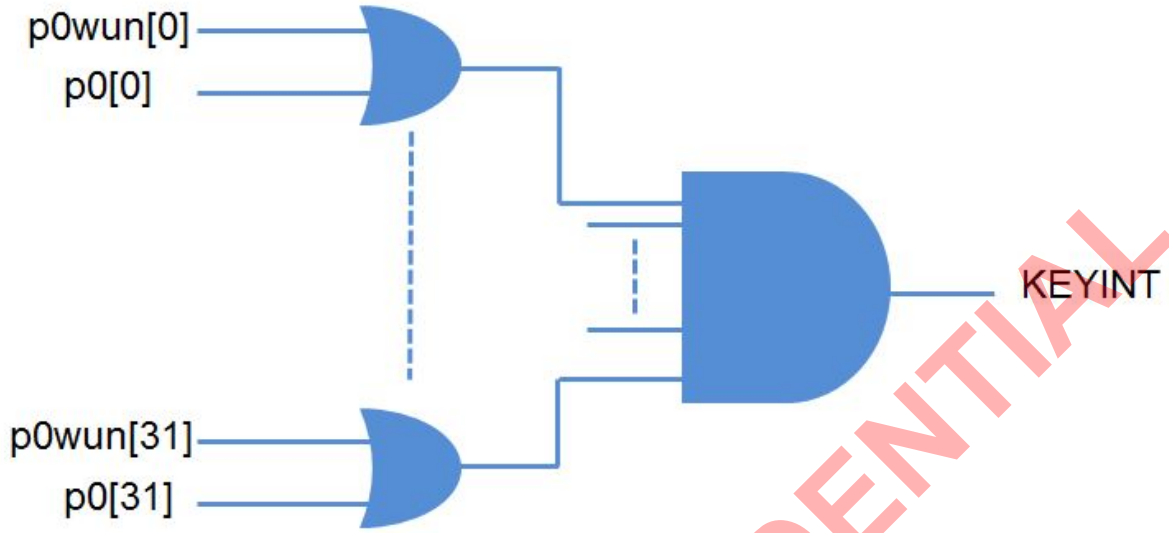


Figure11.2 Key interrupt block diagram

11.3 GPIO interrupt trigger

The P0 port interrupt generation support level / edge interrupt trigger. The register is as below:

Interrupt Type (0x40010028h):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40010028h P0TYPE[31:24]	R/W	P0TYPE[31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40010028h P0TYPE [23:16]	R/W	P0TYPE [23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40010028h P0TYPE [15:8]	R/W	P0TYPE [15:8]							
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**
0x40010028h P0TYPE [7:0]	R/W	P0TYPE [7:0]							
RESET		0	0	0	0	0	0	0	0

P0TYPE[31:0] : Interrupt type set
[0] : level
[1] : edge

Interrupt Polarity-level,edge (0x40010030h):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40010030h P0POL [31:24]	R/W	P0POL [31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**

0x40010030h P0POL [23:16]	R/W	P0POL [23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40010030h P0POL [15:8]	R/W	P0POL [15:8]							
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**
0x40010030h P0POL [7:0]	R/W	P0POL [7:0]							
RESET		0	0	0	0	0	0	0	0

P0POL[31:0] : Polarity-level, edge IRQ configuration set
[0] : low level or falling edge
[1] : high level or rising edge

Type	Pol	Interrupt
0	0	Low level
0	1	High Level
1	0	Falling Edge
1	1	Rising Edge

12 Timer 0 & 1

A8107M0 contains two 32-bit timer/counters, Timer 0 and Timer 1.

12.1 Overview

- an interrupt request signal, **TIMERINT,** when the counter reaches 0.
  The interrupt request is held until it is cleared by writing to the INTCLEAR Register
- If the timer count reaches 0 and, at the same time, the software clears a previous interrupt status, the interrupt status is set to 1.
- A timer enable, **EXTIN(P0_8/P0_9),** can use the zero to one transition of the external input signal
- The external clock, **EXTIN**, must be slower than half of the peripheral clock because it is sampled by a double flip-flop and then goes through edge-detection logic when the external inputs act as a clock

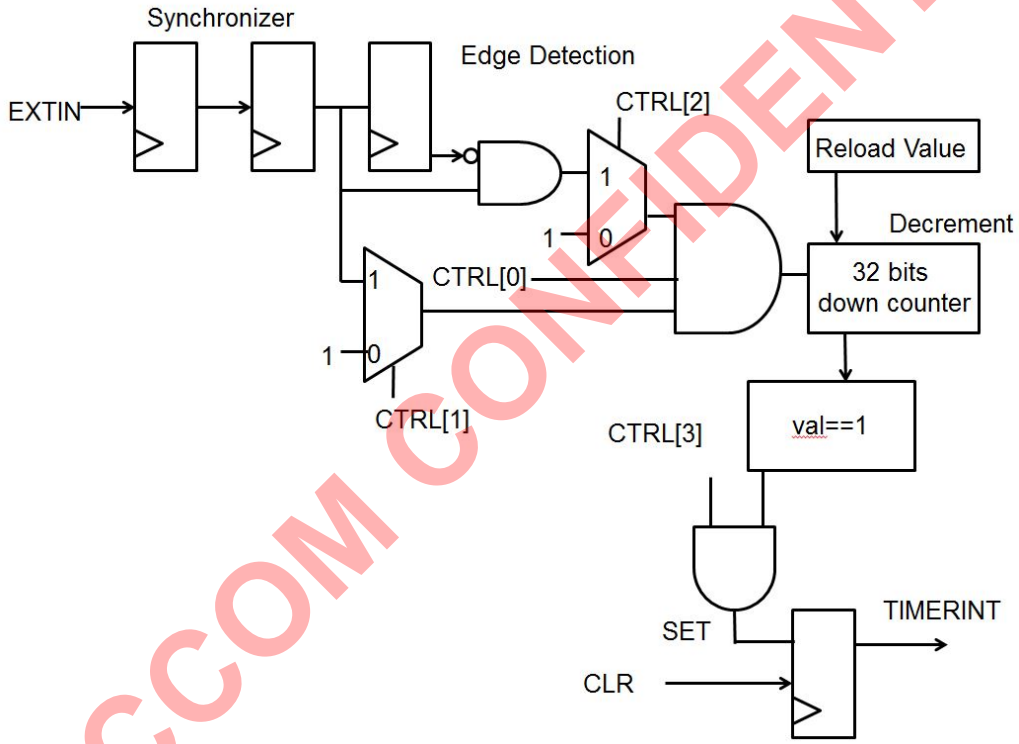


Figure12.1 Timer 32-Bit Timer/Counter

12.2 Timer 0 & 1 Register

Timer 0 & 1 Control (0x40000000 & 0x40001000):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
CTRL[31:24]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
CTRL[23:16]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
CTRL[15:8]	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
CTRL[7:0]	R/W					CTRL3	CTRL2	CTRL1	CTRL0
RESET		0	0	0	0	0	0	0	0

CTRL0: Enable.
CTRL1: Select external input as enable.
CTRL2: Select external input as clock.
CTRL3: Timer interrupt enable.

Timer 0 & 1 Value (0x40000004 & 0x40001004):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
VALUE[31:24]	R/W	VALUE[31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
VALUE[23:16]	R/W	VALUE[23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
VALUE[15:8]	R/W	VALUE[15:8]							
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**
VALUE[7:0]	R/W	VALUE[7:0]							
RESET		0	0	0	0	0	0	0	0

VALUE[31:0]: Current value.

Timer 0 & 1 Reload (0x40000008 & 0x40001008):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
RELOAD [31:24]	R/W	RELOAD [31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
RELOAD [23:16]	R/W	RELOAD [23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
RELOAD [15:8]	R/W	RELOAD [15:8]							
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**
RELOAD [7:0]	R/W	RELOAD [7:0]							
RESET		0	0	0	0	0	0	0	0

RELOAD [31:0]: Reload value. Set the current value.

Timer 0 & 1 Interrupt Status (0x4000000C & 0x4000100C):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
INT_STATUS [31:24]	R/W	--							
RESET		0	0	0	0	0	0	0	0

Address Name	R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
INT_STATUS [23:16]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
INT_STATUS [15:8]	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**
INT_STATUS [7:0]	R/W								INTSTATUS
RESET		0	0	0	0	0	0	0	0

INTSTATUS: Timer interrupt.
[1]: Clear.

13 Dual Timer

A8107M0 contains two 32-bit down-counters., dual-input timer module, is an APB slave module consisting of two programmable 32-bit down-counters that can generate interrupts when they reach 0.

13.1 Dual Timer FUNCTIONALITY

Two timers are defined by default, although you can easily expand this using extra instantiations of the FRC block. The same principle of simple expansion is used for the register configuration, to enable you to use more complex counters. For each timer, the following MODEs of operation are available:

- **Free-running MODE**
  The counter wraps after reaching its zero value, and continues to count down from the maximum value. This is the default MODE.

- **Periodic timer MODE**
  The counter generates an interrupt at a constant interval, reloading the original value after wrapping past zero.

- **One-shot timer MODE**
  The counter generates an interrupt once. When the counter reaches 0, it halts until you reprogram it. You can do this using one of the following:
  - Clearing the one-shot count bit in the control register, in which case the count proceeds according to the selection of Free-running or Periodic MODE.
  - Writing a new value to the Load Value register.

13.2 Dual Timer Operation

Each timer has an identical set of registers The operation of each timer is identical. The timer is loaded by writing to the load register and, if enabled, counts down to 0. When a counter is already running, writing to the load register causes the counter to immediately restart at the new value. Writing to the background load value has no effect on the current count. The counter continues to decrement to 0, and then restarts from the new load value, if in periodic MODE, and one-shot MODE is not selected.

When 0 is reached, an interrupt is generated. You can clear the interrupt by writing to the clear register. If you selected one-shot MODE, the counter halts when it reaches 0 until you deselect one-shot MODE, or write a new load value.

Otherwise, after reaching a zero count, if the timer is operating in free-running MODE, it continues to decrement from its maximum value. If you selected periodic timer MODE, the timer reloads the count value from the Load Register and continues to decrement. In this MODE, the counter effectively generates a periodic interrupt.

You select the MODE using a bit in the Timer Control Register. At any point, you can read the current counter value from the Current Value Register. You can enable the counter using a bit in the Control Register.

At reset, the counter is disabled, the interrupt is cleared, and the load register is set to 0. The MODE and prescale values are set to free-running, and clock divide of one respectively. Figure 4-7 shows a block diagram of the free-running timer module.

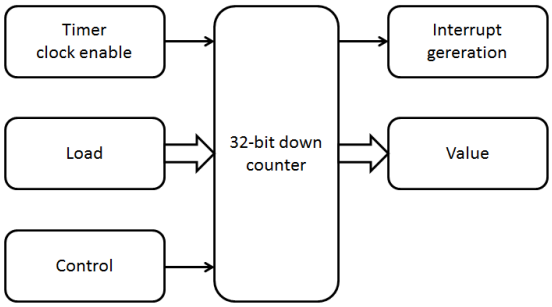


Figure13.1 Timer 32-Bit Timer/Counter

The timer clock enable is generated by a prescale unit. The counter then uses the enable to create a clock with one of the following timings:

- The system clock.
- The system clock divided by 16, generated by 4 bits of prescale.
- The system clock divided by 256, generated by 8 bits of prescale.

Figure 4-8 shows how the timer clock frequency is selected in the prescale unit. This enables you to clock the timer at different frequencies.

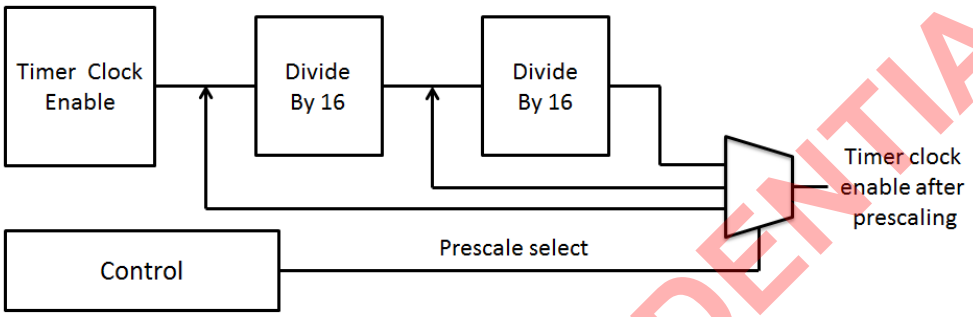


Figure13.2 Timer 32-Bit Timer/Counter

Interrupt generation

An interrupt is generated when the full 32-bit counter reaches 0, and is only cleared when the TimerXClear location is written to. A register holds the value until the interrupt is cleared. The most significant carry bit of the counter detects the counter reaching 0.

You can mask interrupts by writing 0 to the Interrupt Enable bit in the Control Register. You can read the following from status registers:

- Raw interrupt status, before masking.
- Final interrupt status, after masking.

The interrupts from the individual counters, after masking, are logically ORed into a combined interrupt, **TIMINTC**. This provides an additional output from the timer peripheral.

13.3 Dual Timer Register

Timer 1 Load (0x40002000):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40002000 T1LOAD [31:24]	R/W	T1LOAD [31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002000 T1LOAD [23:16]	R/W	T1LOAD [23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002000 T1LOAD [15:8]	R/W	T1LOAD [15:8]							
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**
0x40002000 T1LOAD [7:0]	R/W	T1LOAD [7:0]							
RESET		0	0	0	0	0	0	0	0

T1LOAD[31:0]: Timer 1 Load.

Timer 1 Value (0x40002004):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40002004 T1Val [31:24]	R	T1Val [31:24]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002004 T1Val [23:16]	R	T1Val [23:16]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002004 T1Val [15:8]	R	T1Val [15:8]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0x40002004 T1Val [7:0]	R	T1Val [7:0]							
RESET		1	1	1	1	1	1	1	1

T1Val [31:0]: Timer 1 value.

Timer 1 Control (0x40002008):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40002008 T1ctrl [31:24]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002008 T1ctrl [23:16]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002008 T1ctrl [15:8]	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**
0x40002008 T1ctrl [7:0]	R/W	T1_EN	T1_MD	T1INT_EN	--	T1_Pre		T1_Slt	T1_OS
RESET		0	0	1	0	0	0	0	0

T1_EN: Timer 1 enable.
[0]: disable.
[1]: enable.

T1_MD: Timer 1 MODE.
[0]: free-running MODE.
[1]: periodic MODE.

T1INT_EN: Timer 1 interrupt enable.
[0]: timer interrupt disable.
[1]: timer interrupt enable.

T1_Pre: Timer 1 prescale.

[00]: 0 stages of prescale, clock is divided by 1.
[01]: 4 stages of prescale, clock is divided by 16
[10]: 8 stages of prescale, clock is divided by 256
[11]: Undefined

T1_SIt: Timer 1 size selects 16-bit or 32-bit counter operation
[0]: 16-bit counter
[1]: 32-bit counter

T1_OS: Timer 1 one-shot count selects
[0]: Wrapping MODE
[1]: One-shot MODE

Timer 1 Interrupt Clear (0x4000200C):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x4000200C T1INTC [31:24]	W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x4000200C T1INTC [23:16]	W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x4000200C T1INTC [15:8]	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**
0x4000200C T1INTC [7:0]	W	T1INTC [7:0]							
RESET		0	0	0	0	1	1	0	0

T1INTC: Timer 1 interrupt clear.
Timer 1 Raw Interrupt Status (0x40002010):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40002010 T1RIS [31:24]	R	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002010 T1RIS [23:16]	R	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002010 T1RIS [15:8]	R	--							
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**
0x40002010 T1RIS [7:0]	R	T1RIS [7:0]							
RESET		0	0	0	0	1	1	0	0

T1RIS[7:0]: Raw timer interrupt status from the counter.

Timer 1 Interrupt Enabled Status (0x40002014):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40002014 T1MIS [31:24]	R	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002014 T1MIS [23:16]	R	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002014 T1MIS [15:8]	R	--							
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**
0x40002014 T1MIS [7:0]	R	T1MIS [7:0]							
RESET		0	0	0	0	1	1	0	0

T1MIS[7:0]: Timer 1 interrupt enabled status from the counter.

Timer 1 BG Load (0x40002018):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40010018 T1BGL [31:24]	R/W	T1BGL [31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002018 T1BGL [23:16]	R/W	T1BGL [23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002018 T1BGL [15:8]	R/W	T1BGL [15:8]							
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**
0x40002018 T1BGL [7:0]	R/W	T1BGL [7:0]							
RESET		0	0	0	0	1	1	0	0

T1BGL[31:0]: Timer 1 BG load.

Timer 2 Load (0x40002020):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x T2LOAD [31:24]	R/W	T2LOAD [31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**

0x T2LOAD [23:16]	R/W	T2LOAD [23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x T2LOAD [15:8]	R/W	T2LOAD [15:8]							
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**
0x T2LOAD [7:0]	R/W	T2LOAD [7:0]							
RESET		0	0	0	0	0	0	0	0

T2LOAD[31:0]: Timer 2 Load.

Timer 2 Value (0x40002024):

Address Name	**R/W**	**Bit 31**	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
0x40002024 T2Val [31:24]	R	T2Val [31:24]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002024 T2Val [23:16]	R	T2Val [23:16]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002024 T2Val [15:8]	R	T2Val [15:8]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
0x40002024 T2Val [7:0]	R	T2Val [7:0]							
RESET		1	1	1	1	1	1	1	1

T2Val [31:0]: Timer 2 value.

Timer 2 Control (0x40002028):

Address name	**R/W**	**Bit 31**	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
0x40002028 T2ctrl [31:24]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002028 T2ctrl [23:16]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002028 T2ctrl [15:8]	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**

0x40002028 T2ctrl [7:0]	R/W	T2_EN	T2_MD	T2INT_EN	--	T2_Pre		T2_Slt	T2_OS
RESET		0	0	1	0	0	0	0	0

T2_EN: Timer 2 enable.
[0]: disable.
[1]: enable.

T2_MD: Timer 2 MODE.
[0]: free-running MODE.
[1]: periodic MODE.

T2INT_EN: Timer 2 interrupt enable.
[0]: timer interrupt disable.
[1]: timer interrupt enable.

T1_Pre: Timer 2 prescale.
[00]: 0 stages of prescale, clock is divided by 1.
[01]: 4 stages of prescale, clock is divided by 16.
[10]: 8 stages of prescale, clock is divided by 256.
[11]: Undefined.

T2_Slt: Timer 2 size selects 16-bit or 32-bit counter operation.
[0]: 16-bit counter.
[1]: 32-bit counter.

T2_OS: Timer 2 one-shot count selects.
[0]: Wrapping MODE.
[1]: One-shot MODE.

Timer 2 Interrupt Clear (0x4000202C):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x4000202C T2INTC [31:24]	W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x4000202C T2INTC [23:16]	W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x4000202C T2INTC [15:8]	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**
0x4000202C T2INTC [7:0]	W	T2INTC [7:0]							
RESET		0	0	0	0	1	1	0	0

T2INTC: Timer 1 interrupt clear

Timer 2 Raw Interrupt Status (0x40002030):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40002030 T2RIS [31:24]	R	--							
RESET		0	0	0	0	0	0	0	0

Address Name	R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
0x40002030 T2RIS [23:16]	R	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002030 T2RIS [15:8]	R	--							
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**
0x40002030 T2RIS [7:0]	R	T2RIS [7:0]							
RESET		0	0	0	0	1	1	0	0

T2RIS[7:0]: Raw timer interrupt status from the counter.

Timer 2 Interrupt Enabled Status (0x40002034):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40002034 T2MIS [31:24]	R	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002034 T2MIS [23:16]	R	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002034 T2MIS [15:8]	R	--							
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**
0x40002034 T2MIS [7:0]	R	T1MIS [7:0]							
RESET		0	0	0	0	1	1	0	0

T2MIS[7:0]: Timer interrupt enabled status from the counter.

Timer 2 BG Load (0x40002038):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40002038 T2BGL [31:24]	R/W	T2BGL [31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002038 T2BGL [23:16]	R/W	T2BGL [23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002038 T2BGL [15:8]	R/W	T2BGL [15:8]							
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
0x40002038 T2BGL [7:0]	R/W	T2BGL [7:0]							
RESET		0	0	0	0	1	1	0	0

T2BGL[31:0]: Timer 2 BG load.

Timer 1 Integration Test MODE Enable (0x40002F00):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40002F00 T1ITME [31:24]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002F00 T1ITME [23:16]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002F00 T1ITME [15:8]	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**
0x40002F00 T1ITME [7:0]	R/W	--							T1ITME
RESET		0	0	0	0	0	0	0	0

T1ITME: Timer 1 integration test MODE enable.

Timer Integration Test MODE Enable (0x40002F00):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40002F00 TIMERITCR [31:24]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002F00 TIMERITCR [23:16]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002F00 TIMERITCR [15:8]	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**
0x40002F00 TIMERITCR [7:0]	R/W	--							T1ITME
RESET		0	0	0	0	0	0	0	0

TIMERITCR: Timer integration test MODE enable.
[1] integration test MODE.

Timer Integration Test Select (0x40002F04):

Address Name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x40002F04 TIMERITOP[31:24]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x40002F04 TIMERITOP[23:16]	R/W	--							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x40002F04 TIMERITOP[15:8]	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**
0x40002F04 TIMERITOP[7:0]	R/W	--						TIMERTOP [1:0]	
RESET		0	0	0	0	0	0	0	0

TIMERITOP[1:0]: Timer Integration Test Select.
TIMERITOP[1]: Integration test TIMINT2 value. Value output on TIMINT2 when in integration test MODE.
TIMERITOP[0]: Integration test TIMINT1 value. Value output on TIMINT1 when in integration test MODE.

14. UART 0 & 1 & 2

A8107M0 contains three UART, UART 0(P0.16/P0.17), UART 1(P0.18/P0.19) and UART 2(P0.20/P0.21), The APB UART, is a simple design that supports 8-bit communication without parity, and is fixed at one stop bit per configuration. Figure 14-1 shows the APB UART module

14.1 Overview

The APB UART contains buffering

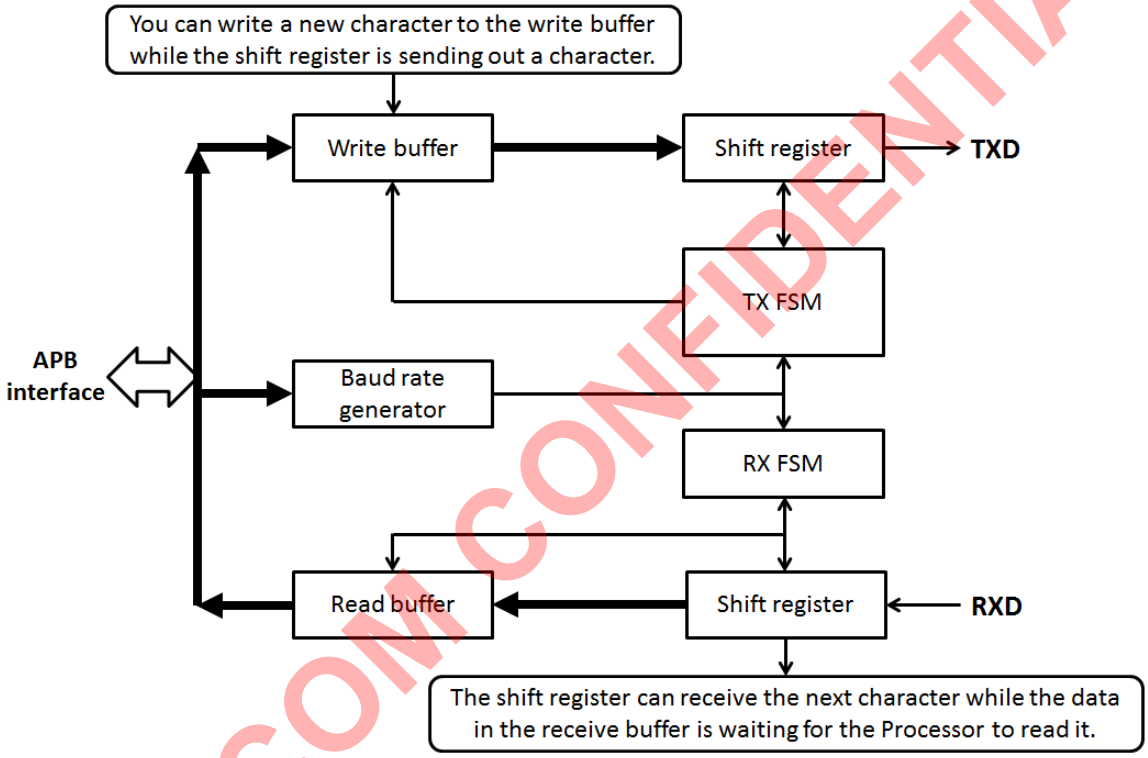


Figure14.1 APB UART module.

PIN	ACTIVE	TYPE	DESCRIPTION
Rxd_0(P0.16)	-	Input / Output	Serial receiver I_0 / O_0
Txd_0(P0.17)	-	Output	Serial transmitter line 0

Table14.1 UART0 pins description

PIN	ACTIVE	TYPE	DESCRIPTION
Rxd_0(P0.18)	-	Input / Output	Serial receiver I_0 / O_0
Txd_0(P0.19)	-	Output	Serial transmitter line 0

Table14.2 UART1 pins description

PIN	ACTIVE	TYPE	DESCRIPTION
Rxd_0(P0.20)	-	Input / Output	Serial receiver I_0 / O_0
Txd_0(P0.21)	-	Output	Serial transmitter line 0

Table14.3 UART2 pins description

This buffer arrangement is sufficient for most simple embedded applications. For example, a processor running at 30MHz with a baud rate of 115200 means a character transfer every 30e6×(1+8+1)/115200=2604 cycles. For duplex communication, the processor might receive an interrupt every 1300 clock cycles. Because the interrupt response time and the handler execution time are usually quite short, this leaves sufficient processing time for the thread.

The design includes a double flip-flop SYNChronization logic for the receive data input.

14.2 UART 0 Register

UART0 base address : 0x40004000h
UART1 base address : 0x40005000h
UART2 base address : 0x40006000h

UART Data (0x40004000h):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
Data[31:24]	R/W								
RESET		--	--	--	--	--	--	--	--
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
Data[23:16]	R/W								
RESET		--	--	--	--	--	--	--	--
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
Data[15:8]	R/W								
RESET		--	--	--	--	--	--	--	--
Address Name	**R/W**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Data[7:0]	R/W	Data[7:0]							
RESET		--	--	--	--	--	--	--	--

Data [7:0] : Data value.
Read : Received data.
Write : Transmit data.

UART State (0x40004004h):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
State[31:24]	R/W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
State[23:16]	R/W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
State[15:8]	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**
State[7:0]	R/W	--	--	--	--	State [3:0]			
RESET		0	0	0	0	0	0	0	0

State[3:0] : UART state.
[3] : RX buffer overrun, write 1 to clear.
[2] : TX buffer overrun, write 1 to clear.
[1] : RX buffer full, read-only.
[0] : TX buffer full, read-only.

UART Ctrl (0x40004008h):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
Ctrl[31:24]	R/W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
Ctrl[23:16]	R/W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
Ctrl[15:8]	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**
Ctrl[7:0]	R/W	--	--	Ctrl [5:0]					
RESET		0	0	0	0	0	0	0	0

Ctrl[5:0] : UART ctrl.
[5] : RX overrun interrupt enable.
[4] : TX overrun interrupt enable.
[3] : RX interrupt enable.
[2] : TX interrupt enable.
[1] : RX enable.
[0] : TX enable.

UART INTStatus&INTClear (0x4000400Ch):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
INT[31:24]	R/W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
INT[23:16]	R/W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
INT[15:8]	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**
INT[7:0]	R/W	--	--	--	--	INT [3:0]			
RESET		0	0	0	0	0	0	0	0

INT[3:0] : UART interrupt.
[3] : RX overrun interrupt. Write 1 to clear.
[2] : TX overrun interrupt. Write 1 to clear.
[1] : RX interrupt. Write 1 to clear.
[0] : TX interrupt. Write 1 to clear.

UART BaudDiv (0x40004010h):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
BaudDiv[31:24]	R/W	--	--	--	--	--	--	--	--

RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
BaudDiv[23:16]	R/W	--	--	--	--	BaudDiv[19:16]			
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
BaudDiv[15:8]	R/W	BaudDiv[15:8]							
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**
BaudDiv[7:0]	R/W	BaudDiv[7:0]							
RESET		0	0	0	0	0	0	0	0

BaudDiv[19:0] : Baud rate divider. The minimum number is 16.

The APB UART supports a high-speed test MODE, useful for simulation during SoC or ASIC development. When **CTRL[6]** is set to 1, the serial data is transmitted at one bit per clock cycle. This enables you to send text messages in a much shorter simulation time. If required, you can remove this feature for silicon products to reduce the gate count. You can do this by removing bit[6] of the **reg_ctrl** signal in the verilog code. The APB interface always sends an OKAY response with no wait state and is two cycles per transfer.

You must program the baud rate divider register before enabling the UART. For example, if the **PCLK** is running at 12MHz, and the required baud rate is 9600, program the baud rate divider register as 12,000,000/9600 = 1250.

The **BAUDTICK** output pulses at a frequency of 16 times that of the programmed baud rate. You can use this signal for capturing UART data in a simulation environment.

The **TXEN** output signal indicates the status of **CTRL[0]**. You can use this signal to switch a bidirectional I/O pin in a silicon device to UART data output MODE automatically when the UART transmission feature is enabled.

The buffer overrun status in the STATE field is used to drive the overrun interrupt signals. Therefore, clearing the buffer overrun status de-asserts the overrun interrupt, and clearing the overrun interrupt bit also clears the buffer overrun status bit in the STATE field.

15. IIC interface

A8107M0'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), Fast plus(1MHz)
- Multi-master systems supported
- Supports 7-bit and 10-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 A8107M0. The alternate function is Port 0.4 and Port 0.5. User can set Altfuncset (0x40010018H) to set up the PIN function. Please refer the Chapter 11 for more detail information.

PIN	TYPE	DESCRIPTION
SCL(P0.4)	INPUT /OUTPUT	I²C clock input /output
SDA(P0.5)	INPUT/ OUTPUT	I²C data input /output

Table15.1 I²C interface pins description.

15.1 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.

15.1.1 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	0x50003000
Control – I²CMCR	0x50003004
Transmitted data I²CBUF	0x50003008
Timer period - I²CMTP	0x5000300C
SCL low time-SCL_LP	0x50003010
SCL high time-SCL_HP	0x50003014
SDA setup time-SDA_SU	0x50003018
Interrupt – I²CMINT	0x5000301C

Table15.2 I²C Registers for writing.

.

Register	Address
Slave address – I²CMSA	0x50003000
Status – I²CMSR	0x50003004
Received data - I²CBUF	0x50003008
Timer period - I²CMTP	0x5000300C
SCL low time-SCL_LP	0x50003010
SCL high time-SCL_HP	0x50003014
SDA setup time-SDA_SU	0x50003018
Interrupt – I²CMINT	0x5000301C

Table15.3 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 + SCL_HP) x CLK_PRD
For example：

- CLK_PRD = 62.5ns (CLK_FRQ = 16MHz) ; - TIMER_PRD =3 ; - SCL_LP = 6; - SCL_HP = 4;
SCL_PERIOD = 2 x (1 + 3) x (6 + 4) x 62.5ns = 5000ns = 5us
SCL_FREQUENCY = 1 / 5us = 200 KHz
SCL_PRD - SCL line period (I^2C clock line)
TIMER PRD -Timer period register value (range 1 to 255)
CLK_PRD - System clock period ($1/f_{clk}$)

I^2CMTP (0x5000300C)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
5000300Ch I^2CMTP	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

SCL_LP (0x50003010)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003010h SCL_LP	R/W	-	-	-	-	lt3	lt2	lt1	lt0
Reset		0	0	0	0	0	1	0	0

SCL_HP (0x50003014)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003014h SCL_HP	R/W	-	-	-	-	ht3	ht2	ht1	ht0
Reset		0	0	0	0	0	0	1	0

SDA_SU (0x50003018)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003018h SDA_SU	R/W	-	-	-	-	dst3	dst2	dst1	dst0
Reset		0	0	0	0	0	0	1	0

■ I^2C 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 I^2C controller and behaves identically as external reset provided by RST pin. Using this bit software application can reinitialize I^2C mater module when some problem is encountered on I^2C bus. In case when I^2C Slave device bloCKS I^2C bus, then SLRST bit should be set along with RUN bit (just after issuing the RSTB). SLRST bit causes that I^2C 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 I^2C 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 I^2C MASTER MODULE core operates in Master receiver MODE the ACK bit must be set normally to logic 1. This cause the I^2C MASTER MODULE bus controller to send acknowledge automatically after each byte. This bit must be reset when the I^2C 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 I^2C MASTER MODULE ACK Polling chapter for details.

I^2CMCR (0x50003004)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003004h I^2CMCR	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	-	-	-	-	-	-	I^2CM module software reset
0	1	0	0	0	0	0	0	1	Reset slaves connected to I^2C 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	-	-	-	-	-	-	I^2CM 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	-	-	-	-	-	-	I^2CM 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.

I^2CMSR (0x50003004)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003004h I^2CMSR	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 I^2C BUS controller is in the IDLE state。
BUSY：This bit indicates that I^2C 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 I^2C 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 I^2C 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).

I^2CMSA (0x50003000)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003000h I^2CMCA	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

■ **I^2C Buffer – RECEIVER AND TRANSMITTER REGISTERS**

I^2C 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).

I^2CBUF (0x50003008)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003007h I^2CBUF	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 (0x50003008)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003008h 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

15.1.2 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

Table15.4 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

Table15.5 I²C MASTER MODULE Timer period values for Fast speed MODE.

15.1.3 I²C MASTER MODULE AVAILABLE COMMAND SEQUENCES

■ **I²C MASTER MODULE SINGLE SEND**

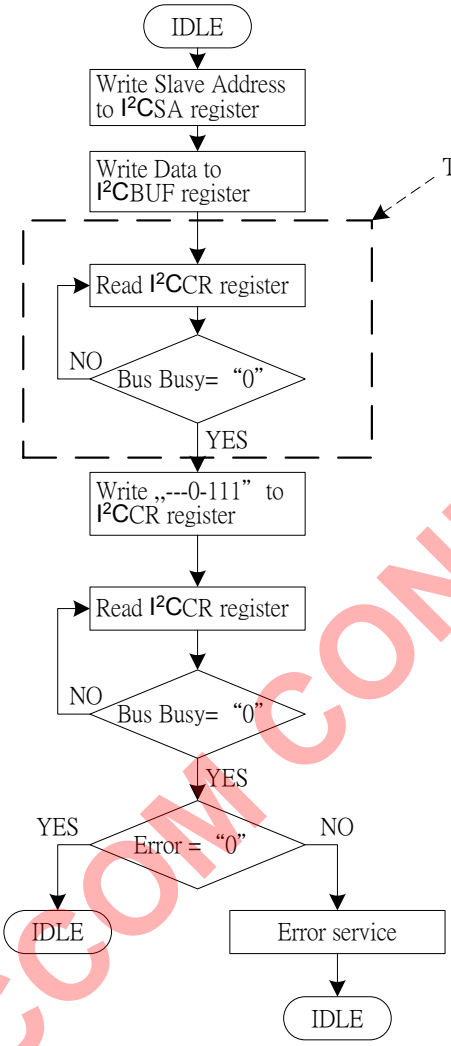


Figure15.1 I²C MASTER MODULE Single SEND flowchart

■ **I²C MASTER MODULE SINGLE RECEIVE**

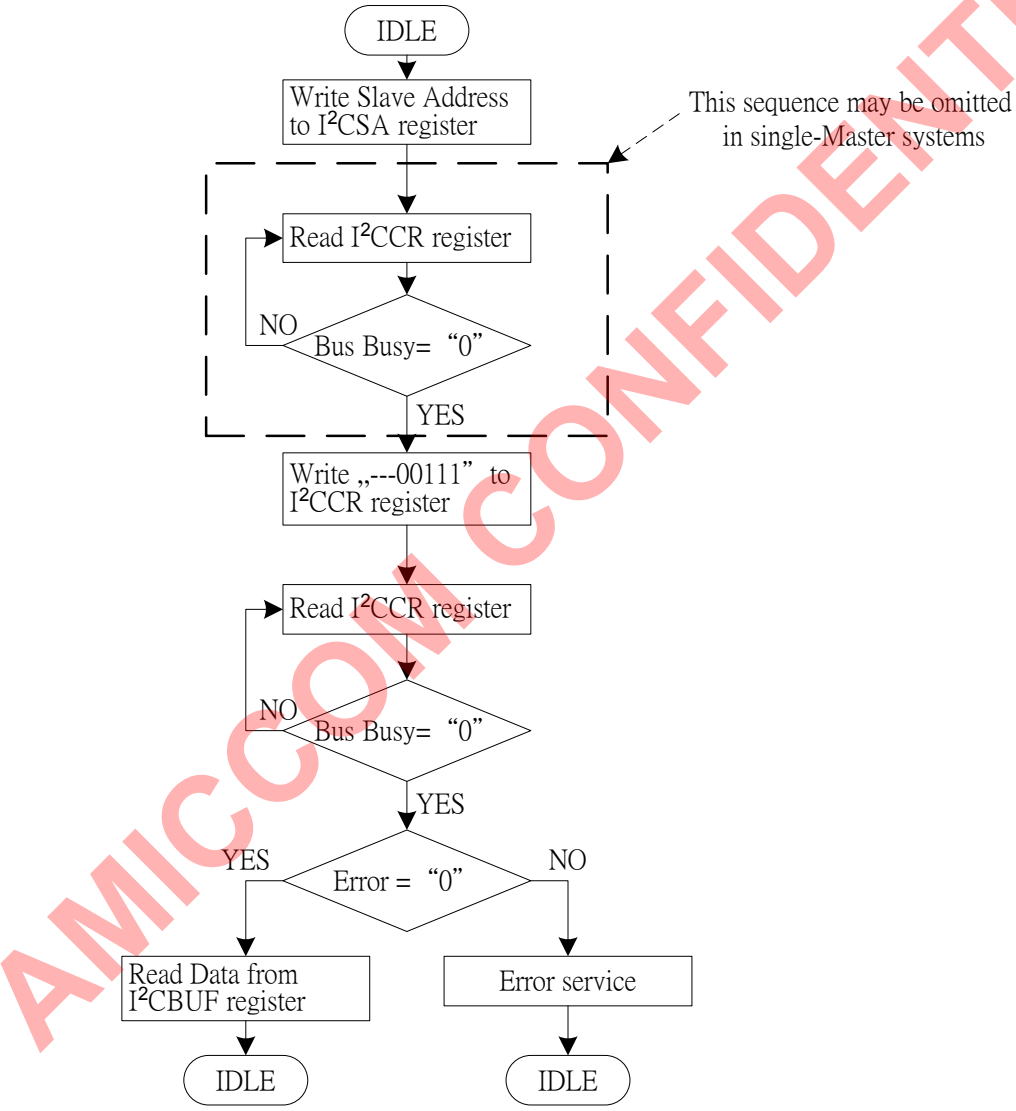


Figure15.2 Single RECEIVE flowchart

■ I²C MASTER MODULE BURST SEND

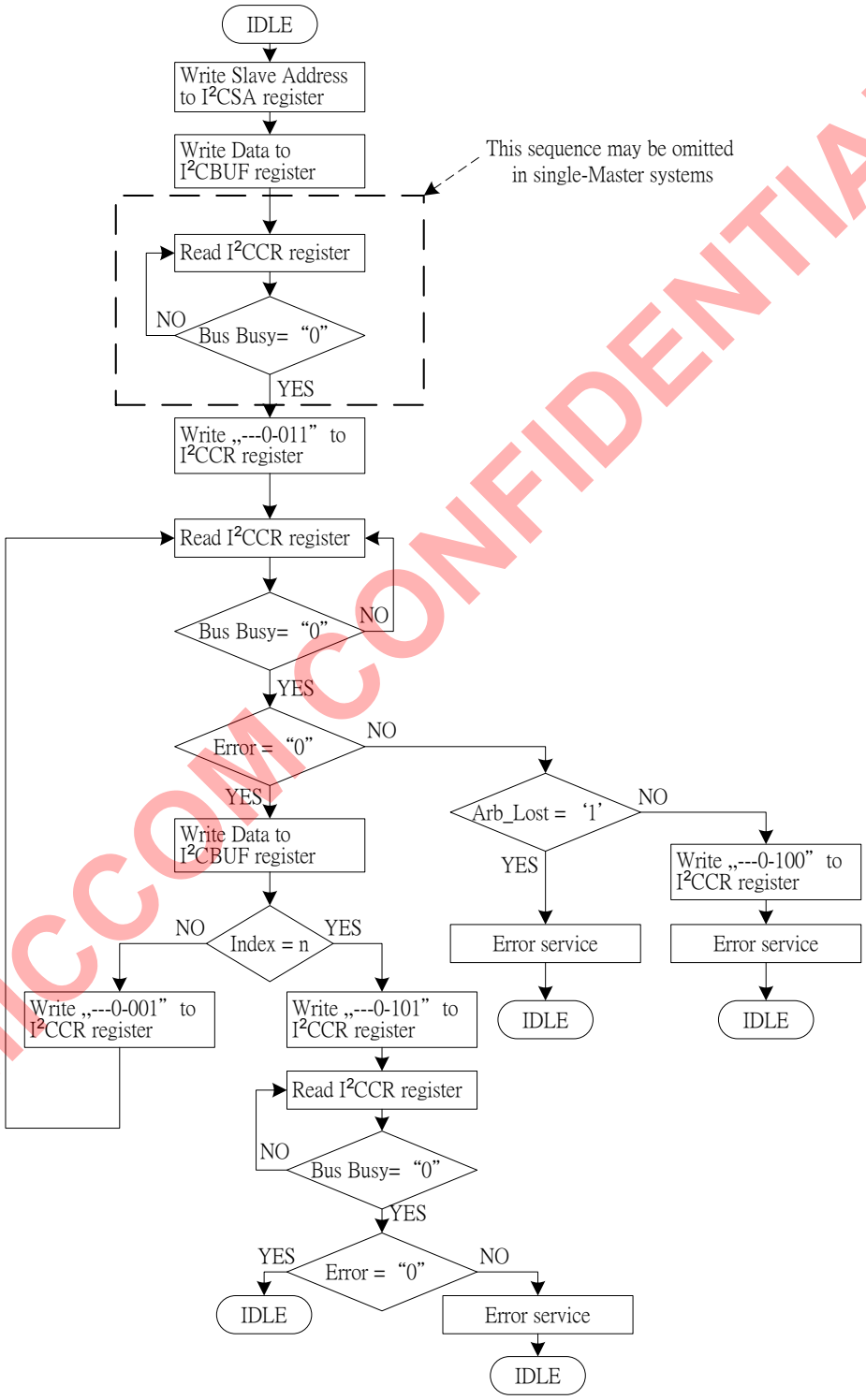


Figure15.3 I²C MASTER MODULE Sending n bytes flowchart

■ I²C MASTER MODULE BURST RECEIVE

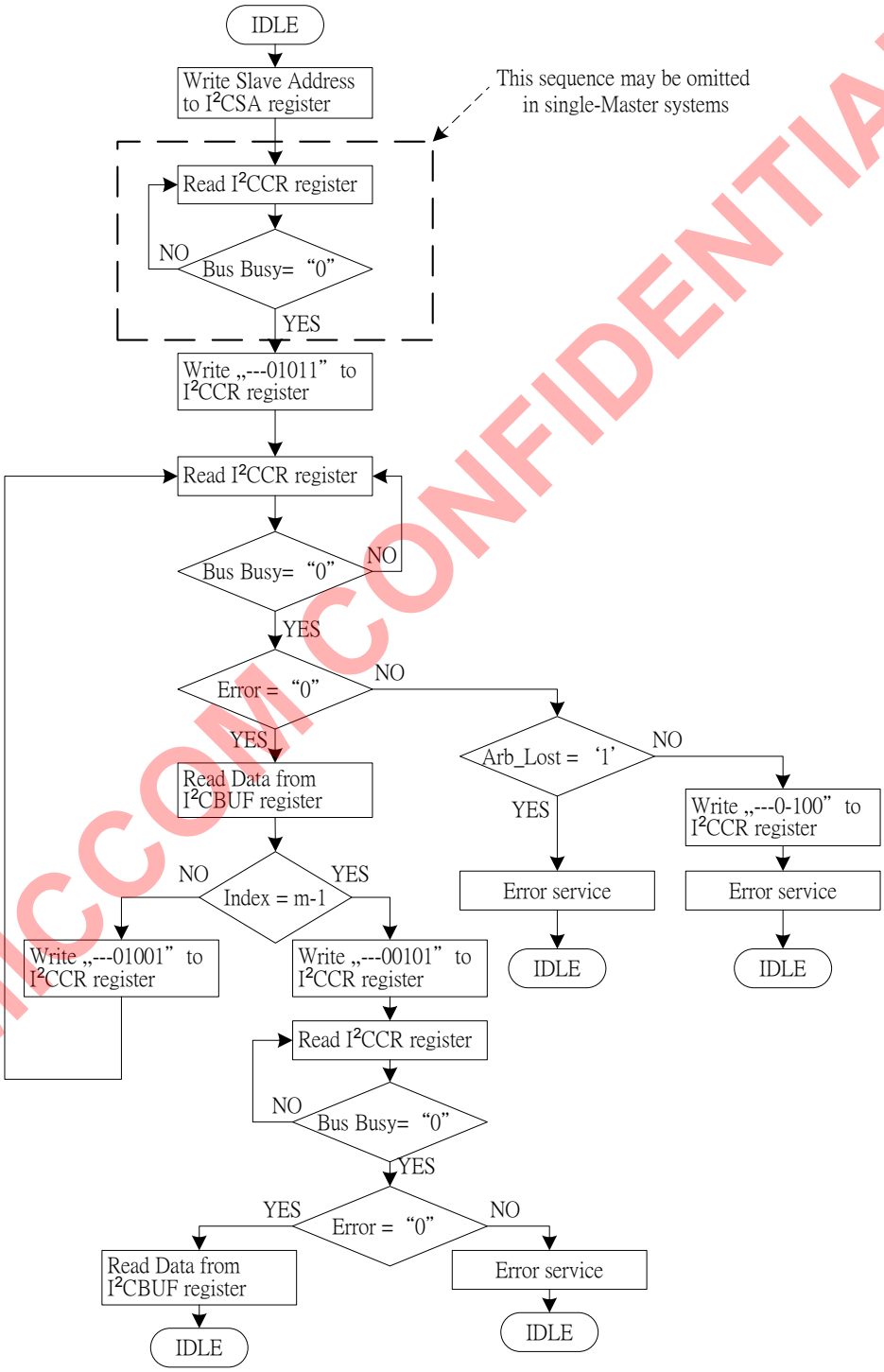


Figure15.4 I²C MASTER MODULE Receiving m bytes flowchart

■ I²C MASTER MODULE BURST RECEIVE AFTER BURST SEND

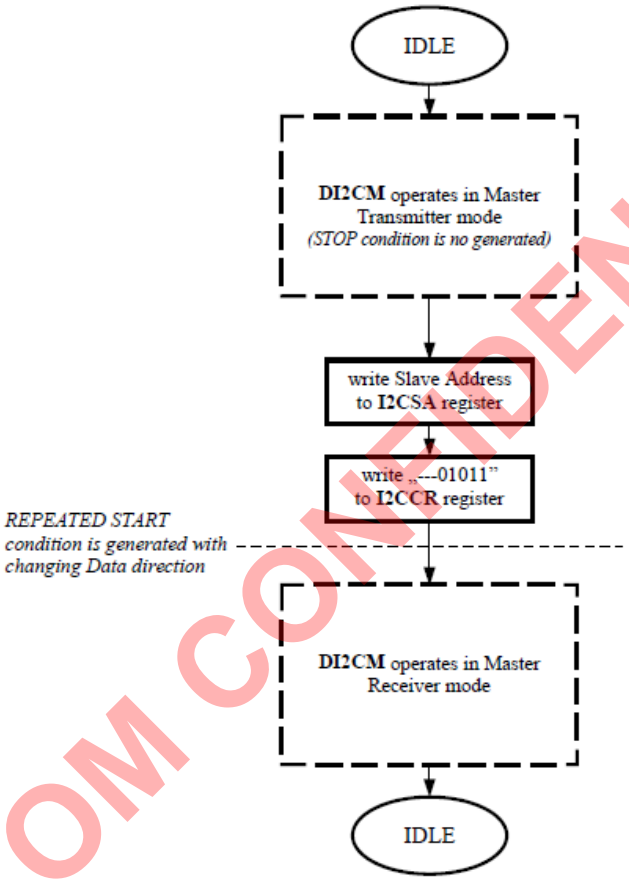


Figure15.5 I²C MASTER MODULE Sending n bytes then Repeated Start and Receiving m bytes flowchart

■ I²C MASTER MODULE BURST SEND AFTER BURST RECEIVE

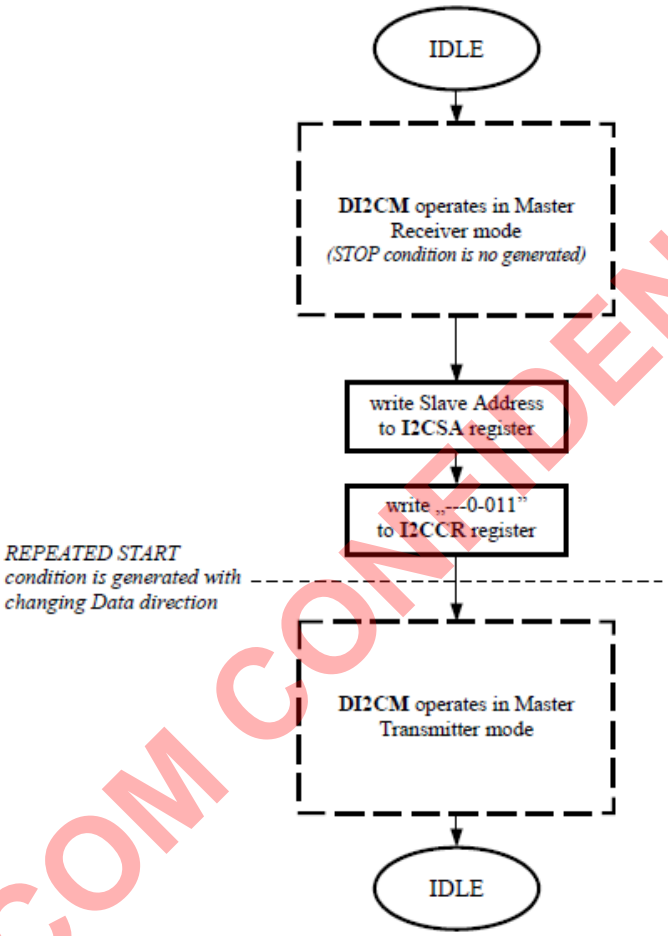


Figure15.6 I²C MASTER MODULE Receiving m bytes then Repeated Start and Sending n bytes flowchart

■ I²C SINGLE RECEIVE WITH 10-BIT ADDRESSING

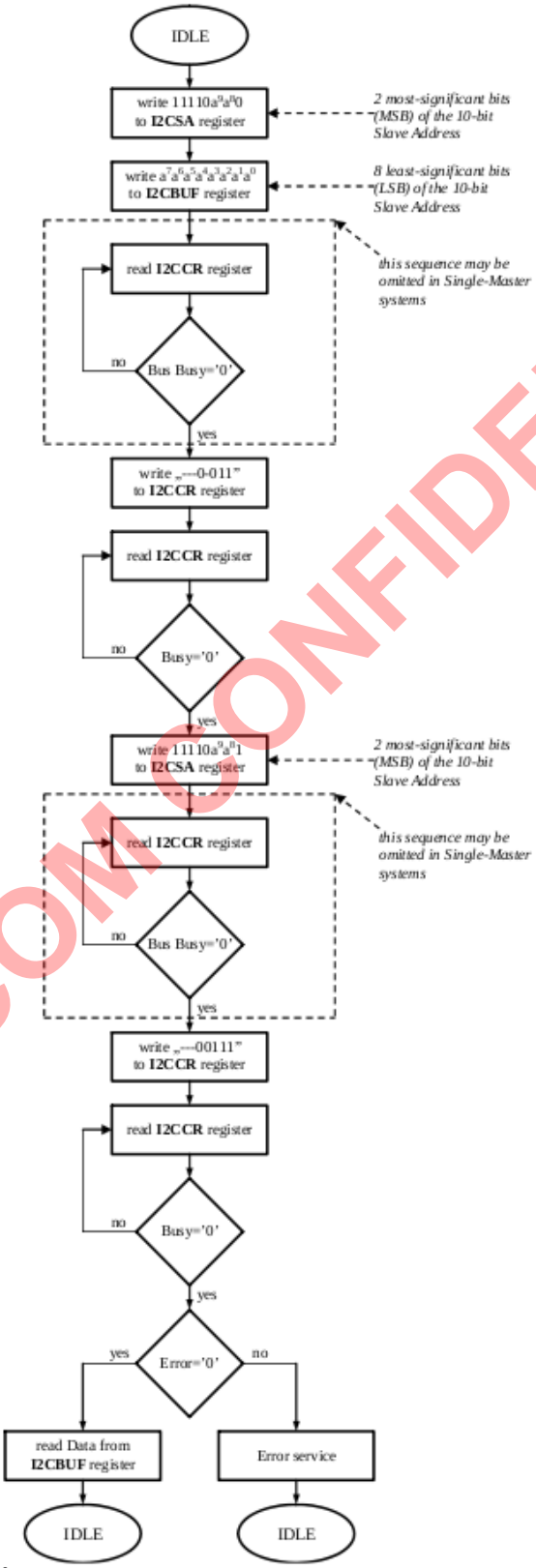


Figure15.7 I²C MASTER MODULE Single RECEIVE with 10-bit addressing flowchart

■ **I²C MASTER MODULE ACK POLLING**

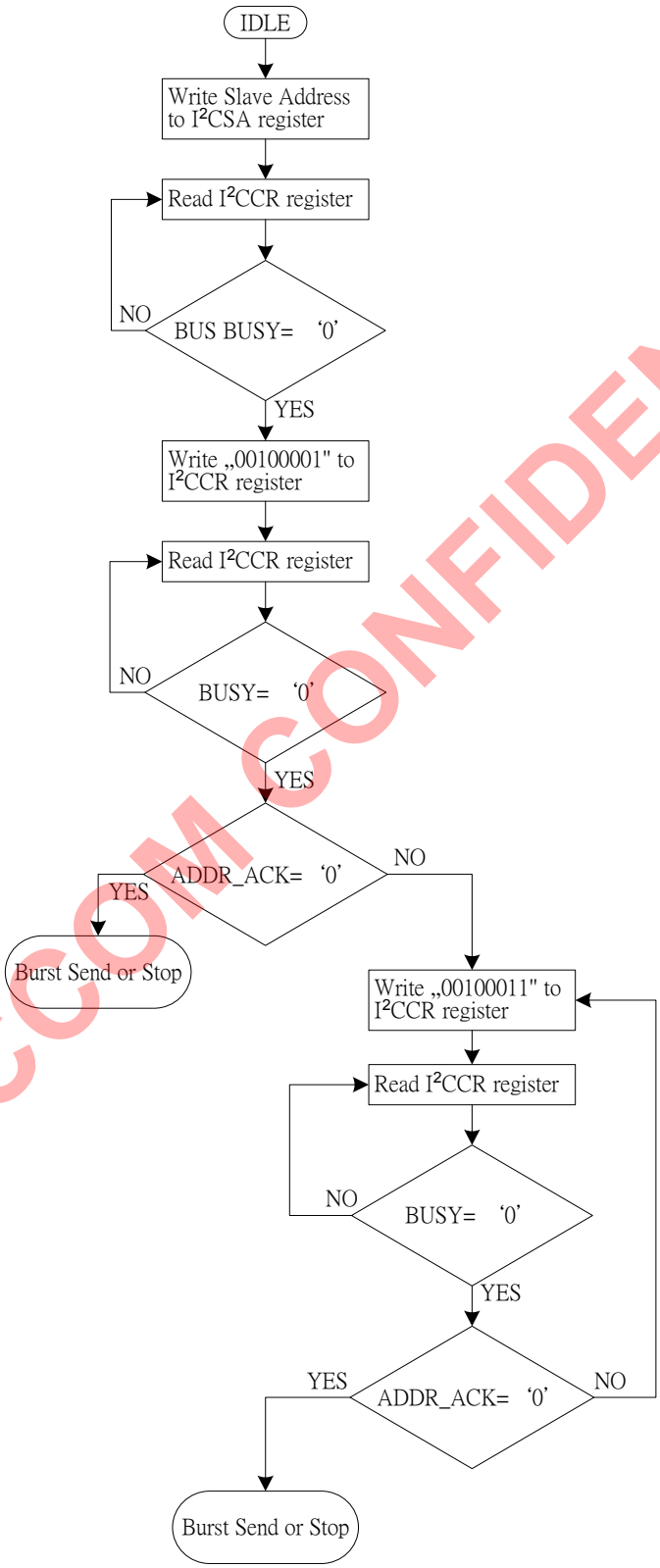


Figure15.8 I²C MASTER MODULE ACK Polling flowchart

15.2 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
I²CMIF	Internal, I²C MASTER MODULE

Table15.6 I²C MASTER MODULE interrupt summary

I²CMIF (0x5000301C)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
5000301Ch I²CMIF	R/W	--	--	--	--	--	--	I²CMIF	I²CMIE
Reset		0	0	0	0	0	0	0	0

I²CMIF：I²C MASTER MODULE interrupt flag

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

I²CMIE：I²C MASTER MODULE interrupt enable

Please refer the Chapter 10.3.2 for more detail information.

15.3 Slave MODE I²C

The I²C module provides an interface between a microprocessor and I²C 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 I²C specification. The I²C module supports all the transmission MODEs: Standard and Fast.

15.3.1 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	0x50003800
Control – I²CSCR	0x50003804
Transmitted data – I²CSBUF	0x50003808
Own address up– I²CSOAUP	0x5000380C
Interrupt – I²CSINT	0x50003010

Table15.7 I²C MODULE Registers for writing

Register	Address
Own address – I²CSOA	0x50003800
Control – I²CSSR	0x50003804
Received data – I²CSBUF	0x50003808
Own address up– I²CSOAUP	0x5000380C
Interrupt – I²CSINT	0x50003810

Table15.8 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 (0x50003800)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003800h 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

I²CSOAUP (0x50003810)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003800h I²CSOA	R/W	-	-	-	-	ten_addr	A.9	A.8	A.7
Reset		0	0	0	0	0	0	0	0

ten_addr：
= 1, enable
= 0, disable

■ 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 MCU write DA=1. This register can be only written at address 0xF2. Reading this address puts status register on data bus – see below.

I²CSCR (0x50003804)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003804h 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 MCU) 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 MCU) 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 MCU 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 (0x50003800)

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 (0x50003808)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003808h I²CSBUF Reset	R/W	D.7 0	D.6 0	D.5 0	D.4 0	D.3 0	D.2 0	D.1 0	D.0 0

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

I²CSBUF (0x50003808)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003808h I²CSBUF Reset	R/W	D.7 0	D.6 0	D.5 0	D.4 0	D.3 0	D.2 0	D.1 0	D.0 0

15.4 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").

15.4.1 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

15.4.2 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

15.4.3 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 I²C module
- ✧ STOP condition

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

15.4.4 I²C module BURST SEND

The figure below shows a set of sequences during Burst Data Send by I²C module. Burst send sequences：

- ✧ Start condition
- ✧ I²C module is addressed by I²C Master as transmitter
- ✧ Address is acknowledged by I²C module
- ✧ (1)Data is transmitted by I²C 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.

15.4.5 AVAILABLE I²C module COMMAND SEQUENCES FLOWCHART

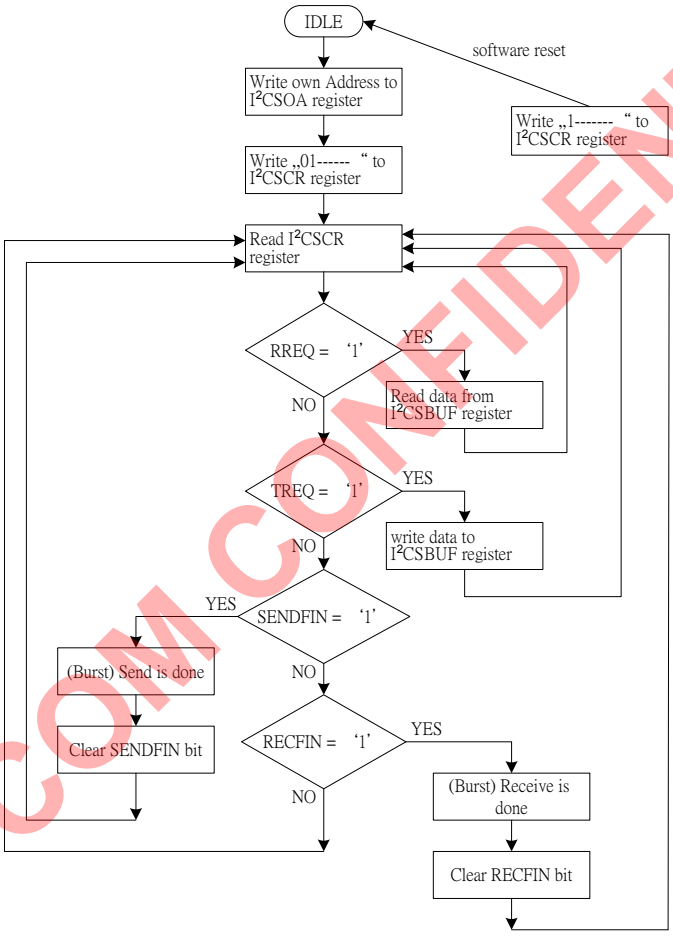


Figure15.9 Available I²C MODULE command sequences flowchart

I15.5 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
I²CSIF	Internal, DI²CS

Table15.9 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.

I²CSIF (0x50003810)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50003810h I^2CSIF	R/W							I^2CSIF	I^2CSIE
Reset		0	0	0	0	0	0	0	0

I^2CSIF : I^2C MODULE interrupt flag

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

I^2CSIE : I^2C MODULE interrupt enable

Please refer the Chapter 10.3.2 for more detail information.

16. 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.

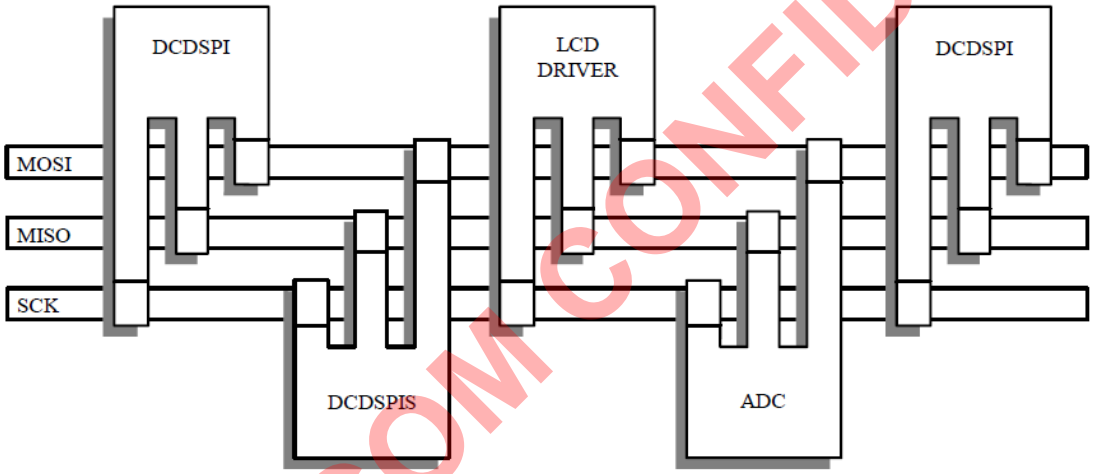
The SPI allows the microcontroller to communicate with serial peripheral devices. It is also capable of inter-processor communications in a multi-master system. A serial clock line (SCK) SYNChronizes shifting and sampling of the information on the two independent serial data lines. SPI data are simultaneously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



16.1 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
  - Bit rate support 1/2 system clock
  - 8 depth * 16 bit FIFO support continuous write.
- 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
- 8 depth * 16 bit FIFO support continuous read

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

16.2 SPI PINS DESCRIPTION

PIN	TYPE	ACTIVE	DESCRIPTION
Scki_Scko(P0.3)	INPUT / OUTPUT	-	SPI clock input / output
SIMO(P0.2)	INPUT / OUTPUT	-	Slave serial data input / Master serial data output
MISO(P0.1)	INPUT / OUTPUT	-	Master serial data input / Slave serial data output
SSO(P0.0)	OUTPUT	low	Slave select output

Table16.1 SPI pins description

16.3 SPI HARDWARE DESCRIPTION

16.3.1 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.

FIFO mode is available, when FIFO mode enable, there is 8 depth * 16 bits FIFO memory to store data, software can write or read continuously.

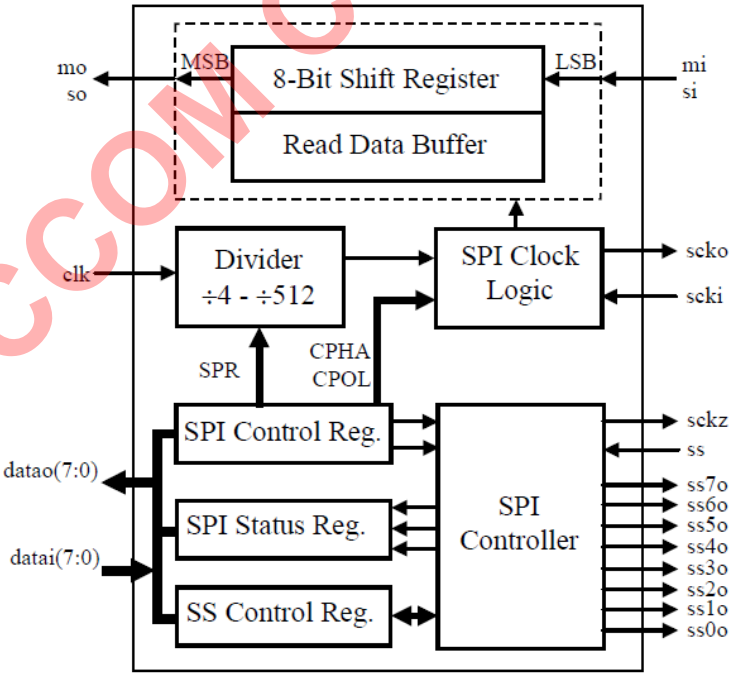


Figure 16.2 SPI Block Diagram

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.

16.3.2 INTERNAL REGISTERS

- **SPI Control Register**

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

SPCR (0x50002000)

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
SPCR[31:24]	R/W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
SPCR [23:16]	R/W	--	rxfifo_il2	rxfifo_il1	rxfifo_il0	--	txfifo_il2	txfifo_il1	txfifo_il0
RESET		0	1	1	1	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
SPCR [15:8]	R/W	slave_rst	--	--	fifoen	bit_len3	bit_len2	bit_len1	bit_len0
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**
SPCR [7:0]	R/W	SPIE	SPE	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0
RESET		0	0	0	0	0	1	0	0

txfifo_il[2:0] : set TX FIFO interrupt level, if txfifo_il[2:0]>=txfifo_rem[2:0], generate interrupt
rxfifo_il[2:0] : set RX FIFO interrupt level, if rxfifo_il[2:0] <=rxfifo_rem[2:0], generate interrupt
fifoen：FIFO mode enable
In TX, We can write FIFOTXBUF continuously by software, the data is stored in FIFO, and the hardware starts to transfer data. when current data is finished, the next data starts to transfer automatically.
In RX, data is stored in FIFO by hardware, we can read FIFORXBUF many times by software.
= 0, SPI FIFO disable, SPI is the same as DCD SPI
= 1, SPI FIFO enable, FIFO is 8 depths * 16 bits
Slave_rst：wirte 1to clear slave TX FIFO
bit_len[3:0] : set number of bits in a shift sequence
= 0000~0011, 4 bits
= 0100, 5 bits
= 0101, 6 bits
= 0110, 7 bits
= 0111, 8 bits
= 1000, 9 bits
= 1001, 10 bits
= 1010, 11 bits
= 1011, 12 bits
= 1100, 13 bits
= 1101, 14 bits
= 1110, 15 bits
= 1111, 16 bits
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	2 (only master to slave)

- **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 (0x5000200C)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
5000200Ch 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 (0x50002004)

Address Name	R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
SPCR [23:16]	R/W	--	--	--	--	SPIEn3	SPIEn2	SPIEn1	SPIEn0
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
SPCR [15:8]	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**
SPCR [15:8]	R/W	SPIF	WCOL	SS_S	MODF	-	-	-	SSCEN
RESET		0	0	0	0	0	0	0	0

SPIEn[3:0] : Reserved for internal usage
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.
SS_S：for slave to read SS line status

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.

FIFOTXSTUS (0x50002010)

Address Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FIFORXBUF [7:0]	R	-	-	-	tx_full	tx_empty	txfifo_rem2	txfifo_rem1	txfifo_rem0
RESET		0	0	0	0	1	0	0	0

tx_full : TX FIFO full, when TX FIFO is full, writing data to FIFOTXBUF is ineffective.
tx_empty : TX FIFO empty, when TX FIFO is empty, SPI will stop transferring.
txfifo_rem[2:0] : TX FIFO remnant, number of data is in the TX FIFO currently.

FIFORXSTUS (0x50002014)

Address Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FIFORXBUF [7:0]	R	-	-	-	rx_full	rx_empty	rxfifo_rem2	rxfifo_rem1	rxfifo_rem0
RESET		0	0	0	0	1	0	0	0

rx_full : RX FIFO full, when RX FIFO is full, but SPI still transfer data to RX, new data will overwrite the old data, and old data will be lost.
rx_empty : RX FIFO empty, if RX FIFO empty, read FIFORXBUF will read 0
rxfifo_rem[2:0] : RX FIFO remnant, number of data is in the RX FIFO currently.

- 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 txbuf15 (MSB).

FIFOTXBUF (0x50002018)

Address Name	R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
FIFOTXBUF [15:8]	R/W	txbuf15	txbuf14	txbuf13	txbuf12	txbuf11	txbuf10	txbuf9	txbuf8
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**
FIFOTXBUF [7:0]	R/W	txbuf7	txbuf6	txbuf5	txbuf4	txbuf3	txbuf2	txbuf1	txbuf0
RESET		0	0	0	0	0	0	0	0

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

FIFORXBUF (0x5000201C)

Address Name	R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
FIFORXBUF [15:8]	R/W	rxbuf15	rxbuf14	rxbuf13	rxbuf12	rxbuf11	rxbuf10	rxbuf9	rxbuf8
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**

FIFORXBUF [7:0]	R/W	rxbuf7	rxbuf6	rxbuf5	rxbuf4	rxbuf3	rxbuf2	rxbuf1	rxbuf0
RESET		0	0	0	0	0	0	0	0

16.4 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.

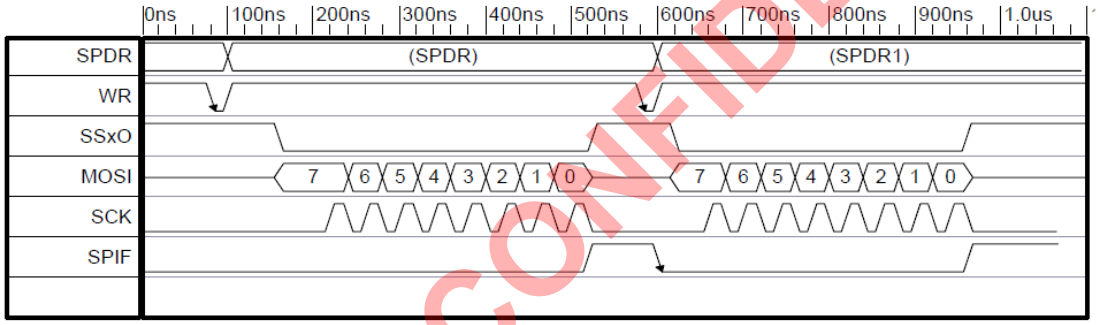


Figure16.3 Automatic slave select lines assertion

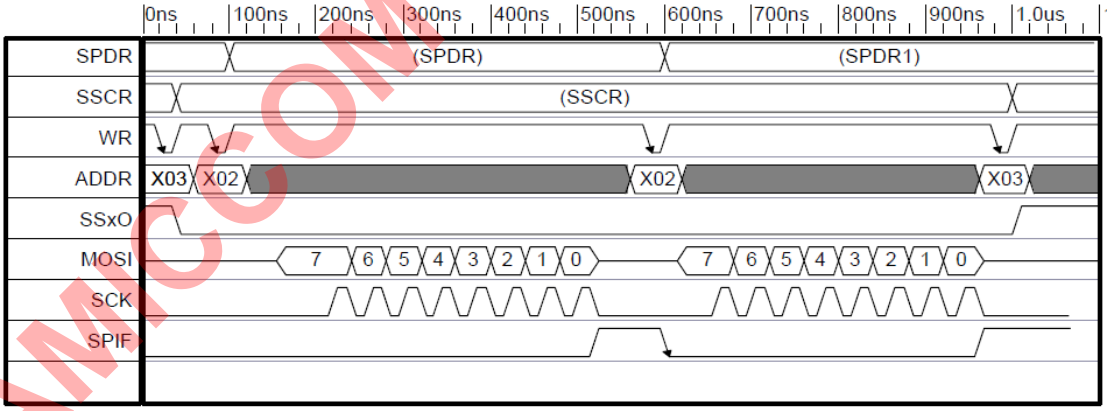


Figure16.4 Software controlled SSxO lines

16.4.1 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 MCU tried to write the SPDR register while transfer was in progress.

◆ **MODE FAULT ERROR**

MODE fault error occurs 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

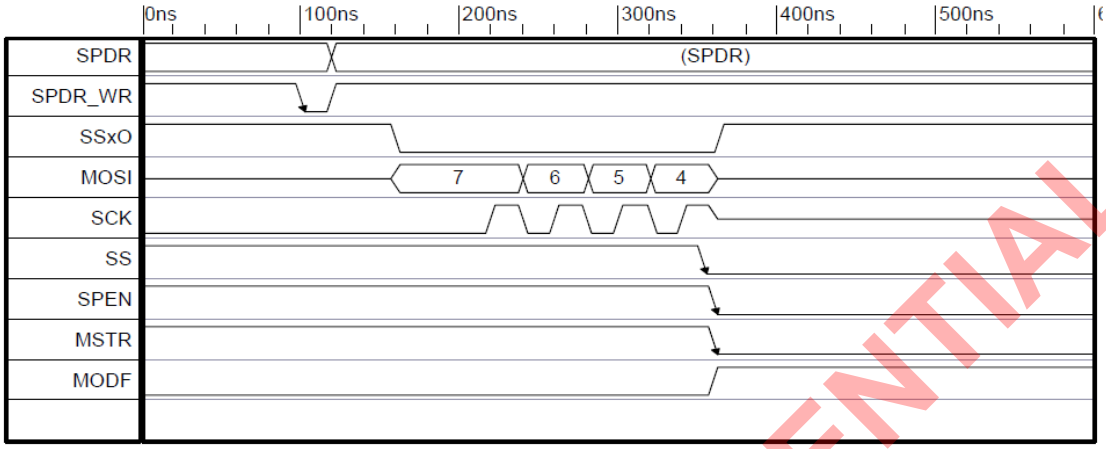


Figure16.5 MODE Fault Error generation

◆ **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)

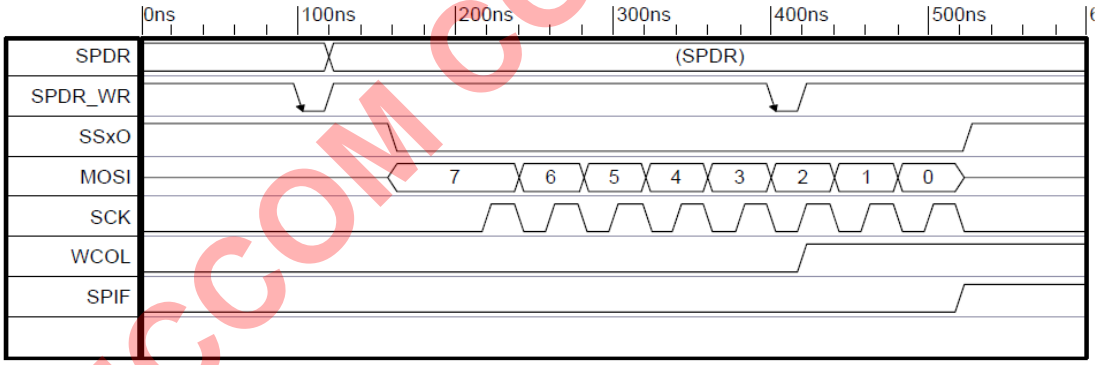


Figure16.6 Write Collision Error in SPI Master MODE

16.5 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.

16.5.1 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)

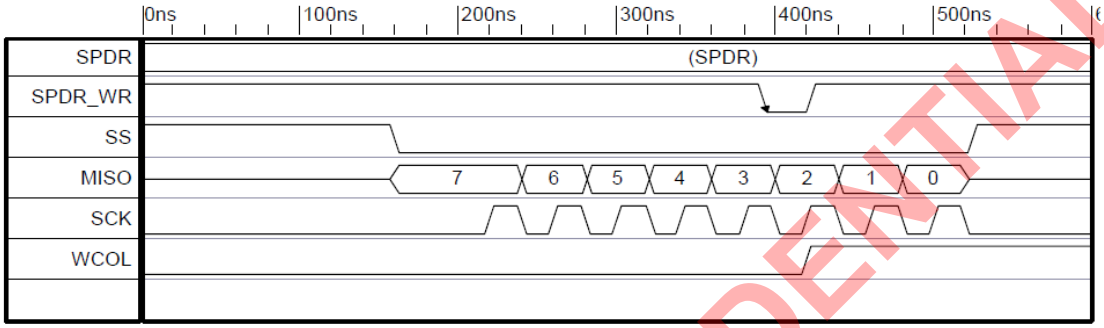


Figure16.7 Write Collision Error – SPI Slave MODE – SPDR write during transfer

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.

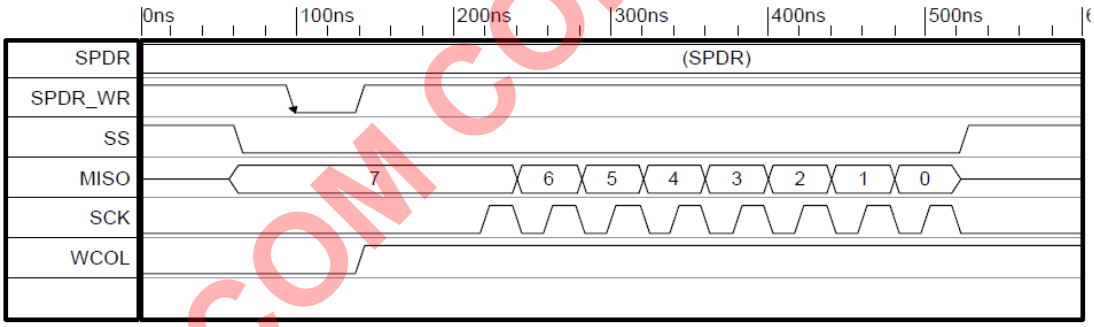


Figure16.8 WCOL Error-SPI Slave MODE-SPDR write when CPHA = 0 and SS = 0

16.6 CLOCK CONTROL LOGIC

16.6.1 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.

16.6.2 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.

16.6.3 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.

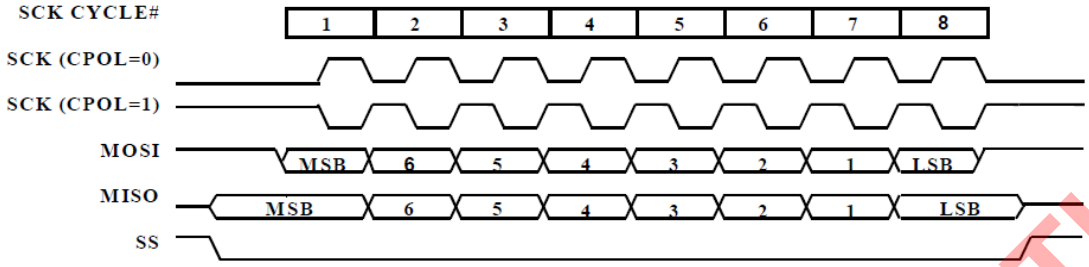


Figure16.9 CPHA Equals Zero SPI Transfer Format

When CPHA = 0, the SS line must be disserted 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.

16.6.4 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.

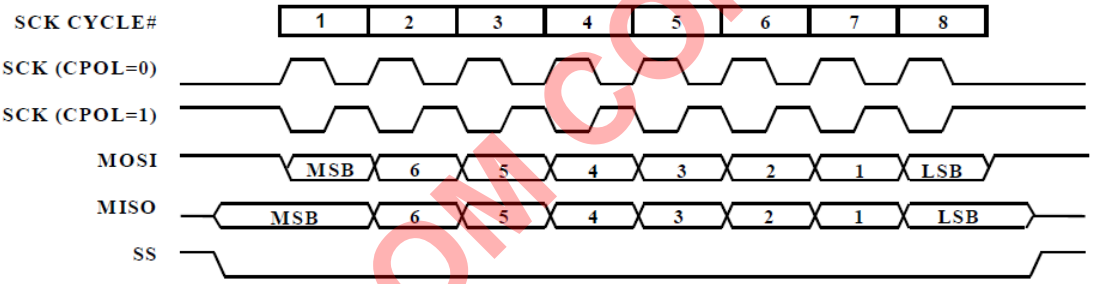


Figure16.10 CPHA Equals One SPI Transfer Format

16.7 SPI DATA TRANSFER

16.7.1 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.

16.7.2 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.

16.8 TIMING DIAGRAMS

16.8.1 MASTER TRANSMISSION

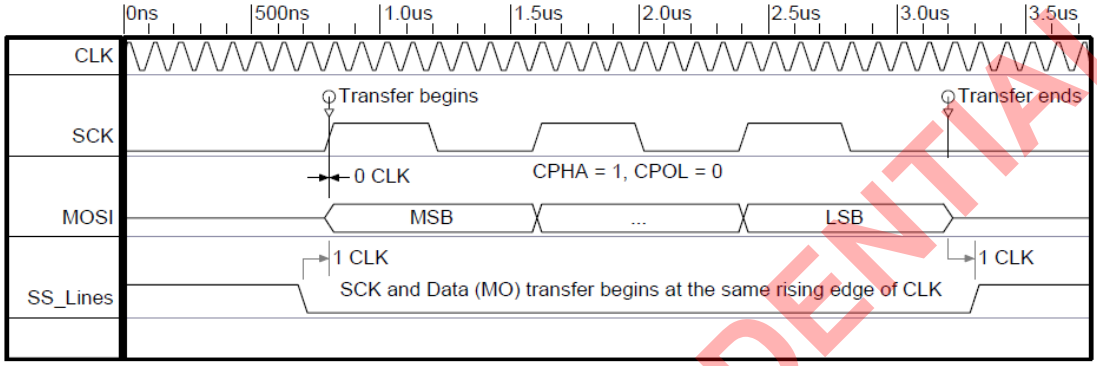


Figure16.11 Master MODE timing diagram

16.8.2 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).

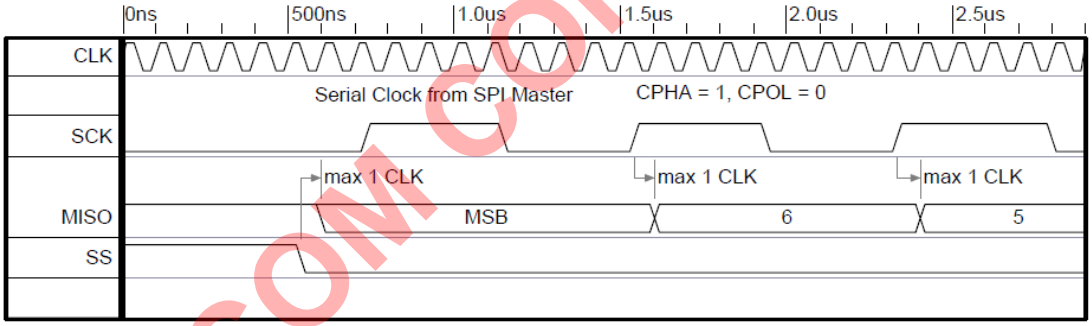


Figure16.12 Slave MODE timing diagram

16.9 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.

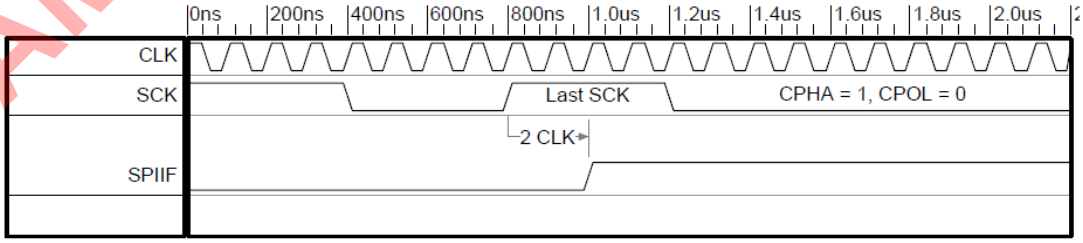


Figure16.13 Interrupt generation

Interrupt flag	Function
SPIIF	Internal, SPI

Table16.1 SPI interrupt summary

SPIINT (0x50002008)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50002008h SPIINT	R/W			rxfifo_int_clr	txfifo_int_clr			rxfifo_intie	txfifo_intie
Reset		0	0	0	0	0	0	0	0

rxfifo_int_clr：RX FIFO interrupt flag, write 0 has no effect, when rxfifo_il[2:0]>=rxfifo_rem[2:0], write 1 to clear
txfifo_int_clr：TX FIFO interrupt flag, write 0 has no effect, when txfifo_il[2:0] <=txfifo_rem[2:0], write 1 to clear
rxfifo_intie：**SPI RXFIFO interrupt enable**
txfifo_intie：**SPI TXFIFO interrupt enable**
Please refer the Chapter 10.3.2 for more detail information.

17. PWM

A8107M0 has four 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

$$\text{Pulse frequency} = \text{System clock} / 2^{pwxclk+1} / (256\text{-PWMxL})$$
$$\text{Duty cycle} = (256\text{-PWMxH}) / 256\text{-PWMxL})$$

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

17.1 PWM FUNCTIONALITY

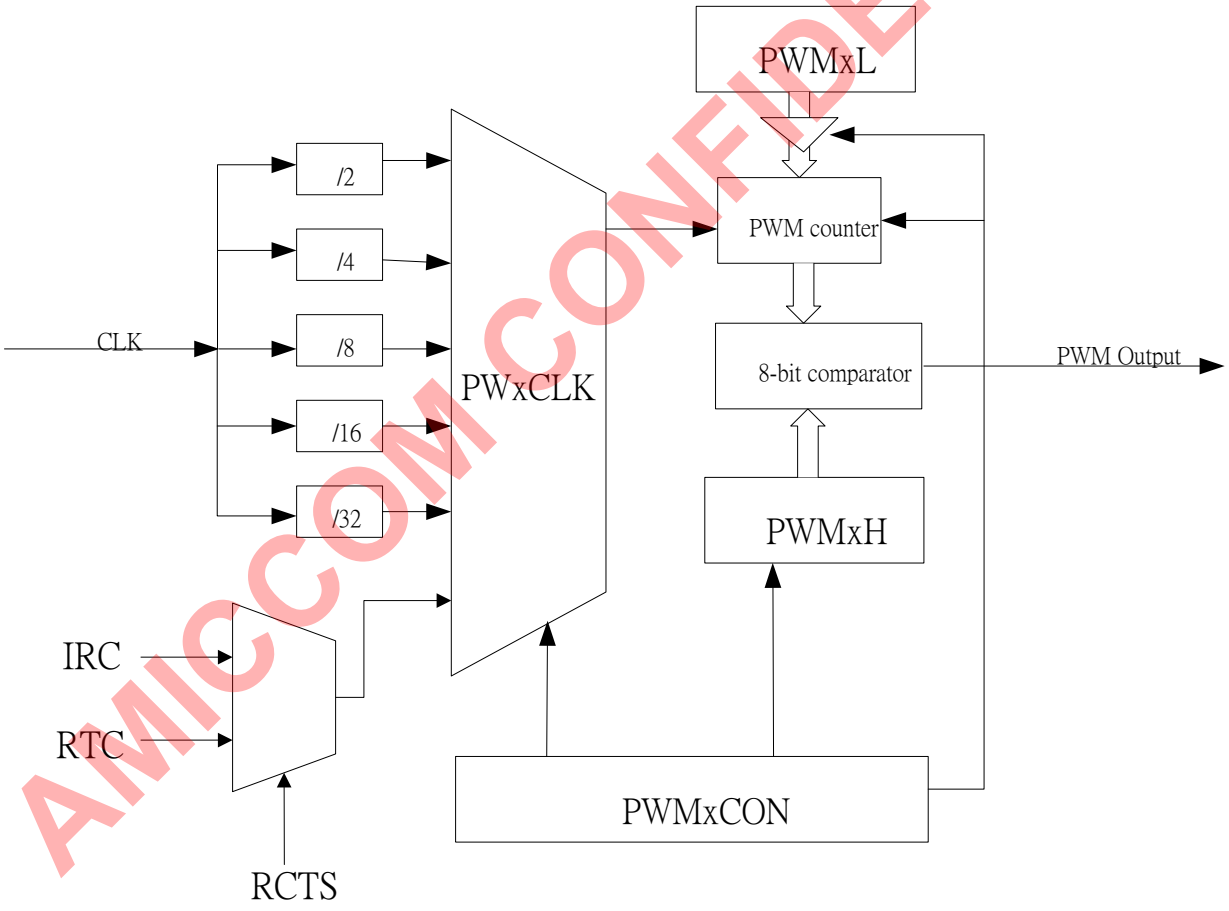


Figure17.1 PWM Block Diagram

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

PIN	ACTIVE	TYPE	DESCRIPTION
PWM0(P0.20)		OUTPUT	PWM 0 output
PWM1(P0.21)		OUTPUT	PWM 1 output
PWM2(P0.10)		OUTPUT	PWM 2 output
PWM3(P0.11)		OUTPUT	PWM 3 output

Table17.1 PWM PIN define

17.1.1 PWM0 Registers

PWM0/1/2/3 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 ,32, 64 and IRC/RTC by setting PWxCLK.

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x50004000h PWM0EN	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x50004000h PWM0EN	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x50004000h PWM0EN	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**
0x50004000h PWM0EN	R/W	PWM0EN	-	-	-	PW0RTC	PW0CLK2	PW0CLK1	PW0CLK0
RESET		0	0	0	0	0	0	0	0

PWM0CON: PWM channel 0 control register

PWM0EN: PWM Channel 0 Enable,
[0]: Disable. **[1]**: Enable.

PW0RTC: PWM Channel 0 Clock Source select,
[0]: MCU Clock. **[1]**: RTC clock.

PWM0CLK[2:0]: PWM Channel 0 Clock select
[000]: PWM Clock /2
[001]: PWM Clock / 4
[010]: PWM Clock / 8
[011]: PWM Clock / 16
[100]: PWM Clock / 32
[101]: PWM Clock / 64

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x50004004h PWM0	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x50004004h PWM0	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x50004004h PWM0	R/W	PWM0H7	PWM0H6	PWM0H5	PWM0H4	PWM0H3	PWM0H2	PWM0H1	PWM0H0
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**
0x50004004h PWM0	R/W	PWM0L7	PWM0L6	PWM0L5	PWM0L4	PWM0L3	PWM0L2	PWM0L1	PWM0L0

RESET		0	0	0	0	0	0	0	0

PWM0: PWM channel 0 control register

PWM0H[7:0]: PWM channel 0 output HIGH register
PWM0L[7:0]: PWM channel 0 frequency setting register

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x50004100h PWM1EN	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x50004100h PWM1EN	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x50004100h PWM1EN	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**
0x50004100h PWM1EN	R/W	PWM1EN	-	-	-	PW1RTC	PW1CLK2	PW1CLK1	PW1CLK0
RESET		0	0	0	0	0	0	0	0

PWM1CON: PWM channel 1 control register

PWM1EN: PWM Channel 1 Enable,
[0]: Disable. **[1]**: Enable.

PW1RTC: PWM Channel 1 Clock Source select,
[0]: MCU Clock. **[1]**: RTC clock.

PWM1CLK[2:0]: PWM Channel 1 Clock select
[000]: MCU Clock / 2
[001]: MCU Clock / 4
[010]: MCU Clock / 8
[011]: MCU Clock / 16
[100]: MCU Clock / 32
[101]: MCU Clock / 64

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x50004104h PWM1	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x50004104h PWM1	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x50004104h PWM1	R/W	PWM1H7	PWM1H6	PWM1H5	PWM1H4	PWM1H3	PWM1H2	PWM1H1	PWM1H0
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**
0x50004104h PWM1	R/W	PWM1L7	PWM1L6	PWM1L5	PWM1L4	PWM1L3	PWM1L2	PWM1L1	PWM1L0

RESET		0	0	0	0	0	0	0	0

PWM1: PWM channel 1 control register

PWM1H[7:0]: PWM channel 1 output HIGH register
PWM1L[7:0]: PWM channel 1 frequency setting register

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x50004200h PWM2EN	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x50004200h PWM2EN	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x50004200h PWM2EN	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**
0x50004200h PWM2EN	R/W	PWM2EN	-	-	-	PW2RTC	PW2CLK2	PW2CLK1	PW2CLK0
RESET		0	0	0	0	0	0	0	0

PWM2CON: PWM channel 2 control register

PWM2EN: PWM Channel 2 Enable,
[0]: Disable. **[1]**: Enable.

PW2RTC: PWM Channel 2 Clock Source select,
[0]: MCU Clock. **[1]**: RTC clock.

PWM2CLK[2:0]: PWM Channel 2 Clock select
[000]: PWM Clock / 2
[001]: PWM Clock / 4
[010]: PWM Clock / 8
[011]: PWM Clock / 16
[100]: PWM Clock / 32
[101]: PWM Clock / 64

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x50004204h PWM2	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x50004204h PWM2	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x50004204h PWM2	R/W	PWM2H7	PWM2H6	PWM2H5	PWM2H4	PWM2H3	PWM2H2	PWM2H1	PWM2H0
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**

0x50004204h PWM2	R/W	PWM2L7	PWM2L6	PWM2L5	PWM2L4	PWM2L3	PWM2L2	PWM2L1	PWM2L0
RESET		0	0	0	0	0	0	0	0

PWM2: PWM channel 2 control register

PWM2H[7:0]: PWM channel 2 output HIGH register
PWM2L[7:0]: PWM channel 2 frequency setting register

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x50004300h PWM3EN	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x50004300h PWM3EN	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x50004300h PWM3EN	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**
0x50004300h PWM3EN	R/W	PWM3EN	-	-	-	PW3RTC	PW3CLK2	PW3CLK1	PW3CLK0
RESET		0	0	0	0	0	0	0	0

PWM3CON: PWM channel 3 control register

PWM3EN: PWM Channel 3 Enable,
[0]: Disable. **[1]**: Enable.

PW3RTC: PWM Channel 3 Clock Source select,
[0]: MCU Clock. **[1]**: RTC clock.

PWM3CLK[2:0]: PWM Channel 3 Clock select
[000]: PWM Clock / 2
[001]: PWM Clock / 4
[010]: PWM Clock / 8
[011]: PWM Clock / 16
[100]: PWM Clock / 32
[101]: PWM Clock / 64

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
0x50004304h PWM3	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
0x50004304h PWM3	R/W								
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
0x50004304h PWM3	R/W	PWM3H7	PWM3H6	PWM3H5	PWM3H4	PWM3H3	PWM3H2	PWM3H1	PWM3H0
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
0x50004304h PWM3	R/W	PWM3L7	PWM3L6	PWM3L5	PWM3L4	PWM3L3	PWM3L2	PWM3L1	PWM3L0
RESET		0	0	0	0	0	0	0	0

PWM3: PWM channel 3 control register

PWM3H[7:0]: PWM channel 3 output HIGH register
PWM3L[7:0]: PWM channel 3 frequency setting register

18. Watchdog Timer

A8107M0 has a special timer, called Watchdog Timer. It is a useful programmable clock counter that serves as a time-base generator, an event timer or system supervisor. User can use be a very long timer with disabled reset function.

18.1 Watchdog timer overview

The Watchdog Free-Running Counter (WdogFrc) is a sub-component of the Watchdog module. It essentially contains:

* The 32-bit free-running down-counter.
* Period load register.
* Control register to enable reset and interrupt signals.
* Interrupt status register.
* Lock register to prevent accidental write access.
* Interrupt and Reset generation logic.

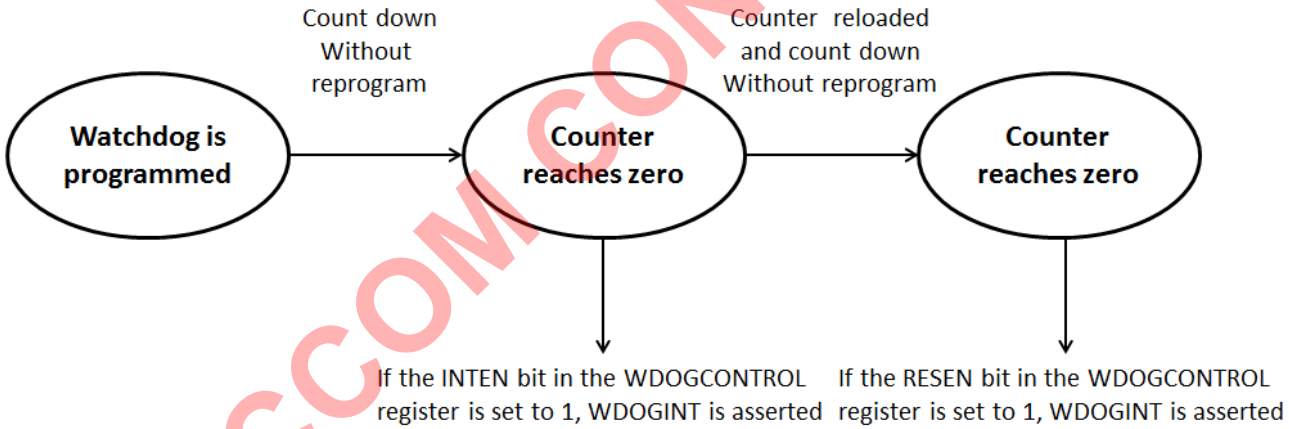


Figure 18.1 Watchdog Timer architecture.

18.2 Watchdog Register

Watchdog Timer LOAD (0x40008000h):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
LOAD[31:24]	R/W	LOAD[31:24]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**Bit 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
LOAD[23:16]	R/W	LOAD[23:16]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
LOAD[15:8]	R/W	LOAD[15:8]							

RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
LOAD[7:0]	R/W	LOAD[7:0]							
RESET		1	1	1	1	1	1	1	1

LOAD [31:0] : The minimum valid value for WDOGLOAD is1.

Watchdog Timer VALUE (0x40008004h):

Address name	**R/W**	**Bit 31**	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
VALUE[31:24]	R	VALUE[31:24]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**VALUE 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
VALUE[23:16]	R	VALUE[23:16]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
VALUE[15:8]	R	VALUE[15:8]							
RESET		1	1	1	1	1	1	1	1
Address Name	**R/W**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
VALUE[7:0]	R	VALUE[7:0]							
RESET		1	1	1	1	1	1	1	1

Watchdog Timer CRTL (0x40008008h):

Address name	**R/W**	**Bit 31**	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
CRTL[31:24]	R/W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**VALUE 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
CRTL[23:16]	R/W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
CRTL[15:8]	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**
CRTL[7:0]	R/W	--	--	--	--	--	--	RESTEN	INTEN
RESET		0	0	0	0	0	0	0	0

RESEN[1] : RESEN enable watchdog reset output, WDOGRES. Acts as a mask for the reset output. Set HIGH to enable the reset, or LOW to disable the reset.

INTEN[0] : INTEN enable the interrupt event, WDOGINT. Set HIGH to enable the counter and interrupt, or LOW to disable the counter and interrupt. Reloads the counter from the value in WDOGLOAD when the interrupt is enabled, after previous being disabled.

Watchdog Timer RAWINTSTAT (0x40008010h):

Address name	**R/W**	**Bit 31**	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**

RAWINTSTAT[23:24]	R	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**VALUE 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
RAWINTSTAT[23:16]	R	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
RAWINTSTAT[15:8]	R	--	--	--	--	--	--	--	--
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**
RAWINTSTAT[7:0]	R	--	--	--	--	--	--	--	RAWINTSTAT0
RESET		0	0	0	0	0	0	0	0

RAWINTSTAT0 : Raw Watchdog Interrupt Raw interrupt status from the counter.

Watchdog Timer MASKINTSTAT (0x40008014h):

Address name	**R/W**	**Bit 31**	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
MASKINTSTAT[31:24]	R	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**VALUE 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
MASKINTSTAT[23:16]	R	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
MASKINTSTAT[15:8]	R	--	--	--	--	--	--	--	--
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**
MASKINTSTAT[7:0]	R	--	--	--	--	--	--	--	MASKINTSTAT0
RESET		0	0	0	0	0	0	0	0

MASKINTSTAT0 : Raw Watchdog Interrupt Raw interrupt status from the counter.

Watchdog Timer LOCK (0x40008C00h):

Address name	**R/W**	**Bit 31**	**Bit 30**	**Bit 29**	**Bit 28**	**Bit 27**	**Bit 26**	**Bit 25**	**Bit 24**
LOCK[31:24]	R/W	LOCK[31:24]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**VALUE 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
LOCK[23:16]	R/W	LOCK[23:16]							
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
LOCK[15:8]	R/W	LOCK[15:8]							
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**
LOCK[7:0]	R/W	LOCK[7:0]							
RESET		0	0	0	0	0	0	0	0

LOCK [31:1] : Enable register writes Enable write access to all other registers by writing 0x1ACCE551. Disable write access by writing any other value.

LOCK [0] : Register write enable status
[0]: Write access to all other registers is enabled. This is the default.
[1]: Write access to all other registers is disabled.

Watchdog Timer ITCR (0x40008F00h):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
ITCR[31:24]	R/W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**VALUE 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
ITCR[23:16]	R/W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
ITCR[15:8]	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**
ITCR[7:0]	R/W	--	--	--	--	--	--	--	ITCR
RESET		0	0	0	0	0	0	0	0

ITCR [0] : Integration Test MODE Enable. When set HIGH, please the watchdog into integration test MODE.

Watchdog Timer ITOP (0x40008F04h):

Address name	R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
ITOP[31:24]	W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 23**	**Bit 22**	**Bit 21**	**VALUE 20**	**Bit 19**	**Bit 18**	**Bit 17**	**Bit 16**
ITOP[23:16]	W	--	--	--	--	--	--	--	--
RESET		0	0	0	0	0	0	0	0
Address Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
ITOP[15:8]	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**
ITOP[7:0]	W	--	--	--	--	--	--	ITOP[1:0]	
RESET		0	0	0	0	0	0	0	0

ITOP [0] : Integration Test WDOGINT value. Value output on WDOGINT when in Integration Test MODE.
ITOP [1] : Integration Test WDOGRES value. Value output on WDOGRES when in Integration Test MODE.

19. ADC (Analog to Digital Converter)

A8107M0 has two built-in ADCs. One is 8-bits ADC do RSSI measurement as well as carrier detection function. The 8-bit ADC converting time is 20 x ADC clock periods. The other is 8-channel 12-bits SAR ADC.

19.1 8-bits ADC

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

Table 19.1 Setting of ADC function

Relative Control Register

ADC Register (Address: 50001240h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADC	R	ERSS	FSARS1	FSARS0	XADS	RSS	ARSSI	CDM	ADCM
	W	ERSS	FSARS1	FSARS0	XADS	RSS	ARSSI	CDM	ADCM
Reset			0	0	0	0	0	0	0
Name	**R/W**	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
ADC	R	RADC	AVSEL1	AVSEL0	MVSEL1	MVSEL0	XASDP	RSM1	RSM-
	W	RADC	AVSEL1	AVSEL0	MVSEL1	MVSEL0	XASDP	RSM1	RSM-
Reset									

Threshold Register (Address: 50001244h)

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

19.1.1 RSSI Measurement

A8107M0 supports 8-bits digital RSSI to detect RF signal strength. RSSI value is stored in ADC [7:0] (1Dh). Fig 19.1 shows a typical plot of RSSI reading as a function of input power. This curve is based on the current gain setting of A8107M0 reference code. A8107M0 automatically averages 8-times ADC conversion a RSSI measurement until A8107M0 exits RX MODE. Therefore, each RSSI measuring time is (8 x 20 x F_{ADC}). Be aware RSSI accuracy is about ± 6dBm.

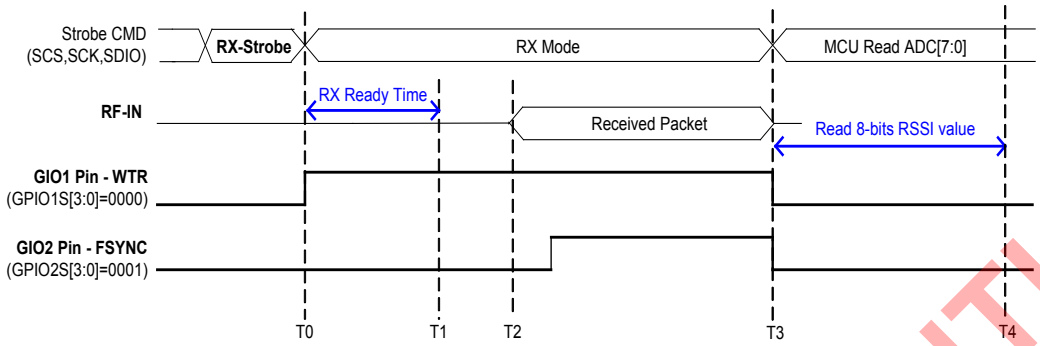
TBD

Figure 19.1 Typical RSSI characteristic.

Auto RSSI measurement for TX Power:

1. Set wanted F_{RXLO}
2. Set RSS= 1 , FSARS= 1 (4MHz ADC clock).
3. Enable ARSSI= 1 .
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 [8:0] (1Dh) for TX power.

In step 6, if A8107M0 is set in direct MODE, MCU shall let A8107M0 exit RX MODE within 40 us to prevent RSSI inaccuracy.

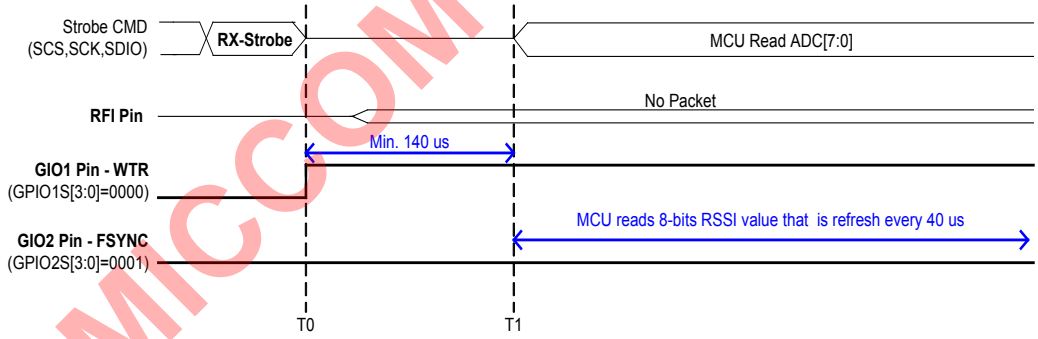


T0-T1: Settling Time
T2-T3: Receiving Packet
T3 : Exit RX mode automatically in FIFO mode
T3-T4: MCU read RSSI value @ ADC [7:0]

Figure 19.2 RSSI Measurement of TX Power.

Auto RSSI measurement for Background Power:

1. Set wanted F_{RXLO}
2. Set RSS= 1 , FSARS= 0 (4MHz ADC clock).
3. Enable ARSSI= 1 .
4. Send RX Strobe command.
5. MCU delays min. 140us.
6. Read digital RSSI value from ADC [8:0] (in 0x81DH and 0x821h) to get background power.
7. Send other Strobe command to let A8107M0 exit RX MODE.



T0-T1: MCU Delay Loop from PLL to RX mode for RSSI measurment
T1 : Auto RSSI Measurment is done by 8-times average.
MCU can read RSSI value from ADC [7:0]

Figure 19.3 RSSI Measurement of Background Power.

19.1.2 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 for absolute RSSI threshold level (ex. RTH = 80d).
2. Set GIO2S = [0010] for Carrier Detect to GIO2 pin.
   (2-1) Set wanted F_{RXLO}
   (2-2) Set RSM= [11] (CDM =0 and hysteresis =6, or CDM =1 and hysteresis =12).
   (2-3) Enable ARSSI= 1 .
   (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 ≧ CDTH, GIO2 = 0.
   (3-2) If ADC ≦ CDTH-CDM, GIO2 = 1.
   (3-3) If ADC locates in hysteresis zone, GIO2 = previous state.
4. Exit RX MODE.

19.2 12-bits SAR ADC

A8107M0 includes a 12-bit successive approximation A/D converter which enables channel selection from 8 channels. The A/D converter has two operating MODEs: single MODE and continuous MODE. The 12-bits A/D converter can be used to perform the analog input of the specified channel or temperature sensor. Fig 19.5 shows a typical plot of temperature reading for 12-bit ADC.

Bit		MODE
DTMP	BUFS	
0	0	Analog Input
1	1	Temperature Sensor

Table 19.2 Setting of 12-bit ADC function

The conversion time in single MODE can be determined as follows:

$$t_{conv} = \frac{1}{4MHz} \times \frac{1}{CKS[1:0]} \times 32$$

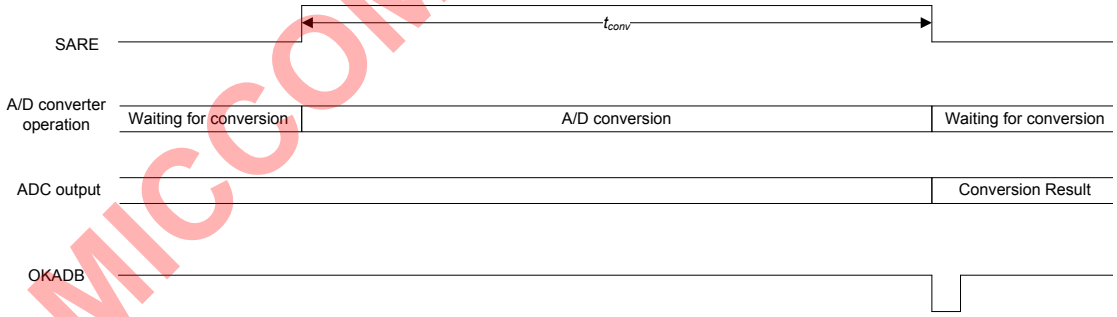


Figure 19.3 Single MODE for A/D Conversion.

Measurement for Analog Input:
1. Set ADCCH (0x50008008h) for selecting ADC channel.
2. Set ENADC (0x50008000h) to enable the SAR ADC.
3. Set MODE (0x50008000h) to select single MODE or continuous MODE.
4. Enable ADCE = 1 (0x50008000h)

Measurement for Temperature:
1. Set ENADC (0x50008000h) to enable the SAR ADC.
2. Set BUFS = 1 (0x50008000h)
3. Set DTMP = 1 (0x50008000) to enable the temperature sensor for 12-bit ADC.
4. Set MODE (0x50008000h) to select single MODE or continuous MODE.
5. Enable ADCE = 1 (0x50008000h)

TBD

Figure 19.4 Typical 12-bit ADC temperature sensor characteristic curve.

20. Battery Detect

A8107M0 has a built-in battery detector to check supply voltage (REGI pin). The detecting range is 1.875V ~ 2.4V in 8 levels.

Relative Control Register

Battery detect Register (Address: 50000000h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Battery detect	W					BVT2	BVT1	BVT0	BDS
	R				BDF	BVT2	BVT1	BVT0	BDS
Reset		0	0	0	0	0	1	1	0

BVT[1:0]: Battery detection threshold.
[000]: 1.875. **[001]:** 1.95V. **[010]:** 2.025V. **[011]:** 2.1V.
[100]: 2.175V. **[101]:** 2.25V. **[110]:** 2.325V. **[111]:** 2.4V.
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 A8107M0 in standby or PLL MODE.
2. Set BVT[2:0] = [011] and enable BDS = 1.
3. After 5 us, BDE is auto clear.
4. MCU reads BDF.
   If REGI pin > 2.1V,
   BDF = 1 (battery high). Else, BDF = 0 (battery low).

21 Power Management

The power consumption of A8107M0 comes from two parts. One is RF part and the other is digital part (includes MCU core and peripherals). In the RF part, the idle MODE use the minimum power and the TX or RX MODE use the maximum power consumptions. Use changes RF status by setting the strobe control,register(0x50001004h). For more detail information, please refer chapter 21.1. In this chapter only introduces digital parts. Low power operation is enabled through different power MODEs setting. A8107M0 has various operating MODE are referred as normal MODE, PM1, PM2, and PM3 (power manager MODE 3). Table 20.1 shows the impact of different power MODEs on systems operation. There are two registers to setting power manager. One is power control register (PCON, 0x50000020h) and the other is power control extend register (PCONE, 0xB9h).

In normal MODE, user selects different clock be MCU core clock.in CLKSEL[2:0] then enable CKSE .User adjusts MCU cloCKS depends on the required power consumption. CLKSEL[2:0] = 001 ~ 110b, the MCU core clock is the clock sources divide 2 ~ 64. User could adjust the MCU speed to trade-off between the performance and the power consumption. **BEWARE, please choice CLKSEL firstly then enable CKSE to avoid glitch. Please refer the reference code or contact AMICCOM's FAE for more details.**

User can enable STOP to freeze MCU core clock and all digital peripherals also stop. MCU can be waked up by hardware reset, KEY wake up, KEYINT or sleep timer (WOR /TWOR). User set sleep timer, WOR or TWOR before enter STOP MODE. In this condition, it is called PM1. In PM1, all digital circuitry is stop and RF circuitry is active by WOR

Note: Please don't enable STOP and CKSE at the same time.

PCON (0x50000020h) Power control

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
50000020h PCON	R/W			CLKSEL2	CLKSEL1	CLKSEL0	SWB	STOP	CKSE
Reset		0	0	0	0	0	0	0	0

SWB (Switchback enable)
[1]: Enable
[0]: Disable
STOP (Stop MODE)
[1]: Enable
[0]: Disable
CKSE (Clock select enable)
[1]: Enable clock select
[0]: Disable clock select
CLKSEL[2:0] (Clock Select), Select clock source when enable clock select.
[000]: Clock source div 64 as MCU clock
[001]: Clock source div 2 as MCU clock
[010]: Clock source div 4 as MCU clock
[011]: Clock source div 8 as MCU clock
[100]: Clock source div 16 as MCU clock
[101]: Clock source div 32 as MCU clock
[110]: Clock source div 64 as MCU clock
[111]: Select RTC as CPU clock when CKSE=0; RTC div 2 as CPU clock when CKSE=1

	MCU speed	16MHz	RAM	Back to Normal	LVR	RF
Normal CKSE=0	16MHz	ON	ON	X	X	X
Normal CKSE=1 (Low speed)	8/4/2/1 MHz IRC/RTC	ON	ON	X	X	X
Idle (PM1)	OFF	OFF	ON	H/W reset / KEYINT / Sleep timer	X	OFF
Sleep (PM2)	OFF	OFF	ON	H/W reset / wakeup key / Sleep timer	X	OFF
Deep Sleep (PM3)	OFF	OFF	256K OFF 2K ON	H/W reset / wakeup key / Sleep timer	OFF	OFF

Table 21.1 Power manager

X: don't care, it can turn on or off by user setting
Note: PM MODE setting refer to chapter 9.2.101
P3.2/P3.3/P3.4/P3.5 at PM MODE, only can setting: output high or input pull-high.

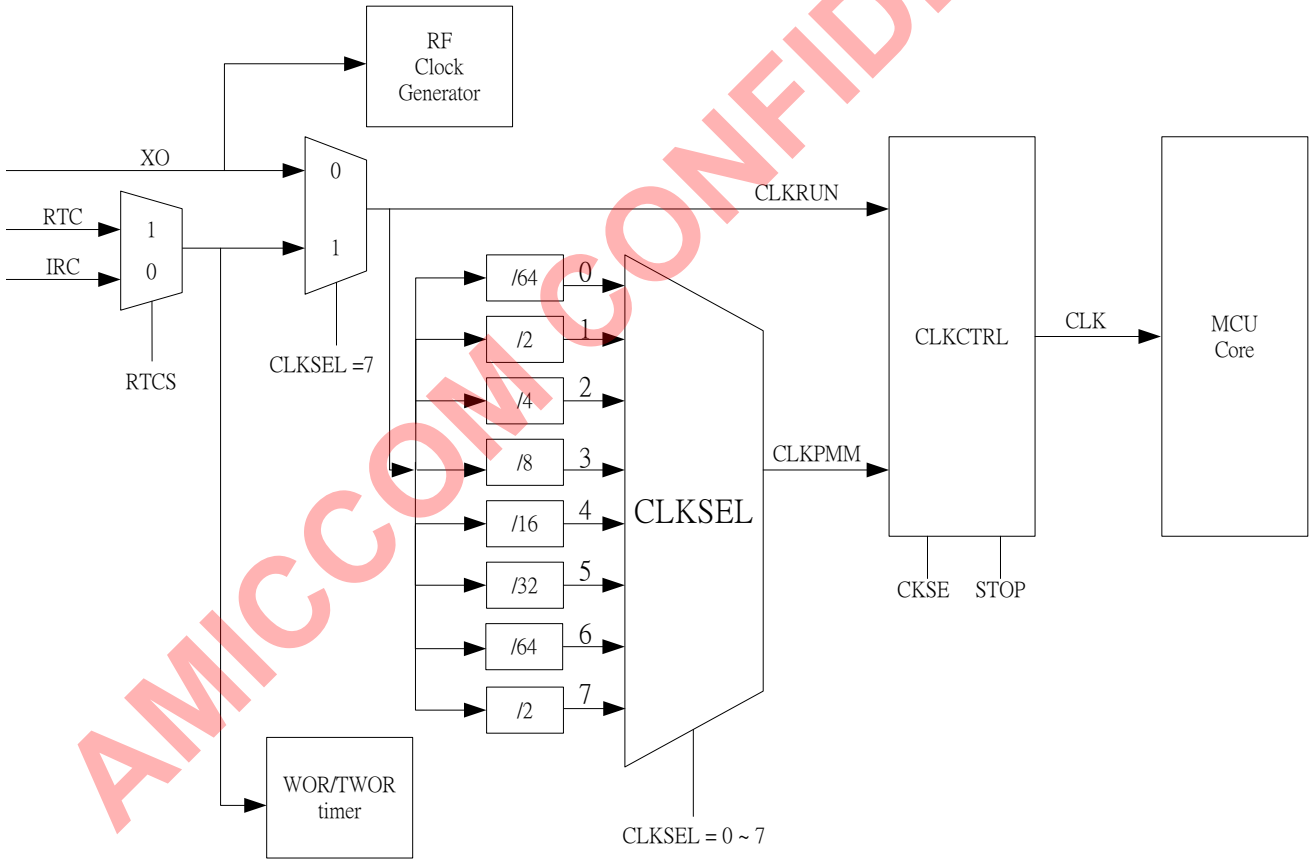


Figure 21.1 Whole chip clock sources.

22 A8107M0 RF

A8107M0 integrate 2.4 Ghz GFSK transceiver and use Strobe control register (50001004h) 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 256Bytes FIFO for data transmitting, receiving. Sleep timer is used for WOR (Wake On Rx) and time-slotted MODE operation.

22.1Strobe Command Register 1 (Address: 0x50001004h)

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

Use strobe command control RF state.
STRB[7:0]: Strobe command register.
0x80: Sleep MODE.
0x90: Idle MODE.
0xA0: Standby MODE.
0xB0: PLL MODE.
0xC0: TX MODE.
0xD0: RX MODE.

Status Register (Address: 0x50001010h)

Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	W								
	R	WTR	FECF	CRCF	CER	XER	PLLER	TRSR	TRER
Reset		--	--	--	--	--	--	--	--

In A8107M0, user can read the RF state from MODE register

CER: RF chip enable status.
[0]: RF chip is disabled. **[1]:** RF chip is enabled.
XER: Internal crystal oscillator enabled status.
[0]: Crystal oscillator is disabled. **[1]:** Crystal oscillator is enabled.
PLLE: PLL enabled status.
[0]: PLL is disabled. **[1]:** PLL is enabled.
TRER: TRX state enabled status.
[0]: TRX is disabled. **[1]:** TRX is enabled.
TRSR: TRX Status Register.
[0]: RX state. **[1]:** TX state.

In A8107M0, user control RF MODE as well as read/write ram. By DPTR access and MOVX instruction, user change RF MODE and know RF status.

22.1.1 Strobe Command - Sleep MODE
Refer to Strobe Control Register, user can write 0x80 to Strobe Control Register directly to set RF into Sleep MODE.

22.1.2 Strobe Command - Idle MODE
Refer to Strobe Control Register, user can write 0x90 to Strobe Control Register directly to set RF into Idle MODE.

22.1.3 Strobe Command - Standby MODE
Refer to Strobe Control Register, user can write 0xA0 to Strobe Control Register directly to set RF into Standby MODE.

22.1.4 Strobe Command - PLL MODE
Refer to Strobe Control Register, user can write 0xB0 to Strobe Control Register directly to set RF into PLL MODE.

22.1.5 Strobe Command - RX MODE

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

22.1.6 Strobe Command - TX MODE

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

22.2 RF Reset Command

In addition to power on reset (POR), A8107M0 could issue software reset (80h) to RF by setting MODE Register (0800h). A8107M0 generates an internal signal “RESETN” to initial RF circuit. After reset command, RF state is in standby MODE and re-calibration is necessary.

22.3 FIFO Accessing Command

Before TX delivery, user only needs to write wanted data into TX FIFO (0x50001400 ~ 0x500014FF) in advance. Similarly, user can read RX FIFO (0x50001500 ~ 0x500015FF) 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.

A8107M0 supports separated 256-bytes TX and RX FIFO. To use A8107M0’s FIFO MODE, user just needs to enable FMS =1 (01h). For FIFO accessing, TX FIFO (write-only) and RX FIFO (read-only) share the same register address 05h. TX FIFO represents transmitted payload. On the other hand, RX circuitry SYNChronizes ID Code and stores received payload into RX FIFO.

22.4 Packet Format of FIFO MODE

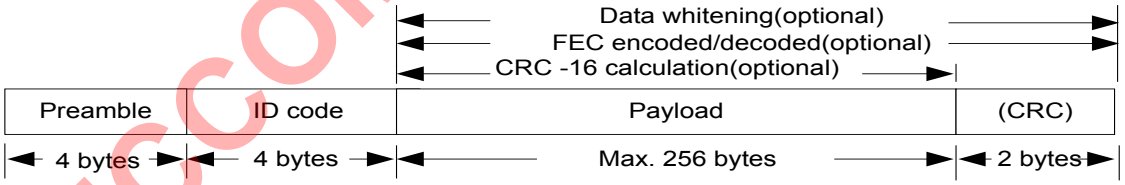


Figure 22.1 Packet Format of FIFO MODE

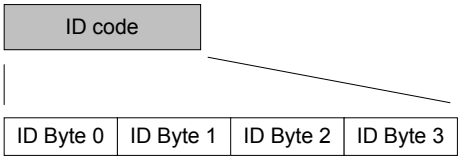


Figure 22.2 ID Code Format

Preamble:
The packet is led by preamble composed of alternate 0 and 1. If the first bit of ID code is 0, preamble shall be 0101…0101. In the contrast, if the first bit of ID code is 1, preamble shall be 1010…1010.
Preamble length is recommended to set 4 bytes by PML [1:0] (20h).

ID code:

ID code is recommended to set 4 bytes by IDL=1 (20h). ID Code is sequenced by Byte 0, 1, 2 and 3. If RX circuitry cheCKS the ID code correct, payload will be written into RX FIFO. In special case, ID code could be set error tolerance (0~ 3bit error) by ETH [1:0] (21h) for ID SYNChronization check.

Payload:
Payload length is programmable by FEP [7:0] (03h). The physical FIFO depth is 256 bytes.

CRC (option):
In FIFO MODE, if CRC is enabled (CRCS=1, 20h), 2-bytes of CRC value is transmitted automatically after payload. In the same way, RX circuitry will check CRC value and show the result to CRC Flag (00h).

22.5 Transceiver Frequency

A8107M0 is a half-duplex transceiver with embedded PA and LNA. The receiver is a low-IF architecture consisting of a LNA, down conversion miXERs, polyphase channel filters and IF limiting amplifiers with RSSI. The transmitter is direct modulation architecture with 6 dBm maximum output power and 35 dB power control range. For TX or RX frequency setting, user just needs to set up one register, CHN (0811h), for frequency agility.

A8107M0's main PLL features are:

- Fractional-N to generate RX/TX frequencies for all ISM 2.4 GHz channels
- Autonomous calibration loops for stable operation within the operating range
- Fast PLL settling to support frequency hopping

During receive operation, the frequency synthesizer works as a local oscillator. During transmit operation, the voltage-controlled oscillator (VCO) is directly modulated to generate the RF transmit signal. The frequency synthesizer is implemented as a fractional-N PLL.

F_{LO} = 2400 + (CHN x 0.5) in [MHz], where CHN is the channel number, addr 0Fh.

A8107M0's LO frequency $F_{LO} = F_{LO\_BASE} + F_{OFFSET}$. Therefore, A8107M0 is very easy to implement frequency hopping by **ONE register setting, (CHN, 0Fh).** In general, user can plan the wanted channels by a CHN Look-Up-Table between master and slaves for two-way frequency hopping. Below is the LO frequency block diagram.

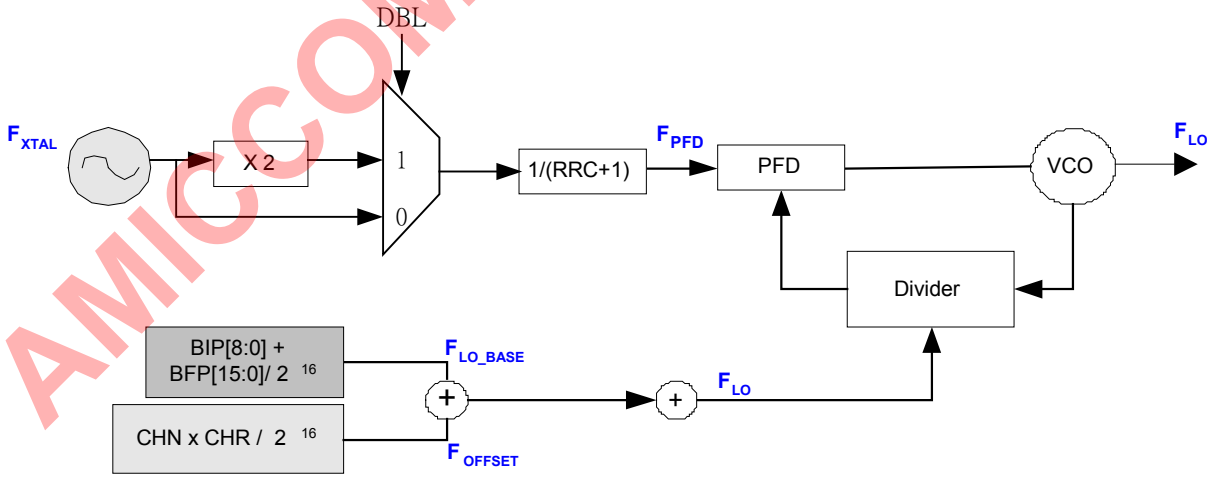


Figure 22.3 Block Diagram of Local Oscillator.

22.5.1 RF Clock

The master clock of A8107M0 (F_{CSCK}= 32/64 MHz) is generated by the PLL clock generator which reference frequency (F_{CGR}= 2/4 MHz) is derived from frequency divider of crystal oscillator.

$F_{CGR} = \frac{F_{XREF}}{(GRC[3:0]+1)}$, where GRC [3:0] (0Eh) is the divide number to get F_{CGR} from crystal oscillator.

Below is block diagram of system clock where F_{XTAL} is the crystal frequency. User can set XS, GRC, CGS to get F_{CSCK} = 32/64MHz. F_{XREF} is a reference clock to generate F_{CGR} and F_{SPLL} . After delay circuitry, F_{CSCK} (32/64MHz) is derived. And with BWS setting, the system clock F_{SYCK} can be fixed to 8MHz.

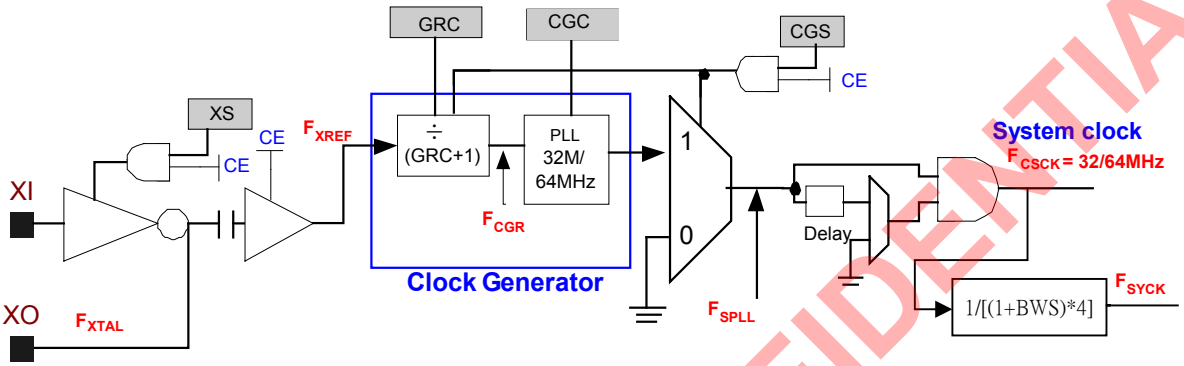


Figure 22.4 RF Clock Block Diagram.

Below is the setting table of system clock for both 1MHz and 2MHz data rate

Data rate	F_{XTAL}	F_{XREF}	F_{CGR}	GRC [3:0]	XS	CGS	CGC	BWS	F_{CSCK}	F_{SYCK}
1M	16 MHz	16 MHz	2 MHz	[0111]	1	1	0	0	32MHz	8MHz
2M	16 MHz	16 MHz	2 MHz	[0011]	1	1	1	1	64MHz	8MHz

22.5.2 LO Frequency Setting

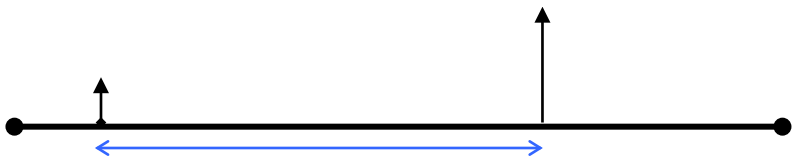
To set up 2.4GHz LO Frequency (F_{LO}), user can refer to below 4 steps.

1. Set the base frequency (**F_{LO_BASE}**) by PLL Register II (0812h) and III (0813h).
   Recommend to set **F_{LO_BASE} ~ 2400.001MHz**.

2. Set channel step F_{CHSP} = 500KHz by PLL Register IV (0814h).

3. Set CHN [7:0] to get offset frequency by PLL Register I (0811h).
   F_{OFFSET} = CHN [7:0] * F_{CHSP}

4. LO frequency is equal to base frequency plus offset frequency.
   F_{LO} = F_{LO_BASE} + F_{OFFSET}



22.5.2.1 How to set F_{LO_BASE}

Regarding to LO frequency setting, Table 21.1 shows 2400.001 MHz base frequency by 16MHz Xtal.

STEP	ITEMS	VALUE	NOTE
1	F_{XTAL}	16 MHz	Crystal Frequency
2	BIP[7:0]	0x96	To get F_{LO_BASE} =2400 MHz
3	BFP[15:0]	0x0004	To get F_{LO_BASE} ~ 2400.001 MHz
4	F_{LO_BASE}	~2400.001 MHz	LO Base frequency

Table 22.1 How to configure F_{LO_BASE}.

22.5.2.2 How to set $F_{LO} = F_{LO\_BASE} + F_{OFFSET}$

Regarding to frequency offset scheme, Table 21.2 shows Channel 11 (2405.001 MHz) by 16MHz Xtal.

STEP	ITEMS	VALUE	NOTE
1	F_{LO_BASE}	~2400.001 MHz	After cofigure BIP and BFP
2	CHR[14:0]	0x0800	To get F_{CHSP}= 500 KHz
3	CHN[7:0]	0x0A	To set channel number = 10
4	F_{OFFSET}	5 MHz	To get F_{OFFSET}= 500 KHz * (CHN) = 5MHz
5	F_{LO}	~2405.001 MHz	To get F_{LO}= F_{LO_BASE} + F_{OFFSET}

Table 22.2 How to configure F_{LO}.

22.6 State machine

In chapter 9.2 and chapter 21.1, user can learn both accessing A8107M0's control registers as well as issuing Strobe commands.

22.6.1 Key states

A8107M0 supports 6 key operation states. Those are,
(1) Standby MODE
(2) Sleep MODE
(3) Idle MODE
(4) PLL MODE
(5) TX MODE
(6) RX MODE

After power on reset or software reset or deep sleep MODE, user has to do calibration process because all control registers are in initial values. The calibration process of A8107M0 is very easy, user only needs to issue Strobe commands and enable calibration registers. After calibration, A8107M0 is ready to do TX and RX operation. User can start wireless transmission.

Strobe Command								Description
b7	b6	b5	b4	b3	b2	b1	b0	
1	0	0	0	x	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

MODE	RF Register retention	RF Regulator	Xtal Osc.	VCO	PLL	RX	TX	Strobe Command
Sleep	Yes	ON	OFF	OFF	OFF	OFF	OFF	(1000-xxxx)b
Idle	Yes	ON	OFF	OFF	OFF	OFF	OFF	(1001-xxxx)b

Standby	Yes	ON	ON	OFF	OFF	OFF	OFF	(1010-xxxx)b
PLL	Yes	ON	ON	ON	ON	OFF	OFF	(1011-xxxx)b
TX	Yes	ON	ON	ON	ON	OFF	ON	(1101-xxxx)b
RX	Yes	ON	ON	ON	ON	ON	OFF	(1100-xxxx)b

Remark: x means “don’t care”

Table 22.3 Operation MODE and strobe command.

22.6.2 FIFO MODE

This MODE is suitable for the requirements of general purpose applications and can be chosen by setting FMS = 1. After calibration, user can issue Strobe command to enter standby MODE where write TX FIFO or read RX FIFO. From standby MODE to packet data transmission, only one Strobe command is needed. Once transmission is done, A8107M0 is auto back to standby MODE. Figure 21.4 and Figure 21.5 are TX and RX timing diagram respectively. Figure 21.6 illustrates state diagram of FIFO MODE.

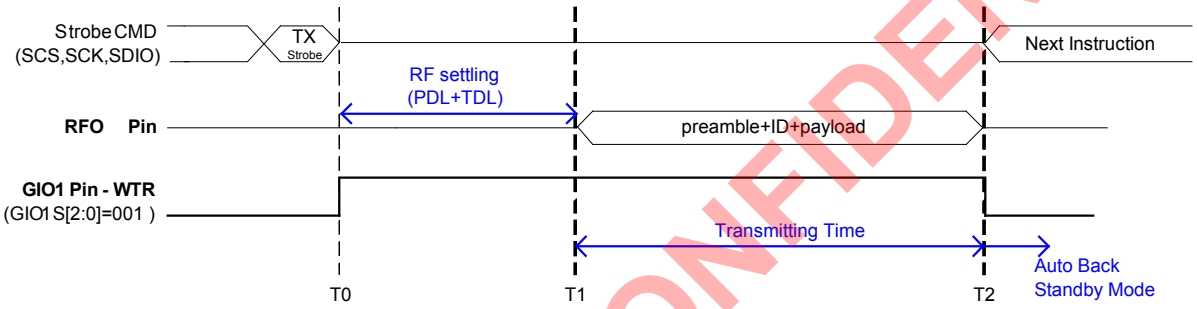


Figure 22.5 TX timing of FIFO MODE.

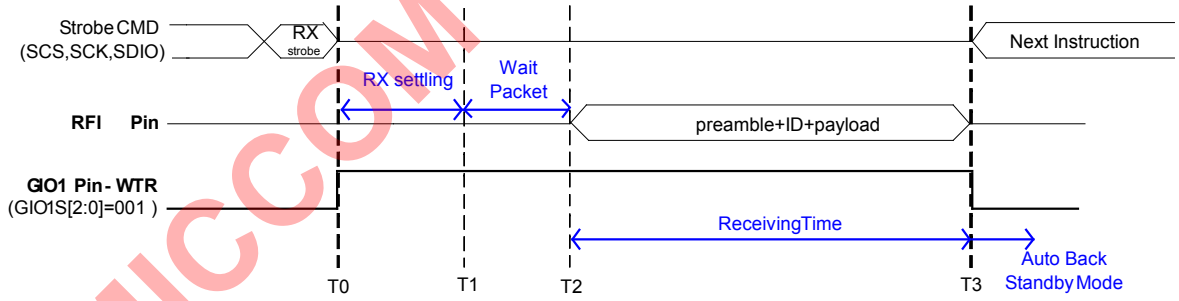


Figure 22.6 RX timing of FIFO MODE.

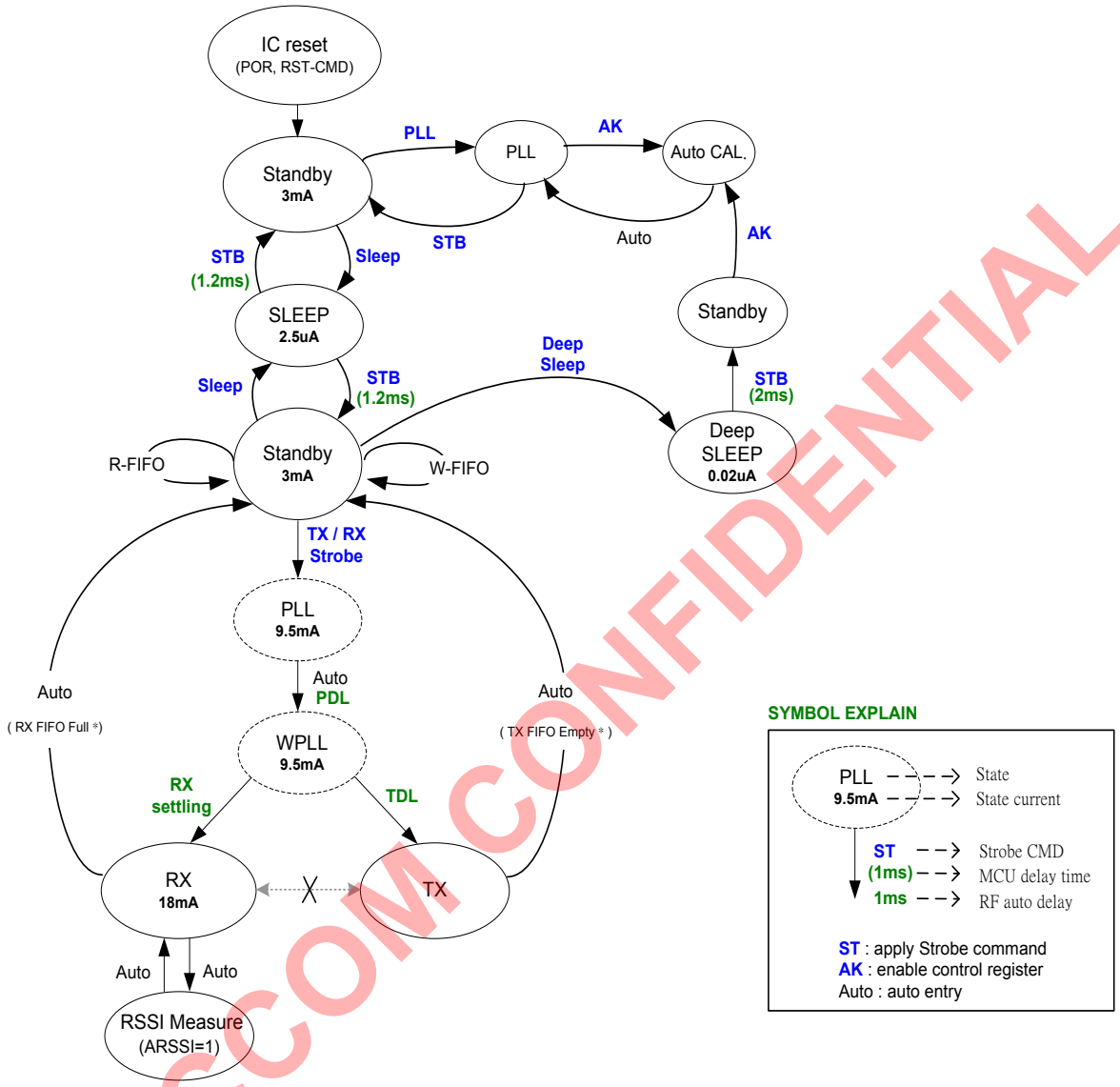


Figure 22.7 State diagram of FIFO MODE.

23. TFT LCD Controller

The TFT LCD controller directly drives LCD panel with 16KB SRAM automatically. See figure 23.1.

23.1 TFT LCD overview

M0 SRAM : 0x20000000~0x20003FFF 16KB
LCD SRAM_0 (local 0x0000~0x1FFF) : 0x20004000~0x20005FFF 8KB
LCD SRAM_1 (local 0x2000~0x3FFF) : 0x20006000~0x20007FFF 8KB

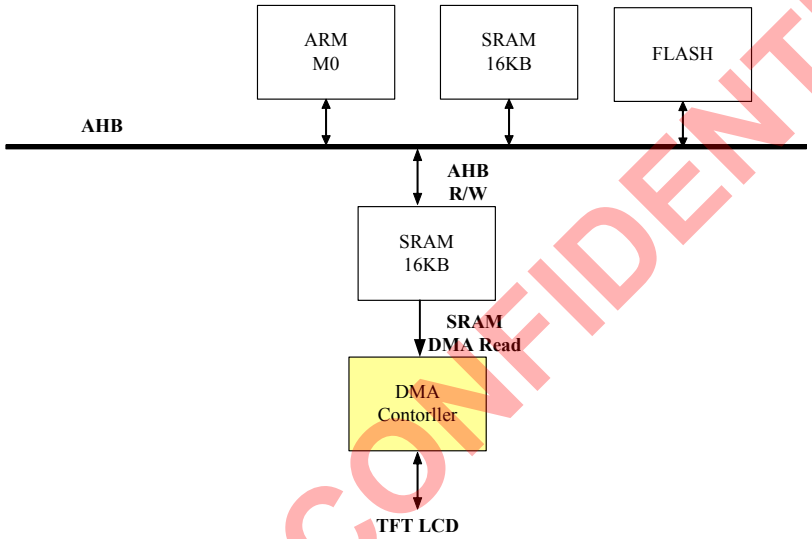


Figure 23.1 LCD diagram.

23.2 TFT LCD SRAM format

TFT LCD SRAM format is RGB565 as below figure 23.2.

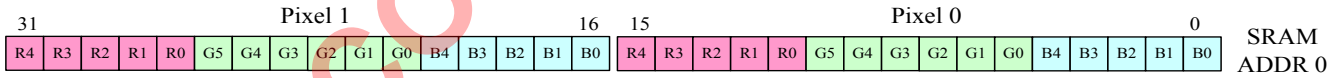


Figure 23.2 LCD SRAM format.

23.3 TFT LCD DMA flow

Operation flow as below:
A. CPU fill the first half of the picture
B. CPU enable DMA and wait for interrupt then fill the second half of the picture
C. CPU enable DMA and wait for interrupt then fill the first half of the picture.

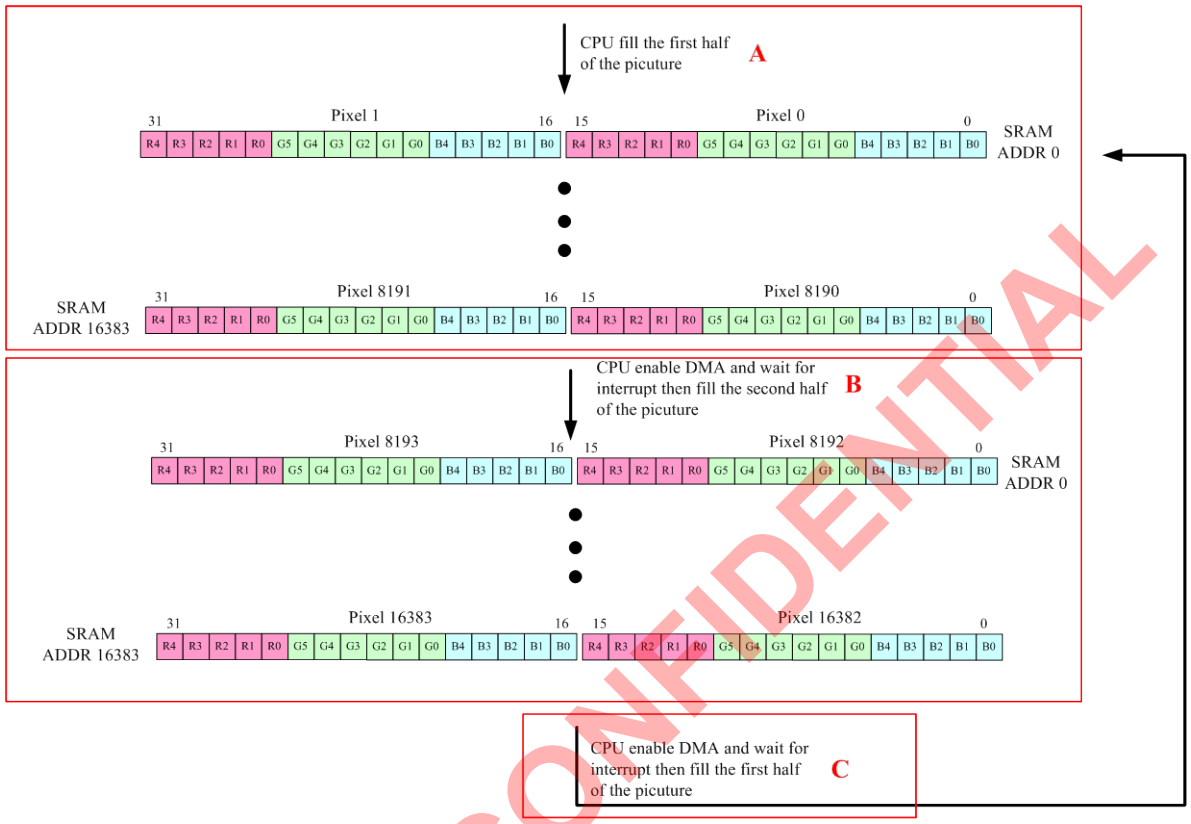
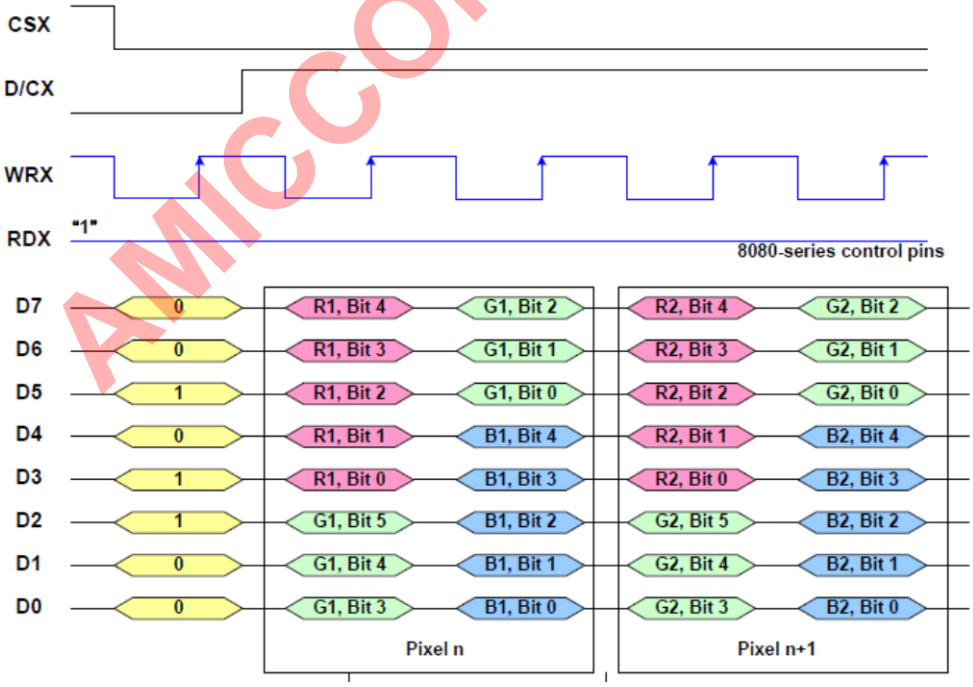


Figure 23.2 LCD DMA flow.

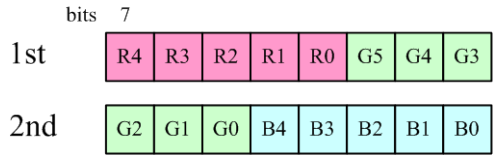
23.4 TFT LCD 8080 output interface(I80)

Only support three interface with RGB565 as below :

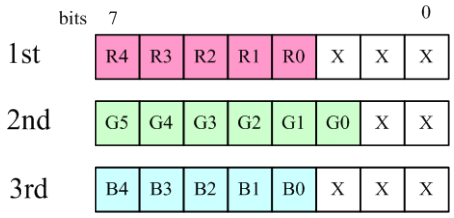
1. Panel data bus : 8-bits bus



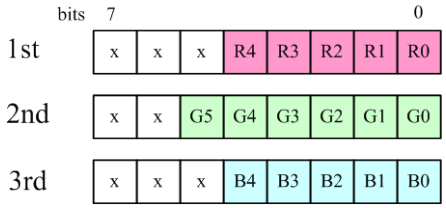
a. RGB_Format 0 : finish a RGB565 pixel with two WRX cycle.



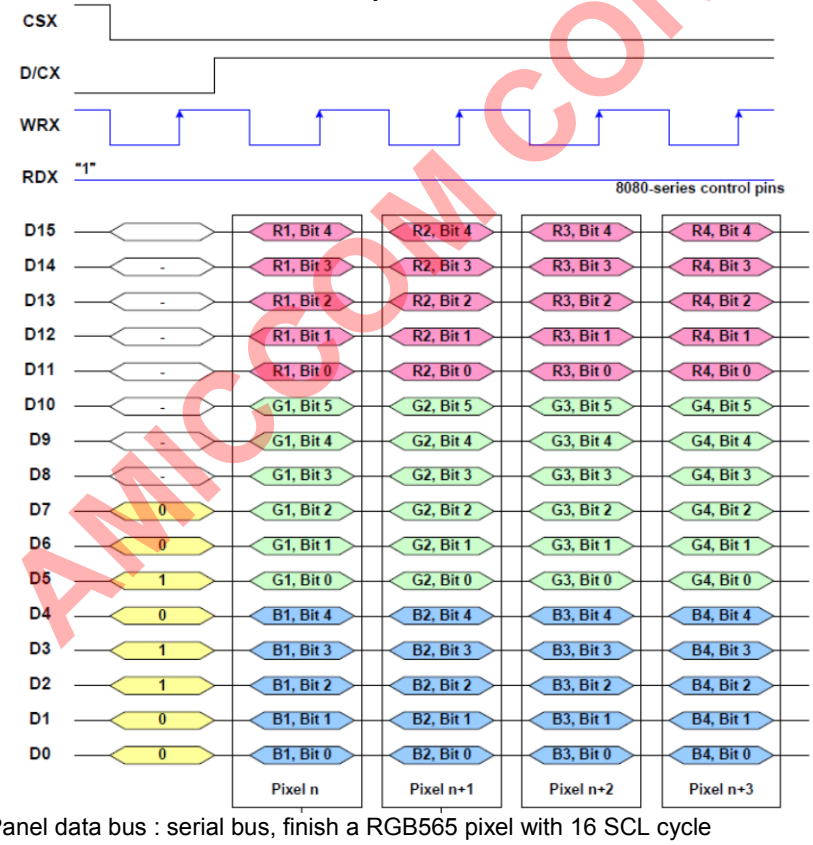
b. RGB_Format 1 : three cycle



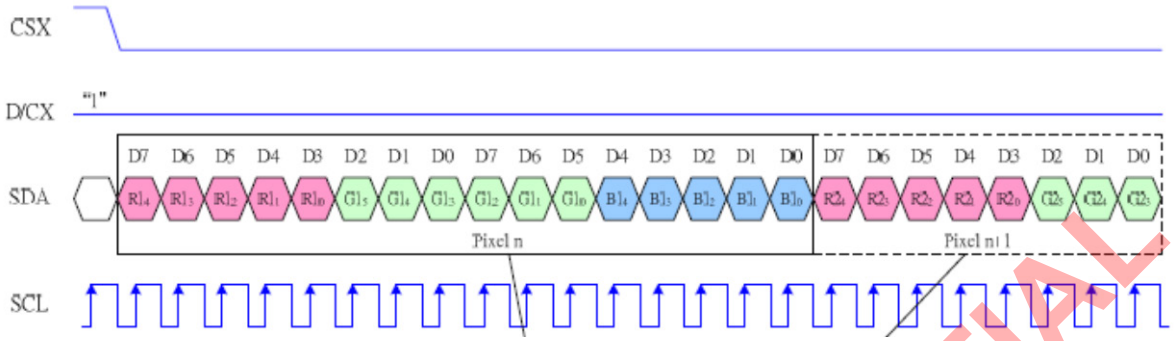
c. RGB_Format 2 : three cycle



2. Panel data bus : 16-bits bus with 1 cycle



3. Panel data bus : serial bus, finish a RGB565 pixel with 16 SCL cycle



24. Encryption and Autherfication

For Bluetooth Low Energy application, it uses AES-128 link layer encryption block with Counter MODE CBC MAC defined in IETF RFC 3610. A8107M0 integrates AES-128 encryption core for user to encrypt data using AES algorithm with 128-bits key. The AES core also supports CBC-MAC for authentication.

24.1 AES

AES (Advanced Encryption Standard) is a symmetric block cipher on 128-bits data bloCKS, it consists 10 encryption rounds during encryption process. Figure 22.1 shows the structure of the AES encryption. AES can be divided into four basic operation block where data are treated at either byte or bit level. The array of bytes organized as a 4x4 matrix is also called “state” and those four basic steps, AddRoundKey, SubBytes, ShiftRows, and MixColumns. These four steps describe one round of the AES operation. The number of rounds is depended on the key length, i.e., 10, 12, and 14 rounds for the key length of 128, 192, and 256 bits respectively. The block diagram of the AES with 128 bit data is shown below in Figure 22.1.

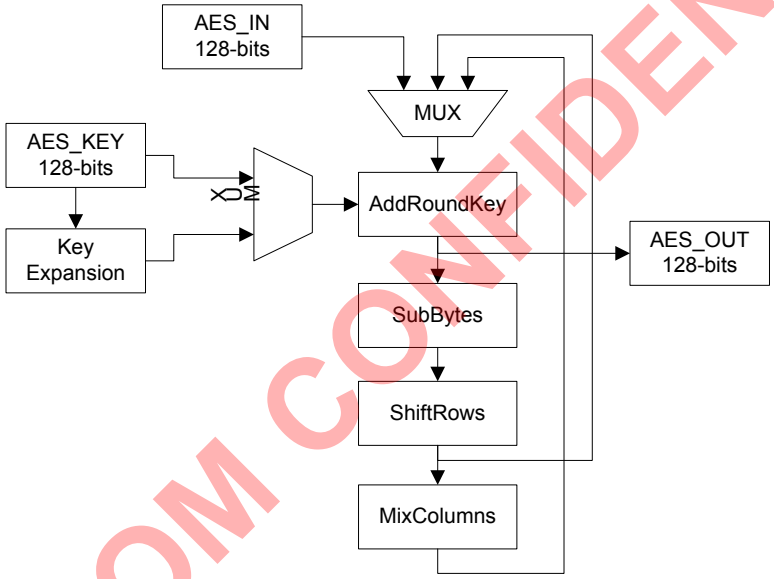


Figure 23.1 Structure of the AES-128 Core.

24.1.1 AddRoundKey

Each byte of the array is added to a byte of the corresponding array of round subkeys. Excluding the first and the last round, the AES with 128-bits round key proceeds for 9 iterations. Round keys are generated by a procedure called round key expansion or key scheduling. Those sub-keys are derived from the original key by XOR the two previous columns. For columns that are in multiples of four, the process involves round constants addition, S-Box and shift operations.

24.1.2 SubBytes

This operation is a non-linear byte substitution. It composes of two sub-transformations; multiplicative inverse and affine transformation. In most implementations, these two sub-steps are combined into a single table lookup called S-Box.

24.1.3 ShiftRows

This step is a simple permutation process, operates on individual rows, i.e. each row of the array is rotated by a CERtain number of byte positions.

24.1.4 MixColumns

The MixColumns transformation is a substitution step that makes of arithmetic over GF(2^8). Column vector is multiplied by a fixed matrix where bytes are treated as a polynomial of degree less than 4.

24.1.5 CCM

CCM is an authenticated encryption algorithm designed to provide both authentication and confidentiality. It is only defined for block ciphers with a block length of 128 bits. It uses encryption algorithm to generate encrypted and authenticated data at the same time. The AES-CCM process is shown in Figure 22.2.

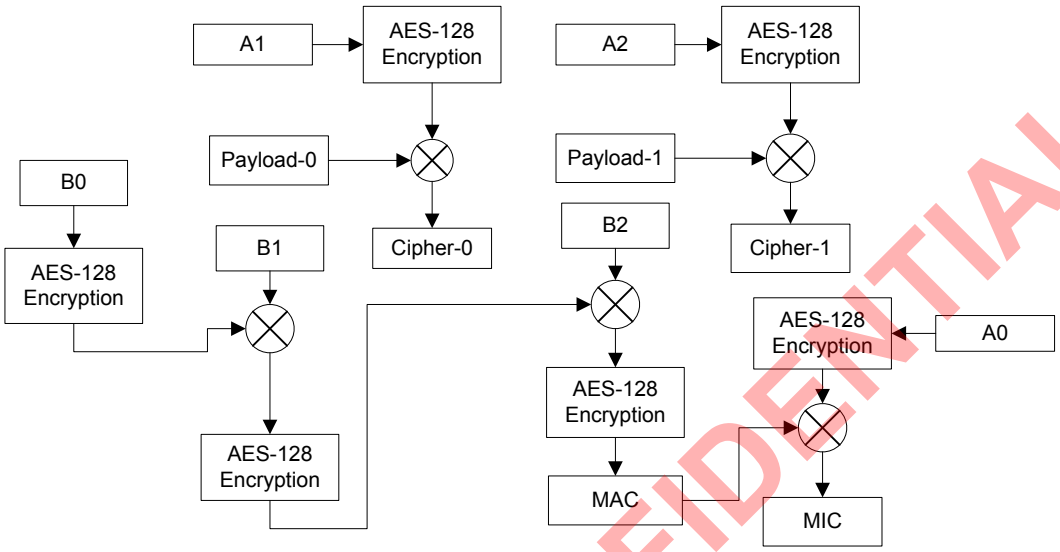


Figure 23.2 CCM Encryption Procedure.

CCM authentication starts by defining a sequence of bloCKS B_0, B_1, and B_2 and thereafter CBC-MAC is applied to those bloCKS so that the authentication field MIC can be obtained. CCM uses the A_0, A_1, and A_2 bloCKS to generate key-stream that is used to encrypt the MIC and the payload. Block A_0 is always used to encrypt and decrypt the MIC. A_1 and A_2 bloCKS are generated as needed for encryption or decryption of the payload.

25. Flash memory controller

25.1 Flash controller command

Flash MODE register (Address: 0x4001_F100)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0	R/W	RBB	TRIM	-	INF	TM[3]	TM[2]	TM[1]	TM[0]
Reset		1	0	0	0	0	0	0	0

TM[3:0] (Flash Test MODE enable)

INF (Flash Information page enable)
[1]: Enable
[0]: Disable

TRIM (Flash trim MODE enable)
[1]: Enable
[0]: Disable

RBB (Flash Ready status output)
[1]: Ready
[0]: Not Ready.

Flash control register (Address: 0x4001_F104)

Address/Name	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0	R/W	CE	F2M	F1M	MERASE	SERASE	PERASE	PROG	WRONLY
Reset		1	0	0	0	0	0	0	0

WRONLY (Flash Page Write enable)
[1]: Enable
[0]: Disable

PROG (Flash Page Program internal Erase and Write enable)
[1]: Enable
[0]: Disable

PERASE (Flash Page Erase enable)
[1]: Enable
[0]: Disable

SERASE (Flash Sector Erase enable)
[1]: Enable
[0]: Disable

MERASE (Flash Mass Erase enable)
[1]: Enable
[0]: Disable

F1M (Flash 0~1M bits select)
[1]: Enable
[0]: Disable

F2M (Flash 1~2M bits select)
[1]: Enable
[0]: Disable

CE (Flash chip enable)

[1]: Enable
[0]: Disable

25.2 Flash controller operations

INF Truth Table

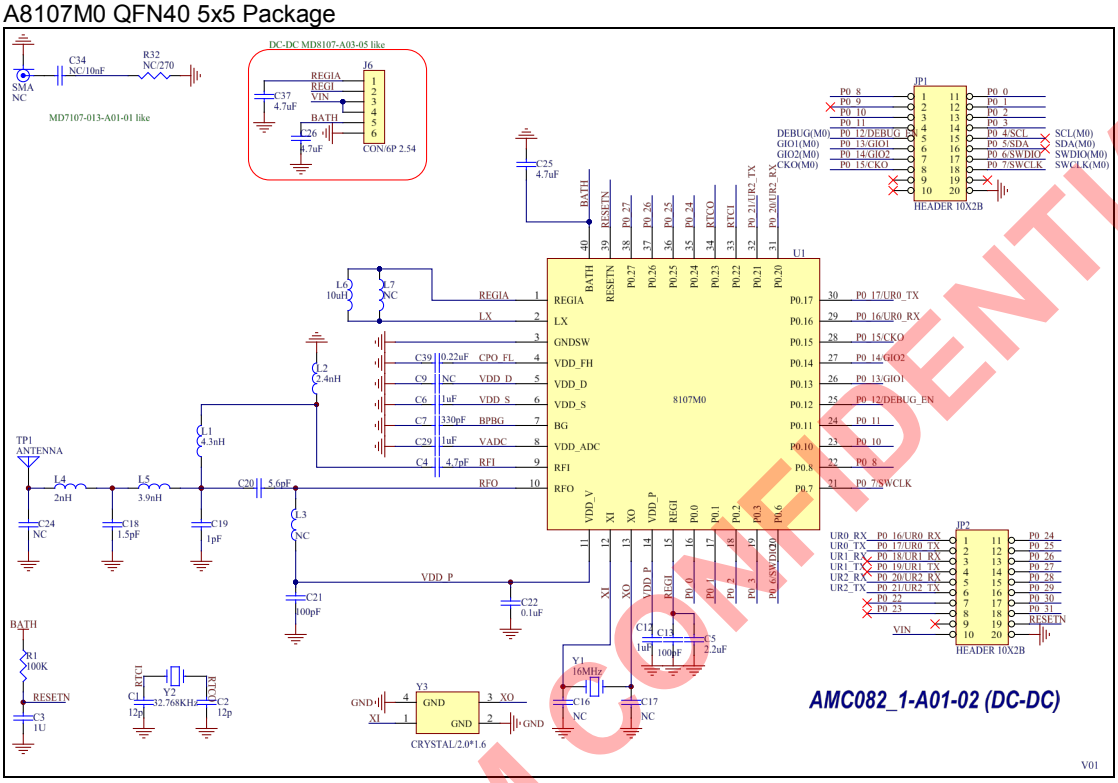
Mode	INF = Logic 1	INF= Logic 0
Read	Read the information page	Read the main memory block
Write	Write the information page	Write the main memory block
Program	Program the information page	Program the main memory block
Page Erase	Erase the information page	Erase the main memory block
Sector Erase	Erase the information page	Erase the main memory block
Mass Erase	Erase both main memory block and information page.	Erase the main memory block

Page Write
The Page Write time (T_{WR}) also includes T_{BUSY} and the internal Program cycle. During this period, the internal Program cycle will start when RBB is set at logic 0 after T_{BUSY}. This operation allows writing page function to change data from “1” to “0”. The data of the selected page need to be erased by performing Page Erase, Sector Erase, or Mass Erase operation before Page Write operation. The data can be written into Assembly Buffer separately, and then be written in main memory block. The writing sequence allows up to 32 X 32 bits in a page written at the same time.
The written page cannot be re-written if Erase operation is not performed in advance; even only partial data in the selected page are written. For example, 8 X 32 bits are written into the selected page by using Page Write command. After this command has been done, the rest 24 X 32 bits cannot be written into the same selected page if Page Erase command is not performed in advance. It is suggested to use Page Write command to change whole page data.
Page/Sector/Mass Erase
Since in Page Write operation, the data will be changed to “0”, Erase operations should be applied before the Page Write operation to reset all the data back to “1”. The whole data in a page, a sector or all the main memory can be erased at a time by performing Page Erase, Sector Erase, or Mass Erase sequences accordingly.
Ready/Busy Port (RBB) During a Write, Program, Erase Cycle
The RBB port provides status of internal operations (Page Write, Page Program, Page Erase, Sector Erase, Mass Erase, Power On Reset, and Low Standby MODEs). Users can know whether the eFlash macro is ready to serve next command by monitoring the RBB. Any other commands are ignored while the RBB is at logic 0 state.

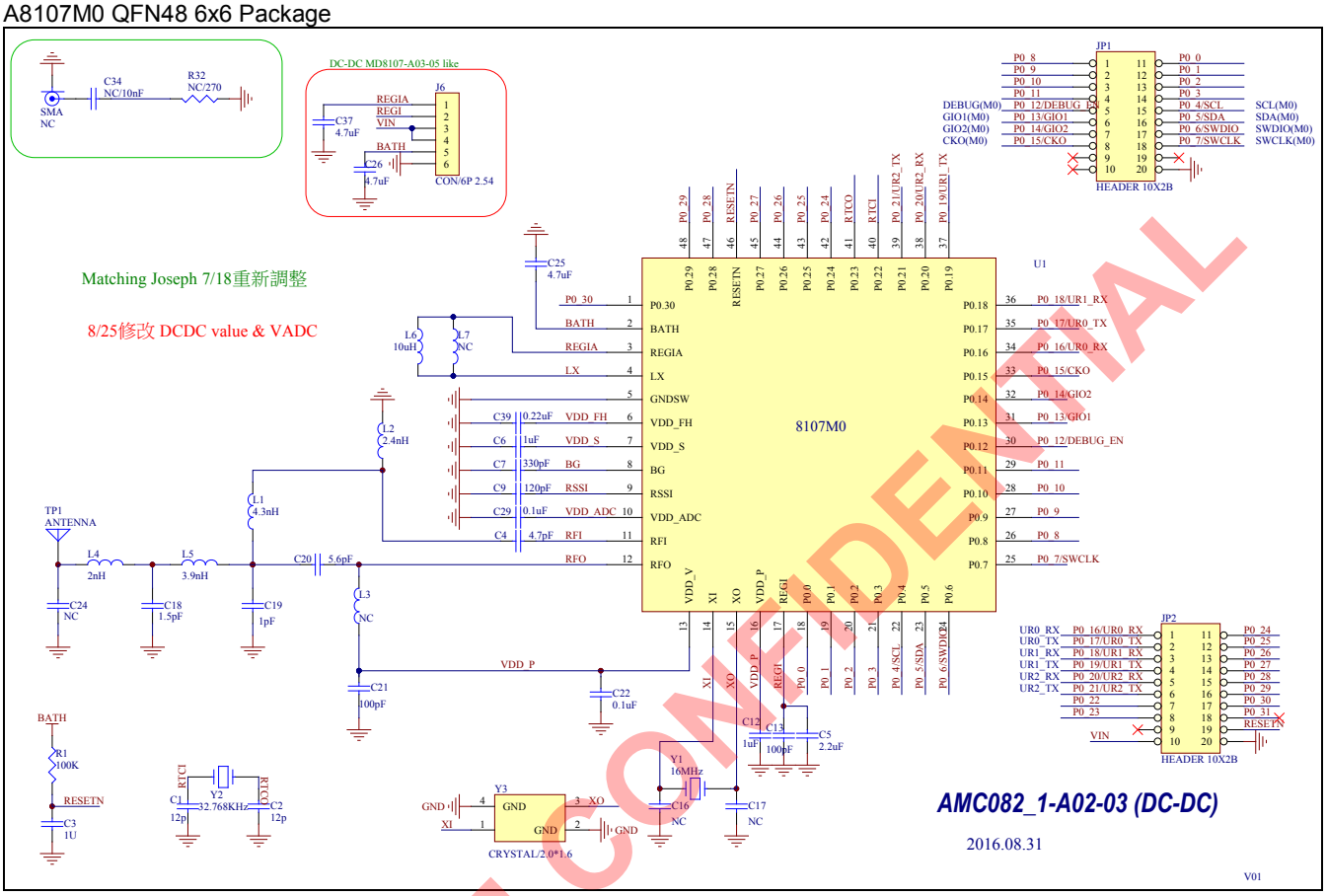
26. Application circuit

Below are AMICCOM's ref. design circuits. For more details, please refer AMICCOM standard module, MD8107M0-Axx or contact AMICCOM's FAE for more details.

A8107M0 QFN40 5x5 Package



A8107M0 QFN48 6x6 Package



27. Abbreviations

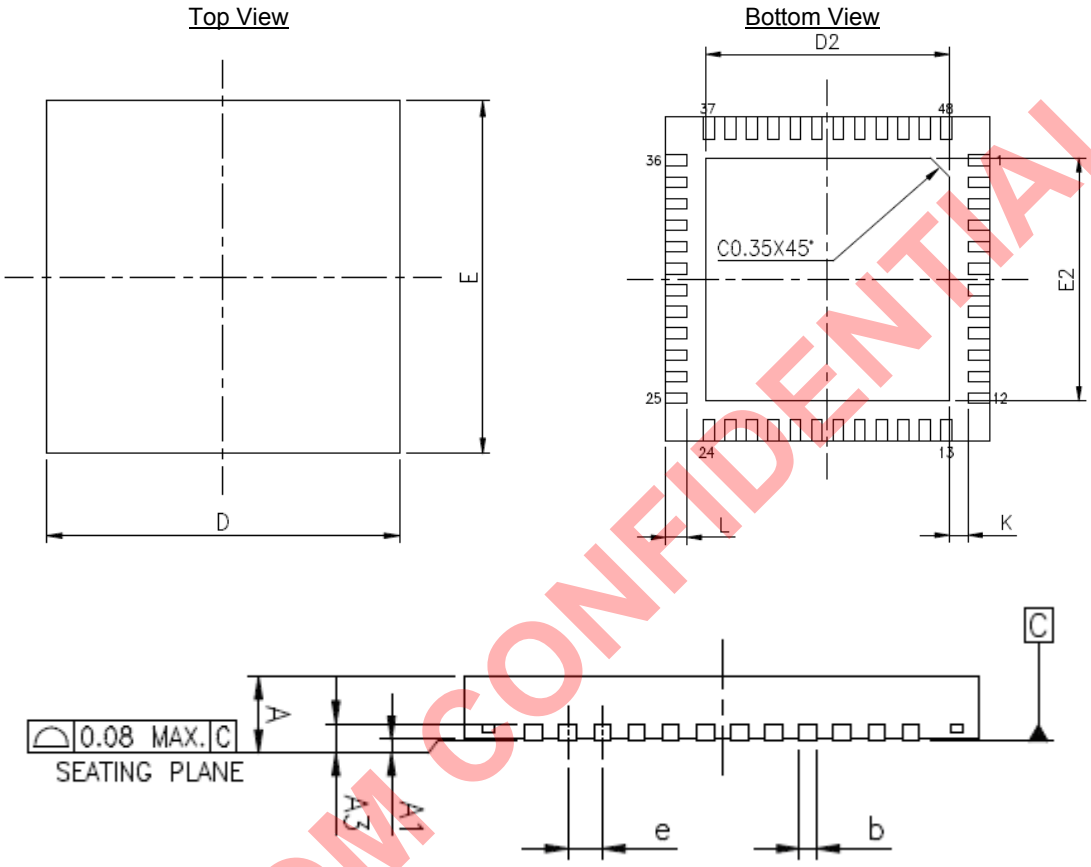
ADC	Analog to Digital Converter
AIF	Auto IF
FC	Frequency Compensation
AGC	Automatic Gain Control
BER	Bit Error Rate
BLE	Bluetooth Low Energy
BW	Bandwidth
CD	Carrier Detect
CHSP	Channel Step
CRC	Cyclic Redundancy Check
DC	Direct Current
FEC	Forward Error Correction
FIFO	First in First out
FSK	Frequency Shift Keying
ID	Identifier
ICE	In Circuit Emulator
I^2C	Inter-Integrated Circuit
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
UART	Universal ASYNChronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator
XOSC	Crystal Oscillator
XREF	Crystal Reference frequency
XTAL	Crystal

28. Ordering Information

Part No.	Package	Units Per Reel / Tray
A81U07F810GAQ6C/Q	QFN48L with 31I/O, DC/DC Pb Free, Tape & Reel, -40℃～85℃	3K
A81U07F810GAQ6C	QFN48L with 31I/O, DC/DC Pb Free, Tray, -40℃～85℃	490EA
A81U07F8102AQ5A/Q	QFN40L with 23I/O, DC/DC Pb Free, Tape & Reel, -40℃～85℃	3K
A81U07F8102AQ5A	QFN40L with 23I/O, DC/DC Pb Free, Tray, -40℃～85℃	490EA

29. Package Information

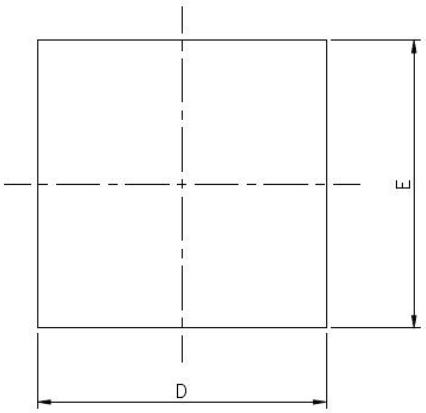
QFN6*6 48L Outline Dimensions



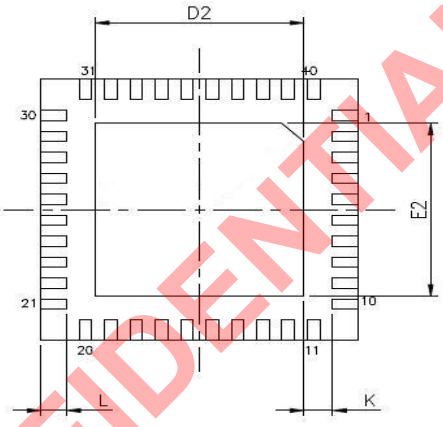
Symbol	Dimensions in inches			Dimensions in mm		
	Min	Nom	Max	Min	Nom	Max
A	0.028	0.030	0.031	0.7	0.75	0.8
A_1	0	0.001	0.002	0.00	0.02	0.05
A_3		0.009 REF.			0.23REF.	
b	0.006	0.008	0.010	0.15	0.2	0.25
D		0.240			6.1 BSC	
D_2	0.146	0.177	0.179	3.70	4.50	4.55
E		0.240			6.1BSC	
E_2	0.146	0.177	0.179	3.70	4.50	4.55
e		0.016BSC			0.4BSC	
L	0.013	0.016	0.020	0.32	0.4	0.48
k		0.008			0.2	

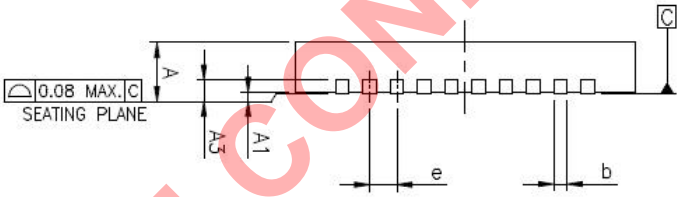
QFN5*5 40L Outline Dimensions

Top View



Bottom View

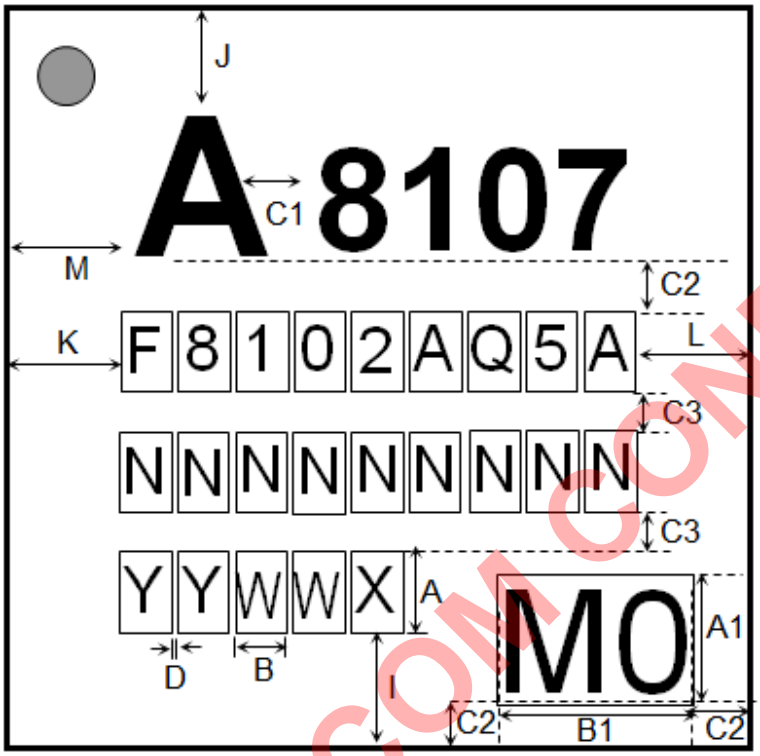




Symbol	Dimensions in inches			Dimensions in mm		
	Min	Nom	Max	Min	Nom	Max
A	0.028	0.030	0.031	0.70	0.75	0.80
A_1	0.000	0.001	0.002	0.00	0.02	0.05
A_3		0.008 REF			0.20 REF	
b	0.006	0.008	0.010	0.15	0.20	0.25
D	0.194	-	0.200	4.924	-	5.1
D_2	0.126	-	0.138	3.20	-	3.50
E	0.194		0.200	4.924		5.1
E_2	0.126	-	0.138	3.20	-	3.50
e		0.016			0.40	
L	0.013	0.016	0.019	0.324	0.40	0.5
k		0.008			0.2	

30. Top Marking Information

- Part No. : **A81U07F8102AQ5A**
- Pin Count : **40**
- Package Type : **QFN**
- Dimension : **5*5 mm**
- Mark Method : **Laser Mark**
- Character Type : **Arial**



❖ **CHARACTER SIZE : (Unit in mm)**

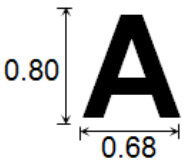
A : 0.55 **A1 : 0.75**
B : 0.36 **B1 : 1.10**
C1 : 0.25 **C2 : 0.3** **C3 : 0.2**
D : 0.03
M : 1.5

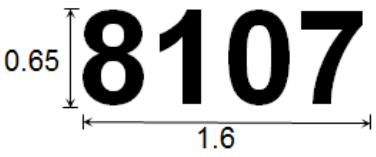
Y Y W W : DATECODE

X : PKG HOUSE ID

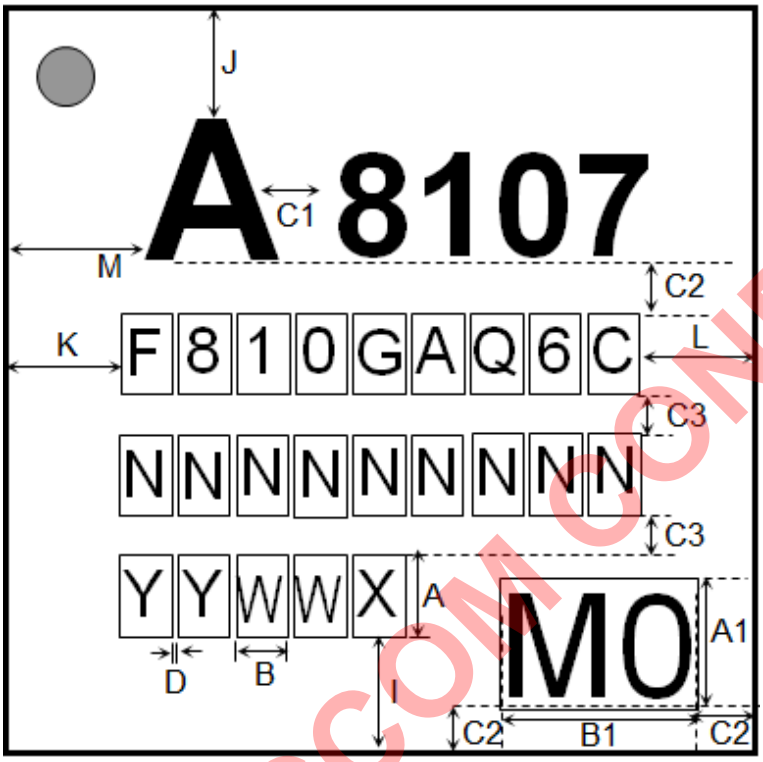
N N N N N N N N N : LOT NO. (max. 9 characters)

I=J
K=L





- Part No. : **A81U07F810GAQ6C**
- Pin Count : **48**
- Package Type : **QFN**
- Dimension :**6*6 mm**
- Mark Method : **Laser Mark**
- Character Type :**Arial**



❖ **CHARACTER SIZE : (Unit in mm)**

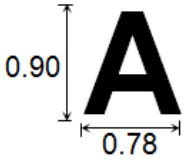
A : 0.65 **A1 : 0.75**
B : 0.45 **B1 : 1.10**
C1 : 0.3 **C2 : 0.4** **C3 : 0.3**
D : 0.03
M : 1.5

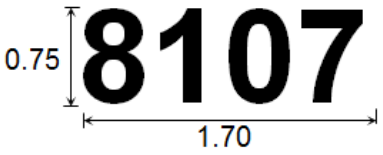
Y Y W W : DATECODE

X : PKG HOUSE ID

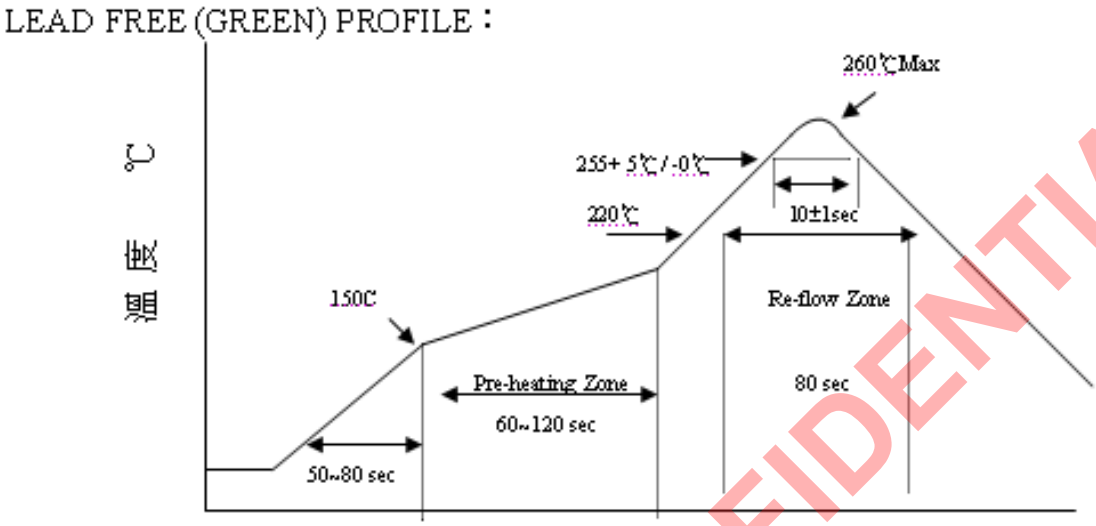
N N N N N N N N N : LOT NO.
(max. 9 characters)

I=J
K=L





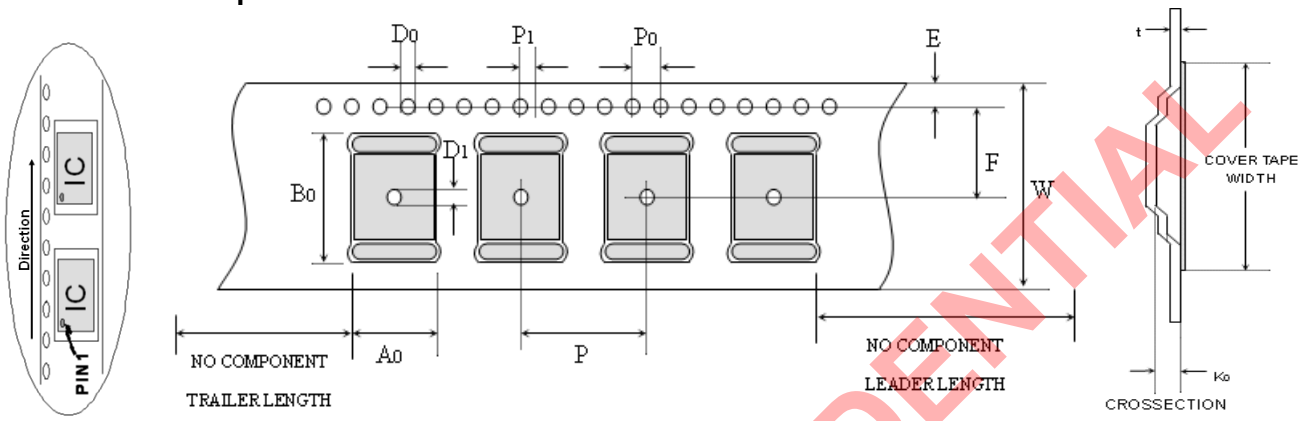
31. Reflow Profile



Actual Measurement Graph

32. Tape Reel Information

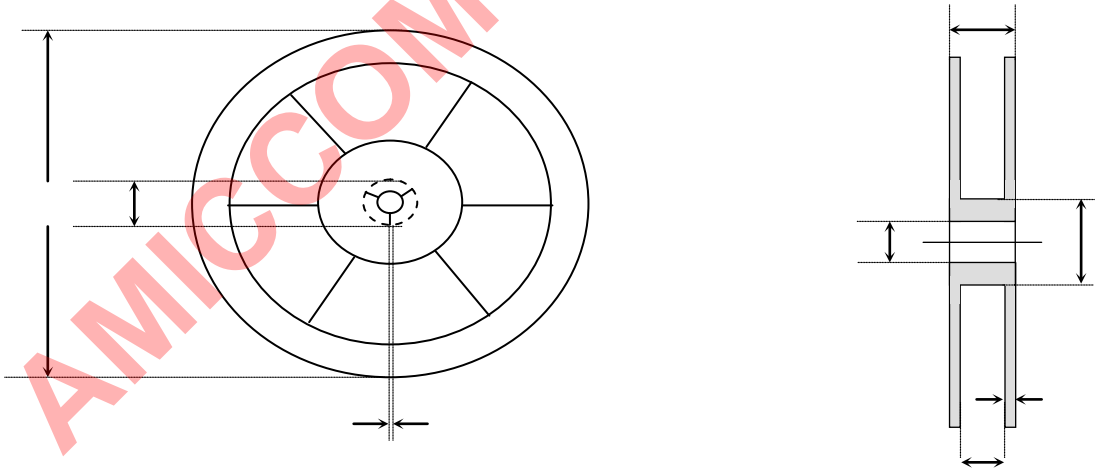
Cover / Carrier Tape Dimension



Unit: mm

TYPE	P	A0	B0	P0	P1	D0	D1	E	F	W	K0	t	Cover tape width
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
QFN 6*6	12±0.1	6.3±0.1	6.3±0.1	4±0.2	2±0.1	1.5±0.1	1.5±0.5	1.75 ±0.1	7.5 ±0.1	16±0.3	1.15 ±0.2	0.3 ±0.05	13.3± 0.1

REEL DIMENSIONS



Unit: mm

TYPE	G	N	M	D	K	L	R
QFN5*5	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
QFN6*6	17±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

33. 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

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