

**Document Title**

A3107M0 Data Sheet, Bluetooth Low Energy SoC

**Revision History**

<u>Rev. No.</u>	<u>History</u>	<u>Issue Date</u>	<u>Remark</u>
0.0	Initial issue.	May, 2018	Objective

AMICCOM CONFIDENTIAL

**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 .....	7
2. Typical Applications .....	7
3. Features .....	8
4. Pin Configurations .....	9
4.1 QFN40 5x5 .....	9
4.2 QFN48 6x6 .....	10
5. Pin Descriptions (I: input; O: output, I/O: input or output) .....	11
5.1 A3107M0 QFN40 5x5 Package .....	11
5.2 A3107M0 QFN48 6x6 Package .....	13
6. Chip Block Diagram .....	15
7. Absolute Maximum Ratings .....	16
8. Electrical Specification .....	17
9. Register List .....	20
9.1 Battery detect Register (Address: 0x50000000) .....	20
9.2 Power control Register 1 (Address: 0x50000008) .....	20
9.3 Power control Register 2 (Address: 0x5000000C) .....	22
9.4 Reset Register (Address: 0x50001000) .....	22
9.5 Strobe Register (Address: 0x50001004) .....	23
9.6 Status Register (Address: 0x50001010) .....	23
9.7 FIFO Register (Address: 0x50001020) .....	24
9.8 RSSI Register (Address: 0x50001244) .....	25
9.9 ADC Control Register I (Address: 0x50008000) .....	25
9.10 ADC Control Register II (Address: 0x50008004) .....	27
9.11 ADC Channel Register (Address: 0x50008008) .....	27
9.12 MODEC Register (Address: 0x50001008) .....	28
9.13 RF GPIO Register (Address: 0x50001028) .....	28
9.14 LO channel Register (Address: 0x500010C0) .....	30
9.15 TX Control Register (Address: 0x50001100) .....	31
9.16 TX Power 1 Register (Address: 0x50001104) .....	31
9.17 TX Modulation Register (Address: 0x5000110C) .....	32
9.18 CODE1 Register (Address: 0x50001180) .....	33
10. A3107M0 RF .....	35
10.1 Strobe Command Register 1 (Address: 0x50001004) .....	35
10.1.1 Strobe Command - Sleep MODE .....	35
10.1.2 Strobe Command - Idle MODE .....	35
10.1.3 Strobe Command - Standby MODE .....	36
10.1.4 Strobe Command - PLL MODE .....	36
10.1.5 Strobe Command - RX MODE .....	36
10.1.6 Strobe Command - TX MODE .....	36
10.2 RF Reset Command .....	36
10.3 FIFO Accessing Command .....	36
10.4 Packet Format of FIFO MODE .....	37
10.5 Transceiver Frequency .....	38
10.5.1 State machine .....	38
10.5.2 Key states .....	38
10.5.3 FIFO MODE .....	39
11. SoC Architectural Overview .....	41
11.1 ARM Cortex-M0 .....	41
11.1.1 Feature .....	41
11.2 Memory Organization .....	43
11.3 Nested Vectored Interrupt Controller (NVIC) .....	43
11.3.1 Feature .....	43
11.3.2 Exception Types and Interrupt Map .....	44
11.4 Reset source .....	46
11.5 Clock Source .....	47
11.6 System Timer (SysTick) .....	47
11.6.1 SysTick Control and Status Register (Address: 0xE000E010) .....	47
11.6.2 SysTick Reload Value Register (Address: 0xE000E014) .....	48

11.6.3 SysTick Current Value Register (Address:0xE000E018) .....	48
11.7 Slow Clock Source .....	48
11.7.1 Relate Register .....	48
11.7.2 Turn on External RTC XTAL .....	50
11.7.3 Turn on Internal RC OSC .....	50
12. I/O Ports .....	51
12.1 FEATURE .....	51
12.2 BLOCK DIAGRAM .....	51
12.3 REGISTER .....	51
12.3.1 Register List .....	51
12.3.2 Register Description .....	52
12.4 FUNCTION DESCRIPTION .....	55
12.4.1 Output Enable (OE) .....	55
12.4.2 WUN interrupt (WUN_INT) .....	55
12.4.3 GPIO0 interrupt (GPIO0_INT) .....	56
12.4.4 High Driving Current Output .....	56
12.4.5 Debug Interface and Flash Mask .....	56
13. Timer .....	57
13.1 FEATURE .....	57
13.2 PINS DESCRIPTION .....	57
13.3 BLOCK DIAGRAM .....	57
13.4 REGISTER .....	57
13.4.1 Base Address List .....	57
13.4.2 Register List .....	57
13.4.3 Register Description .....	58
13.5 FUNCTION DESCRIPTION .....	59
13.5.1 VALUE and RELOAD .....	59
13.5.2 Input-edge Counter .....	59
13.5.3 Time Capture .....	59
13.5.4 Timer Interrupt .....	60
14. Dual Timer .....	61
14.1 FEATURE .....	61
14.2 BLOCK DIAGRAM .....	61
14.3 REGISTER .....	61
14.3.1 Base Address List .....	61
14.3.2 Register List .....	61
14.3.3 Register Description .....	62
14.4 FUNCTION DESCRIPTION .....	63
14.4.1 Operation Mode .....	63
14.4.2 LOAD and BGLOAD .....	63
14.4.3 Interrupt .....	64
15. Sleep Timer .....	65
15.1 FEATURES .....	65
15.2 BLOCK DIAGRAM .....	65
15.3 REGISTER .....	65
15.3.1 Register list .....	65
15.3.2 Register Description .....	66
15.4 FUNCTION DESCRIPTION .....	68
15.5 PROCEDURE .....	68
15.5.1 Sleep Timer0 .....	68
15.5.2 Sleep Timer1 .....	68
16. Watchdog Timer .....	70
16.1 FEATURE .....	70
16.2 BLOCK DIAGRAM .....	70
16.3 REGISTER .....	70
16.3.1 Register List .....	70
16.3.2 Register Description .....	70
16.4 FUNCTION DESCRIPTION .....	72
16.4.1 Watchdog Interrupt .....	72
16.4.2 Watchdog Reset .....	72
16.5 PROCEDURE .....	72
16.5.1 Watchdog Enable .....	72

16.5.2 Watchdog Reset Enable .....	72
17. RTC (Real Time Counter) .....	73
17.1 FEATURE .....	73
17.2 BLOCK DIAGRAM .....	73
17.3 REGISTER .....	73
17.3.1 Register List .....	73
17.3.2 Register Description .....	74
17.4 FUNCTION DESCRIPTION .....	76
17.4.1 Write RTC Register .....	76
17.4.2 Bit RTCON .....	77
17.4.3 Bit RTCEN .....	77
17.4.4 Alarm .....	77
17.4.5 Manual Count .....	78
17.4.6 Interrupt .....	79
18. PWM .....	80
18.1 FEATURES .....	80
18.2 PINS DESCRIPTION .....	80
18.3 BLOCK DIAGRAM .....	80
18.4 REGISTER .....	80
18.4.1 Register list .....	80
18.4.2 Register Description .....	81
18.5 FUNCTION DESCRIPTION .....	82
18.6 PROCEDURE .....	82
19. UART .....	83
19.1 FEATURE .....	83
19.2 PINS DESCRIPTION .....	83
19.3 BLOCK DIAGRAM .....	83
19.4 REGISTER .....	84
19.4.1 Base Address List .....	84
19.4.2 Register List .....	84
19.4.3 Register Description .....	84
19.5 FUNCTION DESCRIPTION .....	86
19.5.1 I/O pin setting .....	86
19.5.2 UART Baud Rate .....	86
19.5.3 UART TX state and interrupt .....	86
19.5.4 UART RX state and interrupt .....	87
19.6 PROCEDURE .....	87
20. I <sup>2</sup> C Interface .....	88
20.1 FEATURE .....	88
20.2 PINS DESCRIPTION .....	88
20.3 BLOCK DIAGRAM .....	88
20.4 REGISTER .....	88
20.4.1 Register List .....	88
20.4.2 I <sup>2</sup> C Master Register Description .....	89
20.4.3 I <sup>2</sup> C Slave Register Description .....	91
20.5 FUNCTION DESCRIPTION .....	93
20.5.1 I <sup>2</sup> C Transmission Format .....	93
20.5.2 I <sup>2</sup> C Transmission Speed .....	94
20.5.3 I <sup>2</sup> C Master .....	96
20.5.4 I <sup>2</sup> C Slave .....	102
20.5.5 10 Bit Address .....	103
21. SPI Interface .....	104
21.1 FEATURES .....	104
21.2 PINS DESCRIPTION .....	104
21.3 BLOCK DIAGRAM .....	104
21.4 REGISTER .....	104
21.4.1 Register list .....	104
21.4.2 Register Description .....	105
21.5 FUNCTION DESCRIPTION .....	108
21.5.1 SPI Master and SPI Slave .....	108
21.5.2 SPI_SCK speed .....	108
21.5.3 SPI Transmission Format .....	109

21.5.4 SPI FIFO.....	110
21.5.5 SPI FIFO interrupt .....	111
21.5.6 SPI Master Transmit .....	111
21.5.7 SPI Slave Transmit .....	113
22. 8-bits ADC.....	115
22.1 FEATURES.....	115
22.2 BLOCK DIAGRAM.....	115
22.3 REGISTER .....	115
22.3.1 Register List.....	115
22.3.2 Register Description .....	115
22.4 FUNCTION DESCRIPTION.....	117
22.4.1 8-bits ADC function:.....	117
22.4.2 RSSI measurement .....	117
22.4.3 Carrier Detect .....	119
23. 12-bits SAR ADC .....	120
23.1 FEATURES.....	120
23.2 PINS DESCRIPTION.....	120
23.3 BLOCK DIAGRAM.....	120
23.4 REGISTER .....	121
23.4.1 Register List.....	121
23.4.2 Register Description .....	121
23.5 FUNCTION DESCRIPTION.....	123
23.5.1 A/D Reference Voltage (Vref):.....	123
23.5.2 ADC Value Calculation:.....	123
23.5.3 A/D Conversion Time:.....	124
23.5.4 Analog Input (Voltage) Measurement.....	124
23.5.5 Internal Temperature Measurement.....	124
24. Battery Detect.....	125
24.1 FEATURES.....	125
24.2 PINS DESCRIPTION.....	125
24.3 BLOCK DIAGRAM.....	125
24.4 REGISTER .....	125
24.4.1 Register list.....	125
24.4.2 Register Description .....	125
24.5 FUNCTION DESCRIPTION.....	126
24.6 PROCEDURE.....	126
25. Power Management.....	127
25.1 BLOCK DIAGRAM.....	127
25.2 REGISTER .....	128
25.2.1 Register List.....	128
25.2.2 Register Description .....	128
25.3 FUNCTION DESCRIPTION.....	128
25.3.1 Normal Mode .....	128
25.3.2 PM Mode .....	129
26. MPULCD.....	130
26.1 FEATURE.....	130
26.2 PINS DESCRIPTION.....	130
26.3 BLOCK DIAGRAM.....	131
26.4 REGISTER .....	131
26.4.1 Register List.....	131
26.4.2 Register Description .....	131
26.5 FUNCTION DESCRIPTION.....	133
26.5.1 MPULCD DMA data Format .....	133
26.5.2 MPULCD output interface.....	133
26.6 PROCEDURE.....	134
26.6.1 MPULCD initial .....	134
26.6.2 MPULCD write data with command mode.....	134
26.6.3 MPULCD read data with command mode .....	134
26.6.4 MPULCD write data with DMA mode.....	135
27. Flash and IAP .....	136
27.1 REGISTER .....	136
27.1.1 Register List.....	136

27.1.2 Register Description .....	136
27.2 FUNCTION DESCRIPTION .....	138
27.2.1 Page Write .....	138
27.2.2 Page Erase .....	138
27.2.3 Sector Erase .....	138
27.3 PROCEDURE .....	138
28. Encryption and Authentication .....	140
28.1 AES .....	140
28.1.1 AddRoundKey .....	140
28.1.2 SubBytes .....	140
28.1.3 ShiftRows .....	140
28.1.4 MixColumns .....	140
28.1.5 CCM .....	141
29. Application circuit .....	142
30. Abbreviations .....	144
31. Ordering Information .....	145
32. Package Information .....	146
33. Top Marking Information .....	148
34. Reflow Profile .....	150
35. Tape Reel Information .....	151
36. Product Status .....	152

AMICCOM CONFIDENTIAL

### **1. General Description**

A3107M0 is a high performance and low cost 2.4GHz FSK/GFSK system-on-chip (SoC) wireless transceiver. With on chip fraction-N synthesizer, it is designed for Bluetooth Low Energy (Bluetooth 5.0 Single mode, master or slave role).

A3107M0 has various powerful functions and excellent performance of a leading 2.4GHz FSK/GFSK RF transceiver. A3107M0 has various operating MODEs, making it highly suited for systems where ultra-low power consumption is required.

A3107M0 supports AES128 engine and CCM. For low current consumption, A3107M0 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.

### **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

#### RF

- Frequency band: 2400 – 2483MHz.
- FSK and GFSK modulation
- High sensitivity:
  - ◆ -97dBm at 500Kbps data rate
  - ◆ -94dBm at 1Mbps data rate
  - ◆ -91dBm at 2Mbps data rate
- Programmable data rate 5K ~ 2Mbps
- Fast settling time synthesizer for frequency hopping system
- 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 to 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

#### Low Power

- Wide Range Operation Voltage from 2.0V ~ 3.6V
- 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 retention (2.1uA)

#### Microcontroller

- High performance ARM-M0 MCU
- 256KB Flash memory with copy protection, 32KB SRAM

#### Peripherals

- LCD controller with DMA and 8080 output interface(I80)
- UART, I<sup>2</sup>C, SPI serial communication
- Operation clock: 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64 of crystal oscillator
- 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
- Built-in thermal sensor for monitoring relative temperature
- Built-in eight channels 12-bits ADC for general purpose analog input (0V ~ 1.8 V)
- Built-in Low Battery Detector

#### Layout

- Package size (QFN5X5, 40 pins/ QFN6X6 48 pins)



### 4. Pin Configurations

#### 4.1 QFN40 5x5

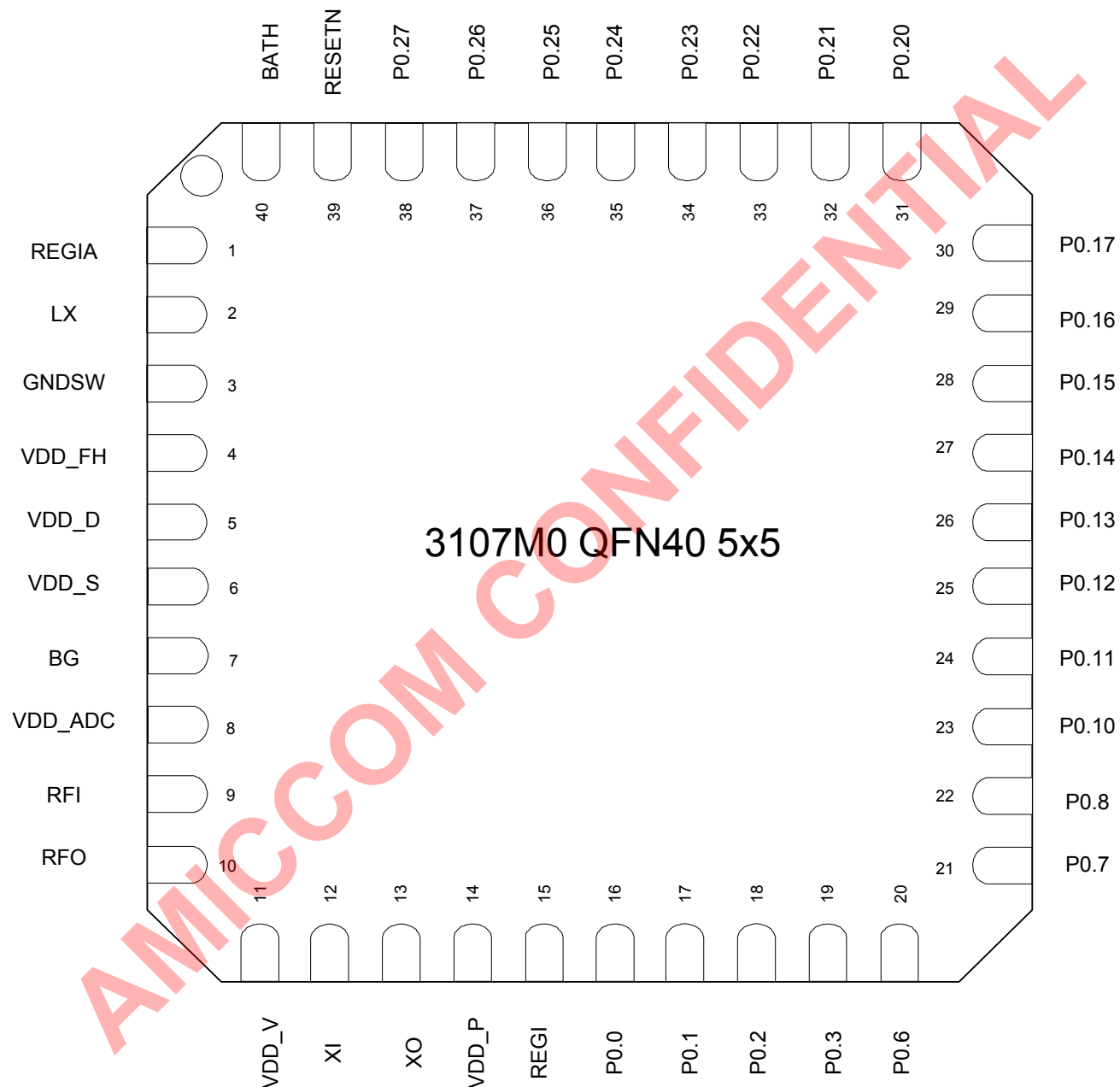


Figure 4.1 A3107M0 QFN40 5x5 Package Top View

### 4.2 QFN48 6x6

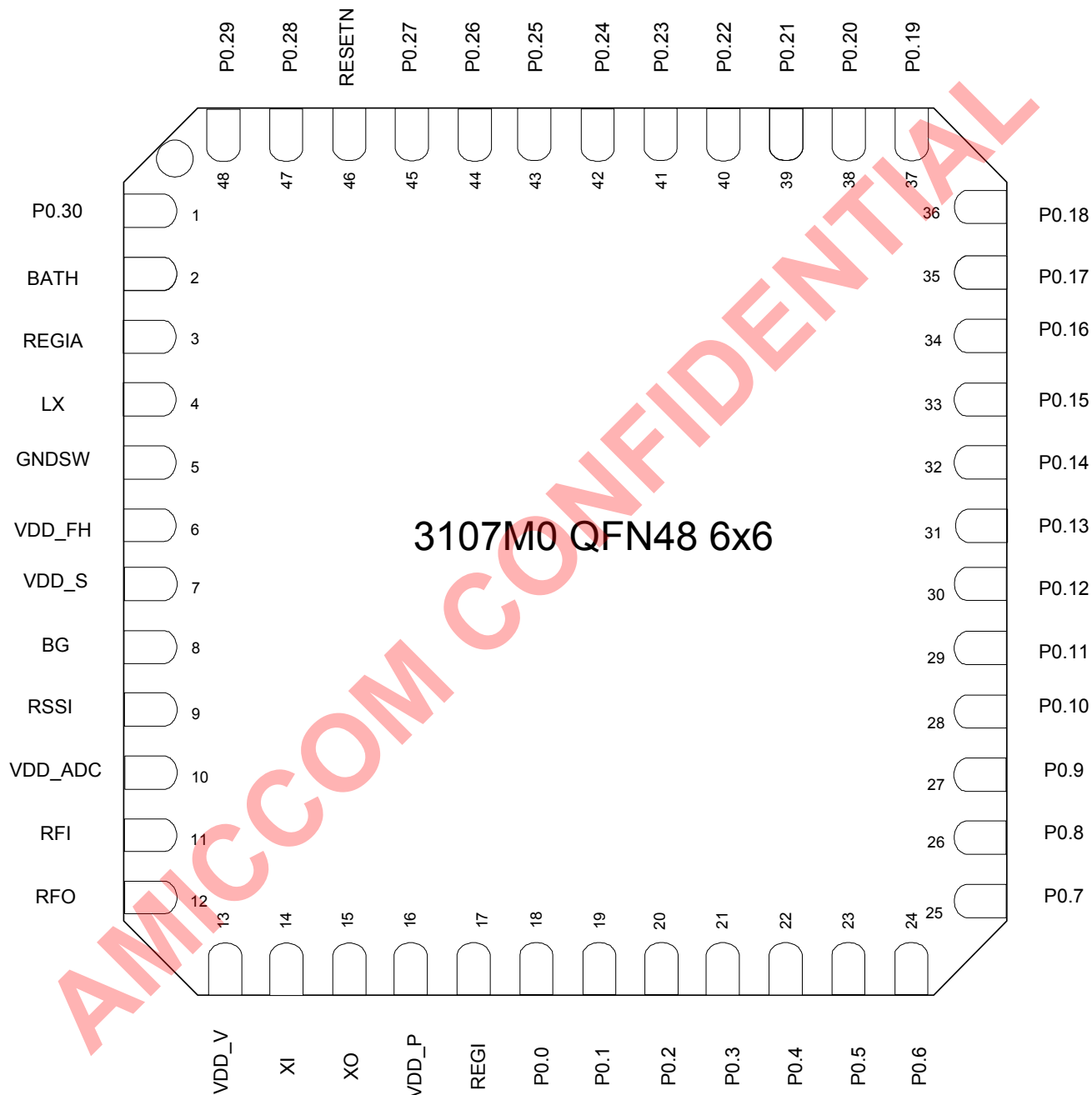


Figure 4.2 A3107M0 QFN48 6x6 Package Top View

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

#### 5.1 A3107M0 QFN40 5x5 Package

Pin No.	Symbol	I/O	Function Description
1	REGIA	AO	DC-DC regulator output for buck
2	LX	AIO	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	SPI_CS
17	P0.1	DIO	SPI_MISO
18	P0.2	DIO	SPI_MOSI
19	P0.3	DIO	SPI_SCK
20	P0.6	DIO	SWDIO
21	P0.7	DIO	SWCLK
22	P0.8	DIO/AI	TIMER0_EIN / ADC2 / BB_GIO1
23	P0.10	DIO	PWM2 / I <sup>2</sup> C_SCL / FLASH_MASK
24	P0.11	DIO	PWM3 / I <sup>2</sup> C_SDA / LCD_TE
25	P0.12	DIO/AI	ADC4 / ICE_MODE / LCD_RDY
26	P0.13	DIO/AI	ADC5/BB_GIO1 / LCD_D[2]
27	P0.14	DIO/AI	ADC6/BB_GIO2 / LCD_D[1]
28	P0.15	DIO/AI	ADC7/BB_CK0 / LCD_D[0] / s_LCD_SDA
29	P0.16	DIO	UART0_RX / LCD_CSX
30	P0.17	DIO	UART0_TX / LCD_D/CX
31	P0.20	DIO	UART2_RX / PWM0 / LCD_WRX/ s_LCD_SCL
32	P0.21	DIO	UART2_TX / PWM1 / LCD_D[7]
33	P0.22	DIO/AIO	RTCI
34	P0.23	DIO/AIO	RTCO
35	P0.24	DIO	General IO / LCD_D[6]
36	P0.25	DIO	General IO / LCD_D[5]

37	P0.26	DIO	General IO / LCD_D[4]
38	P0.27	DIO	General IO / LCD_D[3]
39	RESETN	AI	RESETN input
40	BATH	DIO/AIO	DC-DC converter voltage input for buck mode

AMICCOM CONFIDENTIAL

### 5.2 A3107M0 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 for buck
4	LX	AIO	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	SPI_CS
19	P0.1	DIO	SPI_MISO
20	P0.2	DIO	SPI_MOSI
21	P0.3	DIO	SPI_SCK
22	P0.4	DIO	I <sup>2</sup> C_SCL
23	P0.5	DIO	I <sup>2</sup> C_SDA
24	P0.6	DIO	SWDIO
25	P0.7	DIO	SWCLK
26	P0.8	DIO/AI	TIMER0_EIN / ADC2 / BB_GIO1
27	P0.9	DIO/AI	TIMER1_EIN / ADC3 / BB_GIO2
28	P0.10	DIO	PWM2 / I <sup>2</sup> C_SCL / FLASH_MASK
29	P0.11	DIO	PWM3 / I <sup>2</sup> C_SDA / LCD_TE
30	P0.12	DIO/AI	ADC4 / ICE_MODE / LCD_RDx
31	P0.13	DIO/AI	ADC5/BB_GIO1 / LCD_D[2]
32	P0.14	DIO/AI	ADC6/BB_GIO2 / LCD_D[1]
33	P0.15	DIO/AI	ADC7/BB_CKO / LCD_D[0] / s_LCD_SDA
34	P0.16	DIO	UART0_RX / LCD_CSX
35	P0.17	DIO	UART0_TX / LCD_D/CX
36	P0.18	DIO/AI	UART1_RX / ADC0
37	P0.19	DIO/AI	UART1_TX / ADC1
38	P0.20	DIO	UART2_RX / PWM0 / LCD_WRX/ s_LCD_SCL

39	P0.21	DIO	UART2_TX / PWM1 / LCD_D[7]
40	P0.22	DIO/AIO	RTCI
41	P0.23	DIO/AIO	RTCO
42	P0.24	DIO	General IO / LCD_D[6]
43	P0.25	DIO	General IO / LCD_D[5]
44	P0.26	DIO	General IO / LCD_D[4]
45	P0.27	DIO	General IO / LCD_D[3]
46	RESETN	AI	RESETN input
47	P0.28	DIO	General IO
48	P0.29	DIO	General IO

AMICCOM CONFIDENTIAL

### 6. Chip Block Diagram

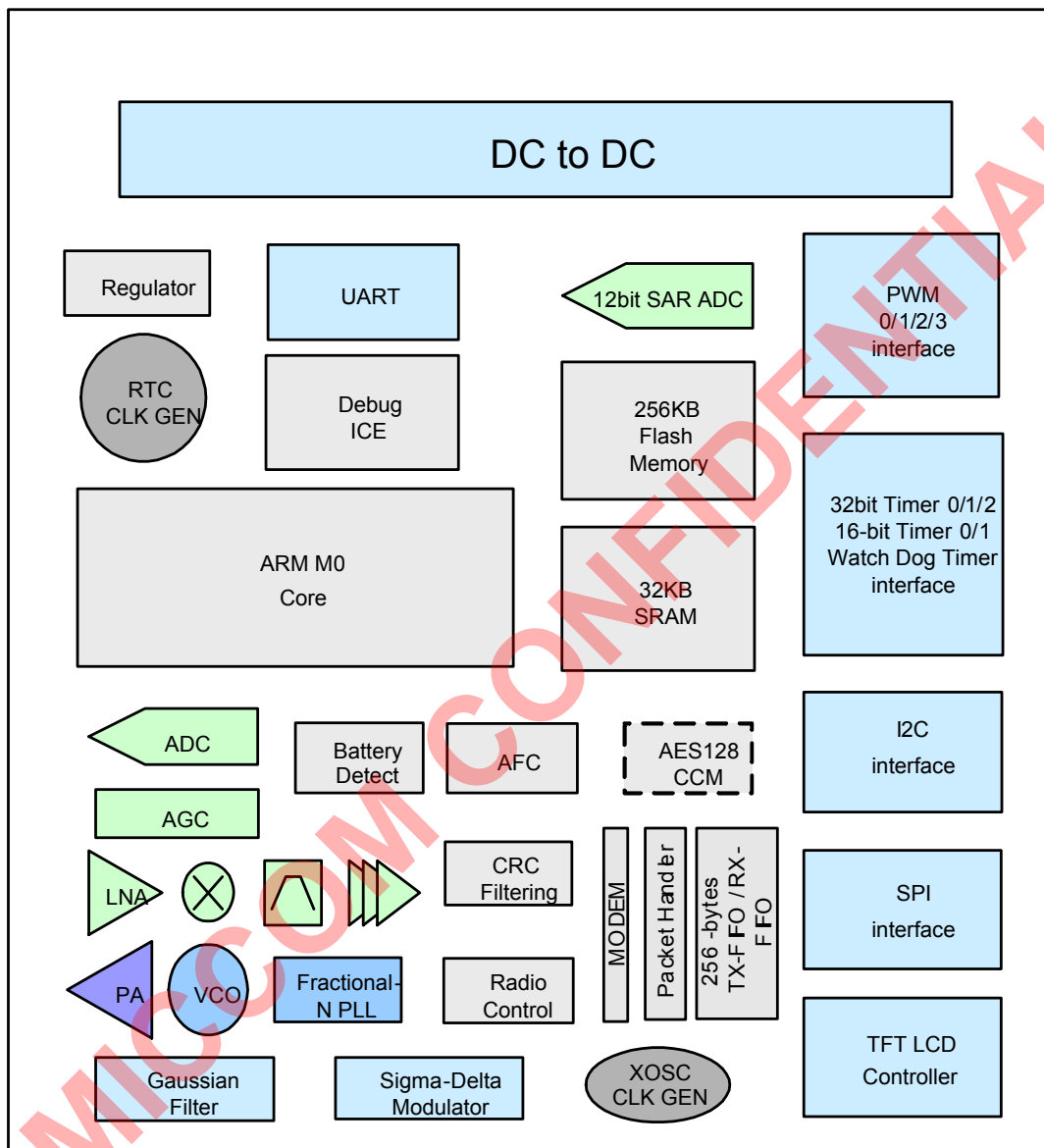


Figure 6.1 A3107M0 Block Diagram

### 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.

\*Pin RFO pass HBM +/- 1.75KV

\*Device is Moisture Sensitivity Level III (MSL 3).





### 8. Electrical Specification

(Ta=25°C, BATH = 3.3V, unless otherwise noted)

Parameter	Description	Min.	Typ.	Max.	Unit
<b>General</b>					
Operating Temperature		-40		85	°C
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
<b>Synthesizer block</b>					
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
<b>Transmitter</b>					
Carrier Frequency		2400		2483.5	MHz
Maximum Output Power			4		dBm
RF Power Control Range			20		dB
Out Band Spurious Emission <sup>1</sup>	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		$\mu$ s
<b>Receiver</b>					
Receiver sensitivity @ BER = 0.1%	Data rate 2M ( $F_{IF} = 2\text{MHz}$ )		-91		dBm
	Data rate 1M ( $F_{IF} = 1\text{MHz}$ )		-94		dBm
	Data rate 500K ( $F_{IF} = 1\text{MHz}$ )		-97		dBm
IF frequency bandwidth			1200/2400		KHz
IF center frequency			1000/2000		KHz
Interference	Co-Channel ( $C/I_0$ )		11		dB
	1 <sup>st</sup> Adjacent Channel ( $C/I_1$ )		2		dB
	2 <sup>nd</sup> Adjacent Channel ( $C/I_2$ )		-18		dB
	3 <sup>rd</sup> 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		$\mu$ s
<b>12Bit SAR ADC</b>					
Input voltage range		0		1.8	V
Reference voltage			1.8		V
Input capacitor			25		pF
Sampling frequency <sup>1</sup>		15.625		125	KHz
EOB, effective number of bits			10		bit
INL			+/- 2		LSB
DNL			+/-1		LSB
Current consumption			0.4		mA
ADC accuracy	With BG trimming		+/-1		%
<b>Regulator</b>					
Regulator settling time			200		$\mu$ s
Band-gap reference voltage			1.21		V
Regulator output voltage			1.21		V
<b>Digital IO DC characteristics</b>					
High Level Input Voltage ( $V_{IH}$ )		0.8*VDD		VDD	V
Low Level Input Voltage ( $V_{IL}$ )		0		0.2*VDD	V
High Level Output Voltage ( $V_{OH}$ )	@ $I_{OH} = -0.5\text{mA}$	VDD-0.4		VDD	V
Source current	@ $V_{OH} = 2.4\text{V}$ , HDV=0		10		mA
Source current	@ $V_{OH} = 2.4\text{V}$ , HDV=1		30		mA
Low Level Output Voltage ( $V_{OL}$ )	@ $I_{OL} = 0.5\text{mA}$	0		0.4	V
Sink current	@ $V_{OL} = 0.4\text{V}$ , HDV=0		5		mA
Sink current	@ $V_{OL} = 0.4\text{V}$ , HDV=1		14.5		mA
<b>DC-DC Buck converter</b>					
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

Note 1 : Crystal frequency is set 16MHz.

### Power on Reset Electrical Characteristics

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		REGI	Condition				
$V_{POR}$	REGI Start Voltage to ensure Power-on Reset	—	—	—	—	100	mV
$R_{POR\ AC}$	REGI Raising Rate to ensure Power-on Reset	—	—	0.04	—	—	V/ms
$t_{POR}$	Minimum Time for REGI to remain at $V_{POR}$ to ensure Power-on Reset	—	—	3	—	—	ms

AMICCOM CONFIDENTIAL

### 9. Register List

A3107M0 contains Peripheral Register and RF Register.

#### 9.1 Battery detect Register (Address: 0x50000000)

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

**BVT [2:0]:** Battery voltage detect threshold.

[000]: 1.875V.

[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.

**BDF:** Battery detection flag.

[0]: Battery voltage less than threshold.

[1]: Battery voltage greater than threshold.

#### 9.2 Power control Register 1 (Address: 0x50000008)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	WUNIE[3]	WUNIE[2]	WUNIE[1]	WUNIE[0]	PM3FLAG	LVDF	RESETNF	PORF
R					PM3FLAG	LVDF	RESETNF	PORF
Reset	0	0	0	0	0	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W	RTCPUN	BATS	REFS	SWR	RGC[2]	RGC[1]	RGC[0]	ECHC
R								
Reset	0	0	0	0	1	0	1	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	CBG[2]	CBG[1]	CBG[0]	PDNS	STA	ENDL[2]	ENDL[1]	ENDL[0]
R								
Reset	1	1	0	1	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

W	QDS	REGAE	CE_3us_ON	MCU_3us	PACTL	RGS	RGV[1]	RGV[0]
R	QDS	REGAE		MCU_3us			RGV[1]	RGV[0]
Reset	0	0	0	0	0	0	0	0

**WUNIE [3:0]:** Reserved for internal usage.

**PM3FLAG (Power MODE 3 flag) :** Reserved for internal usage

Write any data to bit[26:24] to clear all bits.

**BATS:** Reserved for internal usage. Shall be set to [0].

**REFS:** Reserved for internal usage. Shall be set to [0].

**LVDF** (Low voltage detect) flag  
 = 1: Occurred Low Voltage Reset  
 = 0: No Low Voltage reset

**RESETNF** (RESETN flag)  
 = 1: Occurred RESETN reset  
 = 0: No RESETN reset

**PORF** (power-on reset flag)  
 = 1: Occurred Power-on Reset  
 = 0: No Power-on Reset

**RTCPUN:** Reserved for internal usage. Shall be set to [0].

**SWR:** Reserved for internal usage.

**RGC[2:0]:** Low power band-gap current select. Recommend RGC = [01]

**ECHC:** Reserved for internal usage.

**CBG[2:0]:** Reserved for internal usage.

**PDNS:** Power manager to turn on REGOD Recommend PDNS = [0]

**STA:** Reserved for internal usage only. Shall be set to [0].

**ENDL[2:0]:** Reserved for internal usage only

**QDS (Quick discharge) :** Reserved for internal usage

**REGAE (RegA Enable) :** Reserved for internal usage

**CE\_3us\_ON:** Normal TRX Wakeup MCU in 3us.

**MCU\_3us:** Wakeup MCU in 3us enable.

**PACTL:** Reserved for internal usage only. Shall be set to [0].

**RGS:** Reserved for internal usage.

**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.

### 9.3 Power control Register 2 (Address: 0x5000000C)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W								
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W								
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	SWE	SWD	SWC	CADS	SVREFF	CBGF2	CBGF1	CBGF0
R								
Reset	0	0	0	0	0	1	1	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	EBOD	ENAV	QDSA	ENDV	QDSD	CEL	SVREF	CELA
R								
Reset	1	1	0	1	0	0	0	0

**SWE:** Reserved for internal usage

**SWD:** Reserved for internal usage

**SWC:** Reserved for internal usage

**CADS:** Reserved for internal usage

**SVREFF:** Reserved for internal usage

**CBGF[2:0]:** Reserved for internal usage.

**EBOD:** Reserved for internal usage.

**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].

**SVREF:** Reserved for internal usage. Recommend SVREF = [0].

**CELA:** Reserved for internal usage.

**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

### 9.4 Reset Register (Address: 0x50001000)

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

Reset	0	0	0	0	0	0	0	0
<b>R/W</b>	<b>Bit 23</b>	<b>Bit 22</b>	<b>Bit 21</b>	<b>Bit 20</b>	<b>Bit 19</b>	<b>Bit 18</b>	<b>Bit 17</b>	<b>Bit 16</b>
W								
R								
Reset	0	0	0	0	0	0	0	0
<b>R/W</b>	<b>Bit 15</b>	<b>Bit 14</b>	<b>Bit 13</b>	<b>Bit 12</b>	<b>Bit 11</b>	<b>Bit 10</b>	<b>Bit 9</b>	<b>Bit 8</b>
W								
R								
Reset	0	0	0	0	0	0	0	0
<b>R/W</b>	<b>Bit 7</b>	<b>Bit 6</b>	<b>Bit 5</b>	<b>Bit 4</b>	<b>Bit 3</b>	<b>Bit 2</b>	<b>Bit 1</b>	<b>Bit 0</b>
W	RESETN	FWPRN	FRPRN	FDATARN	BFCRN			
R								
Reset	0	0	0	0	0	0	0	0

**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)

**FDATARN:** FIFO Data Reset. (Write “1” to reset)

**BFCRN:** IF Filter Bank Calibration Software Reset. (Write “1” to reset)

### 9.5 Strobe Register (Address: 0x50001004)

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

**STRB[7:0]:** Strobe command register.

[0x80]: Sleep MODE.

[0x90]: Idle MODE.

[0xA0]: Standby MODE.

[0xB0]: PLL MODE.

[0xC0]: RX MODE.

[0xD0]: TX MODE.

Reserve for other settings.

### 9.6 Status Register (Address: 0x50001010)

<b>R/W</b>	<b>Bit 31</b>	<b>Bit 30</b>	<b>Bit 29</b>	<b>Bit 28</b>	<b>Bit 27</b>	<b>Bit 26</b>	<b>Bit 25</b>	<b>Bit 24</b>
W								

R								
Reset	0	0	0	0	0	0	0	0
<b>R/W</b>	<b>Bit 23</b>	<b>Bit 22</b>	<b>Bit 21</b>	<b>Bit 20</b>	<b>Bit 19</b>	<b>Bit 18</b>	<b>Bit 17</b>	<b>Bit 16</b>
W								
R								
Reset	0	0	0	0	0	0	0	0
<b>R/W</b>	<b>Bit 15</b>	<b>Bit 14</b>	<b>Bit 13</b>	<b>Bit 12</b>	<b>Bit 11</b>	<b>Bit 10</b>	<b>Bit 9</b>	<b>Bit 8</b>
W				EOPF				
R				EOPF	FSYNC	FPF	ENCF	SYNC
Reset	0	0	0	0	0	0	0	0
<b>R/W</b>	<b>Bit 7</b>	<b>Bit 6</b>	<b>Bit 5</b>	<b>Bit 4</b>	<b>Bit 3</b>	<b>Bit 2</b>	<b>Bit 1</b>	<b>Bit 0</b>
W								
R	WTR	FECF	CRCF	CER	XER	PLLER	TRSR	TRER
Reset	0	0	0	0	0	0	0	0

**EOPF:** EOP status flag.

**FSYNC:** FSYNC status flag.

**FPF:** FPF status flag.

**SYNC:** SYNC status flag.

**WTR:** WTR status flag.

**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.

### 9.7 FIFO Register (Address: 0x50001020)

<b>R/W</b>	<b>Bit 31</b>	<b>Bit 30</b>	<b>Bit 29</b>	<b>Bit 28</b>	<b>Bit 27</b>	<b>Bit 26</b>	<b>Bit 25</b>	<b>Bit 24</b>
W	FPM1	FPM0						
R	FPM1	FPM0						



Reset	0	1	0	0	0	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W			PSA5	PSA4	PSA3	PSA2	PSA1	PSA0
R			PSA5	PSA4	PSA3	PSA2	PSA1	PSA0
Reset	0	0	0	0	0	0	0	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W								
R								
Reset	0	0	0	0	0	0	0	0
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
Reset	0	0	1	1	1	1	1	1

**FPM [1:0]:** FIFO Pointer Margin

**PSA [5:0]:** Used for Segment FIFO.

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

### 9.8 RSSI Register (Address: 0x50001244)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W								
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W								
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W								
R								
Reset	0	0	0	0	0	0	0	0
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
Reset	0	0	0	0	0	0	0	0

**RTH [7:0]:** Carrier detect threshold.

**ADC [7:0]:** ADC output value of temperature, RSSI or external voltage measurement.

ADC input voltage= 0.9 \* ADC [7:0] / 256 V.

### 9.9 ADC Control Register I (Address: 0x50008000)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	ADC12RN							
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W								ADC1_Clear
R								ADC1_STATE
Reset	0	0	0	0	0	0	0	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8

W	ADCIE	VADS[2]	VADS[1]	VADS[0]	ADIVL	ADCYC	ENADC	DTMP
R								
Reset	0	0	0	0	0	0	0	0
<b>R/W</b>	<b>Bit 7</b>	<b>Bit 6</b>	<b>Bit 5</b>	<b>Bit 4</b>	<b>Bit 3</b>	<b>Bit 2</b>	<b>Bit 1</b>	<b>Bit 0</b>
W	BUFS	CKS1	CKS0	MODE	MVS2	MVS1	MVS0	ADCE
R	--	--	--	MODE	MVS2	MVS1	MVS0	ADCE
Reset	0	1	0	0	0	0	0	0

**ADC12RN: 12-bits ADC Reset. (Write “1” to reset)**

[0]: Disable.

[1]: Enable

**ADCIE** : 12-bits interrupt enable.

[0]: 0.

[1]: 2048

**VADS[2:0]**: Reserved for internal usage only. Shall be set to [0].

[0]: Disable.

[1]: Enable

**ADIVL**: 12-bits sargen initial value select.

[0]: 2048.

[1]: 0

**ADCYC**: 12-bits sargen clock counter select.

[0]: 31.

[1]: 32.

**ENADC**: Enable ADC.

[0]: Disable.

[1]: Enable

**DTMP**: 12-bits temperature select. Refer to chapter 23 for details.

[0]: Disable.

[1]: Enable

**BUFS**: input buffer select for 12 bit ADC. Refer to chapter 23 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.

[0]: Disable.  
[1]: Enable

### 9.10 ADC Control Register II (Address: 0x50008004)

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

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

**ADC [11:0]:** ADC output value

### 9.11 ADC Channel Register (Address: 0x50008008)

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

[0]: Disable.  
[1]: Enable.

### 9.12 MODEC Register (Address: 0x50001008)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W								
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W								
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W						BREV	FIFOREV	IDREV
R						BREV	FIFOREV	IDREV
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W			AIF	DFCD		FMT	FMS	
R			AIF	CD		FMT	FMS	
Reset	0	0	0	0	0	0	0	0

**BREV:** Bit reverse enable.

[0]: Disable.  
[1]: Enable

**FIFOREV:** FIFO reverse enable.

[0]: Disable.  
[1]: Enable

**IDREV:** ID reverse enable.

[0]: Disable.  
[1]: Enable

**AIF (Auto IF Offset):** RF LO frequency will auto offset one IF frequency while entering RX MODE.

[0]: Disable.  
[1]: Enable.

**CD / DFCD:**

**DFCD (Data Filter by CD):** The received package will be filtered out if Carrier Detector signal is inactive.

[0]: Disable.  
[1]: Enable.

**CD (Read):** Carrier detector signal.

[0]: Input power below threshold.  
[1]: Input power above threshold.

**FMT:** Reserved for internal usage only. Shall be set to [0].

**FMS:** Direct/FIFO MODE select.

[0]: Direct MODE.  
[1]: FIFO MODE.

### 9.13 RF GPIO Register (Address: 0x50001028)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W								GIOBSEL
R								

Reset	0	0	0	0	0	0	0	0
<b>R/W</b>	<b>Bit 23</b>	<b>Bit 22</b>	<b>Bit 21</b>	<b>Bit 20</b>	<b>Bit 19</b>	<b>Bit 18</b>	<b>Bit 17</b>	<b>Bit 16</b>
W	CKOE	CKOI	ECKOE		CKOS3	CKOS2	CKOS1	CKOS0
R								
Reset	0	0	1	0	0	1	1	1
<b>R/W</b>	<b>Bit 15</b>	<b>Bit 14</b>	<b>Bit 13</b>	<b>Bit 12</b>	<b>Bit 11</b>	<b>Bit 10</b>	<b>Bit 9</b>	<b>Bit 8</b>
W	GIO2E	GIO2I			GIO2S3	GIO2S2	GIO2S1	GIO2S0
R								
Reset	0	0	0	0	0	1	0	0
<b>R/W</b>	<b>Bit 7</b>	<b>Bit 6</b>	<b>Bit 5</b>	<b>Bit 4</b>	<b>Bit 3</b>	<b>Bit 2</b>	<b>Bit 1</b>	<b>Bit 0</b>
W	GIO1E	GIO1I			GIO1S3	GIO1S2	GIO1S1	GIO1S0
R								
Reset	0	0	0	0	0	0	0	0

**GIOBSEL:: RF GIO pin output select.**

[0]: Disable.

[1]: Enable.

**CKOE: Clock Output Enable.**

[0]: Disable.

[1]: Enable.

**CKOI: CKO pin output signal invert.**

[0]: Non-inverted output.

[1]: Inverted output.

**ECKOE: External Clock Output Enable for CKOS [3:0]= [0100] ~ [0111].**

[0]: Disable.

[1]: Enable.

**CKOS[3:0]: CKO pin output select.**

[0000]: DCK (TX data clock).

[0001]: RCK (RX recovery clock).

[0010]: FPF (FIFO pointer flag).

[0011]: EOP, EOVCB, EOFBC, EOADC, EOVCB, OKADC, RSSC\_OK (Internal usage only).

[0100]: External clock output =  $F_{SYCK}$ .

[0101]: External clock output / 2 =  $F_{SYCK} / 2$ .

[0110]: External clock output / 4 =  $F_{SYCK} / 4$ .

[0111]: External clock output / 8 =  $F_{SYCK} / 8$ .

[1000]: WCK (4KHz)

[1001]: PF8M (8MHz)

[1010]: TMRCK (32KHz)

[1011]: SYCK (8Mhz)

[1100]: TMRCK\_OVF (Timer clock)

[1101]: CKOSI, Reserved.

[1110]: SWNL, Reserved.

[1111]: DSTL, Reserved.

**GIO2I: GIO2 pin output signal invert.**

[0]: Non-inverted output.

[1]: Inverted output.

**GIO2S [3:0]: GIO2 pin function select.**

GIO2S	TX state	RX state
[0000]	ARCWTR (Wait until TX or RX finished)	
[0001]	EOAC (end of access code)	FSYNC (frame SYNC)
[0010]	TMEO or TMDEO (TX modulation enable)	CD (carrier detect)
[0011]	Preamble Detect Output (PMDO)	
[0100]	MCU wakeup signal (TWWS)	
[0101]	Quadrature phase demodulator input (DMIQ)	

[0110]	Reserved
[0111]	TRXD In/Out (Direct MODE)
[1000]	RXD (Direct MODE)
[1001]	TXD (Direct MODE)
[1010]	Quadrature phase demodulator external input (EXDI1)
[1011]	External FSYNC input in RX direct MODE
[1100]	DEC
[1101]	PDN_TX
[1110]	IPNL
[1111]	End flag of CCM decode.

**GIO1E:** GIO1 pin output enable.

[0]: Enable

[1]: Disable.

**GIO1I:** GIO1 pin output signal invert.

[0]: Non-inverted output.

[1]: Inverted output.

**GIO1S [3:0]:** GIO1 pin function select.

GIO1S [3:0]	TX state	RX state
[0000]	WTR (Wait until TX or RX finished)	
[0001]	EOAC (end of access code)	FSYNC (frame SYNC)
[0010]	TMEO or TMDEO (TX modulation enable)	CD (carrier detect)
[0011]	Preamble Detect Output (PMDO)	
[0100]	MCU wakeup signal (TWWS)	
[0101]	In phase demodulator input (DMII)	
[0110]	Reserved	
[0111]	TRXD In/Out (Direct MODE)	
[1000]	RXD (Direct MODE)	
[1001]	TXD (Direct MODE)	
[1010]	In phase demodulator external input (EXDI0)	
[1011]	External FSYNC input in RX direct MODE	
[1100]	INC	
[1101]	PDN_RX	
[1110]	INT5	
[1111]	CCM fail flag	

### 9.14 LO channel Register (Address: 0x500010C0)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W								
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W								
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W								
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	CHN7	CHN6	CHN5	CHN4	CHN3	CHN2	CHN1	CHN0
R	CHN7	CHN6	CHN5	CHN4	CHN3	CHN2	CHN1	CHN0
Reset	0	0	0	0	0	0	0	0

CHN [7:0]: LO channel number select.

### 9.15 TX Control Register (Address: 0x50001100)

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

**GDR:** Gaussian Filter Over-sampling Rate Select.

[0]: BT= 1.

[1]: BT= 0.5

**GDR=0.**

FPS[2:0]	7	6	5	4	3	2	1	0
BT	1.4	1.3	1.2	1.1	0.75	0.7	0.65	0.6

**GDR=1.**

FPS[2:0]	7	6	5	4	3	2	1	0
BT	0.7	0.65	0.6	0.55	0.7	0.65	0.6	0.55

**TMDE:** TX Modulation Enable for VCO Modulation.

[0]: Disable.

[1]: Enable.

**TXDI:** TX data invert. Recommend TXDI = [0].

[0]: Non-invert.

[1]: Invert.

**TME:** TX modulation enable.

[0]: Disable.

[1]: Enable.

**FS: Filter select.**

The filter shape is Gaussian filter

[0]: disable.

[1]: enable.

### 9.16 TX Power 1 Register (Address: 0x50001104)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W								
R								
Reset	0	0	0	0	0	0	0	0

R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W								
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W								
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W		PWORS	TXCS	PAC1	PAC0	TBG2	TBG1	TBG0
R								
Reset	0	0	0	1	0	1	1	1

**PWORS:** TX high power setting.

**[0]:** Disable.

**[1]:** Enable.

**TXCS:** TX Current Setting.

**PAC [1:0]:** PA Current Setting.

**TBG [2:0]:** TX Buffer Setting.

### 9.17 TX Modulation Register (Address: 0x5000110C)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	DAMV1	DAMV0	DEVFD2	DEVFD1	DEVFD0	DEV2	DEV1	DEV0
R								
Reset	1	0	1	0	1	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W			DEVGD2	DEVGD1	DEVGD0	FPS2	FPS1	FPS0
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W						FDP2	FDP1	FDP0
R								
Reset	0	0	0	0	0	1	1	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
R								
Reset	0	0	1	0	1	1	1	1

**DAMV [1:0]:** Demodulator D/A Voltage Range Select. Recommend DAMV = [10].

**[00]:** 1/32\*1.2.

**[01]:** 1/16\*1.2.

**[10]:** 1/8\*1.2.

**[11]:** 1/4\*1.2.

**DEVFD [2:0]:** VCO Modulation Data Delay by 8x over-sampling Clock. Recommend DEVFD = [101].

**DEV2 [2:0]:** VCO Modulation Data Delay by XCPCK Clock. Recommend DEV2 = [000].

**DEVGD [2:0]:** Sigma Delta modulator data delay setting.

**FPS [2:0]:** Gaussian filter BT setting.

**FDP [2:0]:** Frequency deviation power setting. Refer to TX control register (Address: 0x50001100h).



**FD [7:0]:** Frequency deviation setting.

$$F_{DEV} = F_{PFD} / 2^{16} \cdot FD \cdot 2^{(FDP-1)}$$

Where  $F_{PFD} = F_{XTAL} \cdot (DBL+1) / (RRC [1:0]+1)$ , PLL comparison frequency.

### 9.18 CODE1 Register (Address: 0x50001180)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W								IDL
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W	ETH2	ETH1	ETH0	PMD1	PMD0	PML2	PML1	PML0
R								
Reset	1	0	0	1	1	1	0	1
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W								
R								
Reset	1	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W			CRCSW	MSCRC	MCS	WHTS	FECS	CRCS
R								
Reset	0	0	0	0	0	0	0	0

**IDL:** ID code length select. Recommend IDL= [1].

[0]: 2 bytes.

[1]: 4 bytes.

**ETH [2:0]:** ID code error tolerance. Recommend ETH = [01].

[000]~[111]: 0~7 bit.

**PMD [1:0]:** Preamble pattern detection length. Recommend PMD = [10].

[00]: 0bit.

[01]: 4bits.

[10]: 8bits.

[11]: 16bits.

**PML [2:0]:** Preamble length select. Recommend PML= [11].

[000]: 1 byte.

[001]: 2 bytes.

[010]: 3 bytes.

[011]: 4 bytes.

[100]: 5bytes.

[101]: 6bytes.

[110]: 7bytes.

[111]: 8bytes

**CRCSW:** CRC 24-bits enable.

[0]: Disable.

[1]: Enable.

**MSCRC:** Mask CRC (CRC Data Filtering Enable).

[0]: Disable.

[1]: Enable.

**MCS:** Manchester Code Enable.

[0]: Disable.

[1]: Enable.

**WHTS:** Data whitening (Data Encryption) select.

[0]: Disable.  
[1]: Enable.

**FECS:** FEC select.  
[0]: Disable.  
[1]: Enable.

**CRCS:** CRC select.  
[0]: Disable.  
[1]: Enable.

AMICCOM CONFIDENTIAL

### 10. A3107M0 RF

A3107M0 integrate 2.4 GHz GFSK transceiver and Strobe control register (50001004h) into the SoC 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.

#### 10.1 Strobe Command Register 1 (Address:0x50001004)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	STRB[7:0]							
R	--							
Reset	1	0	1	0	0	0	0	0

Use strobe command control RF state.

**STRB:** Strobe command register.

**0x80:** Sleep MODE.

**0x90:** Idle MODE.

**0xA0:** Standby MODE.

**0xB0:** PLL MODE.

**0xC0:** RX MODE.

**0xD0:** TX MODE.

#### Status Register (Address:0x50001010)

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

In A3107M0, 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.

**PLLER:** PLL enabled status.

**[0]:** PLL is disabled.

**[1]:** PLL is enabled.

**TRSR:** TRX Status Register.

**[0]:** RX state.

**[1]:** TX state.

**TRER:** TRX state enabled status.

**[0]:** TRX is disabled.

**[1]:** TRX is enabled.

##### 10.1.1 Strobe Command - Sleep MODE

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

##### 10.1.2 Strobe Command - Idle MODE

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

### 10.1.3 Strobe Command - Standby MODE

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

### 10.1.4 Strobe Command - PLL MODE

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

### 10.1.5 Strobe Command - RX MODE

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

### 10.1.6 Strobe Command - TX MODE

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

## 10.2 RF Reset Command

In addition to power on reset (POR), A3107M0 could issue software reset (80h) to RF by setting Reset Register (0x50001000h). A3107M0 generates an internal signal "RESETN" to initialize RF circuit. After reset command, RF state is in standby MODE and re-calibration is necessary.

**Reset Register (Address: 0x50001000)**

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	RESETN	FWPRN	FRPRN	FDATARN	BFCRN	--	--	--
R	--	--	--	--	--	--	--	--
Reset	0	0	0	0	0	0	0	0

**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)

## 10.3 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.

A3107M0 supports separated 256-bytes TX and RX FIFO. To use A3107M0's FIFO MODE, user just needs to enable FMS =1 (01h). For FIFO accessing, TX FIFO (write-only) and RX FIFO (read-only). TX FIFO represents transmitted payload. On the other hand, RX circuitry synchronizes ID Code and stores received payload into RX FIFO.

## Relative Control Register

### FIFO Register (Address: 0x50001020)

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

**FPM [1:0]:** FIFO Pointer Margin

**PSA [5:0]:** Used for Segment FIFO.

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

## 10.4 Packet Format of FIFO MODE

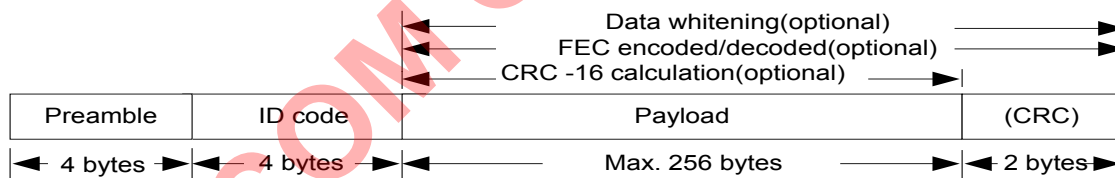


Figure 10.1 Packet Format of FIFO MODE

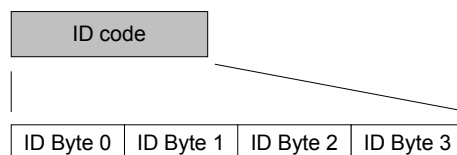


Figure 10.2 ID Code Format

### 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].

### ID code:

ID code is recommended to set 4 bytes by IDL=1. 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] for ID Synchronization check.

### Payload:

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

### CRC (option)

In FIFO MODE, if CRC is enabled (CRCS=1), 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.

## 10.5 Transceiver Frequency

A3107M0 is a half-duplex transceiver with embedded PA and LNA.

The receiver is a low-IF architecture consisting of a LNA, down conversion mixers, poly-phase 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, Channel register (0x500010C0), for frequency agility.

A3107M0'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.

### 10.5.1 State machine

In chapter 10.1, user can learn both accessing A3107M0's control registers as well as issuing Strobe commands.

### 10.5.2 Key states

A3107M0 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 reset values. The calibration process of A3107M0 is very easy, user only needs to issue Strobe commands and enable calibration registers. After calibration, A3107M0 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 10.1 Operation MODE and strobe command.

### 10.5.3 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, A3107M0 is auto back to standby mode. Figure 10.3 and Figure 10.4 are TX and RX timing diagram respectively. Figure 10.5 State diagram of FIFO mode.

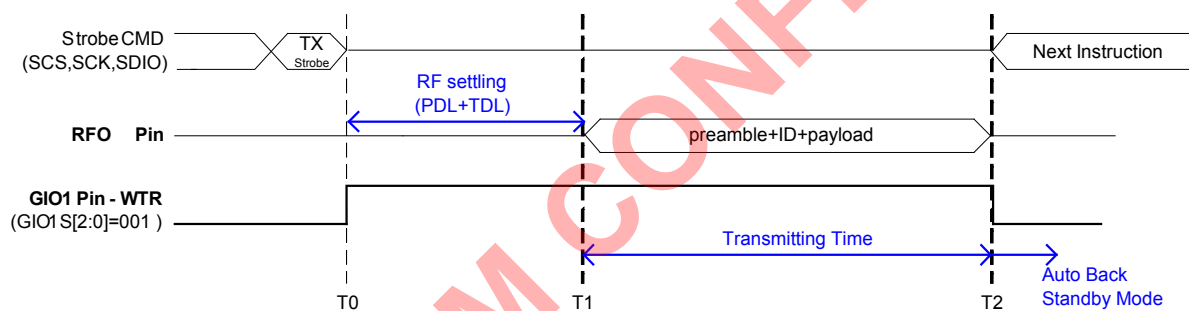


Figure 10.3 TX timing of FIFO MODE.

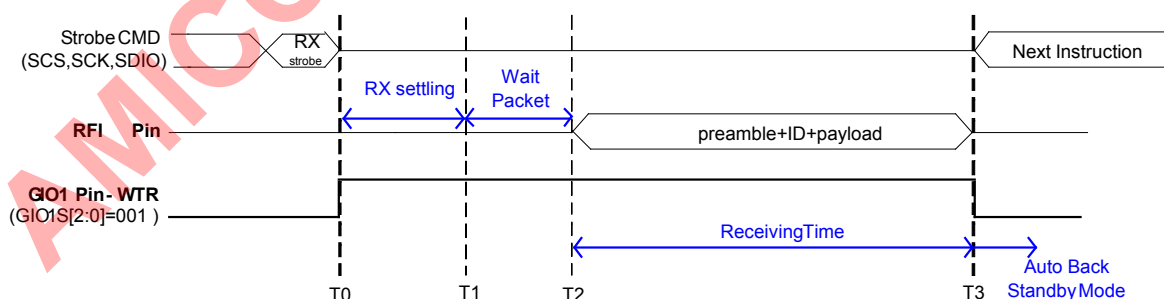


Figure 10.4 RX timing of FIFO MODE.

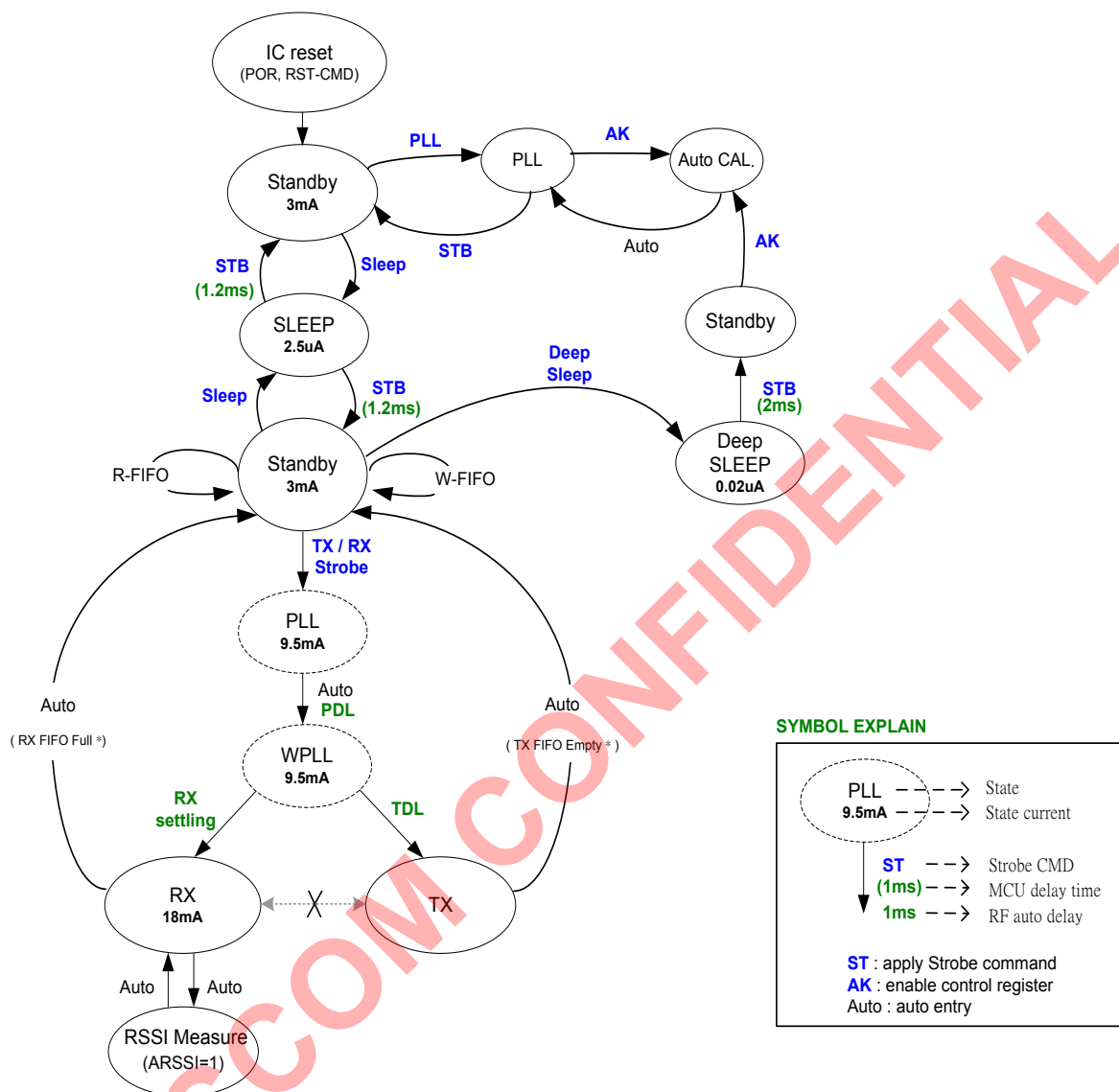


Figure 10.5 State diagram of FIFO MODE.



### 11. SoC Architectural Overview

A3107M0 microcontroller is instruction set compatible with Cortex™-M0 profile processors.

A3107M0 integrates many features such as three 8/16bit counters/timers, watchdog timer, RTC, UART, SPI interface, I<sup>2</sup>C interface, 4 channels PWM, 8 channels ADC and battery detector, The interrupt controller is extended to support 6 interrupt source such as watchdog timer, RTC, SPI, I<sup>2</sup>C, ADC and RF.

A3107M0 includes SWD debug circuitry that provides full time, real-time, in-circuit debugging.

#### 11.1 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.

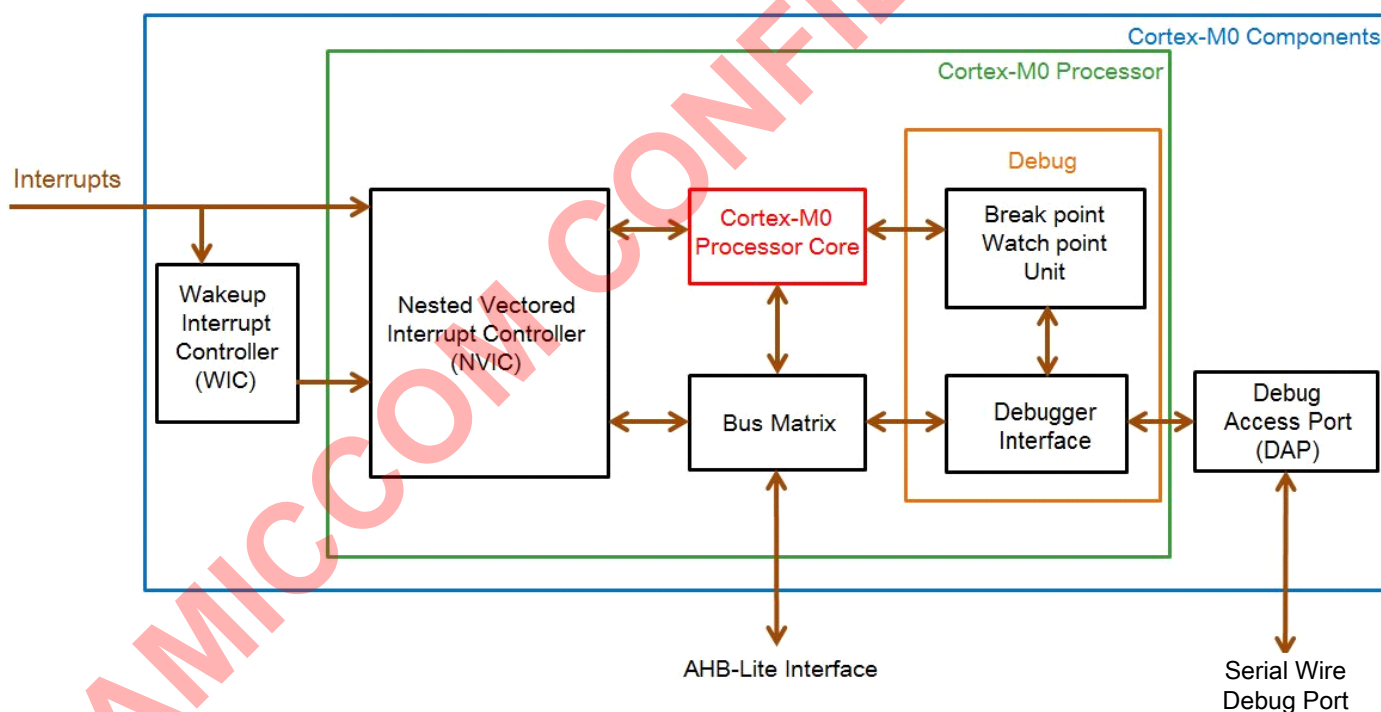


Figure 11.1 Core-M0 block diagram

#### 11.1.1 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 MODEI:  
This is the ARMv6-M, C Application Binary Interface (C-ABI) compliant exception MODEI 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)

### 11.2 Memory Organization

The memory organization is shown as figure 11.2.

0xFFFF_FFFF	Reserved
0xE010_0000	Private Peripheral Bus
0xE000_0000	Reserved
0x5000_0000	AHB Peripherals
0x4008_0000	Reserved
0x4000_0000	APB Peripherals
	Reserved
0x2000_0000	RAM
	Reserved
0x1000_1000	UICR
	Reserved
0x1000_0000	FICR
	Reserved
0x0000_0000	Code

Figure 11.2 Memory Organization

### 11.3 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.

#### 11.3.1 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.

### 11.3.2 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 11.1 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

Table 11.1 Exception Types

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	GPIO0_INT	GPIO 0 combined interrupt for AHB GPIO and I/O port GPIO
23	7	--	--
24	8	TIMER0_INT	Timer0 interrupt
25	9	TIMER1_INT	Timer1 interrupt
26	10	Dual_Timer_INT	Dual Timer interrupt
27	11	MPU_LCD_INT	MPU_LCD interrupt
28	12	--	--
29	13	UART1_INT	UART1 Tx/Rx/Overflow interrupt
30	14	--	--
31	15	WUN_INT	WUN interrupt
32	16	SPI_INT	SPI interrupt
33	17	I <sup>2</sup> C_INT	I <sup>2</sup> CM/I <sup>2</sup> CS 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	SLPTMR1_INT	Sleep timer1 interrupt
41	25	--	--
42	26	--	--
43	27	--	--
44	28	--	--
45	29	--	--
46	30	--	--
47	31	EXT_INT	External signal interrupt from 0x50001300 bit[5]

Table 11.2 Interrupt Map Vector Table

### 11.4 Reset source

Reset circuitry allows A3107M0 to be easily placed in a predefined default condition. LVD, Reset, POR, NVIC reset, and Watchdog signal will reset A3107M0 when they happen.

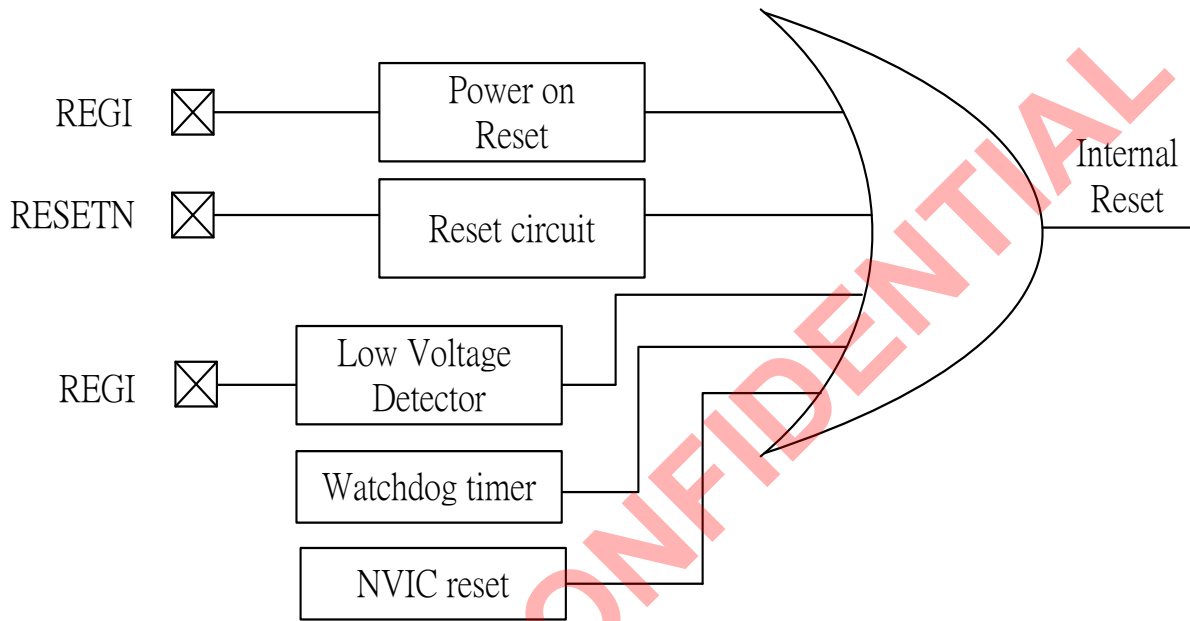


Figure 11.3 Reset source

### 11.5 Clock Source

A3107M0 has three clock sources like crystal oscillator (XI, XO), RTC crystal (P0.22, P0.23/ RTC\_I, RTC\_O) and internal RC oscillator. In the MCU part (digital peripherals), user chooses the suitable clock source by power consumptions and performance. In the RF part, the clock source only comes from XTAL.

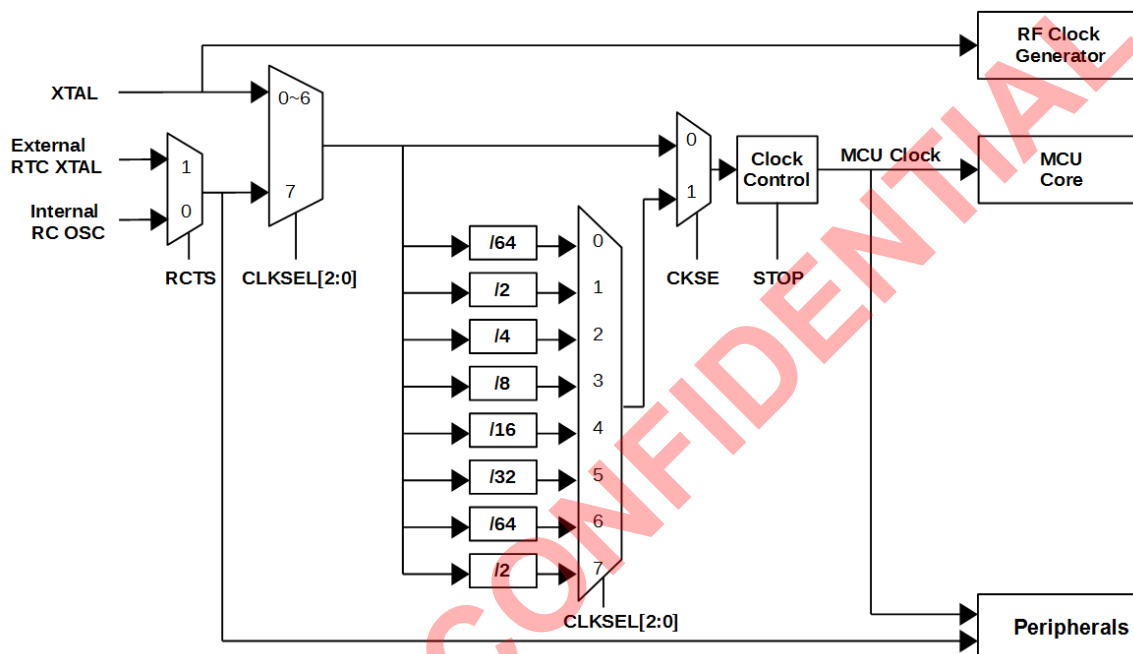


Figure 11.4 Whole chip clock

### 11.6 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.

#### 11.6.1 SysTick Control and Status Register (Address:0xE000E010)

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

**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.

CLKSOURCE needs to be set 1 when uses SysTick.

**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.

### 11.6.2 SysTick Reload Value Register (Address:0xE000E014)

R/W	Bit 31	-----	Bit 24	Bit 23	-----	Bit 0
W	--					
R	--				RELOAD[23:0]	
Reset	0x0				0x000	

**RELOAD[23:0]:** Value to load into the SysTick when the counter is enabled and when it reaches 0.

### 11.6.3 SysTick Current Value Register (Address:0xE000E018)

R/W	Bit 31	-----	Bit 24	Bit 23	-----	Bit 0
W	--					
R	--				CURRENT[23:0]	
Reset	0x0				0x000	

**CURRENT[23:0]:** Reads return the current value of the SysTick counter.

## 11.7 Slow Clock Source

A3107M0 support two slow clock sources: internal RC oscillator and extern RTC crystal. User can set RCTS=0 or 1 to select internal RC-OSC or external RTC-XTAL. Slow clock source is the clock source for slow peripherals to keep working when MCU enter PM (power management) mode. For example: Sleep Timer, Real Time Counter.

### 11.7.1 Relate Register

**RC Control Register 1 (Address: 0x50000040)**

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--	--	MCALS	RTCIOSEL	RCTS	TMRE	MAN	ENCAL
R	--	--						
Reset	0	0	0	0	0	0	0	0

**MCALS:** Enable Continuous RC-OSC Calibration.

[0]: Continuous mode.

[1]: Single mode.

**RTCIOSEL:** RTC XTAL I/O select

[1]: RTC XTAL

[0]: Normal I/O

**RCTS:** Internal Oscillator / External Crystal selection.

[0]: Internal RC OSC



[1]: External RTC XTAL.

**TMRE:** Internal RC OSC / External RTC XTAL enable.

[0]: Disable.

[1]: Enable.

**MAN:** Internal RC OSC Calibration type.

[0]: Auto calibration

[1]: Manual calibration

**ENCAL:** Enable Internal RC OSC calibration.

[0]: Disable

[1]: Enable (Auto clear to 0 when calibration finish)

### RC Control Register 2 (Address: 0x50000044)

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

**RC\_WSEL [1:0]:** Clock select for internal RC oscillator Calibration

[00]: 16 MHz

[01]: 8 MHz

[10]: 4 MHz

[11]: 2MHz

**MVS[1:0]:** RCOSC Calibration sample clock select.

[00]: 1/2.

[01]: 1/4.

[10]: 1/8.

[11]: 1/16

**RCOT[2:0]:** RCOSC current select. Recommend set value is 4.

**MRCT [9:0]:** WOR/RCOSC manual setting value.

**RCOC [9:0]:** WOR Calibration value.

### RC Target Register (Address: 0x50000048)

R/W	Bit 15	-----	Bit 12	Bit 11	-----	Bit 0
W	--			TGNUM[11:0]		
R	--			NUMLH[11:0]		
Reset	0x0			0x000		

**TGNUM[11:0]:** Target Number for RC OSC Calibration for RCOSC auto-calibration.

**NUMLH[11:0]:** WOR Calibration value.

### 11.7.2 Turn on External RTC XTAL

User can follow the step as blow to turn on external RTC XTAL:

1. P0ALTFUNCSET (0x40010018) = (1<<22) | (1<<23)
2. RCCTRL1 (0x50000040) = 0x00
3. RCCTRL1 (0x50000040) = 0x18 (Set RCTS=1, RTCIOSEL=1)
4. RCCTRL1 (0x50000040) = 0x1C (Set TMRE=1)
5. Delay some time for External RTC XTAL stable (about 100ms, depends on RTC XTAL)

### 11.7.3 Turn on Internal RC OSC

User can follow the step as blow to turn on internal RC OSC:

1. RCCTRL1 (0x50000040) = 0x00
2. RCCTRL2 (0x50000044) = 0x03040012
3. RCTARGET (0x50000048) = 977 (for 32.768KHz) or 1000 (for 32KHz)
4. RCCTRL1 (0x50000040) = 0x04 (Set TMRE=1)
5. RCCTRL1 (0x50000040) = 0x05 (Set ENCAL=1)
6. Check RCCTRL1 until ENCAL auto clear to 0 (about 10ms)

AMICCOM CONFIDENTIAL

## 12. I/O Ports

### 12.1 FEATURE

- Tri-state GPIO
  - ◆ For tri-state in PM3 mode, user need to set RFAR[5] = 1 as P0.0 P0.3, P0.11, P0.16, P0.17, P0.26, P0.27
- Support Pull-up resistor
- Support Wakeup pin
- Support 4 types GPIO Interrupt
- Support High Driving Current Output

### 12.2 BLOCK DIAGRAM

Each GPIO has the same structure as below:

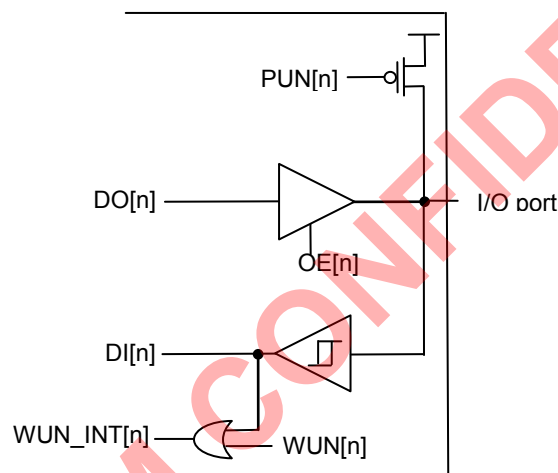


Figure 12.1 I/O Block Diagram

### 12.3 REGISTER

#### 12.3.1 Register List

Address	Name	DESCRIPTION
0x40010000	P0DATA	Port 0 Data Register
0x40010004	P0DATAOUT	Port 0 Data Out Register
0x40010008	P0PUN	Port 0 Pull Up Not Register
0x4001000C	P0WUN	Port 0 Wakeup Enable Not Register
0x40010010	P0OUTENABLESET	Port 0 Output Enable Set Register
0x40010014	P0OUTENABLECLR	Port 0 Output Enable Clear Register
0x40010018	P0ALTFUNCSET	Port 0 Alternative Function Set Register
0x4001001C	P0ALTFUNCCLR	Port 0 Alternative Function Clear Register
0x40010020	P0INTENSET	Port 0 Interrupt Enable Set Register
0x40010024	P0INTENCLR	Port 0 Interrupt Enable Clear Register
0x40010028	P0INTTYPESET	Port 0 Interrupt Type Set Register
0x4001002C	P0INTTYPECLR	Port 0 Interrupt Type Clear Register
0x40010030	P0INTPOLSET	Port 0 Interrupt Polarity Set Register
0x40010034	P0INTPOLCLR	Port 0 Interrupt Polarity Clear Register
0x40010038	P0INTSTATUS	Port 0 Interrupt Request Status Register
0x5000004C	HDV	Port 0 High Driving Current Output Register

Table 12.1 GPIO0 Register list

### 12.3.2 Register Description

#### Port 0 Data Register (Address: 0x40010000)

R/W	Bit 31	-----	Bit 0
W		P0DATAOUT [31:0]	
R		P0DATA [31:0]	
Reset		0x00000000	

**P0DATAOUT[31:0]:** Port 0 data out (output data, DO)

**P0DATA[31:0]:** Port 0 data (input data, DI)

#### Port 0 Data Out Register (Address: 0x40010004)

R/W	Bit 31	-----	Bit 0
W		P0DATAOUT [31:0]	
R			
Reset		0x00000000	

**P0DATAOUT[31:0]:** Port 0 data out (output data, DO)

#### Port 0 Pull Up Not Register (Address: 0x40010008)

R/W	Bit 31	-----	Bit 0
W		P0PUN [31:0]	
R			
Reset		0x00000000	

**P0PUN[31:0]:** Port 0 Pull Up Not

**[0]:** Pull up resistor enable (input setting usually).

**[1]:** Pull up resistor disable (output setting usually)

#### Port 0 Wakeup Enable Not Register (Address: 0x4001000C)

R/W	Bit 31	-----	Bit 0
W		P0WUN [31:0]	
R			
Reset		0xFFFFFFFF	

**P0WUN[31:0]:** Port 0 Wake Up Enable Not

**[0]:** IO pin wakeup enable

**[1]:** IO pin wakeup disable

#### Port 0 Output Enable Set Register (Address: 0x40010010)

R/W	Bit 31	-----	Bit 0
W		P0OUTENABLESET [31:0]	
R		P0OUTENABLE [31:0]	
Reset		0x00000000	

**P0OUTENABLESET [31:0]:** Port 0 Output Enable Set

**[0]:** No effect

**[1]:** Output enable set. (Set IO pin to output)

**P0OUTENABLE[31:0]:** Port 0 Output Enable

**[0]:** IO pin is input

**[1]:** IO pin is output.

#### Port 0 Output Enable Clear Register (Address: 0x40010014)

R/W	Bit 31	-----	Bit 0
W		P0OUTENABLECLR [31:0]	
R		P0OUTENABLE [31:0]	
Reset		0x00000000	

**P0OUTENABLECLR [31:0]:** Port 0 Output Enable Clear

[0]: No effect

[1]: Output enable clean (set IO pin to input).

**P0OUTENABLE[31:0]:** Port 0 Output Enable

[0]: IO pin is input

[1]: IO pin is output.

### Port 0 Alternative Function Set Register (Address: 0x40010018)

R/W	Bit 31	-----	Bit 0
W		P0ALTFUNCSET [31:0]	
R		P0ALTFUNC [31:0]	
Reset		0x00000000	

**P0ALTFUNCSET [31:0]:** Alternative function set

[0]: No effect

[1]: Alternative function set (alternative function enable).

**P0ALTFUNC[31:0]:** Alternative function status

[0]: Alternative function disable (GPIO).

[1]: Alternative function enable (Multi-function IO)

### Port 0 Alternative Function Clear Register (Address: 0x4001001C)

R/W	Bit 31	-----	Bit 0
W		P0ALTFUNCCLR [31:0]	
R		P0ALTFUNC [31:0]	
Reset		0x00000000	

**P0ALTFUNCCLR [31:0]:** Alternative function clear

[0]: No effect

[1]: Alternative function clean (alternative function disable)..

**P0ALTFUNC[31:0]:** Alternative function status

[0]: Alternative function disable (GPIO)

[1]: Alternative function enable (Multi-function IO)

### Port 0 Interrupt Enable Set Register (Address: 0x40010020)

R/W	Bit 31	-----	Bit 0
W		POINTENSET [31:0]	
R		POINTEN [31:0]	
Reset		0x00000000	

**POINTENSET[31:0]:** Port 0 interrupt enable set

[0]: No effect

[1]: Interrupt enable set (interrupt enable)

**POINTEN[31:0]:** Port 0 interrupt enable status

[0]: Interrupt disable

[1]: Interrupt enable

### Port 0 Interrupt Enable Clear Register (Address: 0x40010024)

R/W	Bit 31	-----	Bit 0
W		POINTENCLR [31:0]	
R		POINTEN [31:0]	
Reset		0x00000000	

**POINTENCLR[31:0]:** Port 0 interrupt enable clear

[0]: No effect

[1]: Interrupt enable clear (interrupt disable)

**POINTEN[31:0]:** Port 0 interrupt enable status

[0]: Interrupt disable

[1]: Interrupt enable

### Port 0 Interrupt Type Set Register (Address: 0x40010028)

R/W	Bit 31	-----	Bit 0
W		POINTTYPESET [31:0]	
R		POINTTYPE [31:0]	
Reset		0x00000000	

**POINTTYPESET[31:0]:** Port 0 interrupt type set

[0]: No effect

[1]: Interrupt type set (interrupt by edge)

**POINTTYPE[31:0]:** Port 0 interrupt type status

[0]: Interrupt by level

[1]: Interrupt by edge

### Port 0 Interrupt Type Clear Register (Address: 0x4001002C)

R/W	Bit 31	-----	Bit 0
W		POINTTYPECLR [31:0]	
R		POINTTYPE [31:0]	
Reset		0x00000000	

**POINTTYPECLR[31:0]:** Port 0 interrupt type clear

[0]: No effect

[1]: Interrupt type clean (interrupt by level).

**POINTTYPE[31:0]:** Port 0 interrupt type status

[0]: Interrupt by level

[1]: Interrupt by edge

### Port 0 Interrupt Polarity Set Register (Address: 0x40010030)

R/W	Bit 31	-----	Bit 0
W		POINTPOLSET [31:0]	
R		POINTPOL [31:0]	
Reset		0x00000000	

**POINTPOLSET[31:0]:** Port 0 interrupt level set

[0]: No effect

[1]: Polarity-level set (high level or rising edge)

**POINTPOL[31:0]:** Port 0 interrupt level status

[0]: Low level or falling edge

[1]: High level or rising edge

### Port 0 Interrupt Polarity Clear Register (Address: 0x40010034)

R/W	Bit 31	-----	Bit 0
W		POINTPOLCLR [31:0]	
R		POINTPOL [31:0]	
Reset		0x00000000	

**POINTPOLCLR[31:0]:** Port 0 interrupt level clear

[0]: No effect

[1]: Polarity-level clean (low level or falling edge)

**POINTPOL[31:0]:** Port 0 interrupt level status

[0]: Low level or falling edge

[1]: High level or rising edge

### Port 0 Interrupt Request Status Register (Address: 0x40010038)

R/W	Bit 31	-----	Bit 0
W		POINTSTATUSCLR [31:0]	

R	POINTSTATUS [31:0]
Reset	0x00000000

**POINTSTATUSCLR[31:0]:** Port 0 interrupt request clear

[0]: No effect

[1]: Clear interrupt request status

**POINTSTATUS[31:0]:** Port 0 interrupt request status

[0]: No interrupt request

[1]: An interrupt request has occurred

**Port 0 High Driving Current Output Register (Address: 0x5000004C )**

R/W	Bit 31	-----	Bit 0
W		HDV [31:0]	
R		--	
Reset		0x00000000	

**HDV[31:0]:** Port0 High Driving Current Output Enable

[0]: Normal Driving Current Output

[1]: High Driving Current Output

## 12.4 FUNCTION DESCRIPTION

### 12.4.1 Output Enable (OE)

Each port has 32pins digital I/O Pins and each pin of port can be defined as general-purpose I/O (GPIO) or peripheral I/O signals connected to the timers, UART, I<sup>2</sup>C and SPI functions. Thus, each pin can also be used to wake up from PM mode.

User can select each pin function by setting register. Each port has its own port register like P0DATA (0x40010000). When reading, the logic levels of the Port's input pins are returned. As shown the below Table 12.2 and Table 12.3, each port has three registers to setting Pull-Up Not (PUN), Output-Enable (OE) and Wakeup enable Not (WUN). As shown the below block diagram, Figure 12.1 I/O Block Diagram. 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.

According to the Table 12.2 all pins can be configured as Output, Input or Input with the pull-up resistor (around 100 Kohm). Please refer the Table 12.2 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.

OE[n]	PUN[n]	I/O port status
0	0	input with pull high
0	1	input without pull high
1	1	output

Table 12.2 I/O Port setting

WUN[n]	Wakeup Interrupt (WUN_INT)
0	Enable*
1	Disable

\* The I/O port must set to input(i.e. OE = 0).

Table 12.3 WUN\_INT setting

### 12.4.2 WUN interrupt (WUN\_INT)

The NVIC IRQn of WUN\_INT is 15. Please refer to Table 11.2 Interrupt Map Vector Table. This interrupt can wake up MCU from PM1, PM2. Port 0.26~Port 0.29 can wakeup MCU and need to initialize all needed peripherals.

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

User can use P0 port as key input and meanwhile these key inputs are low to event a WUN\_INT to wakeup MCU. It is a helpful use to design a remote controller and low power consumption with power saving mode setting.

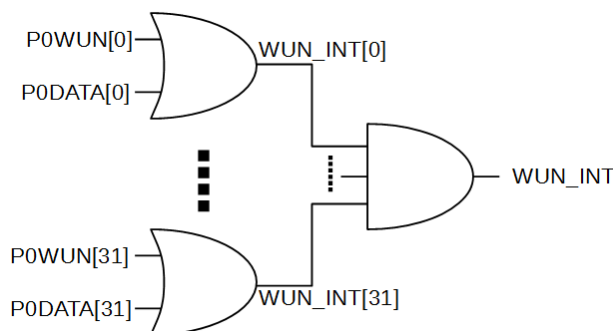


Figure 12.2 WUN\_INT block diagram

### 12.4.3 GPIO0 interrupt (GPIO0\_INT)

The NVIC IRQn of GPIO0\_INT is 6. Please refer to Table 11.2 Interrupt Map Vector Table. This interrupt can't wakeup MCU. GPIO0\_INT supports 4 types IO interrupt. It needs work on normal mode. The setting of GPIO0\_INT shows as Table 12.4.

P0INTTYPE	P0INTPOL	Interrupt by
0	0	Low level
0	1	High Level
1	0	Falling Edge
1	1	Rising Edge

Table 12.4 Port 0 interrupt setting table

### 12.4.4 High Driving Current Output

The HDV register (0x5000004C) can set Prots0 IO driving more current when output high and output low. Please refer to chapter 8 Electrical Specification.

### 12.4.5 Debug Interface and Flash Mask

There are four IO pins need to notice: P0\_06, P0\_07, P0\_10 and P0\_12.

- P0\_06, P0\_07 will be set to SWDIO and SWCLK when P0\_12 = 0 at reset.
- Flash will be mask (read out 0xFFFFFFFF) when P0\_12=0 and P0\_10=0 at reset.

GPIO pins	At Reset				After Rest	
	P0_12	P0_06	P0_07	P0_10	IO Pins	Flash
P0_12	1	X	X	X	P0_12	Flash not mask
P0_06					P0_06	
P0_07					P0_07	
P0_10					P0_10	
P0_12	0	X	X	1	P0_12	Flash not mask
P0_06					SWDIO	
P0_07					SWCLK	
P0_10					P0_10	
P0_12	0	X	X	0	P0_12	Flash MASKED
P0_06					SWDIO	
P0_07					SWCLK	
P0_10					P0_10	

Table 12.5 The pins for debug interface and flash mask

Flash mask is a special measure when the MCU can't download code because of a program execution. User can use Flash Mask to mask flash ROM and erase flash ROM. Please pay attention to the use of restrictions when planning the circuit.



### 13. Timer

#### 13.1 FEATURE

- Programmable 32-bits Timer
- Interrupt generator
- Support input-edge count or time-capture

#### 13.2 PINS DESCRIPTION

PIN	GPIO	TYPE	DESCRIPTION
TIMER0EIN	P0_08	INPUT	EXTIN (external input) for Timer0
TIMER1EIN	P0_09	INPUT	EXTIN (external input) for Timer1

#### 13.3 BLOCK DIAGRAM

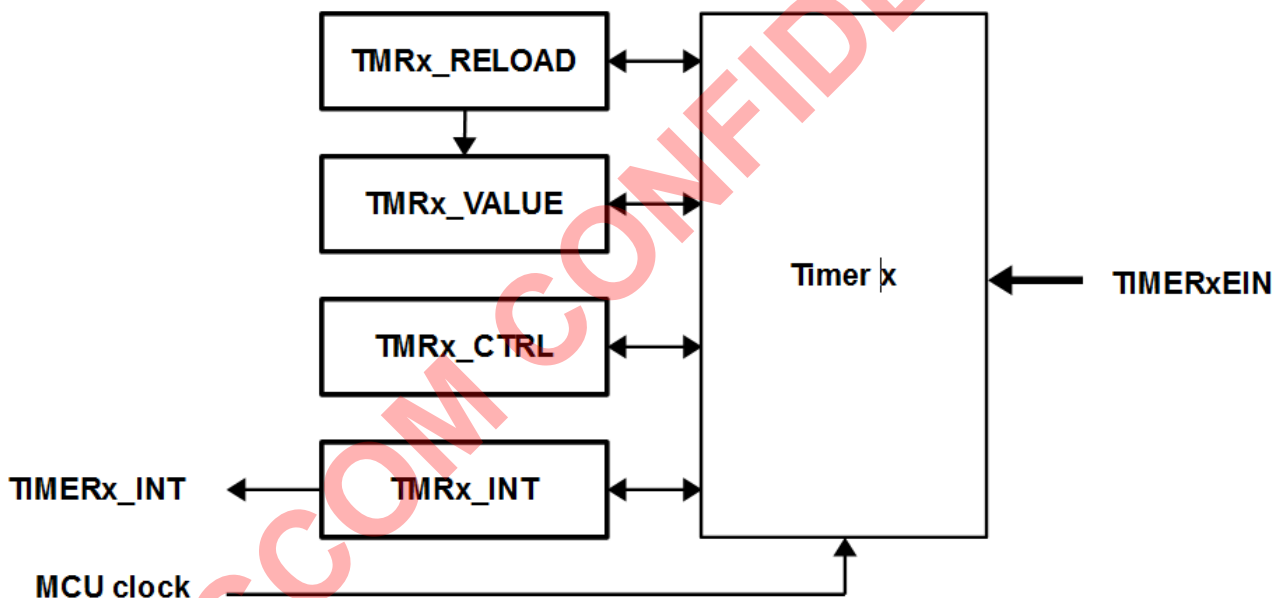


Figure 13.1 32-Bit Timer/Counter block diagram

#### 13.4 REGISTER

##### 13.4.1 Base Address List

Base Address	DESCRIPTION
0x40000000	TIMER0 Base address
0x40001000	TIMER1 Base address

Table 13.1 Base address list

##### 13.4.2 Register List

offset	Name	DESCRIPTION
0x000	TMRx_CTRL	Timer x Control Register
0x004	TMRx_VALUE	Timer x Current Value Register
0x008	TMRx_RELOAD	Timer x Reload Register
0x00C	TMRx_INT	Timer x Interrupt Status Register

Table 13.2 Register list

### 13.4.3 Register Description

#### Timer x Control Register (Offset: 0x000)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--				CTRL [3:0]			
R								
Reset	0	0	0	0	0	0	0	0

**CTRL[0]:** Enable decrement control

[0]: Disable (Timer stop decrement)

[1]: Enable (Timer start decrement)

**CTRL[1] :** Select external input as enable

[0]: Disable

[1]: Enable (Timer stop when EXTIN=0)

**CTRL[2] :** Select external input as clock

[0]: Disable

[1]: Enable (Timer decrement when EXTIN rising edge)

**CTRL[3] :** Timer interrupt enable.

[0]: Disable

[1]: Enable

#### Timer x Current Value Register (Offset: 0x004)

R/W	Bit31	-----	Bit 0
W	VALUE [31:0]		
R			
Reset	0x00000000		

**VALUE[31:0]:** Current counter value.

#### Timer x Reload Register (Offset: 0x008)

R/W	Bit31	-----	Bit 0
W	RELOAD [31:0]		
R			
Reset	0x00000000		

**RELOAD [31:0]:** Reload value. VALUE[31:0] will be wrote when write RELOAD [31:0] immediately.

#### Timer x interrupt Status Register (Offset: 0x00C )

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--							INTCLEAR
R								INTSTATUS
Reset	0	0	0	0	0	0	0	0

**INTCLEAR:**

[0]: No effect

[1]: Clear INTSTATUS

**INTSTATUS:** interrupt status (TIMERINT status)

[0]: Interrupt not occur

[1]: Interrupt occur

### 13.5 FUNCTION DESCRIPTION

#### 13.5.1 VALUE and RELOAD

When any value write to RELOAD[31:0], the VALUE[31:0] will be wrote to the same value immediately. When timer CTRL[0]=1, the VALUE[31:0] start count-down with MCU clock or EXTIN rising edge. When VALUE[31:0] count-down to 0, the VALUE[31:0] will be wrote with RELOAD[31:0] in next clock or rising edge. Please refer to Figure 13.2.

#### 13.5.2 Input-edge Counter

When CTRL[2]=1, Timer is in input-edge count mode. The value of Timer will count down when EXTIN rising edge. The interrupt will occur when VALUE[31:0] count down to 0 and VALUE[31:0] will set with RELOAD[31:0] in next EXTIN rising edge. Figure 13.2 and Figure 13.3 are examples for input-edge counter.

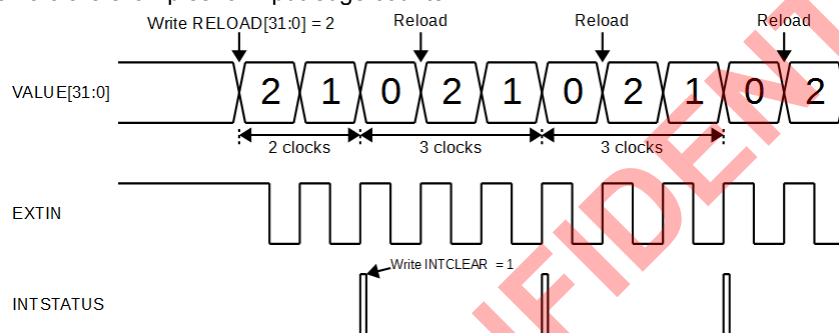


Figure 13.2 A example for input-edge count

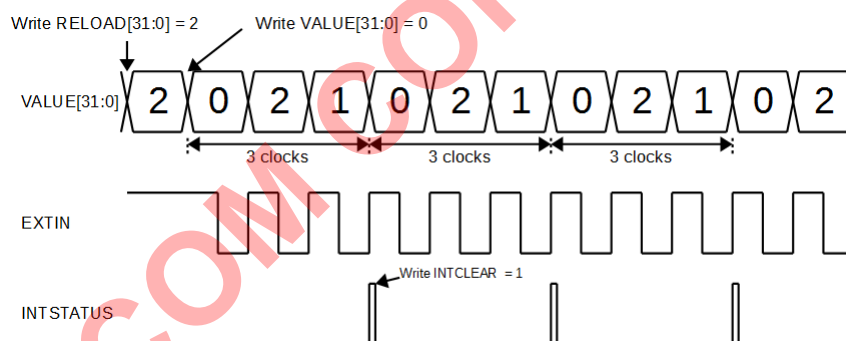


Figure 13.3 Another example for input-edge count

Note: EXTIN must be slower than half of the MCU clock.

#### 13.5.3 Time Capture

When CTRL[1]=1, Timer is in time-capture mode. In this mode, timer will stop count-down when EXTIN=0. User can use this mode to capture time of EXTIN high.

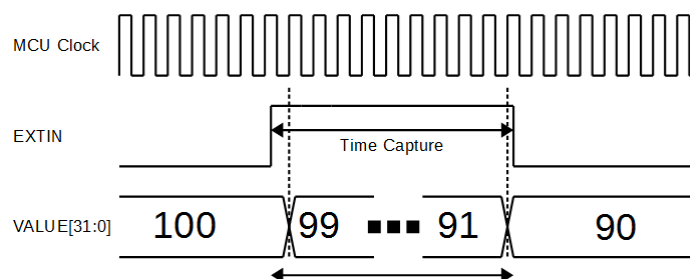


Figure 13.4 A example for time capture

#### **13.5.4 Timer Interrupt**

When VALUE[31:0] count-down to 0, the INTSTATUS will set to 1. The interrupt will occur if CTRL[3]=1. User need write INTCLEAR=1 to clear INTSTATUS. Please refer to Figure 13.2..

The NVIC IRQn of TIMER0 and TIMER1 is 8 and 9. Please refer to Table 11.2 Interrupt Map Vector Table.

AMICCOM CONFIDENTIAL

### 14. Dual Timer

Dual Timer is two programmable 32-bit down-counters with MCU clock. Interrupt will generate when any dual timer count-down to 0.

#### 14.1 FEATURE

- Three clock pre-scale: 1, 16, 256
- Two counter sizes: 16-bits and 32-bits
- Three modes: free-running, one-shot and periodic
- One interrupt for two dual timers.
- Background Load

#### 14.2 BLOCK DIAGRAM

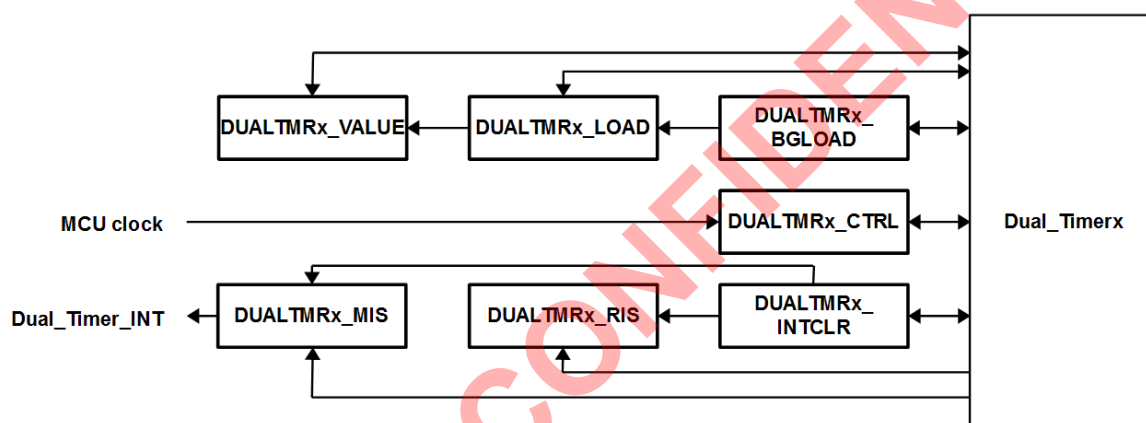


Figure 14.1 Dual Timer block diagram

#### 14.3 REGISTER

##### 14.3.1 Base Address List

Base Address	DESCRIPTION
0x40002000	Dual Timer 1 Base address
0x40002020	Dual Timer 2 Base address

Table 14.1 Base address list

##### 14.3.2 Register List

Offset	Name	DESCRIPTION
0x00	DUALTMRx_LOAD	Dual Timer x Load register
0x04	DUALTMRx_VALUE	Dual Timer x current Value register
0x08	DUALTMRx_CTRL	Dual Timer x Control register
0x0C	DUALTMRx_INTCLR	Dual Timer x Interrupt Clear register
0x10	DUALTMRx_RIS	Dual Timer x Raw Interrupt register
0x14	DUALTMRx_MIS	Dual Timer x Mask Interrupt register
0x18	DUALTMRx_BGLOAD	Dual Timer x Back Ground Load register

Table 14.2 Register list

### 14.3.3 Register Description

#### Dual Timer x Load Register (Offset: 0x00)

R/W	Bit31	-----	Bit 0
W	LOAD[31:0]		
R			
Reset	0x00000000		

**LOAD[31:0]:** Dual timer x Load value. Write LOAD[31:0] will reset the VALUE[31:0] immediately.

#### Dual Timer x current Value Register (Offset: 0x04)

R/W	Bit31	-----	Bit 0
W	VALUE[31:0]		
R			
Reset	0x00000000		

**VALUE[31:0]:** Dual timer x current count value.

#### Dual Timer x Control Register (Offset: 0x08)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	EN	MD	INTEN	--	PRE[1:0]		SLT	OS
R								
Reset	0	0	1	0	0	0	0	0

**EN:** Dual timer x Enable

**[0]:** Disable

**[1]:** Enable (Dual timer start count down)

**MD:** Dual timer Mode

**[0]:** Free-running mode

**[1]:** Periodic mode

**INTEN:** Dual timer interrupt enable.

**[0]:** Dual Timer interrupt disable.

**[1]:** Dual Timer interrupt enable.

**PRE[1:0]:** Dual timer prescale.

**[00]:** Clock is divided by 1.

**[01]:** Clock is divided by 16

**[10]:** Clock is divided by 256

**[11]:** Undefined

**SLT:** Dual timer size selects 16-bit or 32-bit counter operation

**[0]:** 16-bit counter

**[1]:** 32-bit counter

**OS:** Dual timer one-shot count selects

**[0]:** Wrapping

**[1]:** One-shot

#### Dual Timer x Interrupt Clear Register (Offset: 0x0C )

R/W	Bit31	-----	Bit 0
W	INTCLR		
R	--		
Reset	0x00000000		

**INTCLR:** Timer interrupt clear. Write any value to this register to clean RIS and MIS.

#### Dual Timer x RIS Register (Offset: 0x10)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W								--
R								RIS
Reset	0	0	0	0	0	0	0	0

**RIS:** Dual timer Raw interrupt status.

**[0]:** VALUE[31:0] not counts down to 0.

**[1]:** VALUE[31:0] had counted down to 0.

### Dual Timer x MIS Register (Offset: 0x14)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W								--
R								MIS
Reset	0	0	0	0	0	0	0	0

**MIS:** Dual timer interrupt enabled status from the counter.

**[0]:** INTEN=0 or RIS=0. Dual Timer interrupt not occur.

**[1]:** INTEN=1 and RIS=1. Dual Timer interrupt occur.

### Dual Timer x Background Load Register (Offset: 0x18)

R/W	Bit31	-----	Bit 0
W	BGLOAD[31:0]		
R	BGLOAD[31:0]		
Reset	0x00000000		

**BGLOAD[31:0]:** Dual timer Background Load. Write BGLOAD[31:0] will set LOAD[31:0] to BGLOAD[31:0] immediately but not effect to VALUE[31:0].

## 14.4 FUNCTION DESCRIPTION

### 14.4.1 Operation Mode

The BIT0 of DUALTMRx\_CTRL is OS and the BIT6 of DUALTMR\_CTRL is MD. These control bits can set dual timer to three operation mode:

OS	MD	Operation	VALUE reload when counts down to 0
0	0	Free-running	0xFFFF or 0xFFFFFFFF
0	1	Period	LOAD
1	X	One-Shot	Dual Timer Halted. Write LOAD to re-start dual timer.

Table 14.3 Dual Timer operation modes

### 14.4.2 LOAD and BGLOAD

Dual timer has two reload register: LOAD[31:0] and BGLOAD[31:0]. VALUE[31:0] will be set to LOAD[31:0] when any value write to LOAD[31:0] immediately. LOAD[31:0] will be set to BGLOAD[31:0] when any value write to BGLOAD[31:0] and the same time, VALUE[31:0] will not any effect. Figure 14.2 is an example for write LOAD[31:0] and write BGLOAD[31:0].

Write LOAD[31:0] can modify period and current counting immediately. Write BGLOAD[31:0] can modify period at next time count-down to 0 but does not affect the current counting.

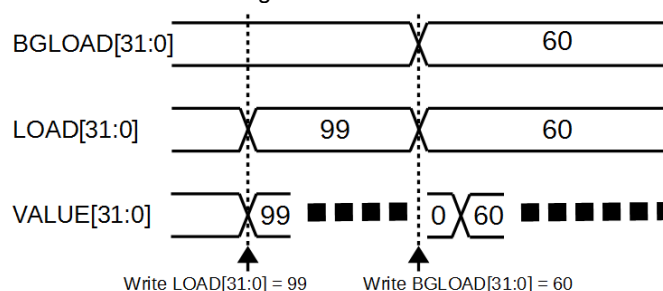


Figure 14.2 An example for write LOAD and BGLOAD

### 14.4.3 Interrupt

Dual timer 1 and Dual timer 2 have the same interrupt handler. Users can check MIS to distinguish interrupt source. The NVIC IRQn of Dual Timer is 10. Please refer to Table 11.2 Interrupt Map Vector Table.

Interrupt will be occurred when VALUE[31:0] count-down to 0 and INTEN=1.

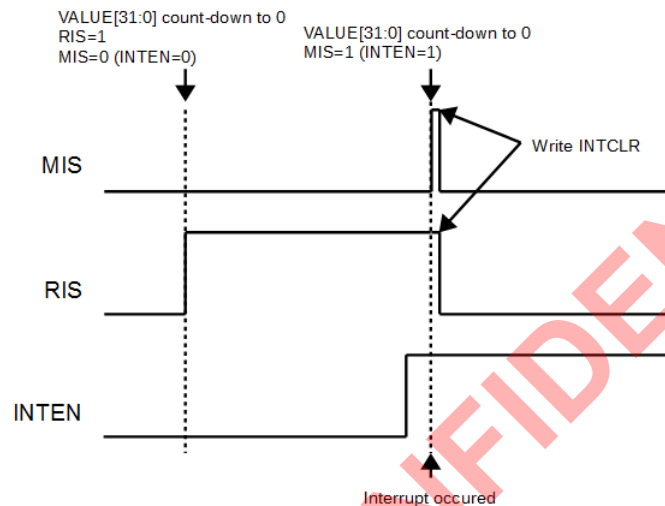


Figure 14.3 Dual Timer interrupt

There is a sample code for Dual timer handler shown as below:

```
void DUALTIMER_Handler(void)
{
    if(DUALTIMER1->MIS & DUALTIMER_MASKINTSTAT_Msk)
    {
        DualTimer_ClearIRQ(DUALTIMER1);
        /* User code */
    }

    if(DUALTIMER2->MIS & DUALTIMER_MASKINTSTAT_Msk)
    {
        DualTimer_ClearIRQ(DUALTIMER2);
        /* User code */
    }
}
```



### 15. Sleep Timer

A3107M0 includes two sleep timers: Sleep Timer0 and Sleep Timer1. Sleep timer 0 wakeup MCU from PM1, PM2, and PM3 mode. Sleep Timer 1 wakeup MCU from PM1 and PM2 mode.

#### 15.1 FEATURES

- Two 16-bit timers and independent interrupts
- Programmable timer interval (ex. RTC clock=32.768KHz, 0.3125 ms~163.84 s)
- Programmable timer0 offset setting for entering Rx mode or Tx mode in advance

#### 15.2 BLOCK DIAGRAM

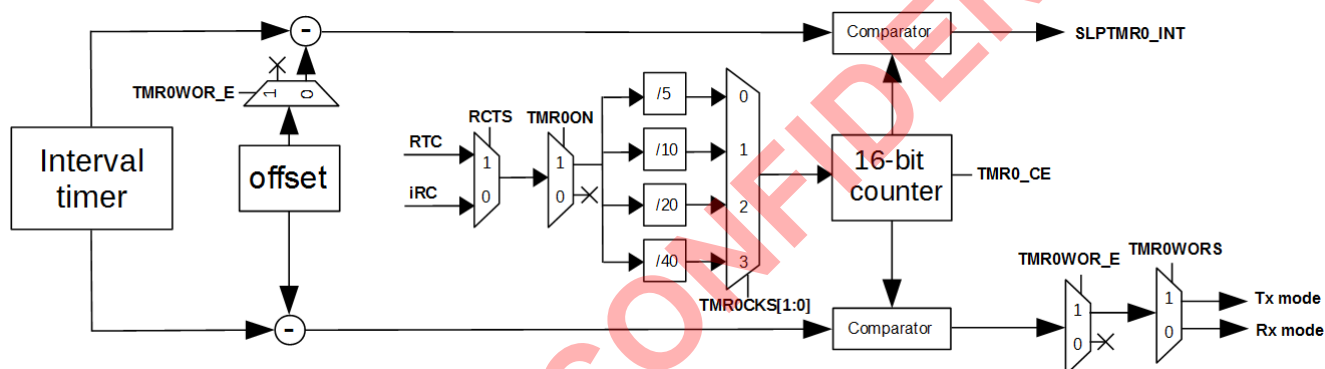


Figure 15.1 Sleep Timer0 Block Diagram

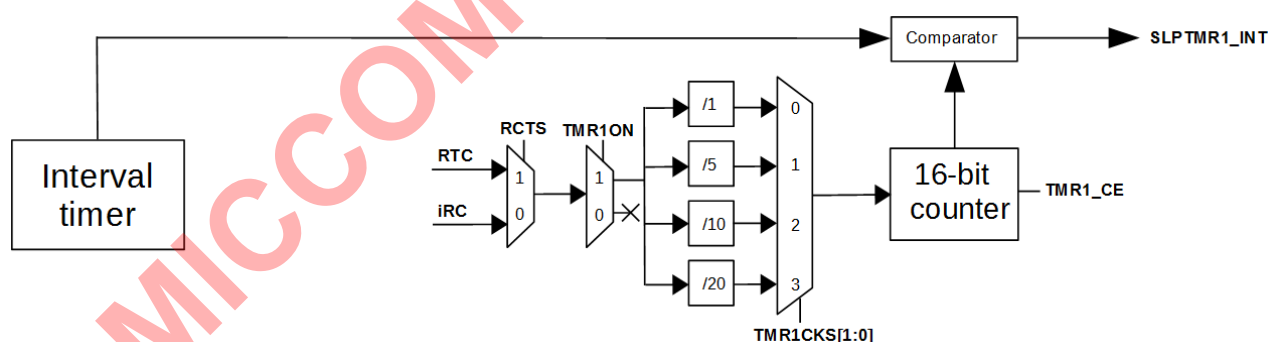


Figure 15.2 Sleep Timer1 Block Diagram

#### 15.3 REGISTER

##### 15.3.1 Register list

Address	Name	DESCRIPTION
0x50006000	ST0IR	Sleep Timer0 Interval Register
0x50006004	ST0CR	Sleep Timer0 Control Register
0x50006010	ST1IR	Sleep Timer1 Interval Register
0x50006014	ST1CR	Sleep Timer1 Control Register

Table 15.1 Sleep Timer Register List

### 15.3.2 Register Description

#### Sleep Timer0 Interval Register (Address: 0x50006000)

R/W	Bit 15	-----	Bit 0
W	TMR0_ITV[15:0]		
R			
Reset	0x00000000		

**TMR0\_ITV [15:0]:** Timer0 interval setting.

Timer0 interval time = 1/ (timer source clock frequency) \* (TMR0\_ITV[15:0] + 1)

#### Sleep Timer0 Control Register (Address: 0x50006004)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	TMR0_RST	--						
R	--							
Reset	0	0	0	0	0	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W	--			TMR0_OFS[4:0]				
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	--			TMR0_PM3	TMR0WORS	TMR0SK	TMR0COR	TMR0WOR_E
R				--				
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	TMR0ON	TMR0IE	TMR0IF	--		TMR0CKS[1:0]		TMR0_CE
R								
Reset	0	0	0	0	0	0	0	0

**TMR0\_RST:** Write to this bit to issue TMR0 reset command. (Write "1" to reset).

**TMR0\_OFS[4:0]:** TMR0 interval offset for 16-bits Timer0.

**TMR0\_PM3:** Reserved for internal usage. Should be set to [0]

**TMR0WORS:** Timer0 WOR / WOT selection.

[0]: Wakeup On Rx

[1]: Wakeup On Tx

**TMR0SK:** Must be set to [1] when TMR0COR set to [1].

[0]: Disable.

[1]: Enable.

**TMR0COR:** Timer0 CLK re-correct when SYNC.

[0]: Disable.

[1]: Enable.

**TMR0WOR\_E:** Timer0 WOR / WOT enable.

[0]: Disable. Interrupt generates at (TMR0\_ITV[15:0] + 1- TMR0\_OFS[4:0])

[1]: Enable. Enter Rx / Tx mode at (TMR0\_ITV[15:0] + 1- TMR0\_OFS[4:0]) and interrupt generates at (TMR0\_ITV[15:0]+1)

**TMR0ON:** Turn on TMR0CK.

[0]: Disable.

[1]: Enable.

**TMR0IE:** Timer0 Interrupt Enable. Must set to [1].

[0]: Disable.

[1]: Enable.

**TMR0IF:** Timer0 Interrupt Flag. (Must write "1" to clear if TMR0IF=1).

[0]: Interrupt not occur  
[1]: Interrupt had occurred.

**TMR0CKS [1:0]:** Select Timer0 Source Clock.

[00]: RTC clock / 5  
[01]: RTC clock / 10  
[10]: RTC clock / 20  
[11]: RTC clock / 40

**TMR0\_CE:** Start Timer0 counter.

[0]: Stop.  
[1]: Start.

### Sleep Timer1 Interval Register (Address: 0x50006010)

R/W	Bit 15	-----	Bit 0
W	TMR1_ITV[15:0]		
R			
Reset	0x00000000		

**TMR1\_ITV [15:0]:** Timer1 interval setting.

Timer1 interval = 1 / (timer source clock frequency) \* (TMR1\_ITV[15:0] + 1)

### Sleep Timer1 Control Register (Address: 0x50006014)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	TMR1_RST	--						
R	--							
Reset	0	0	0	0	0	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W	--							
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	--							
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	TMR1ON	TMR1IE	TMR1IF	--		TMR1CKS[1:0]		TMR1_CE
R								
Reset	0	0	0	0	0	0	0	0

**TMR1\_RST:** Write to this bit to issue TMR1 reset command. (Write "1" to reset).

**TMR1ON:** Turn on TMR1CK.

[0]: Disable.  
[1]: Enable.

**TMR1IE:** Timer1 Interrupt Enable. Must set to [1].

[0]: Disable.  
[1]: Enable.

**TMR1IF:** Timer1 Interrupt Flag. (Must write "1" to clear if TMR1IF=1).

[0]: Interrupt not occur  
[1]: Interrupt had occurred.

**TMR1CKS [1:0]:** Select Timer1 Source Clock

[00]: RTC clock  
[01]: RTC clock / 5  
[10]: RTC clock / 10  
[11]: RTC clock / 20

**TMR1\_CE:** Start Timer1 counter

[0]: Stop

[1]: Start

## 15.4 FUNCTION DESCRIPTION

A3107M0 has two 16-bit sleep timers. Each sleep timer has a control register (STxCR), a time interval setting register (STxIR) and an independent interrupt. TMRxRST is setting for clear counter number. After writing target number into TMRx\_ITV[15:0], user must enable TMRxIE select the clock source by setting TMRxCKS, turn on the clock source by setting TMRxON, and start timerx counter - up by setting TMRx\_CE. When the counter target number is reached, counter number will turn into 0, then generate an interrupt signal. If MCU is in the PM mode, the interrupt signal will wake up MCU immediately. User need write TMRxIF =1 to clear INTSTATUS if TMRxIF =1. In addition, Sleeper timer0 has extra offset functions that can be used for entering Rx or Tx mode earlier by setting TMR0\_OFS[4:0], TMR0WOR\_E and TMR0WORS.

List some example setting in table 15.2

RTC Clock = 32.768KHz					
Sleep timer	TMRxCKS	TMRx_ITV[15:0]	TMR0_OFS[4:0]	Timer0 interval offset	Timerx interval
Sleep Timer0	RTC clock / 5	655	0	0ms	100.098ms
Sleep Timer0	RTC clock / 5	655	31	4.73ms	100.098ms
Sleep Timer1	RTC clock	3276	0	0ms	100.006ms
Sleep Timer1	RTC clock / 5	655	0	0ms	100.098ms

Table 15.2 Some example setting

The NVIC IRQn of Sleep timer0 and Sleep timer1 is 2 and 24. Please refer to Table 11.2.

## 15.5 PROCEDURE

### 15.5.1 Sleep Timer0

Below is the procedure to set sleep timer0 (ex. Timer Interval 100.098ms, Tx mode offset 4.73ms)

- Step1: Set A3107M0 in STBY MODE.
- Step2: Set on RTC / IRC.
- Step3: Set on sleep timer0 NVIC IRQn.
- Step4: Set ST0CR (0x50006004), TMR0\_RST=1.
- Step5: Set ST0CR (0x50006004), TMR0CKS [1:0]=0, TMR0IE=1, TMR0ON=1.
- Step6: Set ST0CR (0x50006004), TMR0\_OFS [4:0]=31, TMR0WORS =1, TMR0WOR\_E =1.
- Step7: Set ST0IR (0x50006000), TMR0\_ITV[15:0]=655.
- Step8: Set ST0CR (0x50006004), TMR0\_CE=1.
- Step9: Interrupt will occur if TMR0IF =1. User need write TMR0IF =1 to clear INTSTATUS.

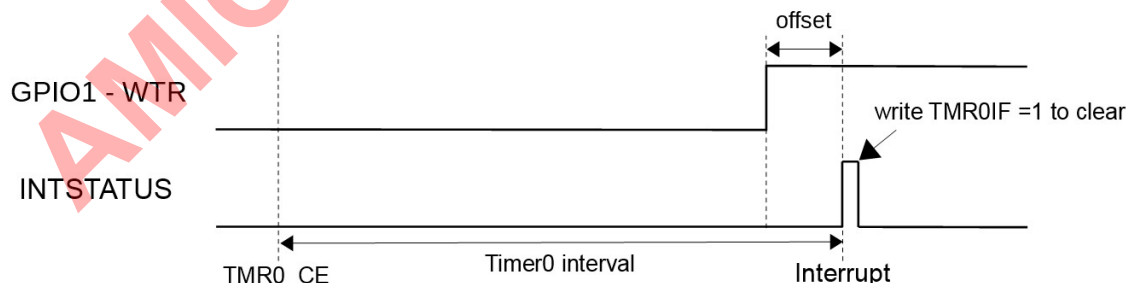


Figure 15.3 Sleep timer0 Function Diagram

### 15.5.2 Sleep Timer1

Below is the procedure to set sleep timer1. (ex. Timer Interval 100.098ms)

- Step1: Set A3107M0 in STBY MODE.
- Step2: Set on RTC / IRC.

Step3: Set on sleep timer1 NVIC IRQn.  
Step4: Set ST1CR (0x50006014), TMR1\_RST=1.  
Step5: Set ST1CR (0x50006014), TMR1CKS [1:0]=1, TMR1IE=1, TMR1ON=1.  
Step6: Set ST1IR (0x50006010), TMR1\_ITV[15:0]=655.  
Step7: Set ST1CR (0x50006014), TMR1\_CE=1.  
Step8: Interrupt will occur if TMR1IF =1. User need write TMR1IF =1 to clear INTSTATUS.

AMICCOM CONFIDENTIAL

## 16. Watchdog Timer

A3107M0 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 it as a very long timer with disabled reset function.

### 16.1 FEATURE

- The 32-bit free-running down-counter
- Lock register to prevent accidental write access
- Interrupt and Reset generation

### 16.2 BLOCK DIAGRAM

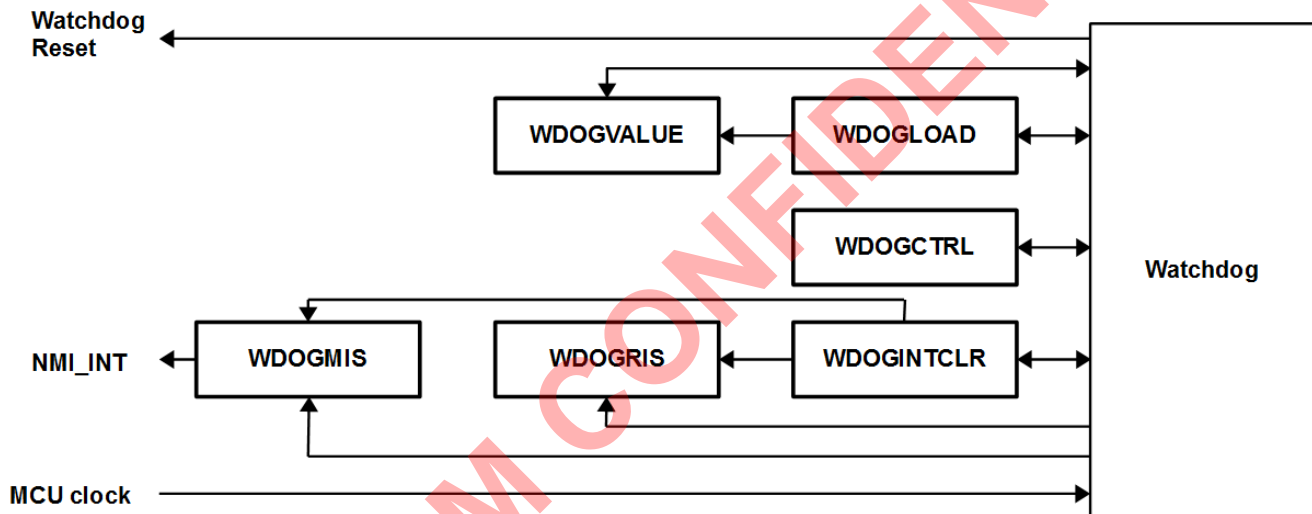


Figure 16.1 Watchdog block diagram

### 16.3 REGISTER

#### 16.3.1 Register List

Address	Name	DESCRIPTION
0x40008000	WDOGLOAD	Watchdog Load Register
0x40008004	WDOGVALUE	Watchdog Value Register
0x40008008	WDOGCTRL	Watchdog Control Register
0x4000800C	WDOGINTCLR	Watchdog Interrupt Clean Register
0x40008010	WDOGRIS	Watchdog Raw Interrupt Status Register
0x40008014	WDOGMIS	Watchdog Enabled Interrupt Status Register
0x40008C00	WDOGLOCK	Watchdog Lock Register

Table 16.1 Watchdog register list

#### 16.3.2 Register Description

##### Watchdog Load Register (Address: 0x40008000)

R/W	Bit 31	-----	Bit 0
W		LOAD[31:0]	
R			
Reset		0xFFFFFFFF	

**LOAD [31:0]:** The minimum valid value for WDOGLOAD is 1.

### WDOGVALUE (Watchdog Value Register)

Address: 0x40008004

R/W	Bit 31	-----	Bit 0
W	VALUE[31:0]		
R			
Reset	0xFFFFFFFF		

**VALUE[31:0]:** Current count value

### Watchdog CTRL Register (Address: 0x40008008)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--						RESTEN	INTEN
R								
Reset	0	0	0	0	0	0	0	0

**RESTEN:** Enable watchdog reset.

**[0]:** Watchdog reset disable

**[1]:** Watchdog reset enable

**INTEN:** Watchdog counter and interrupt enable.

**[0]:** Disable counter and interrupt

**[1]:** Enable counter and interrupt

### Watchdog Interrupt Clean Register (Address: 0x4000800C)

R/W	Bit 32	-----	Bit 0
W	INTCLR		
R			
Reset	0x00000000		

**INTCLR:** A write of any value to clear the watchdog interrupt, and reloads the counter from the value in LOAD.

### Watchdog RAWINTSTAT Register (Address: 0x40008010)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--							--
R								RAWINTSTAT
Reset	0	0	0	0	0	0	0	0

**RAWINTSTAT:** Raw Watchdog Interrupt Raw interrupt status from the counter.

### Watchdog MASKINTSTAT Register (Address: 0x40008014)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--							--
R								MASKINTSTAT
Reset	0	0	0	0	0	0	0	0

**MASKINTSTAT:** Watchdog Interrupt Enabled interrupt status from the counter.

### Watchdog LOCK Register (Address: 0x40008C00)

R/W	Bit31	-----	Bit 0
W	LOCK[31:0]		
R			
Reset	0x0000 0000		

**LOCK:** 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. (Default)

**[1]:** Write access to all other registers is disabled.

### 16.4 FUNCTION DESCRIPTION

#### 16.4.1 Watchdog Interrupt

The interrupt of Watchdog is NMI (Non Maskable Interrupt). It's always working. Please refer to Table 11.1 Exception Types..

Watchdog uses two control bit INTEN and RESTEN. When INTEN=1, the watchdog counter start count-down and interrupt enable. When INTEN=0, the watchdog counter stop count-down and interrupt disable.

RAWINTSTAT=1 when VALUE[31:0] count-down to 0.

MASKINTSTAT=1 when VALUE[31:0] count-down to 0 and RESTEN=1.

#### 16.4.2 Watchdog Reset

Watchdog reset will be occurred when VALUE[31:0] count-down to 0 and MASKINTSTAT=1.

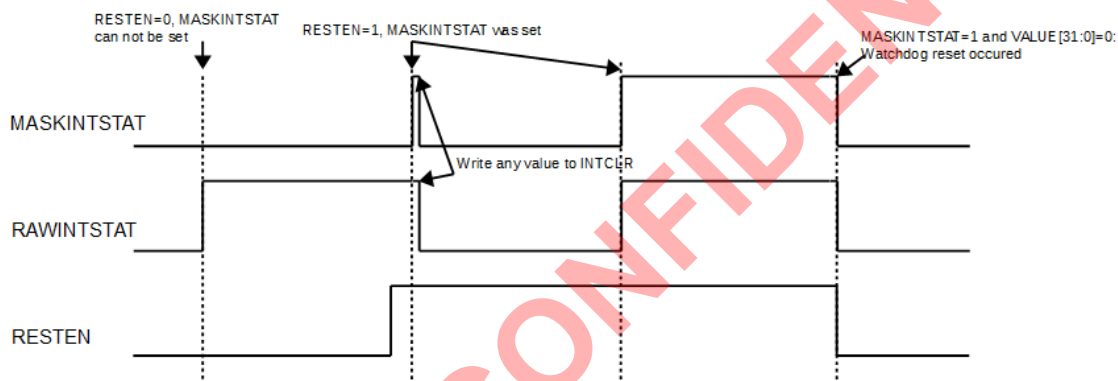


Figure 16.2 A example of watchdog reset

### 16.5 PROCEDURE

#### 16.5.1 Watchdog Enable

1. Unlock watchdog access, Write 0x1ACCE551 to watchdog lock register
2. Set watchdog load value
3. Enable INTEN=1.
4. Lock watchdog access, Write 0 to watchdog lock register
5. Waiting for watchdog interrupt to occur.
6. Unlock watchdog access, Write 0x1ACCE551 to watchdog lock register.
7. Write 0 to the Watchdog INTCLR register clears the watchdog interrupt, and reloads the counter from the value in watchdog LOAD value.
8. Lock watchdog access, Write 0 to watchdog lock register
9. Repeat step 5~8

#### 16.5.2 Watchdog Reset Enable

1. Unlock watchdog access, Write 0x1ACCE551 to watchdog lock register
2. Set watchdog load value
3. Enable INTEN=1 and RESTEN=1.
4. Lock watchdog access, Write 0 to watchdog lock register
5. Waiting for watchdog interrupt to occur.
6. Unlock watchdog access, Write 0x1ACCE551 to watchdog lock register.
7. Write 0 to the Watchdog INTCLR register clears the watchdog interrupt, and reloads the counter from the value in watchdog LOAD value.
8. Lock watchdog access, Write 0 to watchdog lock register
9. Repeat step 5~8

Watchdog timer that automatically generates a system reset if the main program neglects to periodically service it.



### 17. RTC (Real Time Counter)

RTC can use 32KHz or 32.768KHz to clock source. It is a simple and useful design to count week, hours, min, sec and manual count. It also has an alarm set by hour and min.

#### 17.1 FEATURE

- Support 32KHz and 32.768KHz clock source
- Many interrupt: 10ms, 0.5sec, 1sec, alarm (by hour and min) and manual count (by 10ms).

#### 17.2 BLOCK DIAGRAM

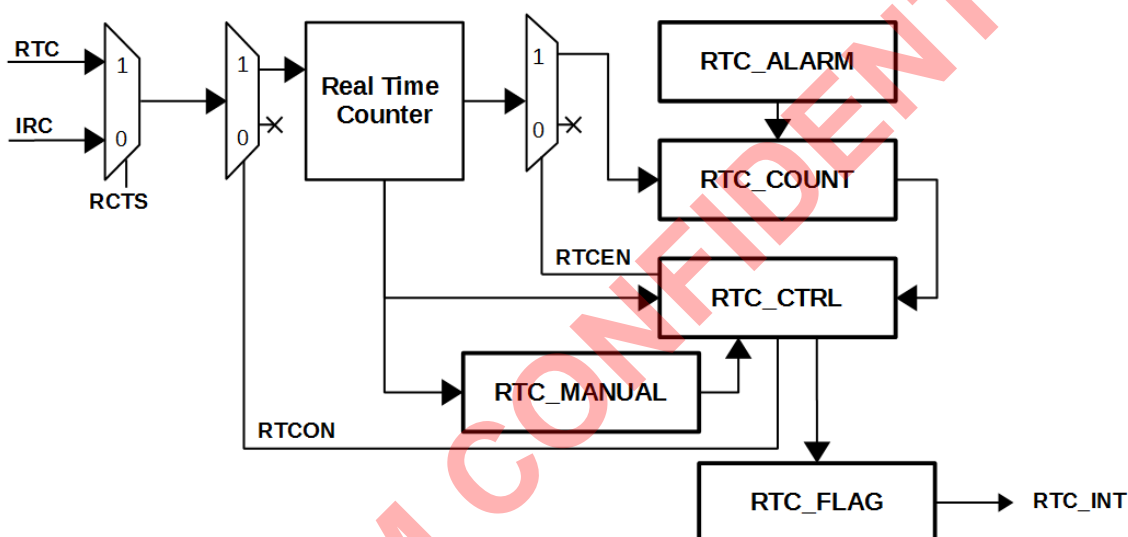


Figure 17.1 RTC Block Diagram

#### 17.3 REGISTER

##### 17.3.1 Register List

Address	Name	DESCRIPTION
0x50005000	RTC_COUNT	RTC Count Register
0x50005004	RTC_ALARM	RTC Alarm setting Register
0x50005008	RTC_CTRL	RTC Control Register
0x5000500C	RTC_FLAG	RTC Flag Register
0x50005010	RTC_MANUAL	RTC Manual count Register

Table 17.1 RTC register list

### 17.3.2 Register Description

#### RTC Count Register (Address: 0x50005000)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	WEEK[2:0]			HOUR[4:0]				
R	CWEEK[2:0]			CHOUR[4:0]				
Reset	0	0	0	0	0	0	0	0
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W	--		MIN[5:0]					
R	--		CMIN[5:0]					
Reset	0	0	0	0	0	0	0	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	--		SEC[5:0]					
R	--		CSEC[5:0]					
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--		--					
R	--		CTMS[6:0]					
Reset	0	0	0	0	0	0	0	0

**WEEK[2:0]:** Write WEEK[2:0] to set CWEEK[2:0]. The range of WEEK[2:0] is 0~6.

**CWEEK[2:0]:** The count value of week. When CWEEK[2:0] count up to 7, CWEEK[2:0] will reset to 0.

**HOUR[4:0]:** Write HOUR[5:0] to set CHOUR[5:0]. The range of HOUR[5:0] is 0~23.

**CHOUR[4:0]:** The count value of hour. When CHOUR[5:0] count up to 24, CHOUR[5:0] will reset to 0 and count up CWEEK[2:0] 1 times.

**MIN[5:0]:** Write MIN[5:0] to set CMIN[5:0]. The range of MIN[5:0] is 0~59.

**CMIN[5:0]:** The count value of min. When CMIN[5:0] count up to 60, CMIN[5:0] will reset to 0 and count up CHOUR[4:0] 1 times.

**SEC[5:0]:** Write SEC[5:0] to set CSEC[5:0]. The range of SEC[5:0] is 0~59.

**CSEC[5:0]:** The count value of sec. When CSEC[5:0] count up to 60, CSEC[5:0] will reset to 0 and count up CMIN[5:0] 1 times.

**CTMS[6:0]:** The count value of ten-ms. CTMS[6:0] will reset to 0 when write any value to RTC\_COUNT. CMS[6:0] will count up 1 times when RTC count 10ms. CTMS[6:0] can't set by user.

#### RTC Alarm Register (Address: 0x50005004)

R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	--			ALARM_HOUR[4:0]				
R	--			ALARM_HOUR[4:0]				
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--		ALARM_MIN[5:0]					
R	--		ALARM_MIN[5:0]					
Reset	0	0	0	0	0	0	0	0

**ALARM\_HOUR[4:0]:** Set alarm hour count

**ALARM\_MIN[5:0]:** Set alarm min count

Alarm interrupt will occur when first time that CHOUR[4:0] equal ALARM\_HOUR[4:0] and CMIN[5:0] equal ALARM\_MIN[5:0].

### RTC Control Register (Address: 0x50005008)

R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	--							MCNTEN
R	--							MCNTEN
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	CLKSEL	RTCON	INTMCNTIE	INT10MSIE	INT05SIE	INT1SIE	ALMIE	RTCEN
R	CLKSEL	RTCON	INTMCNTIE	INT10MSIE	INT05SIE	INT1SIE	ALMIE	RTCEN
Reset	0	0	0	0	0	0	0	0

**MCNTEN:** Manual Count Enable

[0]: Manual Count disable.

[1]: Manual Count enable.

**CLKSEL:** RTC clock select

[0]: RTC clock source is 32KHz

[1]: RTC clock source is 32.768KHz

**RTCON:** RTC clock on

[0]: RTC stop count.

[1]: RTC start count.

**INTMCNTIE:** Manual Count interrupt enable

[0]: Disable

[1]: Enable

**INT10MSIE:** 10ms count interrupt enable

[0]: Disable

[1]: Enable

**INT05SIE:** 0.5sec count interrupt enable

[0]: Disable

[1]: Enable

**INT1SIE:** 1sec count interrupt enable

[0]: Disable

[1]: Enable

**ALMIE:** Alarm interrupt enable

[0]: Disable

[1]: Enable

**RTCEN:** RTC\_COUNT enable.

[0]: RTC Disable. RTC\_COUNT, RTC\_MANUAL count stop.

[1]: RTC Enable. RTC\_COUNT, RTC\_MANUAL count start.

### RTC Flag Register (Address: 0x5000500C)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--	--	CLRMCNTIF	CLRTMSIF	CLRHSECIF	CLRSECIF	CLRALMIF	CLRINTIF
R	--	--	MCNTIF	TMSIF	HSECIF	SECIF	ALMIF	INTIF
Reset	0	0	0	0	0	0	0	0

**CLRMCNTIF:** Clear Manual Count interrupt Flag

[0]: No effect

[1]: Clear MCNTF

**MCNTIF:** Manual Count interrupt Flag

[0]: Manual Count interrupt not occur

[1]: Manual Count interrupt was occurred

**CLRTMSIF:** Clear 10-ms interrupt Flag

[0]: No effect

[1]: Clear TMSIF

**TMSIF:** 10ms interrupt Flag

[0]: RTC 10ms interrupt not occur

[1]: RTC 10ms interrupt was occurred

**CLRHSECIF:** Clear half-sec (0.5sec) interrupt Flag

[0]: No effect

[1]: Clear HSECIF

**HSECIF:** Half-sec (0.5sec) interrupt Flag

[0]: RTC half-sec interrupt not occur

[1]: RTC half-sec interrupt was occurred

**CLRSECIF:** Clear sec interrupt Flag

[0]: No effect

[1]: Clear SECIF

**SECIF:** 1 sec interrupt Flag

[0]: RTC 1 sec interrupt not occur

[1]: RTC 1 sec interrupt was occurred

**CLRALMIF:** Clear alarm interrupt Flag

[0]: No effect

[1]: Clear ALMIF

**ALMIF:** Alarm interrupt Flag

[0]: RTC alarm interrupt not occur

[1]: RTC alarm interrupt was occurred

**CLRINTIF:** Clear interrupt Flag

[0]: No effect

[1]: Clear INTIF

**INTIF:** Interrupt Flag

[0]: RTC any interrupt not occur

[1]: RTC any interrupt was occurred

NOTE: User need clear interrupt flag when interrupt occurred.

### RTC Manual count Register (Address: 0x50005010)

R/W	Bit 15	-----	Bit 0
W	MANUAL_COUNT[15:0]		
R	MANUAL_COUNT[15:0]		
Reset	0x00000000		

**MANUAL\_COUNT[15:0]:** MANUAL\_COUNT[15:0] will reset to 0 when MANUAL\_COUNT[15:0] equal MANUAL\_COUNT[15:0].

**MANUAL\_COUNT[15:0]:** The current count of manual counter. MANUAL\_COUNT[15:0] will count up 1 times when RTC count 10ms and MCNTEN=1. MANUAL\_COUNT[15:0] will reset to 0 when MCNTEN=0. Writing MANUAL\_COUNT[15:0] was not effect to MANUAL\_COUNT[15:0].

## 17.4 FUNCTION DESCRIPTION

### 17.4.1 Write RTC Register

When user writes any RTC register, the register needs RTC clock to update. Max. wait time is 1 / (RTC clock source frequency).

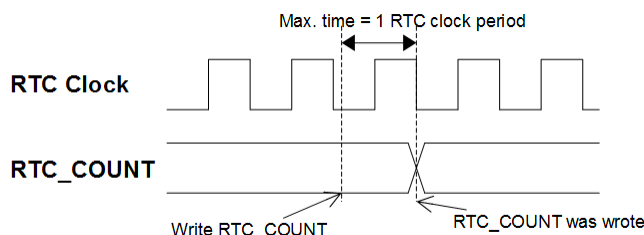


Figure 17.2 The timing for write RTC\_COUNT

### 17.4.2 Bit RTCON

RTC start count after RTCON=1. The 10ms interrupt, 0.5sec interrupt, 1sec interrupt and manual count interrupt will occur if interrupt enable.

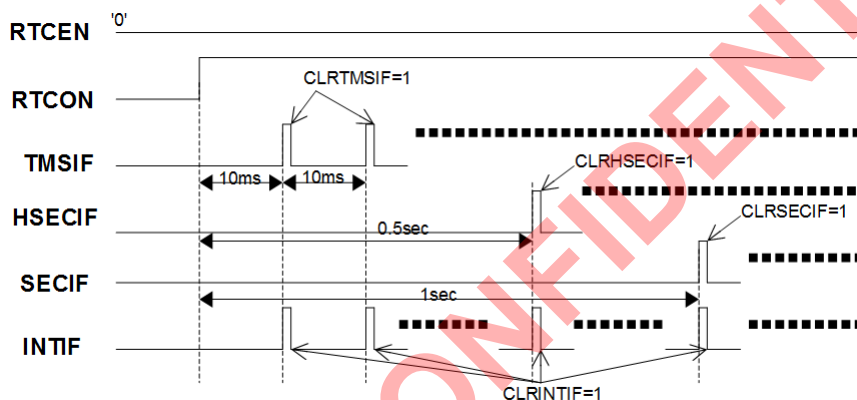


Figure 17.3 An example for RTCON=1, RTCEN=0 and all interrupt enable.

### 17.4.3 Bit RTCEN

RTC\_COUNT will count sec, min, hour and week after RTCEN=1 and RTCON=1. CTMS[6:0] will set to 0 when RTCEN=0.

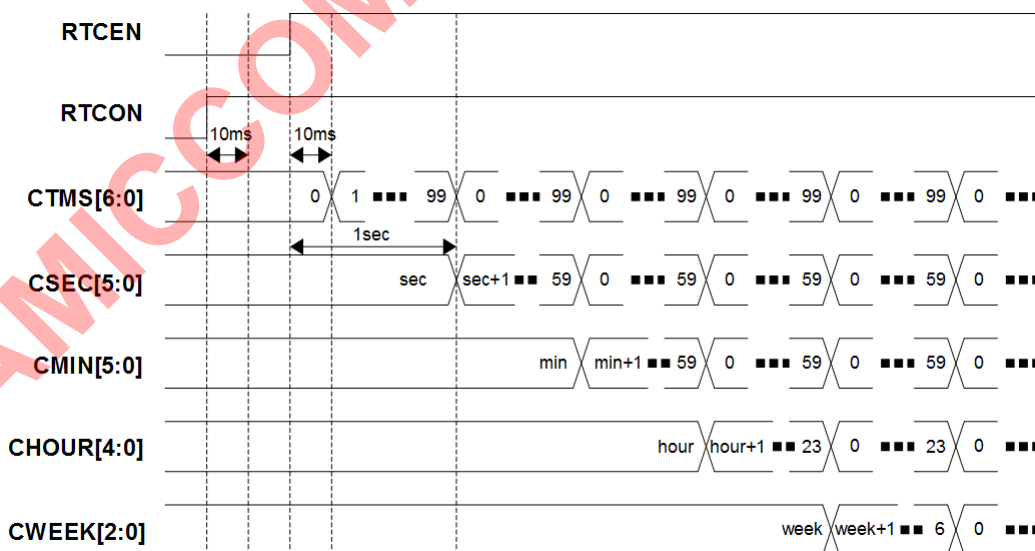


Figure 17.4 An example for RTCON=1, RTCEN=1, RTC\_COUNT will count up

### 17.4.4 Alarm

Register RTC\_ALARM can set hour and min. Alarm interrupt will occur when first time that hour and min of RTC\_COUNT match hour and min of RTC\_ALARM.

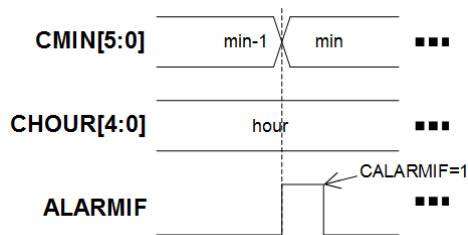


Figure 17.5 An example for alarm set "hour, min"

### 17.4.5 Manual Count

RTC build-in a 16bits manual count that clock source is RTC count 10ms. The interval of manual count is  $RTC\_MANUAL \times 10ms$ . Min interval is 10ms ( $RTC\_MANUAL=0x0001$ ). Max. interval is 655.36sec ( $RTC\_MANUAL = 0x0000$ ). Manual count start count when bit MCNTEN = 1. RTCEN=0 or 1 is no effect to manual count action.

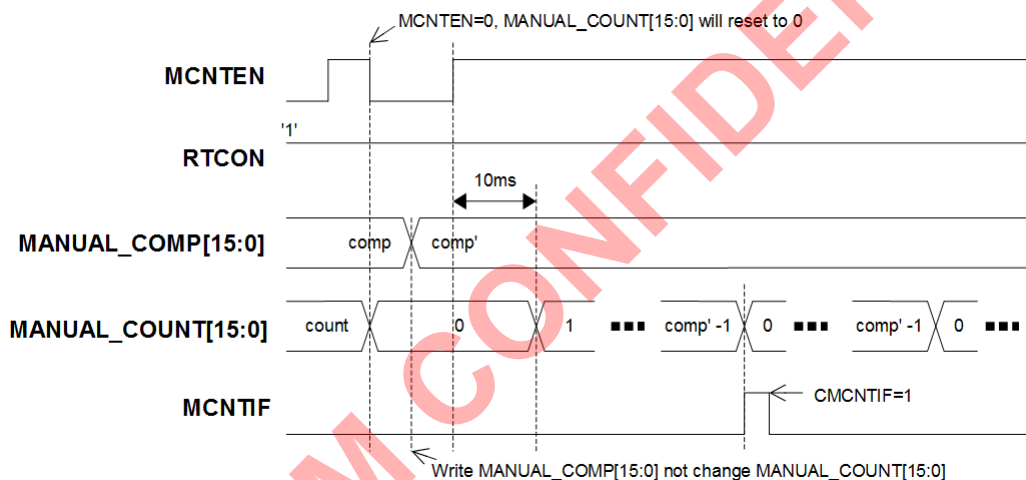


Figure 17.6 An example for manual count.

### 17.4.6 Interrupt

The NVIC IRQn of RTC is 18. Please refer to Table 11.2 Interrupt Map Vector Table.

There is a sample code for RTC handler shown as below:

```
void RTC_Handler(void)
{
    uint32_t flag;
    flag = RTC->FLAG;           // store FLAG
    RTC->FLAG = flag;           // clean all FLAG
    if(flag & RTC_FLAG_MANUALCOUNT_Msk) // manual count interrupt
    {
    }
    if(flag & RTC_FLAG_TMS_Msk)    // 10ms interrupt
    {
    }
    if(flag & RTC_FLAG_HSEC_Msk)   // 0.5sec interrupt
    {
    }
    if(flag & RTC_FLAG_SEC_Msk)    // 1sec interrupt
    {
    }
    if(flag & RTC_FLAG_ALARM_Msk)  // Alarm interrupt
    {
    }
    if(flag & RTC_FLAG_RTC_Msk)    // When any RTC interrupt occurred
    {
    }
}
```

### 18. PWM

PWM (Pulse Width Modulation) is the method to control analog circuits with a processor's digital outputs. It generates variable pulse frequency and the duty cycle of the signal with digital means. Through PWM function, it can be used for a wide variety of control applications.

#### 18.1 FEATURES

- 4 channels PWM output
- Programmable PWM frequency (1/2, 1/4, 1/8, 1/16, 1/32, 1/64 of MCU clock and that of RTC clock)
- Programmable PWM duty cycle

#### 18.2 PINS DESCRIPTION

PIN	GPIO	TYPE	DESCRIPTION
PWM0	P0_20	OUTPUT	PWM channels 0
PWM1	P0_21	OUTPUT	PWM channels 1
PWM2	P0_10	OUTPUT	PWM channels 2
PWM3	P0_11	OUTPUT	PWM channels 3

Table 18.1 PWM pins description

#### 18.3 BLOCK DIAGRAM

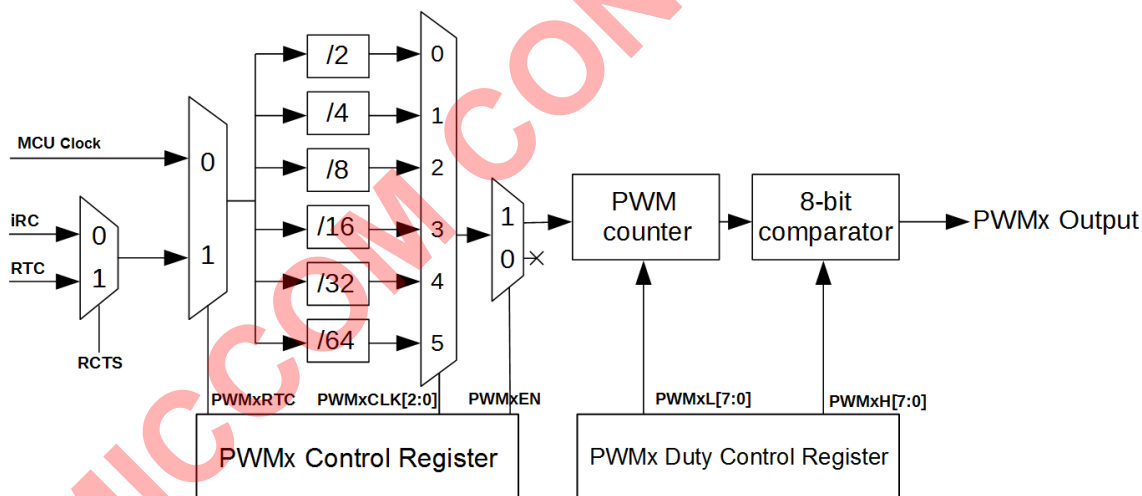


Figure 18.1 PWMx Block Diagram

#### 18.4 REGISTER

##### 18.4.1 Register list

Base Address	DESCRIPTION
0x50004000	PWM0 Base address
0x50004100	PWM1 Base address
0x50004200	PWM2 Base address
0x50004300	PWM3 Base address

Table 18.2 Base address of each PWM

offset	Name	DESCRIPTION
0x000	PWMxCR	PWMx Control Register
0x004	PWMxDCR	PWMx Duty Control Register



Table 18.3 PWM Register List

### 18.4.2 Register Description

#### PWMx Control Register (Offset: 0x000)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	PWMxEN	--			PWMxRTC	PWMxCLK[2:0]		
R								
Reset	0	0	0	0	0	0	0	0

**PWMxEN:** PWM Channel x Enable,

[0]: Disable.

[1]: Enable.

**PWxRTC:** PWM Channel x Clock Source select,

[0]: MCU Clock.

[1]: RTC clock.

**PWxCLK[2:0]:** PWM Channel x 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

[110]: Not allowed to use.

[111]: Not allowed to use.

#### PWMx Duty Control Register ( Offset: 0x004)

R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	PWMxH [7:0]							
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	PWMxL [7:0]							
R								
Reset	0	0	0	0	0	0	0	0

**PWMxH[7:0]:** PWMx output HIGH register

**PWMxL[7:0]:** PWMx frequency setting register

### 18.5 FUNCTION DESCRIPTION

A3107M0 has four channels PWM output. Each channel PWM has an 8-bit counter with comparator, a control register (PWMxCR) and two setting registers (PWMxH and PWMxL). User can select clock source by setting PWMxCR. It divide MCU clock and RTC clock to 2,4,8,16,32,64 by setting PWMxCLK. Enable PWM output and function by setting PWMxEN = 1; otherwise disable PWM output and function by setting PWMxEN = 0. When user sets PWMxEN=1, it outputs LOW single and loads the PWMxL to itself. When the up 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 equations

$$\text{Pulse frequency} = \text{Clock Source} / 2^{(\text{PWMxCLK}+1)} / (256-\text{PWMxL})$$

$$\text{Duty cycle} = (256-\text{PWMxH}) / (256-\text{PWMxL})$$

Noted: PWMxH must be larger than PWMxL. Otherwise, PWM output is always HIGH.  
List some example setting in Table 18.4 Some example setting.

MCU Clock = 16MHz RTC Clock = 32.768KHz				
Offset 0x000	Offset 0x004	Clock Source	PWM Frequency	Duty Cycle
0x80	0xFFFE	16MHz	4MHz	50%
0x80	0xFFFD	16MHz	2.667KHz	33.3%
0x85	0xFFFE	16MHz	125KHz	50%
0x80	0x8000	16MHz	31.25KHz	50%
0x88	0x8000	32.768KHz	64Hz	50%
0x88	0xFFFD	32.768KHz	5.461KHz	33.3%
0x8D	0xFFFE	32.768KHz	256Hz	50%

Table 18.4 Some example setting

### 18.6 PROCEDURE

Below is the procedure to set PWM0 output (ex. PWM Frequency 4MHz, Duty Cycle 50%)

- Step10: Set A3107M0 in STBY, PLL, TX or RX MODE.
- Step11: Set Port 0 Output Enable Set Register (0x40010010), P0\_20=1.
- Step12: Set Port 0 ALTFUNCSET Register (0x40010018), P0\_20=1.
- Step13: Set PWM0CR (0x50004000), PWM0CLK[2:0]=0, and PWM0RTC=0.
- Step14: Set PWM0DCR (0x50004004), PWM0L [7:0]=254, and PWM0H [7:0]=255.
- Step15: Set PWM0CR (0x50004000), PWM0EN=1

PWM0(P0\_20) generates 4MHz Pulse frequency and 50% duty cycle of the signal.

Counter >= PWMxH : output high  
Counter < PWMxH : output low

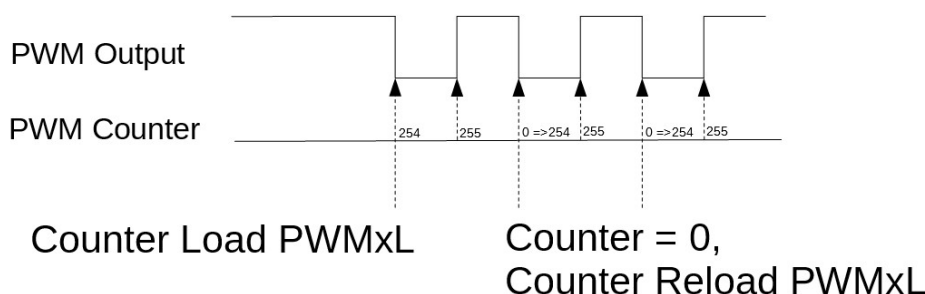
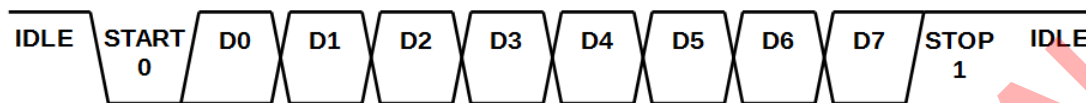


Figure 18.2 An example for PWM

### 19. UART

The UART implements a universal asynchronous receiver/transmitter function. It supports flexible baudrate generation. It's a simple design that only supports 8-bit communication without parity, and is fixed at one stop bit per configuration (N-8-1).



START: Start bit, always low  
D0~D7: Data bit  
STOP: Stop bit, always high

Figure 19.1 UART transmission format.

#### 19.1 FEATURE

- Three UARTs (RX/TX)
- Support format: 8 bit data, 1 start, 1 stop bit and no parities (N-8-1).
- Baud rate derived from system clock.

#### 19.2 PINS DESCRIPTION

PIN	GPIO	TYPE	DESCRIPTION
UART0_RX	P0_16	INPUT	UART0 Receiver Input pin
UART0_TX	P0_17	OUTPUT	UART0 Transmitter Output pin
UART1_RX	P0_18	INPUT	UART1 Receiver Input pin
UART1_TX	P0_19	OUTPUT	UART1 Transmitter Output pin
UART2_RX	P0_20	INPUT	UART2 Receiver Input pin
UART2_TX	P0_21	OUTPUT	UART2 Transmitter Output pin

#### 19.3 BLOCK DIAGRAM

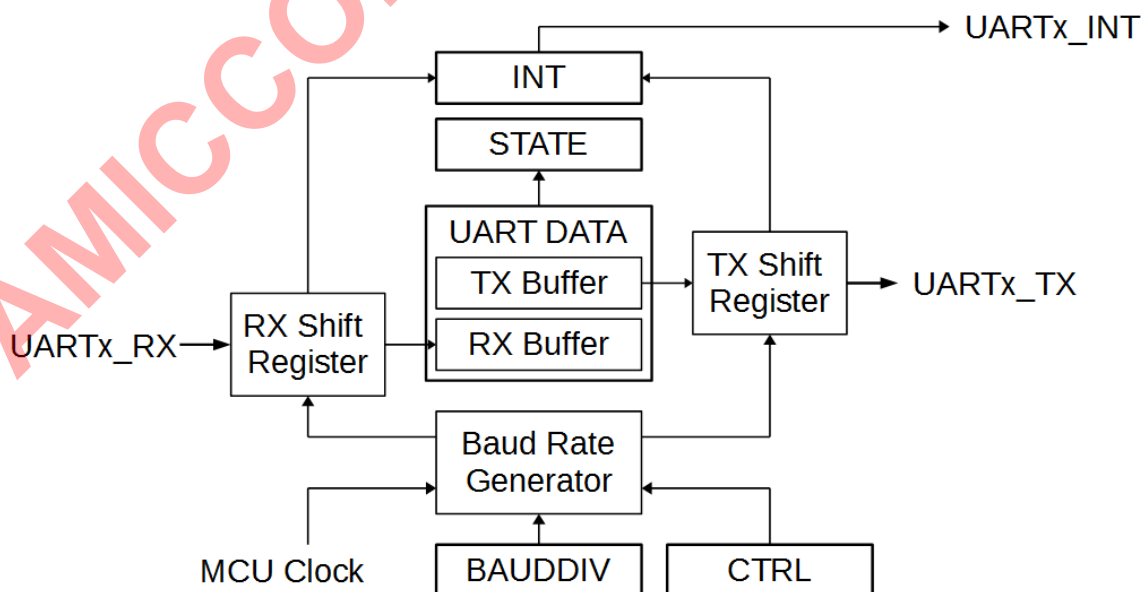


Figure 19.2 UART block diagram

### 19.4 REGISTER

#### 19.4.1 Base Address List

Base Address	DESCRIPTION
0x40004000	UART0 Base Address
0x40005000	UART1 Base Address
0x40006000	UART2 Base Address

Table 19.1 Base address list

#### 19.4.2 Register List

Offset	Name	DESCRIPTION
0x000	UART_DATA	UART Data Register
0x004	UART_STATE	UART Status Register
0x008	UART_CTRL	UART Control Register
0x00C	UART_INT	UART interrupt state and clear Register
0x010	UART_BAUDDIV	UART Baud rate divider register

Table 19.2 Register list

#### 19.4.3 Register Description

##### UART Data Register (Offset: 0x000)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	DATA[7:0]							
R								
Reset	--	--	--	--	--	--	--	--

**DATA[7:0]:** UART Data. Write data to TX Buffer or read data from RX Buffer.

##### UART Status Register (Offset: 0x004)

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

**STATE[3]:** RX buffer overrun.

**[0]:** RX Buffer does not overrun.

**[1]:** RX Buffer has overrun. Write 1 to clean this bit.

**STATE[2]:** TX buffer overrun.

**[0]:** TX Buffer does not overrun.

**[1]:** TX Buffer has overrun. Write 1 to clean this bit.

**STATE[1]:** RX buffer full.

**[0]:** RX buffer not full.

**[1]:** RX buffer full.

**STATE[0]:** TX buffer full.

**[0]:** TX buffer not full.

**[1]:** TX buffer full.

##### UART Control Register (Offset: 0x008)

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

**CTRL[5]** : RX overrun interrupt enable.  
**[0]**: Disable UART RX overrun interrupt.  
**[1]**: Enable UART RX overrun interrupt.

**CTRL[4]** : TX overrun interrupt enable.  
**[0]**: Disable UART TX overrun interrupt.  
**[1]**: Enable UART TX overrun interrupt.

**CTRL[3]** : RX interrupt enable.  
**[0]**: Disable UART RX interrupt.  
**[1]**: Enable UART RX interrupt.

**CTRL[2]** : TX interrupt enable.  
**[0]**: Disable UART TX interrupt.  
**[1]**: Enable UART TX interrupt.

**CTRL[1]** : RX enable.  
**[0]**: Disable UART RX.  
**[1]**: Enable UART RX.

**CTRL[0]** : TX enable.  
**[0]**: Disable UART TX.  
**[1]**: Enable UART TX.

### UART Interrupt Register (Offset: 0x00C )

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

**INT[3]** : RX overrun interrupt.  
**[0]**: RX overrun interrupt does not occur.  
**[1]**: RX overrun interrupt has occurred. Write 1 to clear this bit.

**INT[2]** : TX overrun interrupt.  
**[0]**: TX overrun interrupt does not occur.  
**[1]**: TX overrun interrupt has occurred. Write 1 to clear this bit.

**INT[1]** : RX interrupt.  
**[0]**: RX interrupt does not occur.  
**[1]**: RX interrupt has occurred. Write 1 to clear this bit.

**INT[0]** : TX interrupt.  
**[0]**: TX interrupt does not occur.  
**[1]**: TX interrupt has occurred. Write 1 to clear this bit.

### UART Baud rate Divider Register (Offset: 0x010)

R/W	Bit 31	-----	Bit 20	Bit 19	-----	Bit 0
W	--			BAUDDIV[19:0]		
R	--			BAUDDIV[19:0]		
Reset	0x000			0x00000		

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

### 19.5 FUNCTION DESCRIPTION

#### 19.5.1 I/O pin setting

To communicate with an external serial interface, the internal UARTx has two external pins known as UARTx\_TX and UARTx\_RX. The UARTx\_TX and UARTx\_RX pins are the UARTx transmitter and receiver pins respectively. The UARTx\_TX and UARTx\_RX pin function should first be selected by the corresponding pin-shared function selection register (ALTFUNCSET Register) before the UARTx function is used. If set, will automatically setup the UARTx\_TX and UARTx\_RX pins to their respective TX output and RX input conditions.

#### 19.5.2 UART Baud Rate

The baud rate is depended on MCU clock frequency ( $F_{mcu}$ ), the equation is

$$\text{Baud Rate} = F_{mcu} / \text{BAUDDIV}[19:0]$$

Usually, we choose the baud rate to find the BAUDDIV[19:0]. The equation is

$$\text{BAUDDIV}[19:0] = F_{mcu} / (\text{Baud rate})$$

For example, if the system clock frequency is 16MHz and the required baud rate is 9600bps, then the BAUDDIV[19:0] = 16,000,000/9,600 = 1666 or 1667. And the actual baud rate is 16,000,000/1666 = 9603.84bps (+0.04% with 9600bps) or 9598.08bps (-0.02% with 9600bps).

#### 19.5.3 UART TX state and interrupt

The data of TX Buffer will send to TX Shift Register automatic if TX Shift Register empty. There has 1 data in TX Shift Register to transmit when STATE[0]=0. The STATE[2]=1 when user write data to TX Buffer and STATE[0]=1. INT[0]=1 when TX Shift Register transmit complete.

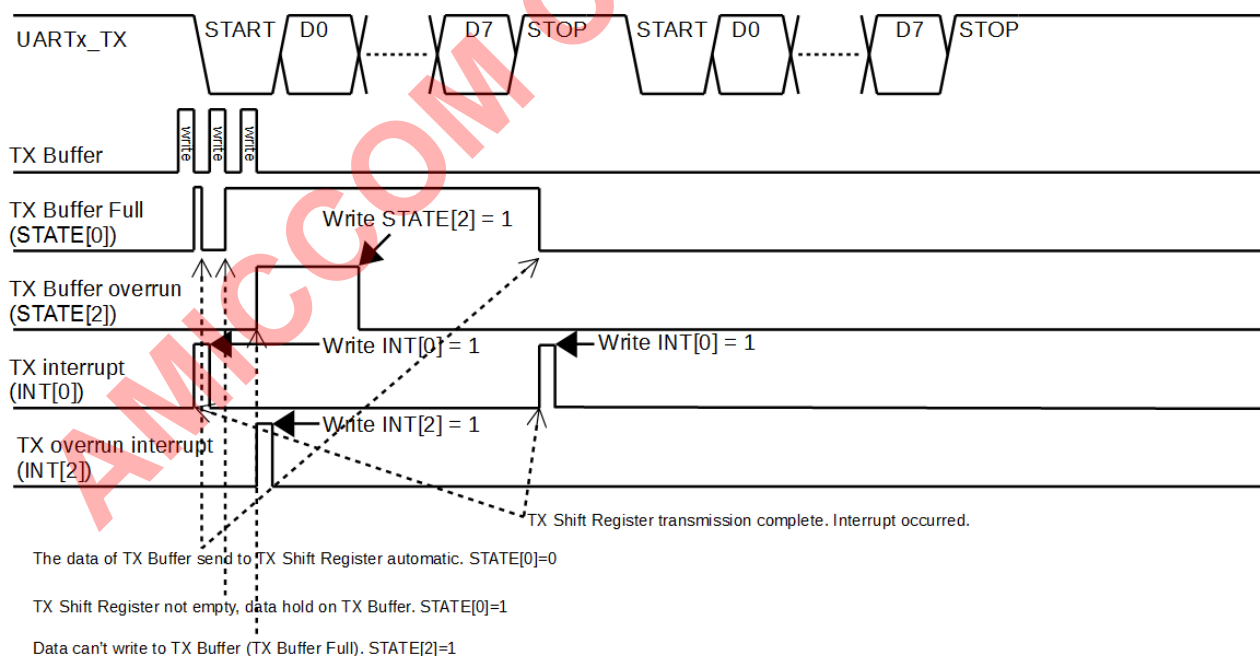


Figure 19.3 A example for UART TX write 3 bytes data

### 19.5.4 UART RX state and interrupt

When data send to UARTx\_RX, data will store in RX Shift Register. This data will send to RX Buffer automatic when RX Shift Register has received one data. And RX interrupt will occur. User need read data from RX Buffer when STATE[1]=1. If RX Shift Register has received one data and STATE[1]=1 then STATE[3]=1 and data will not send to RX Buffer.

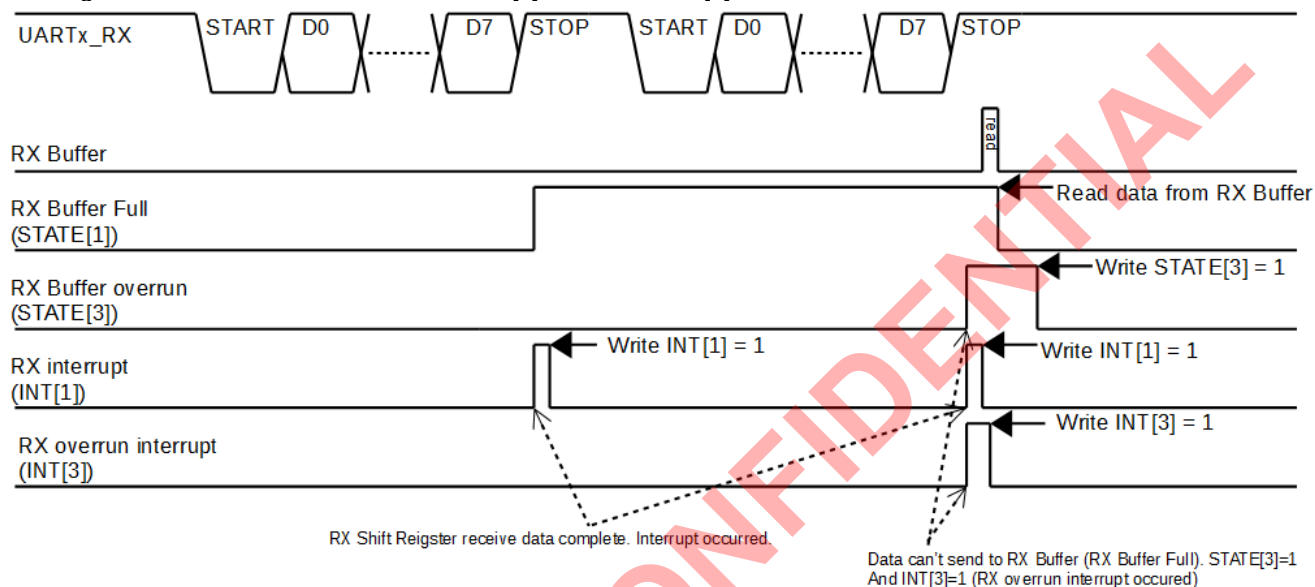


Figure 19.4 A example for UART RX receive data

### 19.6 PROCEDURE

#### UARTx TX :

1. Set wanted UART I/O port.
2. Set Baud Rate.
3. Enable UART TX and enable TX interrupt.
4. Fill in data to UART DATA Register to transmit data.
5. Waiting for TX interrupt to occur and interrupt clear.
6. Repeat step 4~5.

#### UARTx RX :

1. Set wanted UART I/O port.
2. Set Baud Rate.
3. Enable UART RX and enable RX interrupt.
4. Waiting for RX interrupt to occur and interrupt clear.
5. Read out data from UART DATA Register.
6. Repeat step 4~5.

## 20. I<sup>2</sup>C Interface

The I<sup>2</sup>C bus provides bidirectional data transfer through a 2-wire design, a serial data line (I<sup>2</sup>C\_SDA) and a serial clock line (I<sup>2</sup>C\_SCL).

## 20.1 FEATURE

- Conforms to v2.1 of the I<sup>2</sup>C specification (published by Philips Semiconductor)
- Master transmit / receive
- Slave transmit / receive
- Flexible transmission speed: Standard (up to 100 Kb/s), Fast (up to 400Kb/s) and Fast plus(1MHz)
- Multi-master systems supported
- Supports 7-bits and 10-bits addressing on the I<sup>2</sup>C bus
- Interrupt generation
- Allows operation from a wide range of input clock frequencies

## 20.2 PINS DESCRIPTION

I<sup>2</sup>C bus uses two signals: I<sup>2</sup>C\_SCL and I<sup>2</sup>C\_SDA. The alternate function is Port 0.4 and Port 0.5 or Port0.10 and Port0.11 for I<sup>2</sup>CBSSEL. User need set ALTFUNCSET (0x40010018) for GPIO0 to setup the PIN function.

PIN	GPIO	TYPE	DESCRIPTION
I <sup>2</sup> C_SCL	P0_04	INPUT / OUTPUT	I <sup>2</sup> C clock input/output (I <sup>2</sup> CBSEL=0)
I <sup>2</sup> C_SDA	P0_05	INPUT / OUTPUT	I <sup>2</sup> C data input/output (I <sup>2</sup> CBSEL=0)
I <sup>2</sup> C_SCL	P0_10	INPUT / OUTPUT	I <sup>2</sup> C clock input/output (I <sup>2</sup> CBSEL=1)
I <sup>2</sup> C_SDA	P0_11	INPUT / OUTPUT	I <sup>2</sup> C data input/output (I <sup>2</sup> CBSEL=1)

Table 20.1 I<sup>2</sup>C interface pins description

### 20.3 BLOCK DIAGRAM

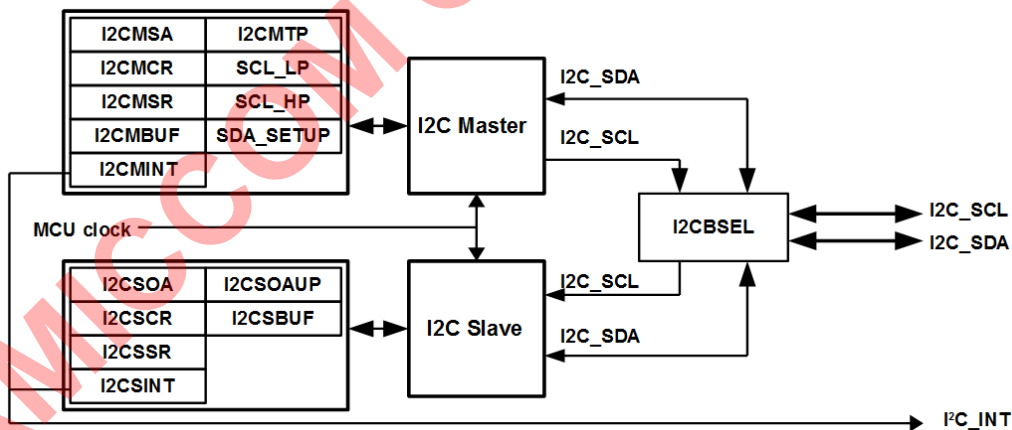


Figure 20.1 I<sup>2</sup>C Interface Block Diagram

## 20.4 REGISTER

### 20.4.1 Register List

Address	Name	DESCRIPTION
0x50003000	I <sup>2</sup> CMSA	I <sup>2</sup> C Master Slave address
0x50003004	I <sup>2</sup> CMSR / I <sup>2</sup> CMSR	I <sup>2</sup> C Master Control Register / I <sup>2</sup> C Master Status Register
0x50003008	I <sup>2</sup> CMBUF	I <sup>2</sup> C Master transmitted/Received data Buffer Register
0x5000300C	I <sup>2</sup> CMTP	I <sup>2</sup> C Master Timer Period Register
0x50003010	SCL LP	I <sup>2</sup> C SCL Low Period Register



0x50003014	SCL_HP	I <sup>2</sup> C_SCL High Period Register
0x50003018	SDA_SETUP	I <sup>2</sup> C_DAT Setup Period Register
0x5000301C	I <sup>2</sup> CMINT	I <sup>2</sup> C Master Interrupt Register
0x50003020	I <sup>2</sup> CBSEL	I <sup>2</sup> C Bus Select Register
0x50003800	I <sup>2</sup> CSOA	I <sup>2</sup> C Slave Own Address Register
0x50003804	I <sup>2</sup> CSCR / I <sup>2</sup> CSSR	I <sup>2</sup> C Slave Control Register / I <sup>2</sup> C Slave Status Register
0x50003808	I <sup>2</sup> CSBUF	I <sup>2</sup> C Slave Transmitted/Received data Buffer Register
0x5000380C	I <sup>2</sup> CSOAUP	I <sup>2</sup> C Slave Own Address UP Register
0x50003810	I <sup>2</sup> CSINT	I <sup>2</sup> C Slave Interrupt register

Table 20.2 I<sup>2</sup>C interface Register List

### 20.4.2 I<sup>2</sup>C Master Register Description

#### I<sup>2</sup>C Master Slave address (Address: 0x50003000)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	SA[6:0]							RS
R								
Reset	0	0	0	0	0	0	0	0

**SA[6:0]:** Slave Address.

**RS:** Receive or Send in START condition. This bit work with START, RUN or HS, RUN.

**[0]:** Transmitter

**[1]:** Receiver

#### I<sup>2</sup>C Master Control Register (Address: 0x50003004)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	RSTB	SLRST	--	HS	ACK	STOP	START	RUN
R	--	--	--	--	--	--	--	--
Reset	0	0	0	0	0	0	0	0

**RSTB:** Reset Bit for I<sup>2</sup>C Master.

**[0]:** No effect.

**[1]:** Reset I<sup>2</sup>C controller.

**SLRST:** Slave Reset.

**[0]:** No effect.

**[1]:** Reset slaves connected to I<sup>2</sup>C bus by generating 9 I<sup>2</sup>C\_SCK clocks followed by STOP. This bit need with RUN to work.

**HS:** Transmission speed switch to High-speed.

**[0]:** No effect.

**[1]:** Send START follow by Slave Address and Switching to High-speed.

**ACK:** Master in Receive mode need to set this bit. This bit must work with RUN bit.

**[0]:** Read data follow by NAK.

**[1]:** Read data follow by ACK.

**STOP:** Send STOP and return to Idle mode, and transmission speed switch to Standard.

**[0]:** No effect.

**[1]:** Send STOP.

**START:** Send START follow by Slave Address and SEND or RECEIVE. This bit need with RUN and RS to work.

**[0]:** No effect.

**[1]:** Send START follow by Slave Address and SEND or RECEIVE.

**RUN:** This bit work with START, STOP, ACK and HS.

**[0]:** No effect.

**[1]:** Run a transmission.

#### I<sup>2</sup>C Master Slave Register (Address: 0x50003004)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--	--	--	--	--	--	--	--
R	--	BUS_BUSY	IDLE	ARB_LOST	DATA_ACK	--	ERROR	BUSY
Reset	0	0	1	0	0	0	0	0

**BUS\_BUSY:** This bit is set by START conditions and reset by STOP conditions.

[0]: Bus not busy.

[1]: Bus is busy.

**IDLE:** I<sup>2</sup>C Bus controller is in the idle state.

[0]: I<sup>2</sup>C Bus controller is not in idle state.

[1]: I<sup>2</sup>C Bus controller is in idle state.

**ARB\_LOST:** Due to the last operation I<sup>2</sup>C Bus controller lost the arbitration.

[0]: not arbitration lost.

[1]: has arbitration lost.

**DATA\_ACK:** The acknowledged of DATA.

[0]: DATA with NAK.

[1]: DATA with ACK.

**ERROR:** Due to the last operation an error occurred, including slave address wasn't acknowledged, transmitted data wasn't acknowledged, or I<sup>2</sup>C Bus controller lost the arbitration.

[0]: Normal.

[1]: Some error occurred.

**BUSY:** I<sup>2</sup>C Master is receiving, or transmitting data on the bus and other bits of I<sup>2</sup>CMSR are no valid.

[0]: I<sup>2</sup>C Master controller not busy.

[1]: I<sup>2</sup>C Master controller is busy.

### I<sup>2</sup>C Master transmitted data Buffer (Address: 0x50003008)

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

**D[7:0]:** I<sup>2</sup>C Master write SEND data or read RECEIVE data.

### I<sup>2</sup>C Master Timer Period (Address: 0x5000300C)

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

**MTP[6:0]:** I<sup>2</sup>C master timer period register. The range of MTP[6:0] is 1~63.

### I<sup>2</sup>C\_SCL Low Period (Address: 0x50003010)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--				SCL_LP[3:0]			
R	--				SCL_LP[3:0]			
Reset	0	0	0	0	0	1	0	0

**SCL\_LP[3:0]:** I<sup>2</sup>C master SCL low time period register. The range of SCL\_LP[3:0] is 1~14.

### I<sup>2</sup>C\_SCL High Period (Address: 0x50003014)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--				SCL_HP[3:0]			
R	--				SCL_HP[3:0]			
Reset	0	0	0	0	0	0	1	0

**SCL\_HP[3:0]:** I<sup>2</sup>C master SCL high time period register. The range of SCL\_HP[3:0] is 2~15.

### I<sup>2</sup>C\_DAT Setup Period (Address: 0x50003018)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--				SDA_SETUP[3:0]			
R	--				SDA_SETUP[3:0]			
Reset	0	0	0	0	0	0	1	0

**SDA\_SETUP[3:0]:** I<sup>2</sup>C Master SDA setup time register. The range of SDA\_SETUP[3:0] is 2~15.

### I<sup>2</sup>C Master Interrupt (Address: 0x5000301C)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--						I <sup>2</sup> CMIF	I <sup>2</sup> CMIE
R	--						I <sup>2</sup> CMIF	I <sup>2</sup> CMIE
Reset	0	0	0	0	0	0	0	0

**I<sup>2</sup>CMIF:** I<sup>2</sup>C MASTER MODULE interrupt flag.

[0]: No effect.

[1]: I<sup>2</sup>C Master Interrupt occurred. Write 1 to clean this bit.

**I<sup>2</sup>CMIE:** I<sup>2</sup>C Master interrupt enable.

[0]: Disable.

[1]: Enable.

### I<sup>2</sup>C Bus Select (Address: 0x50003020)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--							I2CBSEL
R	--							I2CBSEL
Reset	0	0	0	0	0	0	0	0

**I<sup>2</sup>CBSEL:** I<sup>2</sup>C Bus select.

[0]: I<sup>2</sup>C\_SCL connect to P0\_04 and I<sup>2</sup>C\_SCL connect to P0\_05.

[1]: I<sup>2</sup>C\_SCL connect to P0\_10 and I<sup>2</sup>C\_SCL connect to P0\_11.

## 20.4.3 I<sup>2</sup>C Slave Register Description

### I<sup>2</sup>C Slave Own Address (Address: 0x50003800)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--							
R	ADDR[6:0]							
Reset	0	0	0	0	0	0	0	0

**ADDR[6:0]:** Slave device 7bits own address.

### I<sup>2</sup>C Slave Control Register (Address: 0x50003804)

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

**RSTB:** I<sup>2</sup>C Slave controller reset.

[0]: No effect.

[1]: Reset I<sup>2</sup>C Slave controller.

**DA:** I<sup>2</sup>C module Device Active.

[0]: I<sup>2</sup>C Slave device inactive.

[1]: I<sup>2</sup>C Slave device active.

**RECFINCLR:** RECFIN clear.

[0]: No effect.

[1]: Clear RECFIN flag.

### SEDNFINCLR:

[0]: No effect.

[1]: Clear SENDFIN flag.

### I<sup>2</sup>C Slave Status Register (Address: 0x50003804)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--	--	--	--	--	--	--	--
R	--	DA	--	BUSACTIVE	RECFIN	SEDNFIN	TREQ	RREQ
Reset	0	0	0	0	0	0	0	0

**DA:** I<sup>2</sup>C slave Device Active.

[0]: I<sup>2</sup>C slave device inactive.

[1]: I<sup>2</sup>C slave device active.

**BUSACTIVE:** BUS ACTIVE.

[0]: Bus no any transmission.

[1]: Bus has any transmission.

**RECFIN:** Receive Finish.

[0]: No effect.

[1]: I<sup>2</sup>C Slave device receive finish. User need write 1 to RECFINCLR to clear this bit.

**SEDNFIN:** Send Finish.

[0]: No effect.

[1]: I<sup>2</sup>C Slave device send finish. User need write 1 to SENDFINCLR to clear this bit.

**TREQ:** Transmit Request.

[0]: No transmit request.

[1]: I<sup>2</sup>C Slave device is addressed as transmitter and requires data from host. User need write data to I2CBUF to clear this bit.

**RREQ:** Receive Request.

[0]: No receive request.

[1]: Receive request occurred. User need read data from I<sup>2</sup>CSBUF to clear this bit.

### I<sup>2</sup>C Slave Transmitted data Buffer (Address: 0x50003808)

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

**D[7:0]:** I<sup>2</sup>C Slave read data from D[7:0] when RREQ occurred, and write data to D[7:0] when TREQ occurred.

### I<sup>2</sup>C Slave Own Address UP (Address: 0x5000380C)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--				TEN_ADDR_EN	ADDR[9:7]		
R								
Reset	0	0	0	0	0	0	0	0

**TEN\_ADDR\_EN:** Ten bits address enable.

[0]: I<sup>2</sup>C Slave device using 7bits address.

[1]: I<sup>2</sup>C Slave device using 10bits address.

**ADDR[9:7]:** The higher 3 bits for I<sup>2</sup>C Slave device 10bits own address.

### I<sup>2</sup>C Slave Interrupt register (Address: 0x50003810)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--						I <sup>2</sup> CSIF	I <sup>2</sup> CSIE
R								
Reset	0	0	0	0	0	0	0	0

**I<sup>2</sup>C\_SIF:** I<sup>2</sup>C Slave device interrupt flag.

[0]: No effect.

[1]: I<sup>2</sup>C Slave device interrupt occurred. Write 1 to clear this bit.

**I<sup>2</sup>C\_SIE:** I<sup>2</sup>C Slave device interrupt enable.

[0]: Disable.

[1]: Enable.

## 20.5 FUNCTION DESCRIPTION

### 20.5.1 I<sup>2</sup>C Transmission Format

I<sup>2</sup>C transmission uses 2 signals: I<sup>2</sup>C\_SCL and I<sup>2</sup>C\_SDA. The transmission format consists of START, STOP, DATA and Acknowledge. These 2 signals are the open-drain signal, therefore, they need to use resistor to pull high. The resistor value is determined by transmission speed and circuit design. In general, 4.7Kohm can be used when transmission speed is 400Kbps.

When I<sup>2</sup>C\_SCL is high, I<sup>2</sup>C\_SDA is changed from high to low, it is called START.

When I<sup>2</sup>C\_SCL is high, I<sup>2</sup>C\_SDA is changed from low to high, it is called STOP.

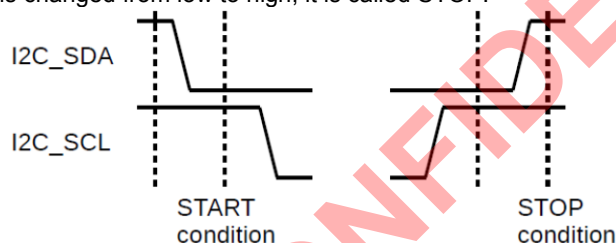


Figure 20.2 START condition and STOP condition

DATA is fixed to 8 bits. When data is sent, I<sup>2</sup>C\_SDA can be changed state only I<sup>2</sup>C\_SCL is low. After each 8 bits DATA is sent, 1 bit Acknowledge should be followed. It is called ACK when Acknowledge=0. It is called NAK when Acknowledge=1:

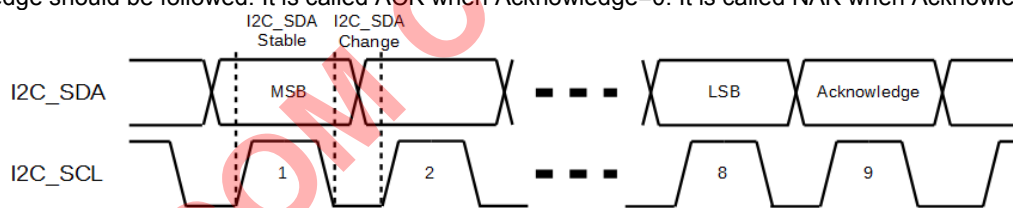


Figure 20.3 Data Validity

I<sup>2</sup>C Bus must be checked and it is idle before sending. When I<sup>2</sup>C bus is idle, START can be sent by I<sup>2</sup>C Master. I<sup>2</sup>C bus will be occupied after sending START and the other I<sup>2</sup>C host will be not transmitted. I<sup>2</sup>C Bus will be released and changed to idle until STOP is sent. The simplest data and successful data of an I<sup>2</sup>C transmission will contain START, SA [7: 0], RS, Acknowledge (0), DATA [7: 0], Acknowledge, STOP at least. Please refer to following figure:

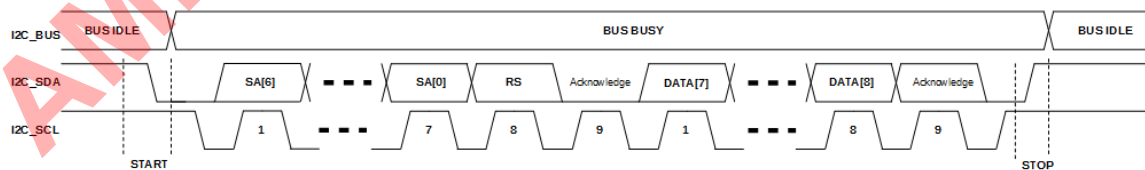


Figure 20.4 A simple I<sup>2</sup>C Transmission

For easy to describe, these symbols will be used later.

**S** represents START.

**Sr** represents Repeat START.

**a** represents Acknowledge to send by device.

**A** represents Acknowledge to send by I<sup>2</sup>C Master.

**P** represents STOP.

After sending START, then slave address and RS will be transmitted. If address of I<sup>2</sup>C Bus matched with device, I<sup>2</sup>C\_SDA will be pulled low and generated ACK by device, transmission will be continued.

S	7 bits Slave Address	RS	0
---	----------------------	----	---

If I<sup>2</sup>C bus does not match with device, I<sup>2</sup>C\_SDA keeps high and generated NAK. STOP will be transmitted by I<sup>2</sup>C Master and transmission will be ended.

S	7 bits Slave Address	RS	1	P
---	----------------------	----	---	---

It represents I<sup>2</sup>C Master to send data to device when RS is 0.

S	7 bits Slave Address	0	0	8 bits data SEND	a
---	----------------------	---	---	------------------	---

ACK will be returned when Device received data normally. STOP will be sent by I<sup>2</sup>C Master to release I<sup>2</sup>C Bus when transmission was ended.

8 bits data SEND	0	---	8 bits data SEND	0	P
------------------	---	-----	------------------	---	---

NAK will be returned when Device cannot receive data or error happened. In general, STOP will be sent by I<sup>2</sup>C Master to release I<sup>2</sup>C Bus.

8 bits data SEND	1	P
------------------	---	---

It represents device sends data to I<sup>2</sup>C Master when RS is 1.

S	7 bits Slave Address	1	0	8 bits data RECEIVE	A
---	----------------------	---	---	---------------------	---

In general, ACK will be set when I<sup>2</sup>C Master keeps receiving data. NAK will be set and STOP will be sent for transmission ending when I<sup>2</sup>C Master does not receive data.

8 bits data RECEIVE	0	---	8 bits data RECEIVE	1	P
---------------------	---	-----	---------------------	---	---

During transmission, if the I<sup>2</sup>C Master wants to change the transmission direction or with other Device to communicate, and does not want to release I<sup>2</sup>C BUS. START is sent again when STOP does not send. The START is called Repeat START. Transmission will be continued after Repeat START.

To use Repeat START read data from Device after I<sup>2</sup>C Master sending data.

---	8 bits data SEND	a	Sr	7 bits Slave Address	1	a	8 bits data RECEIVE	A	---
-----	------------------	---	----	----------------------	---	---	---------------------	---	-----

To use Repeat START send data to Device after I<sup>2</sup>C Master reading data.

---	8 bits data RECEIVE	0	Sr	7 bits Slave Address	0	a	8 bits data SEND	a	---
-----	---------------------	---	----	----------------------	---	---	------------------	---	-----

Via combine these formats, I<sup>2</sup>C transmission can be completed.

## 20.5.2 I<sup>2</sup>C Transmission Speed

A lot of transmission speed is defined by I<sup>2</sup>C protocol, including

- Standard Mode (<= 100K bps)
- Fast Mode (<= 400 Kbps)
- Fast Mode Plus (<= 1M bps)
- High Speed Mode (<= 3.4 Mbps)

Standard mode, fast mode and fast mode plus do not require extra control. The high speed mode requires a specific Master Code to control.

### 20.5.2.1 Normal Speed

In Normal Speed, I<sup>2</sup>C Master supports these transmission speed : Standard Mode (<= 100K bps), Fast Mode (<= 400 Kbps) and Faster Mode Plus (<= 1M bps) of I<sup>2</sup>C. MTP[6:0] (0x5000300C), SCL\_LP[3:0] (0x50003010) and SCL\_HP[3:0] (0x50003014) can be combined transmission speed wanted by user. The formula is as follows:

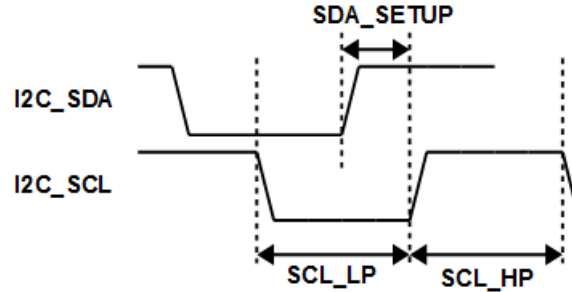
$$\begin{aligned} \text{SCL\_PERIOD} &= 2 \times (1 + \text{MTP}[6:0]) \times (\text{SCL\_LP}[3:0] + \text{SCL\_HP}[3:0]) \times \text{CLK\_PRD} \\ \text{SCL\_FREQUENCY} &= 1 / \text{SCL\_PERIOD} \end{aligned}$$

For example, if CLK\_FRQ = 16MHz, CLK\_PRD = 62.5ns. MTP[6:0] = 3, SCL\_LP[3:0] = 6, SCL\_HP[3:0] = 4.

Then SCL\_PERIOD = 2 x (1 + 3) x (6 + 4) x 62.5ns = 5000ns = 5us

And SCL\_FREQUENY = 1 / 5us = 200 KHz

SCL\_LP [3:0] is used to determine the duration of I<sup>2</sup>C\_SCL at low.  
 SCL\_HP[3:0] is used to determine the duration of I<sup>2</sup>C\_SCL at high.  
 SDA\_SETUP [3: 0] is used to determine the time point of I<sup>2</sup>C\_SDA conversion.



NOTE: SCL\_LP > SDA\_SETUP

Figure 20.5 SCL\_LP, SCL\_HP and SDA\_SETUP

### 20.5.2.2 High Speed

When Device supports High Speed Mode, Master Code (0x00001XXX) can be sent after START, followed a forced NAK Acknowledge. And then, Speed will be changed to High Speed. High Speed will be back to Normal Speed until transmission end.

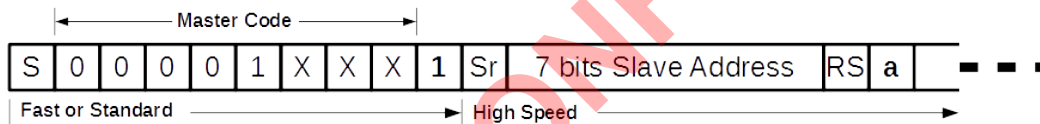


Figure 20.6 Switching High Speed Mode

Please be noted, the Master Code is provided by the Device that supports High Speed Mode.

In I<sup>2</sup>C Master, bit HS controls whether to enter the High Speed Mode or not. Transmission speed of High Speed Mode is fixed as follows:

$$\text{SCL\_PERIOD} = 12 \times \text{CLK\_PRD}$$

$$\text{SCL\_FREQUENCY} = \text{MCU Clock} / 12$$

### 20.5.3 I<sup>2</sup>C Master

I<sup>2</sup>C Master will be switched at Idle State, Master Transmitter and Master Receiver.

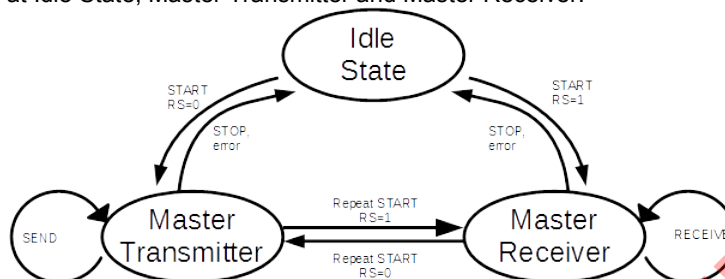


Figure 20.7 State of I<sup>2</sup>C Master

#### 20.5.3.1 I<sup>2</sup>C Master Control

I<sup>2</sup>C Master can control by the main use of I<sup>2</sup>CMSA (0x50003000) and I<sup>2</sup>CMCR (0x50003004). Bit RS is included in I<sup>2</sup>CMSA. These bits RSTB, SLRST, HS, ACK, STOP, START and RUN are included in I<sup>2</sup>CMCR. These control bits can be combined the following transmission formats:

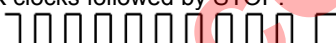

R S	R S T B	S L R S T	H S	A C K	S T O P	S T A R T	R U N	OPERATION	Next I <sup>2</sup> C Master State	Next Speed Mode
X	1	0	X	X	X	X	X	I <sup>2</sup> CM module software reset	Idle	Normal
0	0	1	0	0	0	0	1	9 SCK clocks followed by STOP: I2C_SCL  I2C_SDA 	Idle	Normal
0	0	0	0	X	0	1	1	S 7 bits Slave Address 0 a 8 bits data SEND a	Transmitter	Continue
0	0	0	0	X	1	1	1	S 7 bits Slave Address 0 a 8 bits data SEND a P	Idle	Normal
1	0	0	0	0	0	1	1	Sr 7 bits Slave Address 1 a 8 bits data RECEIVE 1	Receiver	Continue
1	0	0	0	0	1	1	1	Sr 7 bits Slave Address 1 a 8 bits data RECEIVE 1 P	Idle	Normal
1	0	0	0	1	0	1	1	Sr 7 bits Slave Address 1 a 8 bits data RECEIVE 0	Receiver	Continue
0	0	0	1	0	0	0	1	S Master Code 0 a	Transmitter	High Speed

Table 20.3 Control bits combinations permitted in Idle state

R S	R S T B	S L R S T	H S	A C K	S T O P	S T A R T	R U N	OPERATION	Next I <sup>2</sup> C Master State	Next Speed Mode
X	1	0	X	X	X	X	X	I <sup>2</sup> CM module software reset	Idle	Normal
X	0	0	0	X	1	0	0	P	Idle	Normal
X	0	0	0	X	0	0	1	8 bits data SEND a	Transmitter	Continue
X	0	0	0	X	1	0	1	8 bits data SEND a P	Idle	Normal
0	0	0	0	X	0	1	1	Sr 7 bits Slave Address 0 a 8 bits data SEND a	Transmitter	Continue
0	0	0	0	X	1	1	1	Sr 7 bits Slave Address 0 a 8 bits data SEND a P	Idle	Normal
1	0	0	0	0	0	1	1	Sr 7 bits Slave Address 1 a 8 bits data RECEIVE 1	Receiver	Continue
1	0	0	0	0	1	1	1	Sr 7 bits Slave Address 1 a 8 bits data RECEIVE 1 P	Idle	Normal
1	0	0	0	1	0	1	1	Sr 7 bits Slave Address 1 a 8 bits data RECEIVE 0	Receiver	Continue

Table 20.4 Control bits combinations permitted in Master Transmitter



R	S	R	S	S	H	A	S	S	R	OPERATION	Next I <sup>2</sup> C Master State	Next Speed Mode
S	T	T	R	S	C	T	O	T	U			
B	B	S	S	T	K	O	P	A	N			
X	1	0	X	X	X	X	X	X	X	I <sup>2</sup> C module software reset	Idle	Normal
X	0	0	0	X	1	0	0		P		Idle	Normal
X	0	0	0	0	0	0	0	1		8 bits data RECEIVE 1	Receiver	Continue
X	0	0	0	0	1	0	1		P	8 bits data RECEIVE 1	Idle	Normal
X	0	0	0	1	0	0	1			8 bits data RECEIVE 0	Receiver	Continue
1	0	0	0	0	0	1	1	S		7 bits Slave Address 1 a 8 bits data RECEIVE 1	Receiver	Continue
1	0	0	0	0	1	1	1	S		7 bits Slave Address 1 a 8 bits data RECEIVE 1 P	Idle	Normal
1	0	0	0	1	0	1	1	S		7 bits Slave Address 1 0 8 bits data RECEIVE 0	Receiver	Continue
0	0	0	0	X	0	1	1	Sr		7 bits Slave Address 0 a 8 bits data SEND a	Transmitter	Continue
0	0	0	0	X	1	1	1	Sr		7 bits Slave Address 0 a 8 bits data SEND a P	Idle	Normal

Table 20.5 Control bits combinations permitted in Master Receiver

### 20.5.3.2 BUSY bit

When Control bit write to I<sup>2</sup>CMCR, the BUSY bit will not change to high immediately. There are about delay 2us.

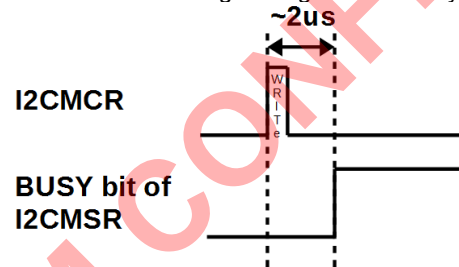


Figure 20.8 Timing for BUSY bit

### 20.5.3.3 I<sup>2</sup>C Master Interrupt

To use the I<sup>2</sup>C Master interrupt, in addition to set I<sup>2</sup>CMIE = 1 in I<sup>2</sup>CMINT (0x5000301C), I<sup>2</sup>C\_INT=1 should be set in NVIC. Please refer to Table 11.2 Interrupt Map Vector Table..

When any transmission is completed by I<sup>2</sup>C Master, I<sup>2</sup>C Master Interrupt will be generated. I<sup>2</sup>CMIF can be determined whether interrupt is generated by I<sup>2</sup>C Master.

User need write I<sup>2</sup>CMIF=1 to clean I<sup>2</sup>CMIF.

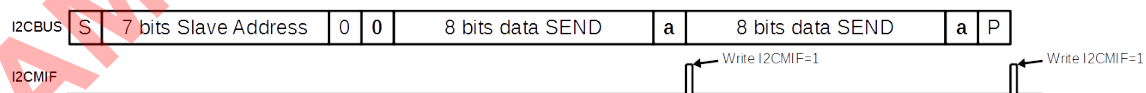


Figure 20.9 I<sup>2</sup>CMIF=1 when transmit finish

### 20.5.3.4 I<sup>2</sup>C Master Transmit

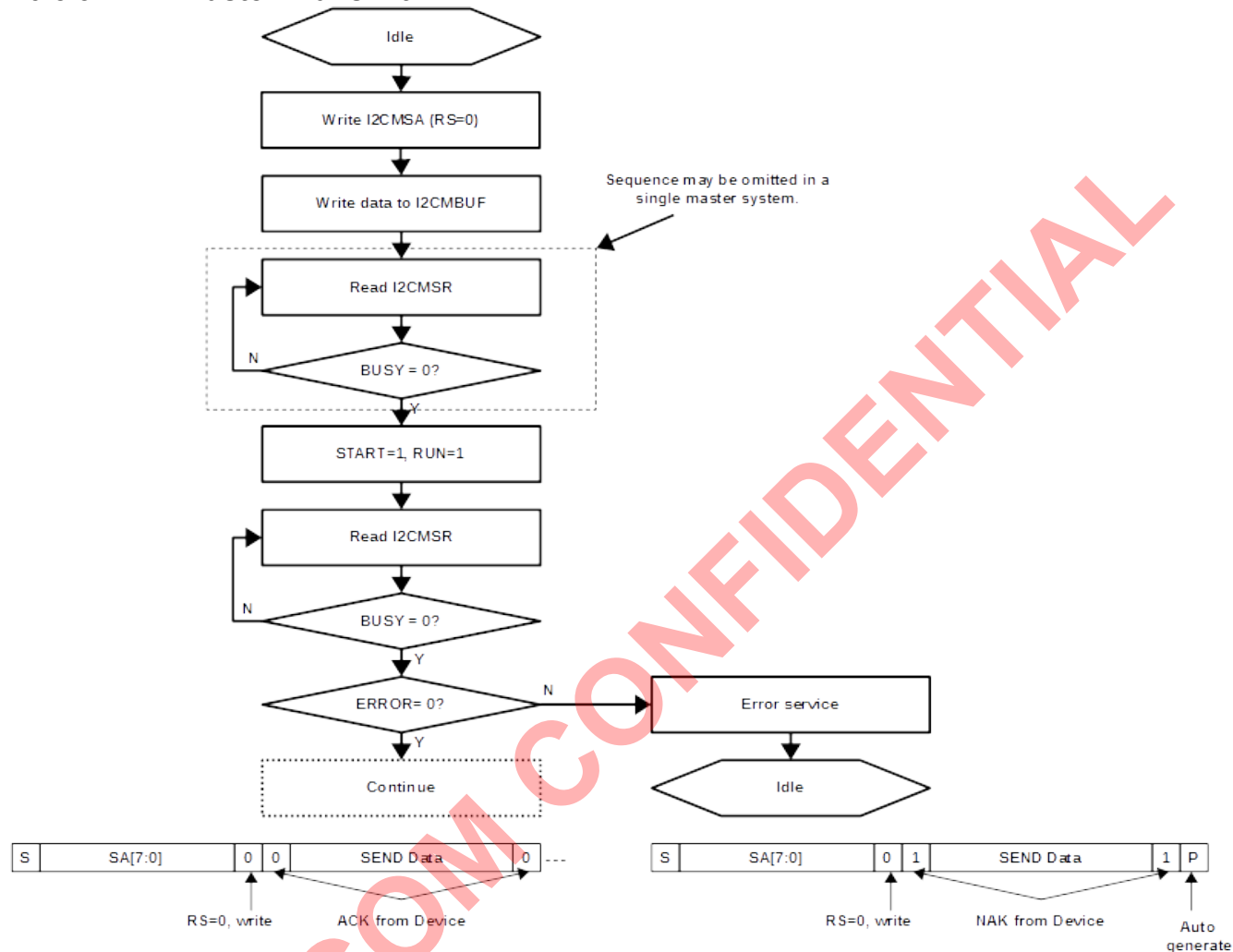


Figure 20.10 The flow chart of I<sup>2</sup>C Master Transmit

### 20.5.3.5 I<sup>2</sup>C Master Receive

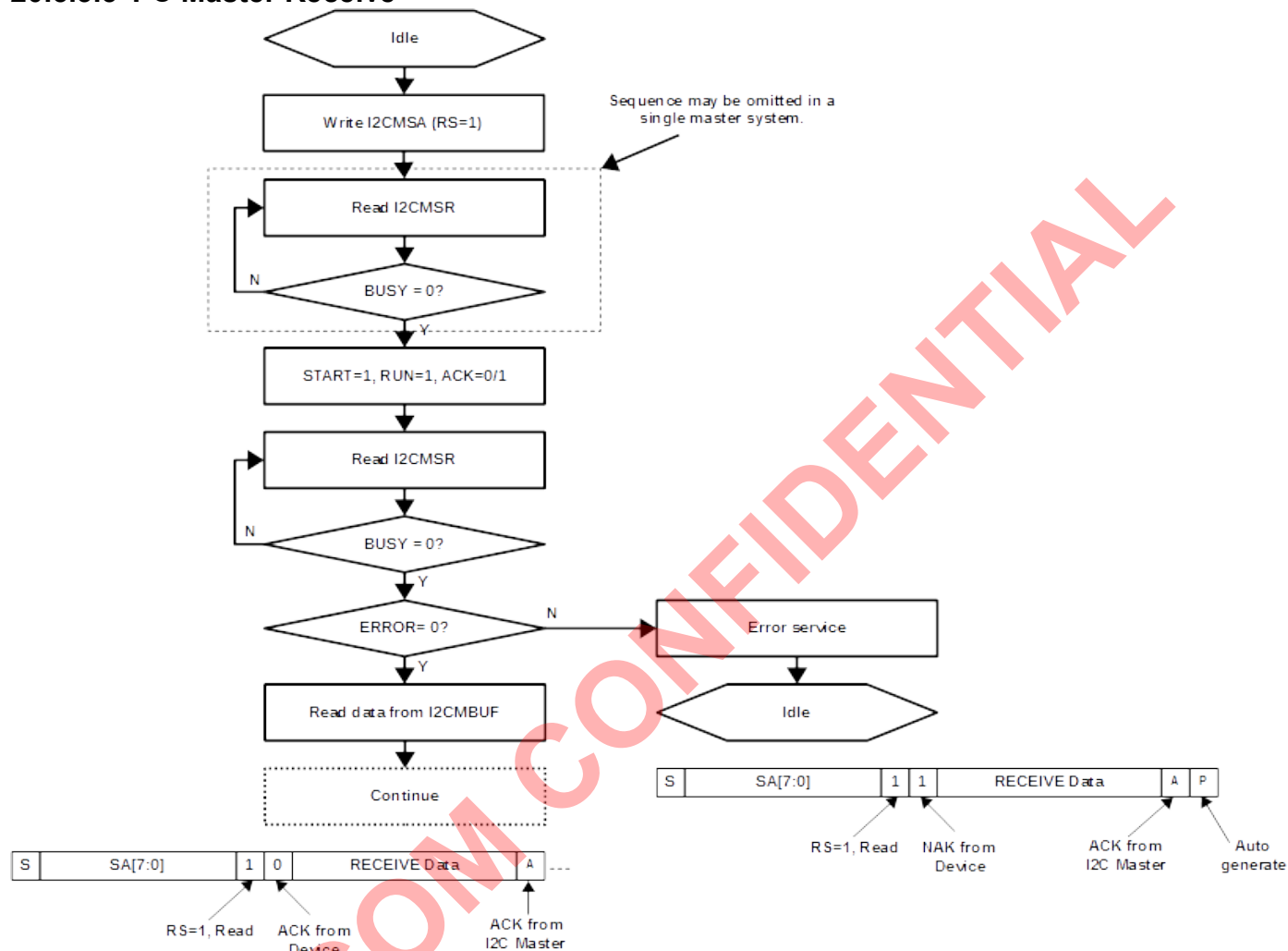


Figure 20.11 The flow chart of I<sup>2</sup>C Master Receive

### 20.5.3.6 I<sup>2</sup>C Master Transmit with Repeat START

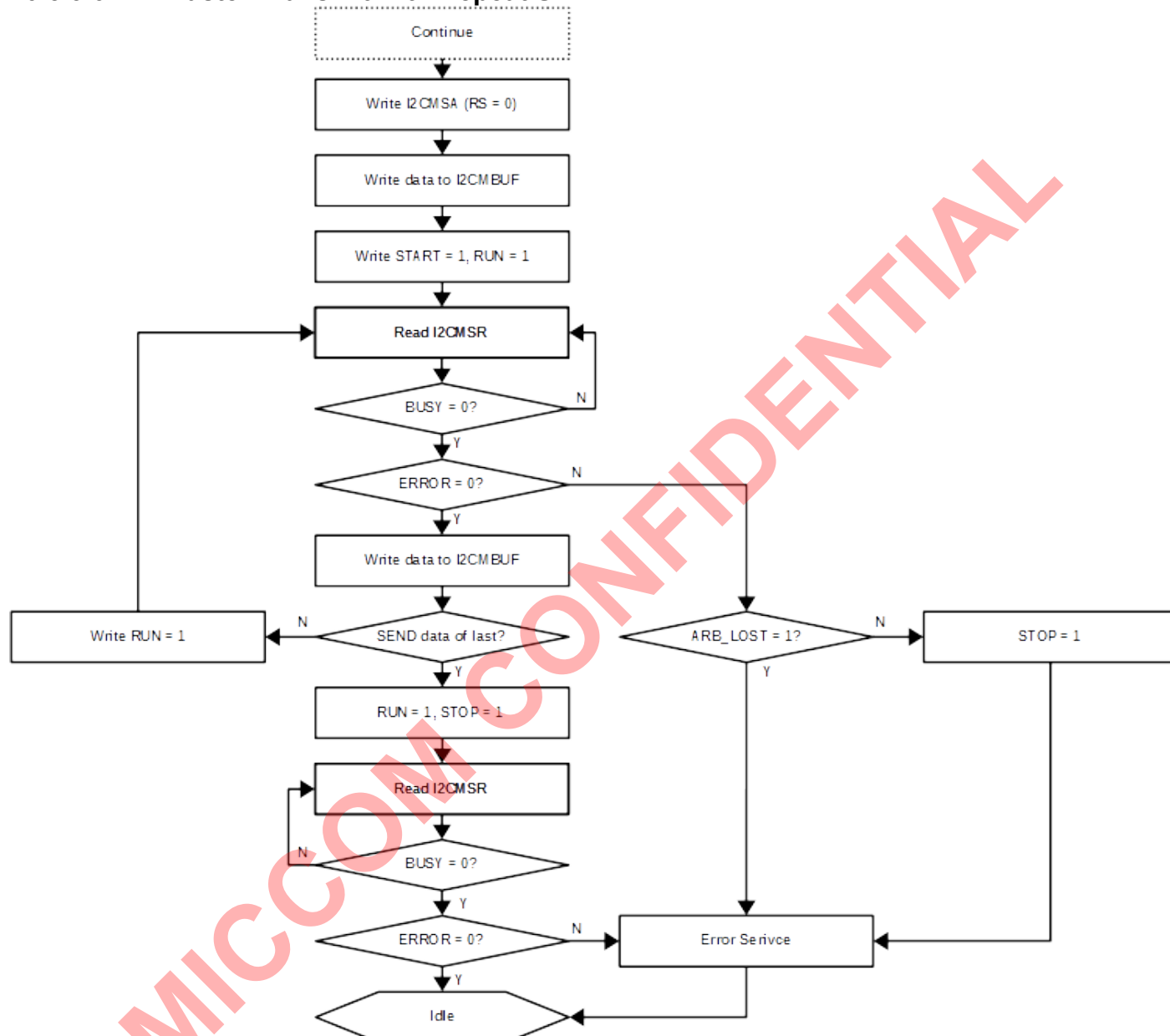


Figure 20.12 The flow chart of I<sup>2</sup>C Master Transmit with Repeat START

### 20.5.3.7 I<sup>2</sup>C Master Receive with Repeat START

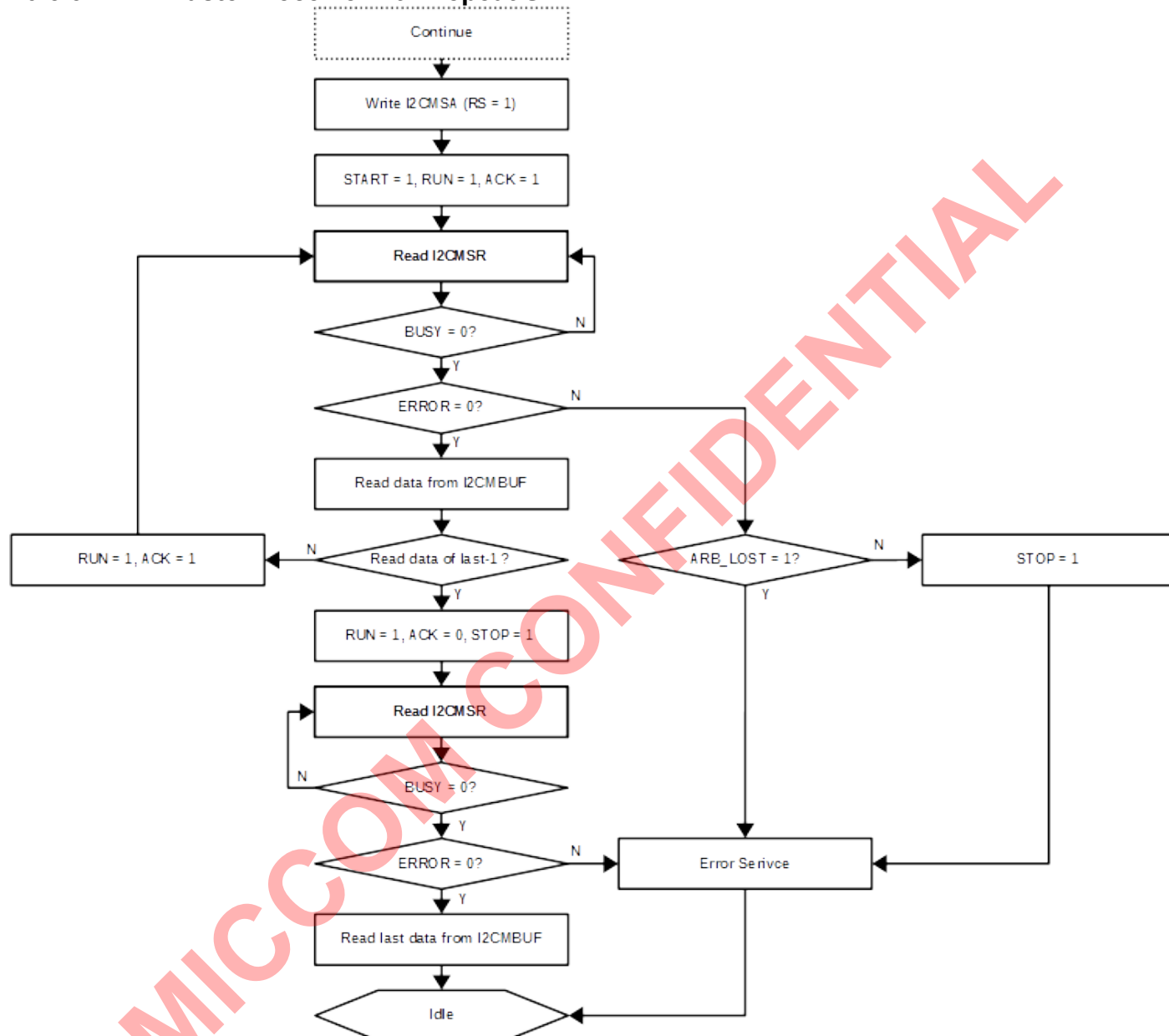


Figure 20.13 The flow chart of I<sup>2</sup>C Master Receive with Repeat START

### 20.5.4 I<sup>2</sup>C Slave

I<sup>2</sup>C Slave will be switched at Idle State, Slave Receiver and Slave Transmitter.

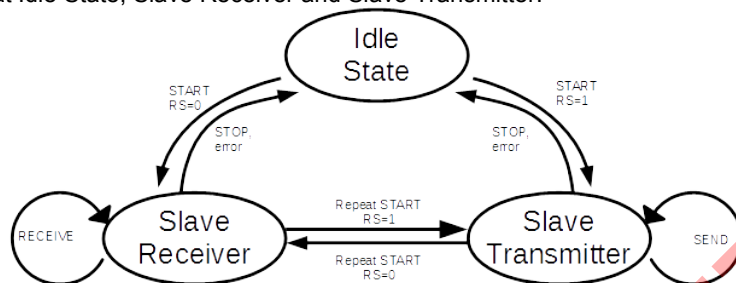


Figure 20.14 State of I<sup>2</sup>C Slave

#### 20.5.4.1 I<sup>2</sup>C Slave Control

Because I<sup>2</sup>C Slave can only be received state passively, therefore, bits need to be controlled are less actually. I<sup>2</sup>C Slave will be ready when after set DA=1. And then, read the state of the I<sup>2</sup>CSSR to determine the action.

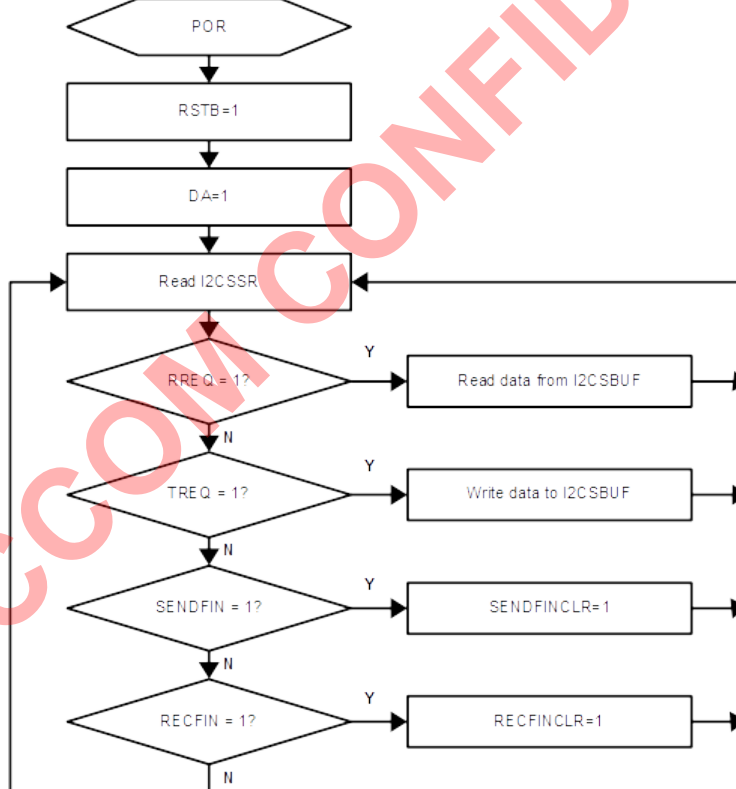


Figure 20.15 The flow chart of I<sup>2</sup>C Slave

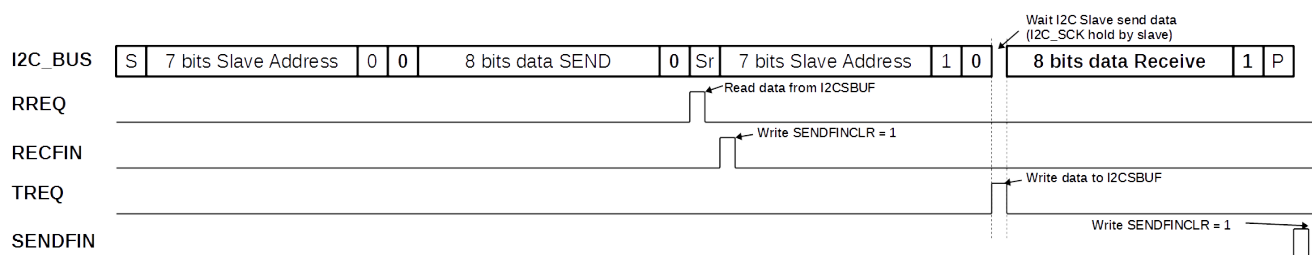


Figure 20.16 Timing for RREQ, TREQ, RECFIN, SENDFIN

### 20.5.4.2 I<sup>2</sup>C Slave Interrupt

To use the I<sup>2</sup>C Slave interrupt, in addition to set I<sup>2</sup>CSIF=1 in I<sup>2</sup>CSINT (0x50003810), I<sup>2</sup>C\_INT=1 should be set in NVIC. Please Refer Table 11.2 Interrupt Map Vector Table.

When any signal of RREQ or TREQ or SENDFIN or RECFIN is generated by I<sup>2</sup>C Slave, I<sup>2</sup>C Slave interrupt will be generated. I<sup>2</sup>CSIF can be determined whether interrupt is generated by I<sup>2</sup>C Slave. After interrupt generation, I<sup>2</sup>CSIF should be written to 1 to clear I<sup>2</sup>CSIF.

### 20.5.5 10 Bit Address

I<sup>2</sup>C protocol uses 10 Bit Address to solve address 7 Bit Address caused by the lack of address space, and it can be compatible with the original 7 Bit Address protocol. Preamble “11110” is used to determine 10 Bit Address and 7 Bit Address by 10 Bit Address. Followed by 2 Bits Slave Address (SA [9: 8]), and then the next packet can be written to the remaining 8 Bits Slave Address (SA [7: 0]).

The following format is used to write and read 10 Bit Address Device generally:

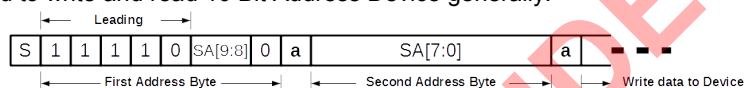


Figure 20.17 Write data to 10 Bit Address Device

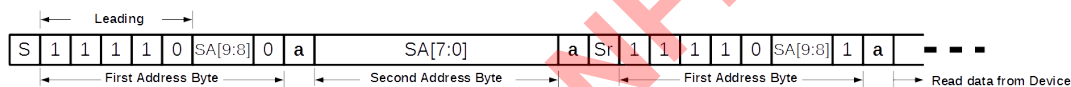


Figure 20.18 Read data from 10 Bit Address Device (It needs to use Repeat Start to direction conversion).

#### 20.5.5.1 10 Bit Address for I<sup>2</sup>C Master

Because 10 Bit Address is compatible with 7 Bit Address, I<sup>2</sup>C Master does not need to set specially. Only according to transmission format of the 10 Bit Address to send the corresponding data.

#### 20.5.5.2 10 Bit Address for I<sup>2</sup>C Slave

TEN\_ADDR\_EN=1 of I<sup>2</sup>CSOAUP (0x5000380C), set ADDR[9:7] and ADDR[6:0]. And then, set DA=1, followed by the same as the original Slave program.

## 21. SPI Interface

### 21.1 FEATURES

- Full duplex Synchronous serial data transfer
- Configurable as a master or a slave on the interface
- Programmable clock bit rate (1/2 (master only), 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256 of MCU clock)
- Programmable data size from 4 bits to 16 bits
- Separate TXFIFO (8x16bits) and RXFIFO (8x16bits).
- Interrupts for TXFIFO and RXFIFO

### 21.2 PINS DESCRIPTION

PIN	GPIO	TYPE	DESCRIPTION
SPI_SCK	P0_03	INPUT / OUTPUT	SPI clock pin
SPI_MOSI	P0_02	INPUT / OUTPUT	Slave serial data input / Master serial data output
SPI_MISO	P0_01	INPUT / OUTPUT	Master serial data input / Slave serial data output
SPI_CS	P0_00	INPUT / OUTPUT	Slave select pin

Table 21.1 SPI pins description

### 21.3 BLOCK DIAGRAM

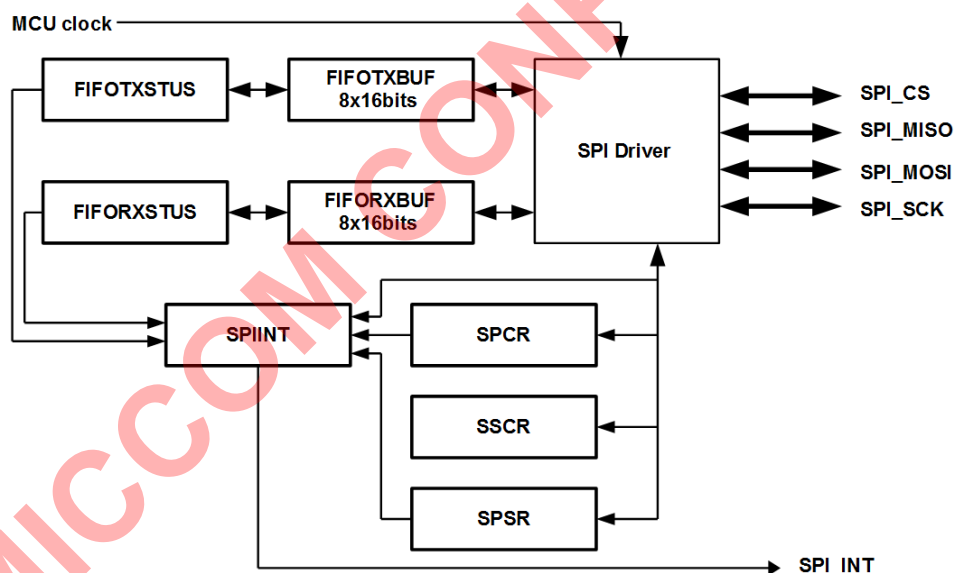


Figure 21.1 SPI Block Diagram

### 21.4 REGISTER

#### 21.4.1 Register list

Address	Name	DESCRIPTION
0x50002000	SPCR	SPI Control Register
0x50002004	SPSR	SPI Status Register
0x50002008	SPIINT	SPI Interrupt Register
0x5000200C	SSCR	Slave Select Control Register
0x50002010	FIFOTXSTUS	SPI TX FIFO Status
0x50002014	FIFORXSTUS	SPI RX FIFO Status
0x50002018	FIFOTXBUF	SPI TX FIFO Buffer
0x5000201C	FIFORXBUF	SPI RX FIFO Buffer

Table 21.2 SPI register list



## 21.4.2 Register Description

### SPI Control Register (Address: 0x50002000)

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

**RXFIFO\_IL[2:0]:** Set SPI RX FIFO interrupt level. If RXFIFO\_IL[2:0] < RXFIFO\_REM[2:0], generate interrupt.

**TXFIFO\_IL[2:0]:** Set SPI TX FIFO interrupt level. If TXFIFO\_IL[2:0] > TXFIFO\_REM[2:0], generate interrupt.

**SLAVE\_RST:** In SPI Slave, write 1 to reset SPI TX FIFO point.

[0]: No effect

[1]: Reset SPI TX FIFO WP (only for SPI Slave)

**FIFOEN:** SPI FIFO mode enable

[0]: NOT SUPPORT

[1]: FIFO mode

**BIT\_LEN[3:0]:** Set number of bits in a shift sequence.

[0~3]: 4 bits data

[4~15]: (BIT\_LEN[3:0] + 1) bits data

**SPE:** SPI Enable

[0]: SPI disable

[1]: SPI enable

**MSTR:** SPI Master/ SPI Slave select

[0]: SPI Slave

[1]: SPI Master

**CPOL:** Clock polarity select

[0]: SPI\_SCK idle low

[1]: SPI\_SCK idle high

**CPHA:** Clock phase select

[0]: Latch data in first SPI\_SCK edge change

[1]: Latch data in second SPI\_SCK edge change

**SPR[2:0]:** SPI clock rate select bits:

SPR2	SPR1	SPR0	MCU 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 for SPI Master)

**SPI Status Register (Address: 0x50002004)**

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--		--	--				SSCEN
R	--		SS_S	--				SSCEN
Reset	0	0	0	0	0	1	0	0

**SS\_S:** In SPI Salve, this bit connected with SPI\_CS.

**SSCEN:** SPI\_CS automatic select setting. This bit work with SS0.

SSCEN	SS0	SPI_CS
0	0	0
0	1	1
1	0	0 when transfer. 1 when idle.
1	1	1

**SPI Interrupt Register (Address: 0x50002008)**

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--		RXFIFO_INT_FLAG	TXFIFO_INT_FLAG	--		RXFIFO_INTEN	TXFIFO_INTEN
R	--		RXFIFO_INT_FLAG	TXFIFO_INT_FLAG	--		RXFIFO_INTEN	TXFIFO_INTEN
Reset	0	0	0	0	0	0	0	0

**RXFIFO\_INT\_FLAG:** RXFIFO interrupt flag

[0]: Interrupt not occurs.

[1]: Interrupt occurred, write 1 to clear.

**TXFIFO\_INT\_FLAG:** TXFIFO interrupt flag

[0]: Interrupt not occurs.

[1]: Interrupt occurred, write 1 to clear.

**RXFIFO\_INTEN:** SPI RXFIFO interrupt enable

[0]: Disable

[1]: Enable

**TXFIFO\_INTEN:** SPI TXFIFO interrupt enable

[0]: Disable

[1]: Enable

NOTE: User need clear RXFIFO\_INT\_FLAG and TXFIFO\_INT\_FLAG before write FIFOTXBUF when SPI interrupt occur.

**SPI Select Control Register (Address: 0x5000200C)**

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--							SS0
R	--							SS0
Reset	1	1	1	1	1	1	1	1

**SS0:** SPI\_CS automatic select setting

SSCEN	SS0	SPI_CS
-------	-----	--------

0	0	0
0	1	1
1	0	0 when transfer. 1 when idle.
1	1	1

### FIFO TX Status (Address: 0x5000200C)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--			--	--	--		
R	--			TX_FULL	TX_EMPTY	TXFIFO_REM[2:0]		
Reset	0	0	0	0	1	0	0	0

**TX\_FULL:** FIFOTXBUF Full

[0]: FIFOTXBUF not full, user can write data to FIFOTXBUF.

[1]: FIFOTXBUF full, user can't write data to FIFOTXBUF.

**TX\_EMPTY:** FIFOTXBUF Empty

[0]: FIFOTXBUF not empty.

[1]: FIFOTXBUF empty.

**TXFIFO\_REM [2:0]:** FIFOTXBUF remnant. The number of data in the FIFOTXBUF

TX_FULL	TX_EMPTY	TXFIFO_REM[2:0]	Number of data in FIFOTXBUF
0	1	0	0
0	0	0	1
0	0	1	2
0	0	2	3
0	0	3	4
0	0	4	5
0	0	5	6
0	0	6	7
1	0	7	8

### FIFO RX Status (Address: 0x50002014)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--			--	--	--		
R	--			RX_FULL	RX_EMPTY	RXFIFO_REM[2:0]		
Reset	0	0	0	0	1	0	0	0

**RX\_FULL:** FIFORXBUF Full

[0]: FIFORXBUF not full.

[1]: FIFORXBUF full.

**RX\_EMPTY:** FIFORXBUF Empty

[0]: FIFOTXBUF not empty.

[1]: FIFOTXBUF empty. Read FIFORXBUF will read as 0.

**RXFIFO\_REM [2:0]:** The numbers of data in the FIFORXBUF

RX_FULL	RX_EMPTY	RXFIFO_REM[2:0]	Numbers of data in FIFORXBUF
0	1	0	0
0	0	1	1
0	0	2	2
0	0	3	3
0	0	4	4
0	0	5	5
0	0	6	6
0	0	7	7
1	0	0	8

**FIFO TX Buffer (Address: 0x50002018)**

R/W	Bit 15	-----	Bit 0
W	TXBUF[15:0]		
R	--		
Reset	0x0000		

**TXBUF[15:0]:** FIFOTXBUF is a FIFO register with a width of 16 bits and depth of 8. User can write data to this FIFO and send out via SPI. Refer to 21.5.4 SPI FIFO for details.

**FIFO RX Buffer (Address: 0x5000201C )**

R/W	Bit 15	-----	Bit 0
W	--		
R	RXBUF[15:0]		
Reset	0x0000		

**RXBUF[15:0]:** FIFORXBUF is a FIFO register with a width of 16 bits and depth of 8. Data will be saved to FIFORXBUF when SPI received data. User can read out data from FIFORXBUF. Refer to 21.5.4 SPI FIFO for details.

## 21.5 FUNCTION DESCRIPTION

### 21.5.1 SPI Master and SPI Slave

SPI Driver can be set to SPI Master (MSTR=1) or SPI Slave (MSTR=0) via MSTR. When SPI Driver is set to SPI Master, SPI\_CS and SPI\_SCK will be controlled by SPI Driver, and send data via SPI\_MOSI, receive data via SPI\_MISO. When SPI Driver is set to SPI Slave, SPI\_CS and SPI\_SCK will be controlled by other SPI, and receive data via SPI\_MOSI, send data via SPI\_MISO.

### 21.5.2 SPI\_SCK speed

Speed of SPI is decided by speed of SPI\_CLK. Speed of SPI\_CLK is related to MCU Clock. Speed can be adjusted by SPR2, SPR1 and SPR0:

SPR2	SPR1	SPR0	MCU 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 for SPI Master)

Table 21.3 SPI pins description

For example:

System Clock is 16MHz and SPR2=0, SPR1=0, SPR0=0, speed of SPI\_CLK is 4MHz (16MHz / 4). Therefore, speed of SPI is 4M bps.

It needs to be noted, the set SPR2=1, SPR1=1, SPR0=1 is only for SPI Master, SPI Slave does not use this set.

### 21.5.3 SPI Transmission Format

#### 21.5.3.1 SPI Transmission Data Length

SPI can set every transmission data length via BIT\_LEN[3:0] of SPCR. Data length can be set during 4~16 Bits.

For example:

Each data of SPI will transmit 4 bits when BIT\_LEN[3:0]=3.

Each data of SPI will transmit 16 bits when BIT\_LEN[3:0]=15.

#### 21.5.3.2 SPI\_CS Setting

In SPI Master, SPI\_CS can be set auto selection mode or manual selection mode. Description is as follows:

Auto selection mode: When SSCEN=1, SS0=0, SPI\_CS will be pulled low automatically at data is transmitting and SPI\_CS will be pulled high automatically at data transmitting is complete. Shown is as following figure:

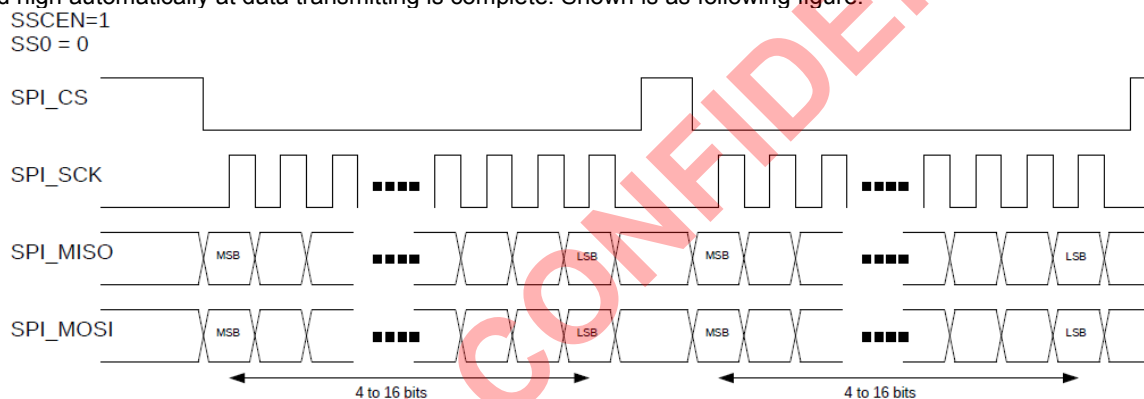


Figure 21.2 SPI\_CS auto selection mode

Manual selection mode: When SSCEN=0, SPI\_CS is controlled by SS0, SPI\_CS will be pulled low when SS0=0; SPI\_CS will be pulled high when SS0=1. Shown is as following figure:

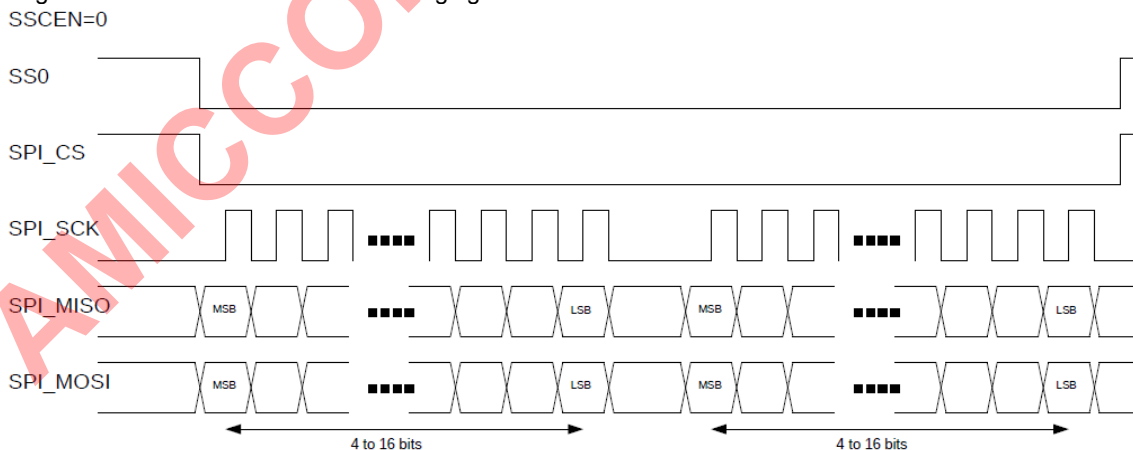


Figure 21.3 SPI\_CS manual selection mode

### 21.5.3.3 SPI\_SCK Setting

SPI can support four kinds of transmission format via CPOL and CPHA of SPCR. SPI\_SCK is set by CPOL at idle state. When CPOL=0, SPI\_SCK is low at idle state. When CPOL=1, SPI\_SCK is high at idle state. CPHA is used to set which number of SPI\_SCK to occur changed and start to latch data. When CPHA=0, start to latch data at SPI\_SCK occurs changed first time. When CPHA=1, start to latch data at SPI\_SCK occurs changed second time.

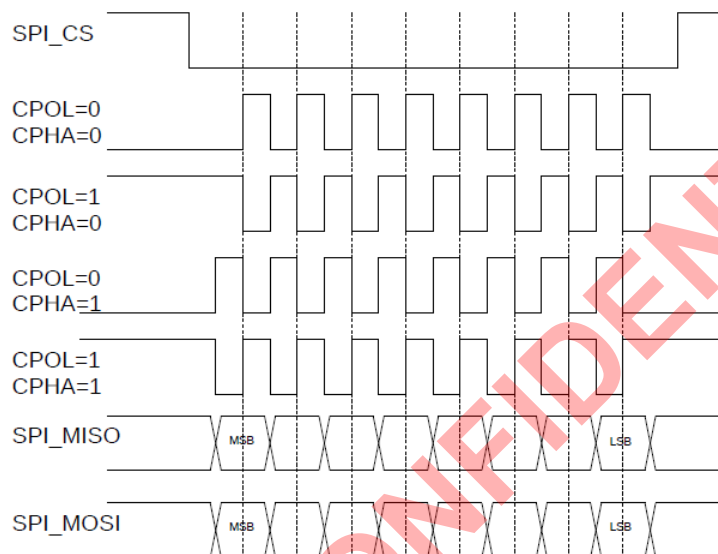


Figure 21.4 The dotted line represents timing to latch data.

### 21.5.4 SPI FIFO

The SPI has two separate FIFO, one is SPI TX FIFO, and the other is SPI RX FIFO. These two FIFO are with a width of 16 bits and depth of 8. They can be used to send/receive data continuously.

WP (write point) will move up when data is written to FIFOTXBUF.

WP (write point) will move down when data of SPI Master's MOSI or data of SPI Slave's MISO is sent out.

RP (read point) will move up when data of SPI Master's MISO or data of SPI Slave's MOSI is received.

RP (read point) will move down when data is read.

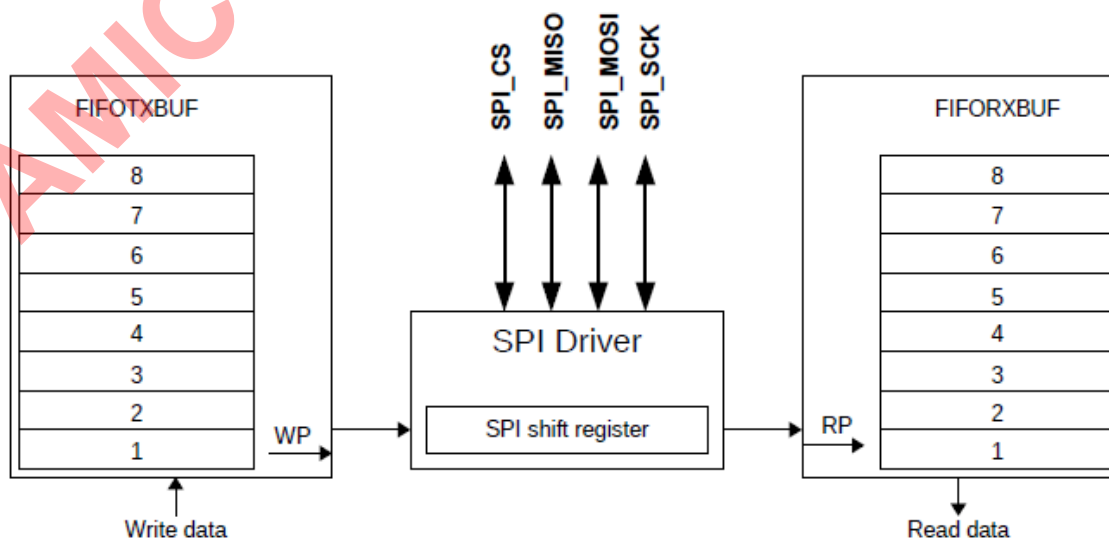


Figure 21.5 SPI FIFO Architecture descriptions

### 21.5.5 SPI FIFO interrupt

SPI FIFO interrupt, it must be  $RXFIFO\_INTEN=1$  of SPIINT or  $TXFIFO\_INTEN=1$  of SPIINT, and  $SPI\_INT=1$  of NVIC. Please refer to Table 11.2 Interrupt Map Vector Table.

And then, when data number of SPI TX FIFO or data number of SPI RX FIFO reaches to set value of  $TXFIFO\_IL[2:0]$  or  $RXFIFO\_IL[2:0]$ . SPI interrupt will be started. User can read  $TXFIFO\_INT\_FLAG$  and  $RXFIFO\_INT\_FLAG$  to determine the interruption is generated by SPI TX FIFO or SPI RX FIFO.

Please be noted,  $TXFIFO\_INT\_FLAG$  and  $RXFIFO\_INT\_FLAG$  should be cleared to zero by user. And then,  $FIFOTXBUF$  can be written data or read data from  $FIFORXBUFF$ .

SPI TX FIFO interrupt condition:  $TXFIFO\_IL[2:0] > TXFIFO\_REM[2:0]$

SPI RX FIFO interrupt condition:  $RXFIFO\_IL[2:0] < RXFIFO\_REM[2:0]$

Please refer to 21.5.6.1 SPI Master Transmit data and 21.5.6.2 SPI Master Receive data.

### 21.5.6 SPI Master Transmit

When SPI Driver is set to SPI Master, data can be written from  $FIFOTXBUF$  to SPI TX FIFO and data can be read from SPI RX FIFO via  $FIFORXBUFF$ . At the same time, state of SPI TX FIFO and SPI RX FIFO can be checked by  $FIFOTXSTUS$  and  $FIFORXSTUS$ , and write/read data at the right time to keep writing/reading data continuously.

#### 21.5.6.1 SPI Master Transmit data

When  $TX\_FULL=0$ , data can be written to  $FIFOTXBUF$  by user. When  $TX\_FULL=1$ , data can't be written to SPI TX FIFO if user keeps writing data continuously. When there is data in SPI TX FIFO, data will be sent by  $SPI\_MOSI$  at SPI Master. At the same time, data will be read by  $SPI\_MISO$  and will be saved to SPI RX FIFO.

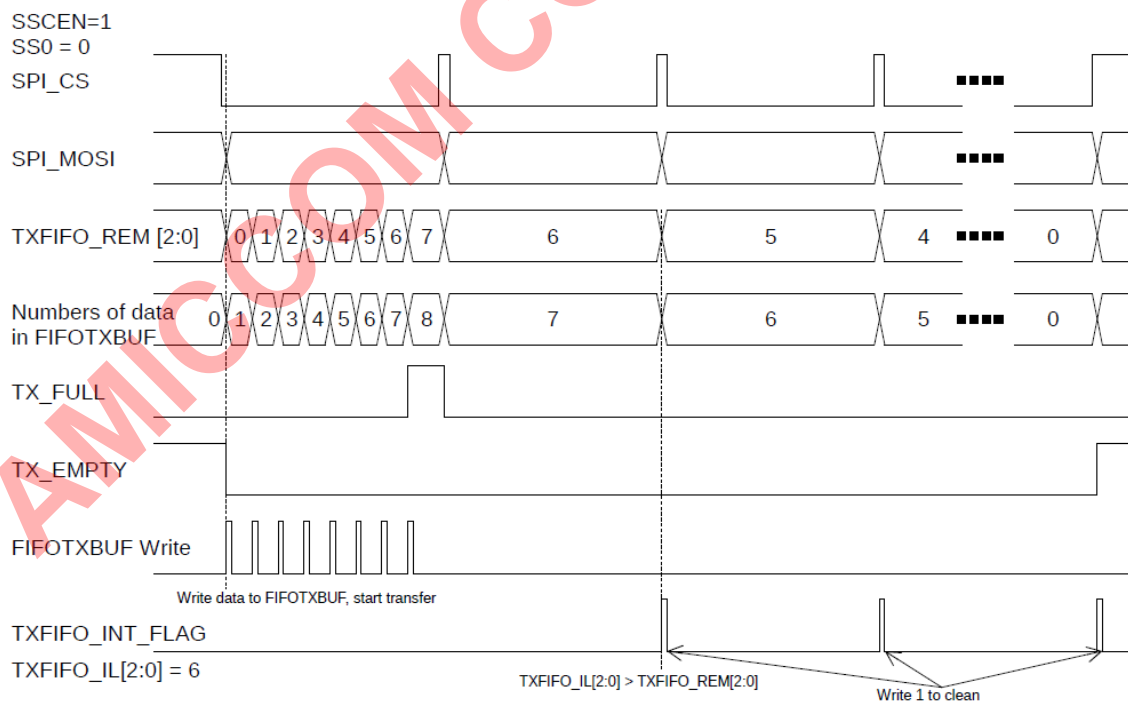


Figure 21.6 An example for SPI Master sent data 8 times and set  $TXFIFO\_IL[2:0]=6$

### 21.5.6.2 SPI Master Receive data

When SPI Master is sending data, SPI will receive data and save to SPI RX FIFO at the same time. When RX\_FULL=1, data will be not written to SPI RX FIFO.

SSCEN=1

SS0 = 0

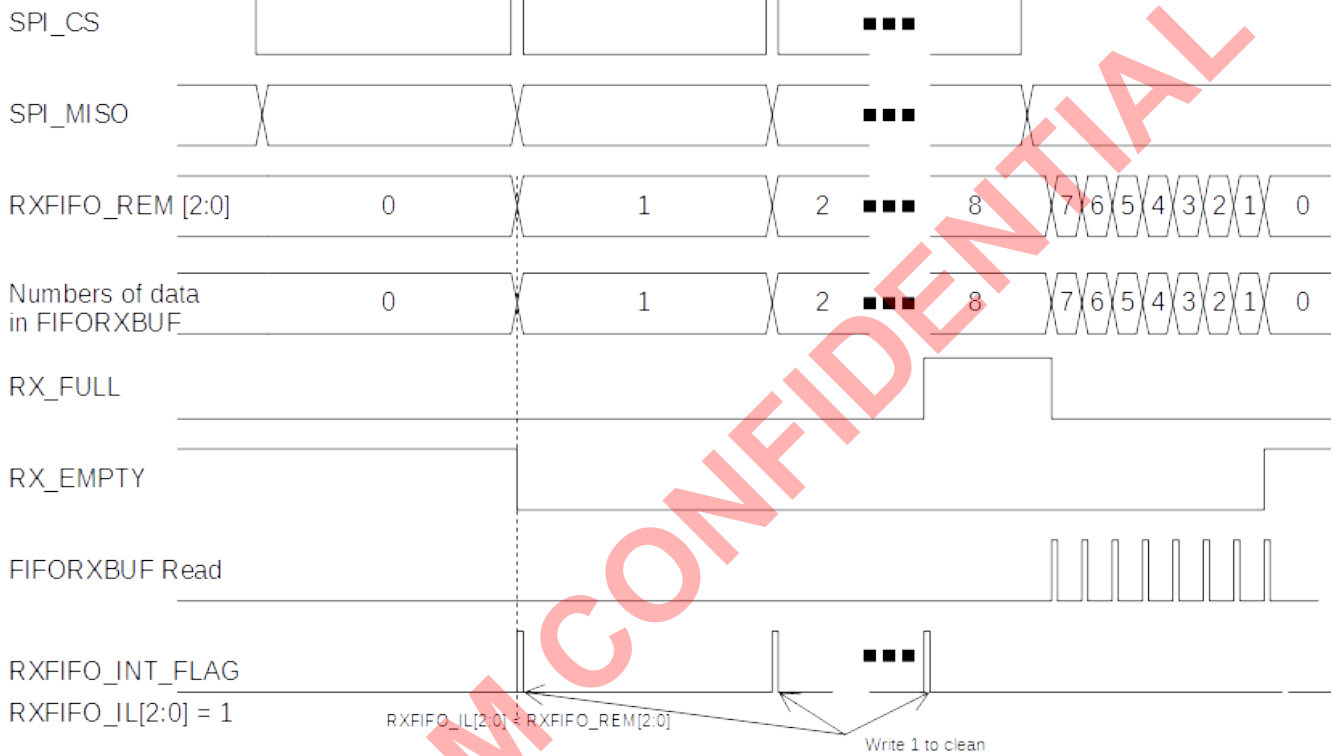


Figure 21.7 An example SPI Master received data 8 times and set RXFIFO\_IL[2:0]=1



### 21.5.7 SPI Slave Transmit

#### 21.5.7.1 SPI Slave Transmit data

In SPI Slave mode, data can be written to FIFOTXBUF at SPI idle (there is no send/receive data). Or, data will be error. When TX\_FULL=1, data cannot be written to SPI TX FIFO if user keeps writing data continuously. SLAVE\_RST=1 can be set by SPI Slave, and SPI TX FIFO WP can be reset to first data position.

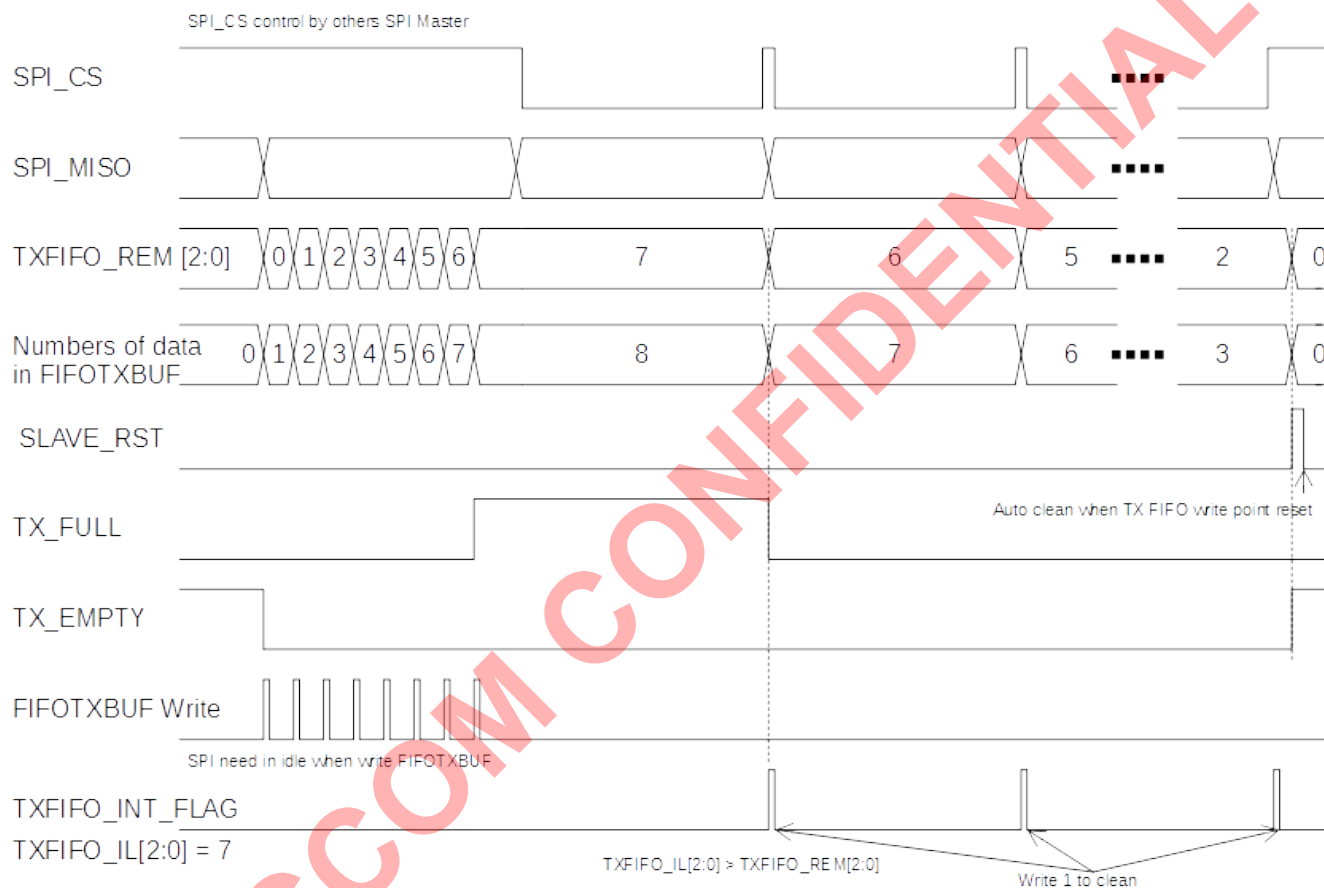


Figure 21.8 An example for SPI Slave wrote data 8 times in advance, and sent out data when received data.

### 21.5.7.2 SPI Slave Receive data

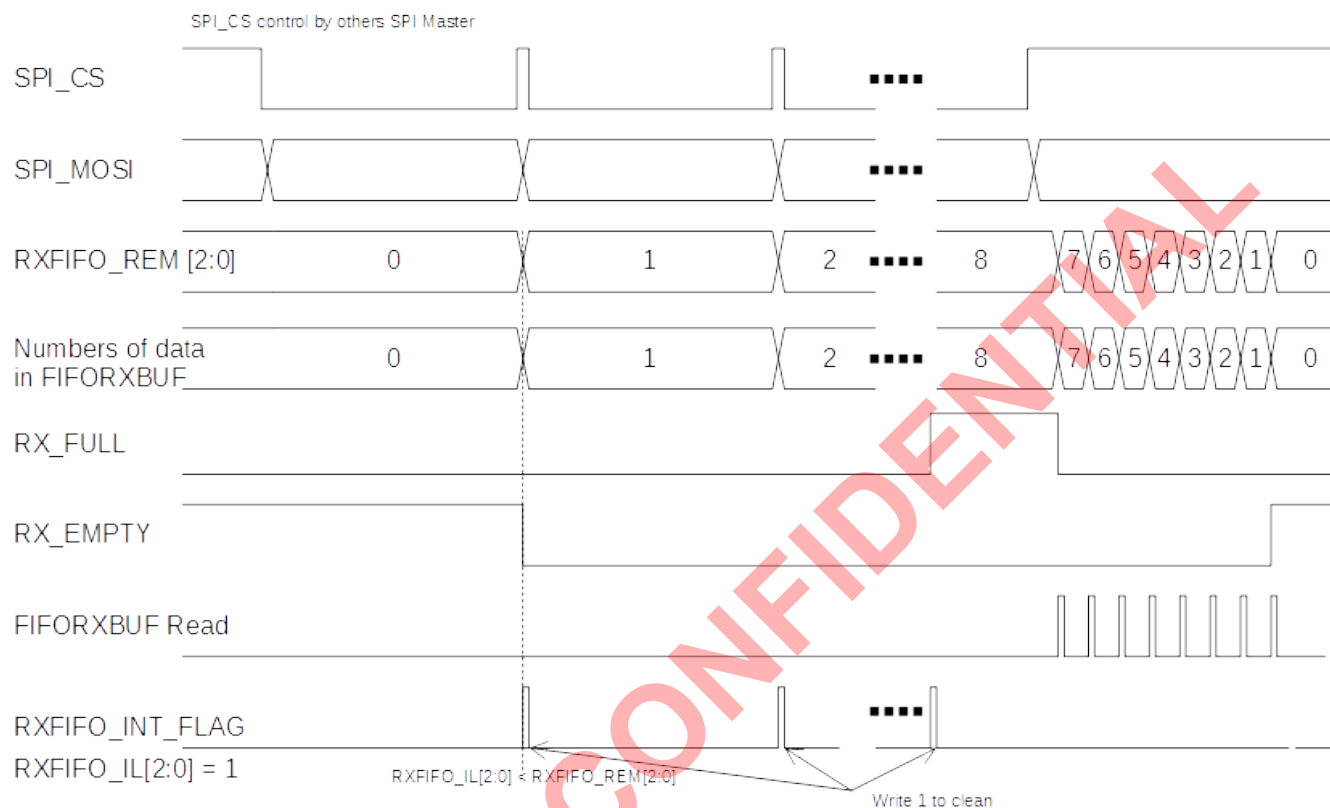


Figure 21.9 An example for SPI Slave received data 8 times

## 22. 8-bits ADC

A3107M0 has built-in 8-bits ADC (Analog to Digital Converter). The 8-bits ADC can be used to measure the carrier detect and RSSI.

### 22.1 FEATURES

- For carrier detect, RSSI measurement

### 22.2 BLOCK DIAGRAM

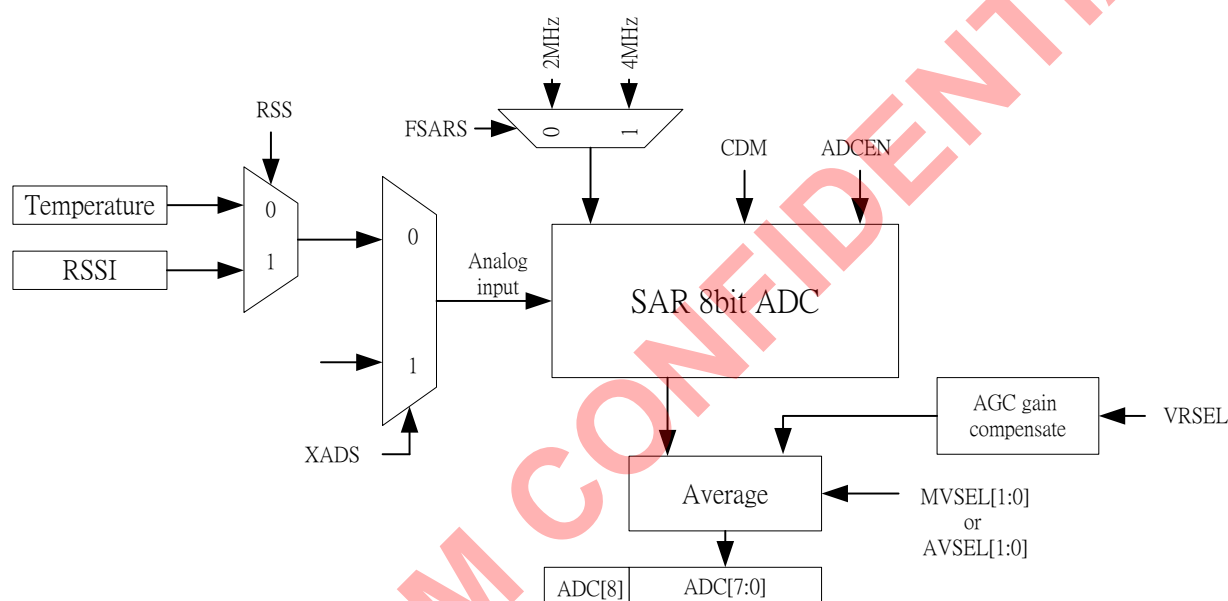


Figure 22.1 8-bits ADC block diagram

### 22.3 REGISTER

#### 22.3.1 Register List

Address	Name	DESCRIPTION
0x50001240	ADC_CTRL	8-bits ADC Control Register
0x50001244	ADC	8-bits ADC Value / Threshold Register

Table 22.1 PWM Register List

#### 22.3.2 Register Description

##### 8-bits ADC Control Register (Address: 0x50001240)

R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	RADC	AVSEL[1:0]		MVSEL[1:0]		--	RSM[1:0]	
R								
Reset	0	1	0	1	0	0	0	1
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	ERSS	--	FSARS	XADS	RSS	ARSSI	CDM	ADCM
R								
Reset	0	0	1	0	1	0	1	0

**RADC:** ADC read out average mode.

**[0]:** 1, 2, 4, 8 average mode. The average number is according to the setting of AVSEL.

[1]: 8, 16, 32, 64 average mode. The average number is according to the setting of MVSEL.

**AVSEL[1:0]:** ADC average times

[00]: No average.

[01]: Average 2 times.

[10]: Average 4 times.

[11]: Average 8 times.

**MVSEL[1:0]:** ADC average times

[00]: Average 8 times.

[01]: Average 16 times.

[10]: Average 32 times.

[11]: Average 64 times.

**RSM[1:0]:** RSSI margin = RTH – RTL.

[00]: 5.

[01]: 10.

[10]: 15.

[11]: 20.

**ERSS:** End enable for RSSI measurement

[0]: RSSI measurement continues until leave off RX mode.

[1]: RSSI measurement will end when carrier detected and ID code word received.

**FSARS:** ADC clock select.

[0]: 2MHz.

[1]: 4MHz.

**XADS:** ADC mode input signal select.

[0]: Convert RSSI.

[1]: Not support

**RSS:** Temperature/RSSI measurement select.

[0]: Temperature measurement.

[1]: RSSI or carrier-detect measurement.

**ARSSI:** Auto RSSI measurement while entering RX mode.

[0]: Disable.

[1]: Enable.

**CDM:** RSSI measurement mode.

[0]: Single mode.

[1]: Continuous mode.

**ADCM:** ADC measurement enable (Auto clear when done for single mode).

[0]: Disable measurement or measurement finished.

[1]: Enable measurement.

### 8-bits ADC Value / Threshold Register (Address: 0x50001244)

R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	--							--
R	--							ADC[8]
Reset								0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	RTH[7:0]							
R	ADC[7:0]							
Reset	0	0	0	0	0	0	0	0

**RTH[7:0]:** Carrier detect threshold.

**ADC[7:0]:** ADC output value of temperature, RSSI

**ADC[8]:** ADC output value for RSSI with AGC on.

## 22.4 FUNCTION DESCRIPTION

### 22.4.1 8-bits ADC function:

The 8-bits ADC provides RSSI and carrier detect measurement.

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

Table 22.2 8-bits ADC setting

### 22.4.2 RSSI measurement

A3107M0 supports 8-bits digital RSSI to detect RF signal strength. RSSI value is stored in ADC [7:0] (0x50001244). Figure 22.2 and Figure 22.3 shows a typical plot of RSSI reading as a function of input power. This curve is based on the current gain setting of A3107M0 reference code. A3107M0 automatically averages 8-times ADC conversion a RSSI measurement until A3107M0 exits RX MODE. Therefore, each RSSI measuring time is  $(8 \times 20 \times F_{ADC})$ . Be aware RSSI accuracy is about  $\pm 6\text{dBm}$ .

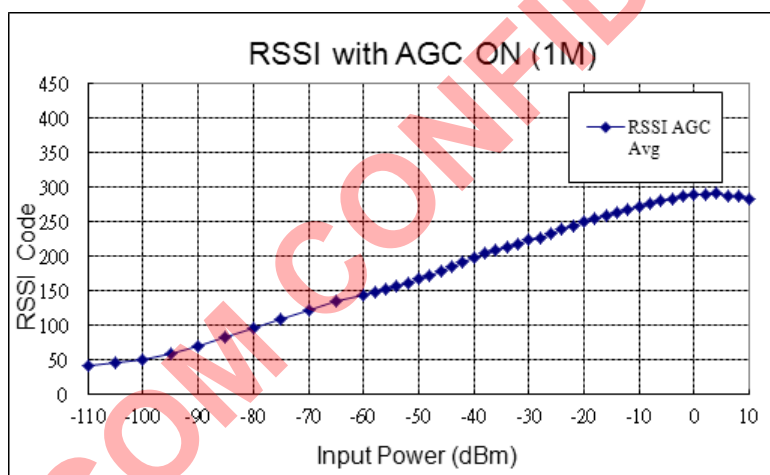


Figure 22.2 RSSI vs RF input Power (AGC On)

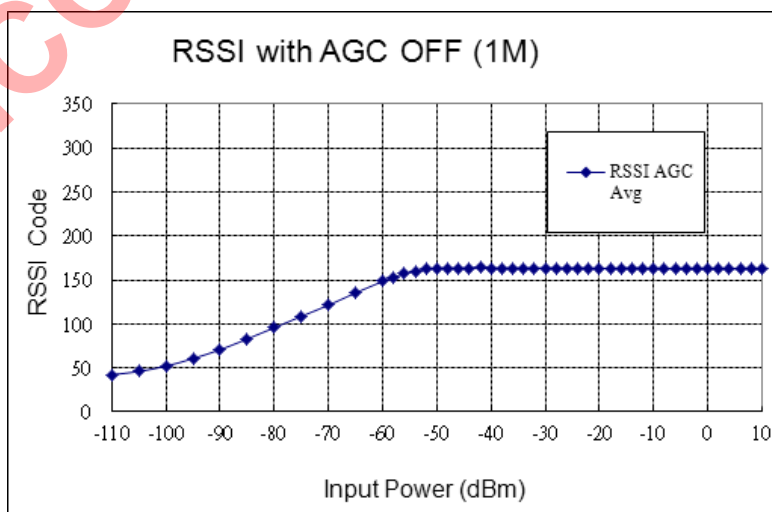


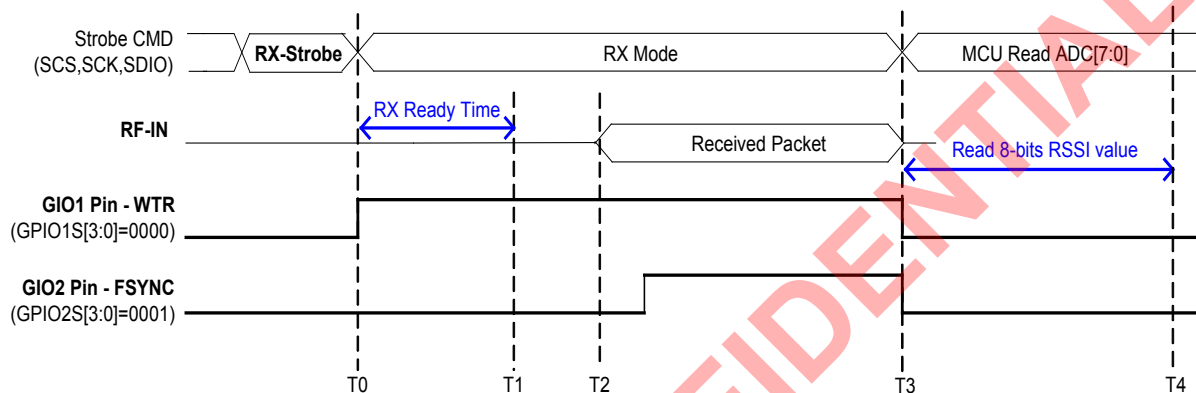
Figure 22.3 RSSI vs RF input Power (AGC Off)

**Auto RSSI measurement for TX Power:**

1. Set wanted Channel (0x500010C0)
2. Set RSS = 1, FSARS= 0 (2MHz ADC clock).
3. Enable ARSSI = 1.
4. Enter RX state.
5. In RX mode, 8-times average a RSSI measurement periodically.
6. Exit RX mode, user can read digital RSSI value from ADC[7:0] (0x50001244) for TX power.

**Note: If AGCE=1 and VRESL=1, RSSI value from ADC[8:0](0x50001244)**

In step 6, if A3107M0 is set in direct MODE, MCU shall let A3107M0 exit RX MODE within 40us to prevent RSSI inaccuracy.

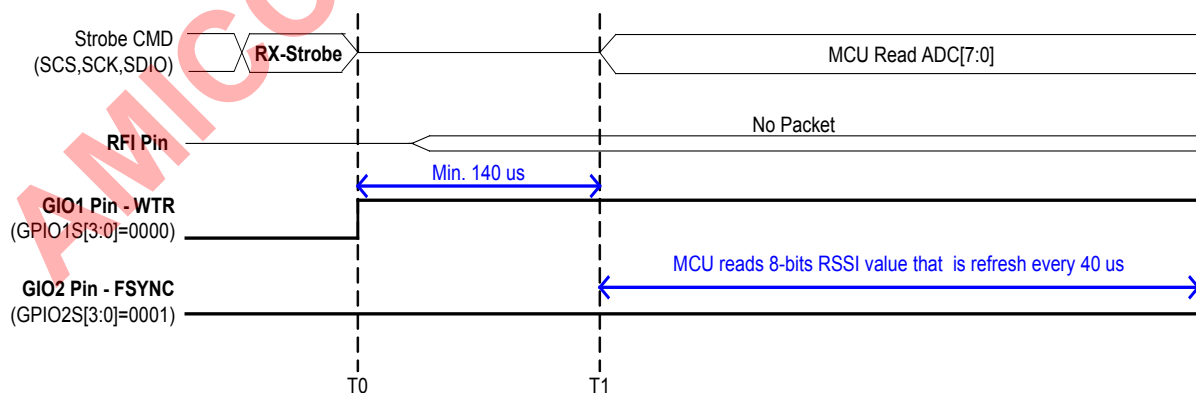


T0-T1: Settling Time  
T2-T3: Receiving Packet  
T3 : Exit RX mode automatically in FIFO mode  
T3-T4: MCU read RSSI value @ ADC [7:0]  
Figure 22.4 RSSI Measurement of TX Power

### Auto RSSI measurement for Background Power:

1. Set wanted Channel (0x500010C0)
2. Set RSS = 1, FSARS= 1 (4MHz ADC clock).
3. Enable ARSSI = 1.
4. Enter RX state.
5. MCU delays min. 140us.
6. Read digital RSSI value from ADC[7:0] (0x50001244) to get background power.
7. Send other Strobe command to let A3107M0 exit RX mode.

**Note: If AGCE=1 and VRESL=1, RSSI value from ADC[8:0](0x50001244)**



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

Figure 22.5 RSSI Measurement of Background Power

### 22.4.3 Carrier Detect

Base on RSSI measurement, user can extend its application to do carrier detect (CD). In Carrier Detect MODE, RSSI is re-fresh every 5us 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 =1.
  - (2-3) Enable ARSSI=1.
  - (2-4) Enter RX mode
  - (2-5) MCU enables a timer delay (min. 100 us).
3. MCU checks GIO2 pin.
  - (3-1) If  $ADC \geq CDTH$ , GIO2 = 1.
  - (3-2) If  $ADC \leq CDTH-CDM$ , GIO2=0
  - (3-3) If ADC locates in hysteresis zone, GIO2 = previous state.
4. Exit RX MODE.

A3107M0 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.

### 23. 12-bits SAR ADC

A3107M0 includes a 12-bits 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 internal temperature sensor.

**\*\*Note:** Under the A3107M0 with DC to DC device, 12-bits ADC function must be used in ByPass mode.

#### 23.1 FEATURES

- External Voltage measure from 8 channels
- Internal Temperature measure
- Programmable internal reference voltage: 1.8V, 2.0V, or external reference voltage
- Application measuring for internal temperature, sensor interface, battery operated system, data acquisition

#### 23.2 PINS DESCRIPTION

PIN	GPIO	TYPE	DESCRIPTION
ADCCH0	P0_18	INPUT	12-bits ADC analog input channel 0
ADCCH1	P0_19	INPUT	12-bits ADC analog input channel 1
ADCCH2	P0_08	INPUT	12-bits ADC analog input channel 2
ADCCH3	P0_09	INPUT	12-bits ADC analog input channel 3
ADCCH4	P0_12	INPUT	12-bits ADC analog input channel 4
ADCCH5	P0_13	INPUT	12-bits ADC analog input channel 5
ADCCH6	P0_14	INPUT	12-bits ADC analog input channel 6
ADCCH7	P0_15	INPUT	12-bits ADC analog input channel 7

Table 23.1 12-bits ADC pins description

#### 23.3 BLOCK DIAGRAM

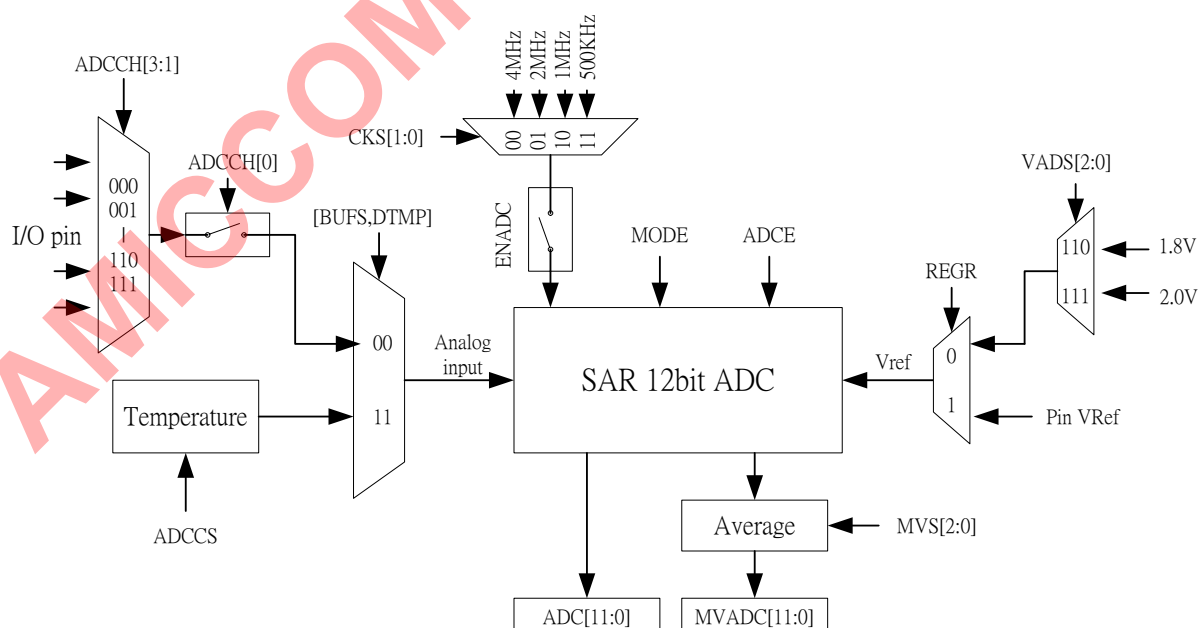


Figure 23.1 12-bits ADC block diagram



### 23.4 REGISTER

#### 23.4.1 Register List

Address	Name	DESCRIPTION
0x50008000	ADC_CTRL_1	12-bit ADC Control Register
0x50008004	ADC_CTRL_2	12-bit ADC Value Register
0x50008008	ADC_CH	12-bit ADC external voltage channel select
0x50000010	PWR_CTRL_3	12-bit ADC reference voltage setting

Table 23.2 PWM Register List

#### 23.4.2 Register Description

##### 12-bits ADC Control Register 1 (Address: 0x50008000)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W	ADC12RN	--						
R	--							
Reset	0	0						
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W	--							
R								
Reset	0							
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	ADCIE	VADS[2:0]			ADIVL	ADCYC	ENADC	DTMP
R	--	--			--	--	--	--
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	BUFS	CKS[1:0]		MODE	MVS[2:0]			ADCE
R	--	--						
Reset	0	1	0	0	0	0	0	0

**ADC12RN:** 12-bits ADC Reset. (Write "1" to reset).

[0]: Disable

[1]: Enable

**ADCIE:** 12-bits interrupt enable (only for single mode).

[0]: Disable

[1]: Enable

**VADS[2:0]:** ADC Reference Voltage.

[110]: 1.8V

[111]: 2.0V

[Others]: Reserved for internal usage only.

**ADIVL:** 12-bits ADC sargen initial value select.

[0]: 2048 (recommend)

[1]: 0

**ADCYC:** 12-bits ADC sargen clock counter select.

[0]: 31 (recommend)

[1]: 32

**ENADC:** Enable ADC clock source.

[0]: Disable

[1]: Enable

**DTMP:** 12-bits temperature select.

[0]: Disable

[1]: Enable

**BUFS:** input buffer select for 12 bit ADC.

[0]: Disable

[1]: Enable

**CKS[1:0]:** ADC clock selected.

[00]: MCU clock / 4

[01]: MCU clock / 8

[10]: MCU clock / 16

[11]: MCU clock / 32

**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.

[0]: Disable

[1]: Enable

### 12-bits ADC Control Register II (Address: 0x50008004)

R/W	Bit 31	-----	Bit 28	Bit 27	-----	Bit 16
W	--				--	
R	--				MVADC[11:0]	
Reset	0x0				0x000	
R/W	Bit 15	-----	Bit 12	Bit 11	-----	Bit 0
W	--				--	
R	--				ADC[11:0]	
Reset	0x0				0x000	

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

**ADC [11:0]:** ADC output value.  $\text{ADC input voltage} = \text{VDD\_ADC} * \text{MVADC [11:0]} / 4095 \text{ V}$

### 12-bits ADC Channel Register (Address: 0x50008008)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W								
R		--					ADCCH[3:0]	
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.

**ADCCH[0]:** ADC input enable.

[0]: Disable.  
[1]: Enable.

### Power Control Register 3 (Address: 0x500000010)

R/W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
W								
R	--							
Reset	--							
R/W	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W	ENVAD	--	--	ECH	LIBM	ADCCS	ECHA	REGR
R	ENVAD	--	--	ECH	LIBM	ADCCS	ECHA	REGR
Reset	0	0	0	1	0	0	1	0
R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	VTRB[3:0]				VMRB[3:0]			
R	--				--			
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--	IFAS	CGC	IWA	IWC	LBG	VCS	VCSW
R	--	--	--	--	--	--	--	--
Reset	0	0	0	0	0	0	0	0

**ADCCS:** Internal temperature measure enable.

[0]: Disable  
[1]: Enable

**REGR:** 12-bit ADC external reference voltage (Vref) input.

[0]: Disable  
[1]: Enable

## 23.5 FUNCTION DESCRIPTION

12-bit ADC provides two functions: one for analog input measurement and the other for internal temperature sensor measurement. The user can set bit DTMP, bit BUFS to switch to the test mode.

Bit		MODE
DTMP	BUFS	
0	0	Analog Input
1	1	Internal temperature Sensor

Figure 23.2 12-bit ADC function

### 23.5.1 A/D Reference Voltage (Vref):

The reference voltage supply to the A/D Converter can be supplied from the positive power supply pin, VDD\_ADC, an internal reference source derived from the Bandgap circuit. The desired selection is made using the VADS[2:0] bits in the ADC Control Register I and relevant pin function control bits.

REGR	VADS[2:0]	Vref	Description
0	110	VDD_ADC	ADC Reference Voltage comes from VDD_ADC=1.8V
0	111	VDD_ADC	ADC Reference Voltage comes from VDD_ADC=2.0V
1	x	Vref	External reference voltage from IC-Pin: RSSI (Max Vref ≤ REGI voltage)

Table 23.3 Reference Voltage

### 23.5.2 ADC Value Calculation:

12 Bit ADC can be used to measure the external input. The input voltage range is from 0V ~ VDD\_ADC. Please care the input voltage if set the input source from external input. Refer to the following formula, input voltage can be calculated from MVADC[11:0] or ADC[11:0].

$$\text{ADC input voltage} = \text{Vref} * \text{MVADC} [11:0] / 4095$$

### 23.5.3 A/D Conversion Time:

The time of A/D conversion 1 times is:

$$t_{conv} = \frac{4 \times 2^{CKS[1:0]}}{\text{Crystal Frequency}} \times 32 \mu$$

### 23.5.4 Analog Input (Voltage) Measurement

#### Measurement for Analog Input:

1. Set ADCCH (0x50008008) for selecting ADC channel.
2. Set MODE=1, ENADC=1, VADS=6, DTMP=0, BUFS=0 (0x50008000) to enable the SAR ADC with continue mode.
3. Set DCDC Control 2 register (0x50000018) =0 to switch to DCDC bypass mode (with DC to DC device function).
4. Set ADCE=1 (0x50008000) to enable ADC.
5. Delay two MVADC conversion time.
6. Read MVADC or ADC value.
7. Set ADCE=0 (0x50008000) to disable ADC.
8. Set DCDC Control 2 register (0x50000018) =241 to switch to DCDC mode (with DC to DC device function).

### 23.5.5 Internal Temperature Measurement

#### Measurement for Temperature:

1. Set RSS=0 (0x50001240) to select temperature sensor.
2. Set ADCCS=1 (0x50000010) to connect temperature sensor. Usually set PWR\_CTRL\_3 (0x50000010) = 0x0094000C.
3. Set MODE=1, ENADC=1, VADS=6, DTMP=1, BUFS=1 (0x50008000) to enable the SAR ADC with continue mode.
4. Set DCDC Control 2 register (0x50000018) =0 to switch to DCDC bypass mode.
5. Set ADCE=1 (0x50008000) to enable ADC.
6. Delay time that over two MVADC conversion time. For example, the delay time need over  $2 \times ((4 \times 1 / 16\text{MHz}) \times 32) \times 8 = 128\mu\text{s}$  if CKS=0, MVS=3.
7. Read MVADC or ADC (0x50008004) value.
8. Set ADCE=0 (0x50008000) to disable ADC.
9. Set DCDC Control 2 register (0x50000018) =241 to switch to DCDC mode.
10. Set ADCCS=0 (0x50000010) to disconnect temperature sensor.

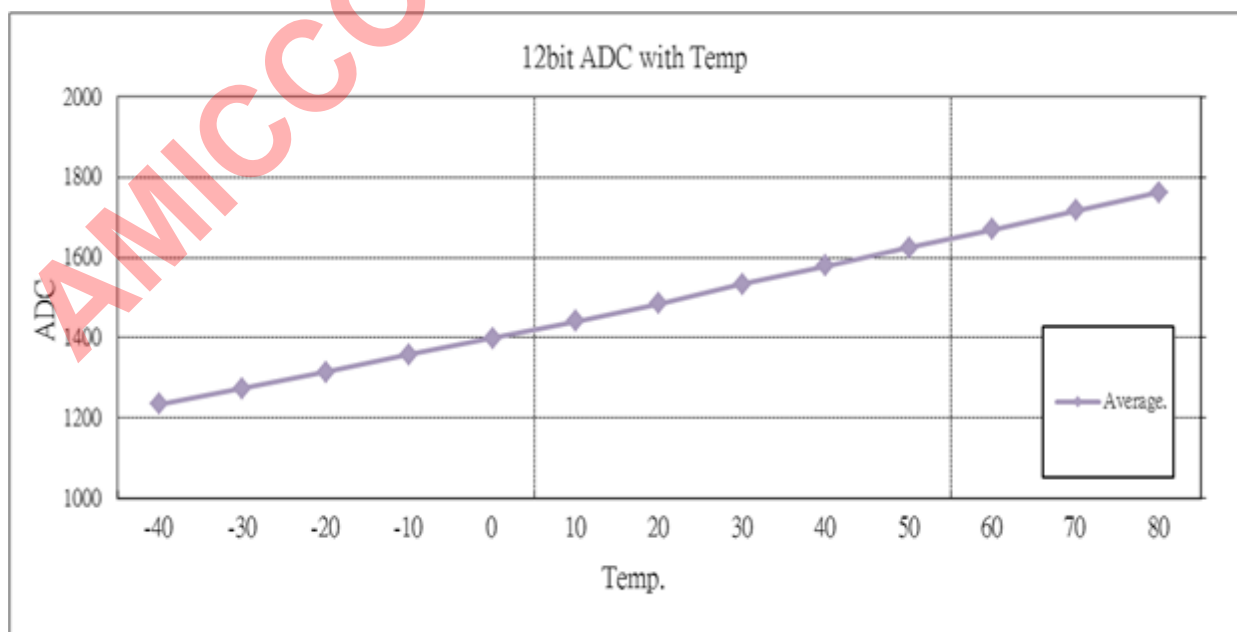


Figure 23.3 Typical 12-bit ADC temperature sensor characteristic curve

## 24. Battery Detect

The Battery detect function can be used to detect a low battery condition with setting voltage threshold. This is useful for warning that regulation voltages could no longer be maintained. Then MCU has the time to provide hardware protection of data stored in the program memory.

### 24.1 FEATURES

- Programmable voltage detection threshold in 8 levels

### 24.2 PINS DESCRIPTION

PIN	GPIO	TYPE	DESCRIPTION
REGI		INPUT	Regulator input

Table 24.1 Battery detect pin description

### 24.3 BLOCK DIAGRAM

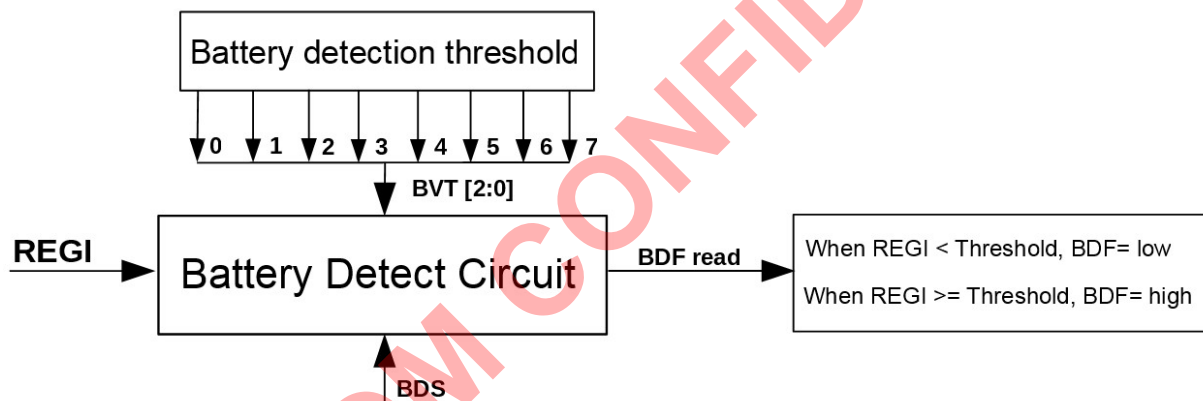


Figure 24.1 Battery detect Block Diagram

### 24.4 REGISTER

#### 24.4.1 Register list

Address	Name	DESCRIPTION
0x50000000	BDR	Battery Detect Register

Table 24.2 Battery detect Register List

#### 24.4.2 Register Description

**Battery Detect Register (Address: 0x50000000)**

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--			--	BVT[2:0]			BDS
R				BDF				
Reset	0	0	0	0	0	1	1	0

**BDF:** Battery detection flag.

**[0]:** Battery voltage less than threshold.

**[1]:** Battery voltage greater than threshold.

**BVT[2:0]:** Battery detection threshold.

**[000]:** 1.875V.

[001]: 1.95V.  
[010]: 2.025V.  
[011]: 2.1V.  
[100]: 2.175V.  
[101]: 2.25V.  
[110]: 2.325V.  
[111]: 2.4V.

**BDS[0]:** Battery detect enable.

[0]: Disable.

[1]: Enable. It will be clear after battery detection done.

## 24.5 FUNCTION DESCRIPTION

A3107M0 has a built-in battery detector to check supply voltage (REGI pin). Battery detect function has a control register (BDR). User can select Battery detection threshold by setting BVT[2:0], and the detecting range is 1.875V ~ 2.4V in 8 levels. Enable Battery detect function by setting BDS = 1, and MCU read bit BDF (Battery detection flag) to detect voltage condition. If Battery voltage less than threshold, BDF will output low; otherwise BDF will output high.

## 24.6 PROCEDURE

Below is the procedure to detect low voltage input (ex. Battery detection threshold 2.1V).

1. Set A3107M0 in STBY, PLL, TX or RX MODE.
2. Set BDR(0x50000000), BVT[2:0] = [011] and enable BDS = 1.
3. After 5 us, BDS is auto clear.
4. MCU reads BDF.
5. If REGI pin voltage  $\geq$  2.1V, BDF = 1 (battery high); otherwise BDF = 0 (battery low). Then back to step (2).

### 25. Power Management

A3107M0 has four power management modes: Normal mode and PMx mode. In normal mode, user selects different clock being MCU core clock.in CLKSEL[2:0] then enable CKSE. User adjusts MCU clocks depending on the required power consumption.

PM mode includes PM1, PM2 and PM3. User can set STOP=1 into PMx mode, so that the clock to the core is stopped and all digital peripherals also stop. MCU can be waked up by hardware reset, wakeup key, sleep timer and RTC.

	MCU speed	16MHz XTAL	16KB RAM	16KB LCD RAM	LVR	RF	Back to Normal (Wakeup)
Normal CKSE=0	16MHz	ON	ON	ON	X	X	X
Normal CKSE=1 (Low speed)	8/4/2/1 MHz, IRC/RTC	ON	ON	ON	X	X	X
PM1	STOP	OFF	ON	OFF	X	OFF	HW reset/ WUN/ Sleep Timer 0, 1/ RTC
PM2	STOP	OFF	ON	OFF	X	OFF	HW reset/ WUN/ Sleep Timer 0, 1/ RTC
PM3	STOP	OFF	ON	OFF	OFF	OFF	HW reset/ WUNIE/ Sleep Timer 0

X: don't care, it can turn on or off by user setting  
LVR: Low Voltage Regulator  
WUN: Wakeup pin, setting by WUN  
WUNIE: Only P0\_26~P0\_29 (Default enable PM3 wakeup function)

Table 25.1 A3107M0 supports Power Management Modes

#### 25.1 BLOCK DIAGRAM

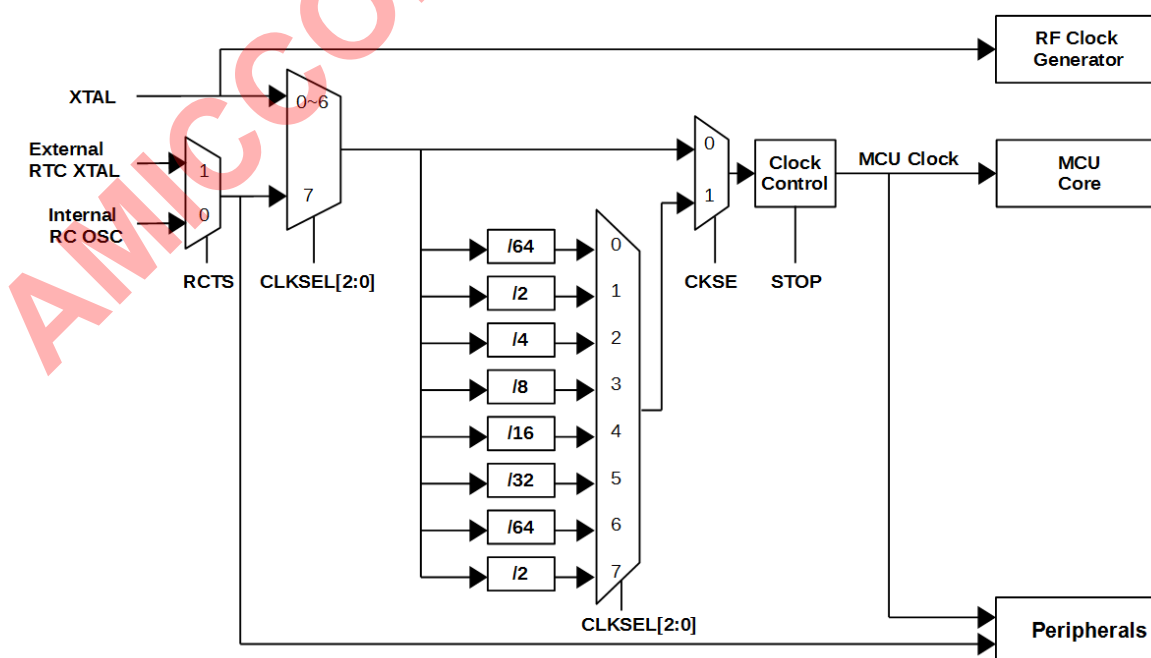


Figure 25.1 Whole chip clock sources

## 25.2 REGISTER

### 25.2.1 Register List

Address	Name	DESCRIPTION
0x5000000C	PWR_CTRL_2	Power Control Register 2
0x50000020	MCU_PWR_CTRL	MCU Power Control Register

Table 25.2 Power Management Register List

### 25.2.2 Register Description

#### Power Control Register 2 (Address: 0x5000000C )

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	1	ENAV	QDSA	ENDV	QDSD	--	--	--
R	--	--	--	--	--	--	--	--
Reset	1	1	0	1	0	0	0	0

**ENAV:** REGOA and REGOS connection.

**QDSA:** Quick discharge select for REGOA.

**ENDV:** REGOA is connected to REGOD.

**QDSD:** Quick discharge select for REGOD.

#### MCU Power Control Register (Address: 0x50000020)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--		CLKSEL[2:0]			--	STOP	CKSE
R	--		CLKSEL[2:0]			--	STOP	CKSE
Reset	0	0	0	0	0	0	0	0

**CLKSEL[2:0]:** Clock Select, Select clock source when CKSE=1.

**[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 MCU clock when CKSE=0; RTC div 2 as CPU clock when CKSE=1

**STOP:** Stop MCU clock

**[0]:** Disable

**[1]:** Enable

**CKSE:** Clock select enable

**[0]:** Disable clock select

**[1]:** Enable clock select

## 25.3 FUNCTION DESCRIPTION

### 25.3.1 Normal Mode

In normal mode, user selects different clock be MCU core clock.in CLKSEL[2:0] then enable CKSE .User adjusts MCU clocks depending on the required power consumption. CLKSEL[2:0] = 001b ~ 110b, the MCU core clock is the clock sources divided 2 ~ 64. User could adjust the MCU speed to trade-off between the performance and the power consumption. **BEWARE, please choose CLKSEL firstly then enable CKSE to avoid glitch.**

To perform Normal mode, follow these steps:



1. CKSE = 0
2. Set CLKSEL[2:0] to select clock source
3. CKSE = 1

### 25.3.2 PM Mode

User can STOP MCU clock and some circuit, peripherals to save current consumption:

PM Mode	PWR_CTRL_2 0x5000000C	Digital Power	Analog Power
PM1	0xD0	ON	ON
PM2	0xF0	ON	OFF
PM3	0xF8	OFF	OFF

Table 25.3 Power control register 2 setting for PM Mode

To perform PMx, follow these steps:

1. If CKSE = 1, set it to 0
2. Set PWR\_CTRL\_2 (0x5000000c) value by PMx mode. Please refer
3. Set FALSHCTRL (0x50000004) = 0x0000A495
4. Set ADC\_CTRL\_1 (0x50008000) = 0x000001FE to set VADS = 0
5. STOP = 1 to enter PMx mode

When wakeup MCU, please follow these steps:

1. Set ADC\_CTRL\_1 (0x50008000) = 0x000061FE to set VADS = 6
2. Set FALSHCTRL (0x50000004) = 0x00006495

### 26. MPULCD

The MPULCD controller driver pixels data to LCD panel directly and automatically. The MPULCD has two 8KB SRAM, MPULCD\_SRAM, for DMA function that address base 0x20004000 and offset 0x0000 and 0x2000. MPULCD\_SRAM also are general SRAM when DMA function not active.

DMA memory	Address Base	Offset	Size
MPULCD_SRAM_0	0x20004000	0x0000	8KB
MPULCD_SRAM_1	0x20004000	0x2000	8KB

Table 26.1 SRAM for MPULCD DMA function

#### 26.1 FEATURE

- Support 8-bits 8080 parallel interface, 16-bits 8080 parallel interface and serial interface
- Build-in two 8KB DMA SRAM

#### 26.2 PINS DESCRIPTION

PIN	GPIO	TYPE	DESCRIPTION
LCD_TE	P0_11	INPUT	Tearing effect input pin to synchronize frame rate.
LCD_CSX	P0_16	OUTPUT	Chip Selection Pin
LCD_D/CX	P0_17	OUTPUT	Display Data / Command selection Pin
LCD_RDX	P0_12	OUTPUT	Read Enable pin
LCD_WRX = s_LCD_SCL	P0_20	OUTPUT	Write Enable in parallel interface Serial clock pin in serial Interface
LCD_D[0] = s_LCD_SDA	P0_15	INPUT / OUTPUT	D[15:0] are 16-bits 8080 parallel interface data bus D[7:0] are 8-bits 8080 parallel interface data bus D[0] is the serial input/output signal in serial interface
LCD_D[1]	P0_14		
LCD_D[2]	P0_13		
LCD_D[3]	P0_27		
LCD_D[4]	P0_26		
LCD_D[5]	P0_25		
LCD_D[6]	P0_24		
LCD_D[7]	P0_21		
LCD_D[8]	P0_31		
LCD_D[9]	P0_30		
LCD_D[10]	P0_29		
LCD_D[11]	P0_28		
LCD_D[12]	P0_05		
LCD_D[13]	P0_04		
LCD_D[14]	P0_19		
LCD_D[15]	P0_18		

Table 26.2 MPULCD pin description

### 26.3 BLOCK DIAGRAM

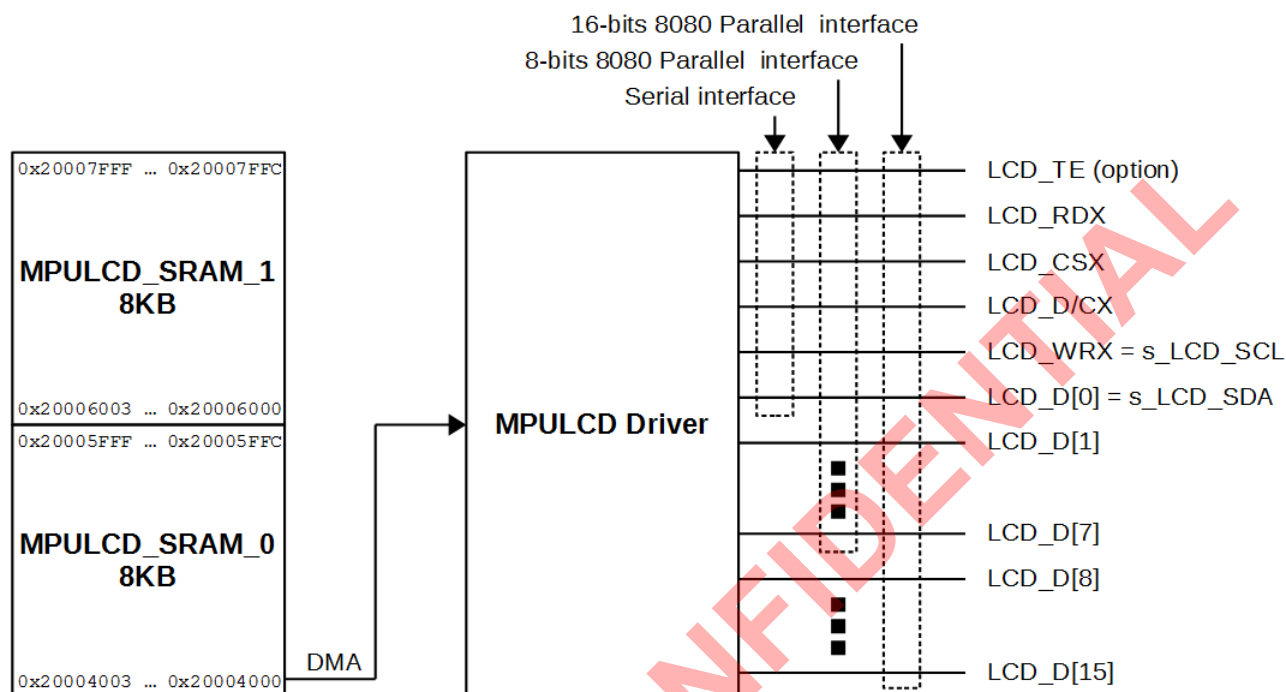


Figure 26.1 MPULCD Block Diagram

### 26.4 REGISTER

#### 26.4.1 Register List

Address	Name	DESCRIPTION
0x5000A000	MPULCD_CTRL	MPULCD Control Register
0x5000A004	MPULCD_WIDTH	MPULCD SRAM Width Register
0x5000A008	MPULCD_HEIGHT	MPULCD SRAM Height Register
0x5000A010	MPULCD_CYCLE	MPULCD Stretch Cycle Register
0x5000A014	MPULCD_DATA	MPULCD Command Data Register
0x5000A018	MPULCD_OFFSET	MPULCD DMA Start Offset Register

Table 26.3 MPULCD Register List

#### 26.4.2 Register Description

##### MPULCD Control Register (Address: 0x5000A000)

R/W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W	Data_Format[1:0]		INT	nCS_IO	Data_Interface[2:0]			--
R								
Reset	0	0	0	0	0	0	0	0
R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--	TE_Pol	TE_En	CMD	Enable	--	RDnWR	A0_Pol
R						Busy		
Reset	0	0	0	0	0	0	0	0

**Data\_Format[1:0]:** The data transfer format when interface is 8-bits 8080 Parallel Interface.

[00]: RGB565.

[01]: RGB666 format 1.

[10]: RGB666 format 2.

**INT:** MPULCD interrupt enable.

[0]: Interrupt Disable.

[1]: Interrupt Enable.

**nCS\_IO:** Control MPU LCD control pin LCD\_CSX.

[0]: Disable.

[1]: Enable. This bit enables need before then MPULCD Enable.

**Data\_Interface[2:0]:** MPULCD transfer interface:

[001]: 8-bits 8080 Parallel Interface.

[010]: 16-bits 8080 Parallel Interface.

[100]: 8-bits serial interface.

**Others:** Not support.

**TE\_Pol:** TE Trigger Level.

[0]: Trigger when LCD\_TE is low.

[1]: Trigger when LCD\_TE is high.

**TE\_En:** TE Trigger Enable.

[0]: Disable TE Trigger.

[1]: Enable TE Trigger. The trigger source from LCD control pin LCD\_TE.

**CMD:** MPULCD transfer data select.

[0]: DMA mode. MPULCD will transfer data from MPULCD\_SRAM (0x20004000).

[1]: Command mode. Only transfer one data from MPULCD\_DATA (0x5000A014).

**Enable:** MPULCD enable.

[0]: Disable. MPULCD stop transfer data.

[1]: Enable. MPULCD start transfer data.

This bit is not auto-clean when transfer finish. User need clean Enable = 0 before next time set Enable=1.

**Busy:** MPULCD busy flag.

[0]: MPULCD is idle, not transfer data.

[1]: MPULCD is busy, still transfer data.

**RDnWR:** Control MPULCD control pin LCD\_RDX and LCD\_WRX.

[0]: Write mode. LCD\_RDX is high and LCD\_WRX is low when MPU\_LCD transfer.

[1]: Read mode. LCD\_RDX is low and LCD\_WRX is high when MPU\_LCD transfer.

**A0\_Pol:** Control MPULCD control pin LCD\_D/CX.

[0]: LCD\_D/CX is low when MPU\_LCD transfer.

[1]: LCD\_D/CX is high when MPU\_LCD transfer.

### MPULCD SRAM Width Register (Address: 0x5000A004)

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

**Width[7:0]:** Set width by pixel.

### MPULCD SRAM Height Register (Address: 0x5000A008)

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

**Height[7:0]:** Set height by pixel.

The MPULCD DMA will transfer Width[7:0] \* Height[7:0] pixel = Width[7:0] \* Height[7:0] \* 2 bytes.

### MPULCD\_CYCLE (MPULCD Stretch Cycle Register)

Address: 0x5000A010

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	--	Cycle[6:0]						
R								
Reset	0	0	0	0	0	1	0	1

**Cycle[6:0]:** Set to adjust the read / write pulse timing width =  $2 * (\text{Cycle}[6:0] + 1) / (\text{MCU clock frequency})$ .

### MPULCD Command Data Register (Address: 0x5000A014)

R/W	Bit 15	-----	Bit 0
W	Data[15:0]		
R	Data[15:0]		
Reset	0x0000		

**Data[15:0]:** MPULCD data for command mode (CMD=1). MPULCD only transfer 8-bits data in 8-bits 8080 parallel interface and serial interface.

### MPULCD DMA Start Offset Register (Address: 0x5000A018)

R/W	Bit 15	-----	Bit 0
W	Offset[15:0]		
R	Offset[15:0]		
Reset	0x0000		

**Offset[15:0]:** DMA access memory address offset. DMA will transfer data start from address  $0x20004000 + \text{Offset}[15:0]$ . Offset[15:0] need aligned by 4bytes. This means Offset[1:0] = 00 always.

## 26.5 FUNCTION DESCRIPTION

### 26.5.1 MPULCD DMA data Format

MPULCD\_SRAM only support RGB565 format for DMA function. Each 32-bits data include 2 pixels data.

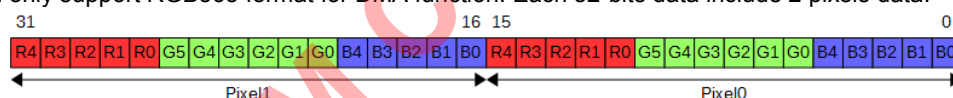


Figure 26.2 RGB565 format

### 26.5.2 MPULCD output interface

MPU LCD support 8-bits 8080 parallel, 16-bits 8080 parallel and serial Interface.

#### 26.5.2.1 8-bits 8080 Parallel Interface

This interface support data out format is RGB565 and RGB666:

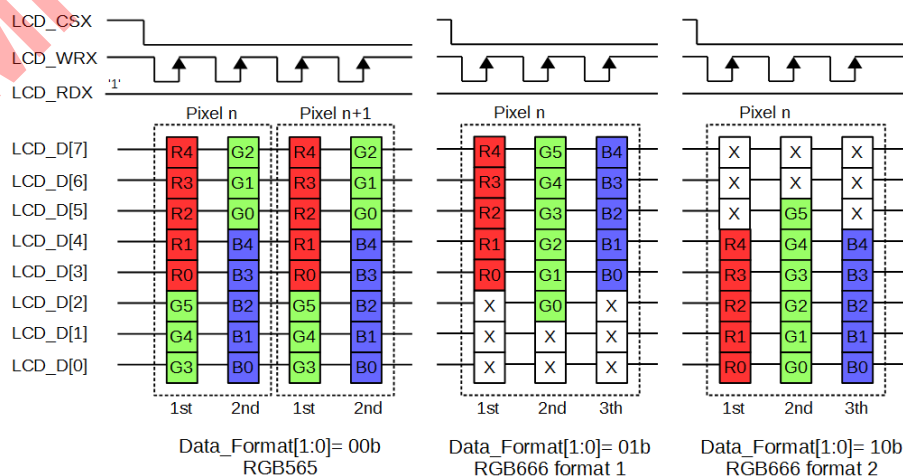


Figure 26.3 8-bits 8080 parallel interface data out

### 26.5.2.2 16-bits 8080 Parallel Interface

This interface only support data out format RGB565, finish a pixel with one LCD\_WRX cycle.

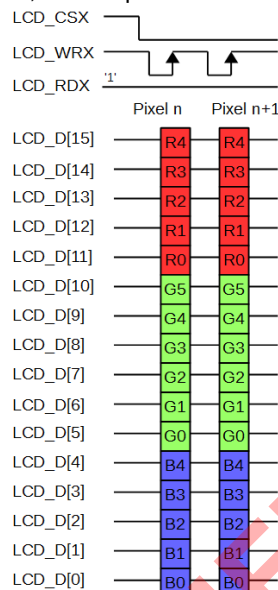


Figure 26.4 16-bits 8080 parallel interface data out

### 26.5.2.3 Serial interface (SPI)

This interface only support data out format RGB565, finish a pixel data with 16 s\_LCD\_SCL cycle.

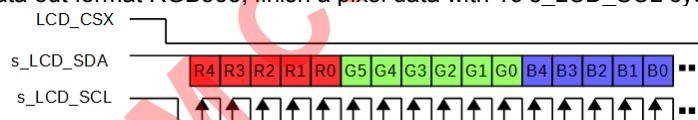


Figure 26.5 Serial Interface data out

## 26.6 PROCEDURE

### 26.6.1 MPULCD initial

- Step1: Set MPULCD\_CYCLE (0 is fastest)
- Step2: Set MPULCD\_CTRL. GPIO will change to MPULCD IO when nCS\_IO=1.

### 26.6.2 MPULCD write data with command mode

- Step1: Set CMD=1 (BIT4 of MPULCD\_CTRL)
- Step2: Set RDnWR=0 (BIT1 of MPULCD\_CTRL) and A0 (BIT0 of MPULCD\_CTRL)
- Step3: Write data to MPULCD\_DATA
- Step4: Set Enable=1 (BIT3 of MPULCD\_CTRL)
- Step5: Wait Busy (BIT2 of MPULCD\_CTRL) from 1 to 0
- Step6: Set Enable=0

### 26.6.3 MPULCD read data with command mode

- Step1: Set CMD=1 (BIT4 of MPULCD\_CTRL)
- Step2: Set RDnWR=1 (BIT1 of MPULCD\_CTRL) and A0 (BIT0 of MPULCD\_CTRL)
- Step3: Set Enable=1 (BIT3 of MPULCD\_CTRL)
- Step4: Wait Busy (BIT2 of MPULCD\_CTRL) from 1 to 0
- Step5: Read data from MPULCD\_DATA
- Step6: Set Enable=0

#### **26.6.4 MPULCD write data with DMA mode**

- Step1: Write data to MPULCD\_SRAM
- Step2: Set MPULCD\_WIDTH and MPULCD\_HEIGHT
- Step3: Set CMD=0 (BIT4 of MPULCD\_CTRL)
- Step4: Set RDnWR=0 (BIT1 of MPULCD\_CTRL) and A0 (BIT0 of MPULCD\_CTRL)
- Step5: Set Enable=1 (BIT3 of MPULCD\_CTRL)
- Step6: Wait Busy (BIT2 of MPULCD\_CTRL) from 1 to 0
- Step7: Set Enable=0

AMICCOM CONFIDENTIAL

## 27. Flash and IAP

A3107M0 build-in 256KB flash ROM. There have two flash chips, F1M and F2M. Each flash chip has total 32 sectors. Each sector has total 32 pages. Each page has total 32 words (32-bits, 4Bytes). Each word has 32 bits. See Table 27.1 Flash organization as below:

Item	Unit			
	Sectors	Pages	Words	Bytes
Flash Chip	32	1K	32K	128K
Flash Sector	1	32	1K	4K
Flash Page		1	32	128
Word			1	4

Table 27.1 Flash organization

Flash ROM support IAP (in application program). User can erase/write flash in application to store data, update code ...etc.

### 27.1 REGISTER

#### 27.1.1 Register List

Address	Name	DESCRIPTION
0x4001F104	FLASH_CTRL	Flash Control Register
0x4001F108	FLASH_PWE	Flash Write Enable Register
0x4001F10C	FLASH_ADDR	Flash Address Register
0x4001F110	FLASH_PWD0	Flash Password0 Register
0x4001F114	FLASH_PWD1	Flash Password1 Register
0x4001F118	FLASH_KEY0	Flash Key0 Register
0x4001F11C	FLASH_KEY1	Flash Key1 Register
0x4001F200	FLASH_BUFFER	Flash Write Buffer Register
0x4001F300	FLASH_LOCK	Flash Lock Register

Table 27.2 Flash register list

#### 27.1.2 Register Description

##### Flash Control Register (Address: 0x4001F104)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	CE	F2M	F1M	--	SERASE	PERASE	--	WRITE
R								
Reset	1	0	0	0	0	0	0	0

**CE:** Flash Chip Enable.

**[0]:** Disable (Not Support)

**[1]:** Enable

**F2M:** Second Flash ROM (address 0x00020000 ~ 0x0003FFFF) select.

**[0]:** Second Flash ROM not selected.

**[1]:** Second Flash ROM selected.

**F1M:** First Flash ROM (address 0x00000000 ~ 0x0001FFFF) select.

**[0]:** First Flash ROM not selected.

**[1]:** First Flash ROM selected.

**SERASE:** Flash Sector Erase.

**[0]:** No effect.

**[1]:** Flash sector erase active.

**PERASE:** Flash Page Erase.

**[0]:** No effect.

**[1]:** Flash page erase active.



**WRITE:** Flash Page write.  
**[0]:** No effect.  
**[1]:** Flash page write active.

### Flash Enable Register (Address: 0x4001F108)

R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
W	PWE	--	LENGTH[5:0]					
R								
Reset	0	0	0	0	0	0	0	0

**PWE:** Flash erase, write enable.  
**[0]:** Flash only can read.  
**[1]:** Flash can erase, write.

**LENGTH[5:0]:** Write LENGTH[5:0] words to flash from FLASH\_BUFFER. The range of LENGTH[5:0] is 1~32.

### Flash Address Register (Address: 0x4001F10C)

R/W	Bit 31	-----	Bit 0
W	ADDR		
R			
Reset	0x00000000		

**ADDR:** Flash IAP controller will erase or write flash by ADDR. The **BIT0** and **BIT1** need to keep to '0' of ADDR.

### Flash Password0 Register (Address: 0x4001F110)

R/W	Bit 31	-----	Bit 0
W	PWD0		
R			
Reset	0x00000000		

**PWD0:** Flash IAP will skip if PWD0 is not equal KEY0.

### Flash Password1 Register (Address: 0x4001F114)

R/W	Bit 31	-----	Bit 0
W	PWD1		
R			
Reset	0x00000000		

**PWD1:** Flash IAP will skip if PWD1 is not equal KEY1. PWD1 will set to reset value when Flash IAP finish or not match KEY1.

### Flash Key0 Register (Address: 0x4001F118)

R/W	Bit 31	-----	Bit 0
W	KEY0		
R			
Reset	0xAAAA5555		

**KEY0:** User can set KEY0 before Flash IAP.

### Flash Key1 Register (Address: 0x4001F11C)

R/W	Bit 31	-----	Bit 0
W	KEY1		
R			
Reset	0xAA55AA55		

**KEY1:** User can set KEY1 before Flash IAP.

### Flash Write Buffer (Address: 0x4001F200 ~ 0x4001F27F)

R/W	Word 31	-----	Word 0
W	BUFFER[31:0]		

R	
Reset	--

**BUFFER[31:0]:** 32 words (128 Bytes) buffer to buffering data for page write.

### Flash Lock Register (Address: 0x4001F300 ~ 0x4001F307)

R/W	Bit 63	-----	Bit 32
W	LOCK[1]		
R	LOCK[1]		
Reset	0xFFFFFFFF		
R/W	Bit 31	-----	Bit 0
W	LOCK[0]		
R	LOCK[0]		
Reset	0xFFFFFFFF		

**LOCK[1]:** Each bit can lock 4KB (1 sector) flash. Protect flash can't write / erase between address 0x20000~0x3FFFF.

**LOCK[0]:** Each bit can lock 4KB (1 sector) flash. Protect flash can't write / erase between address 0x00000~0x1FFFF.

User needs clean lock bit to 0 before flash erase / write. LOCK[1] and LOCK[0] will reset to reset value when IAP finish.

## 27.2 FUNCTION DESCRIPTION

### 27.2.1 Page Write

User can write data to flash with Page Write. A page will be wrote when page write finish, even only write 1 word to flash. This function can't over page.

### 27.2.2 Page Erase

User can erase 1 page flash with Page Erase. The flash need erase before Page Write. After Page Erase, all words will erase to 0xFFFFFFFF.

### 27.2.3 Sector Erase

This function is same as Page Erase but erase 1 sector.

## 27.3 PROCEDURE

User needs to set FLASH\_KEY0 and FLASH\_KEY1 before Flash IAP.

User needs to unlock flash protect when do flash IAP. User can refer the way as below that show unlock 1 sector of address:

```
unlock = (1 << ((address & 0x1FFFF) / 4096)) ^ 0xFFFFFFFF;
if (address < 0x20000)
    LOCK[0] = unlock;
else
    LOCK[1] = unlock;
```

The general Flash IAP show as below:

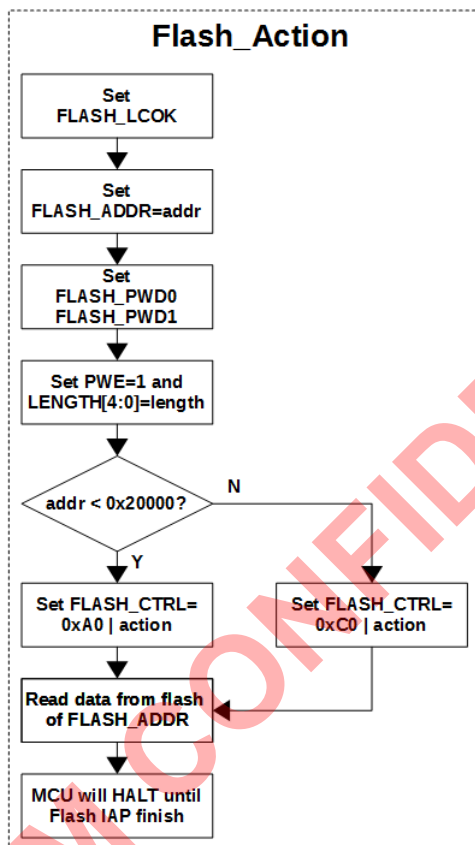


Figure 27.1 Flash\_Action flow chart

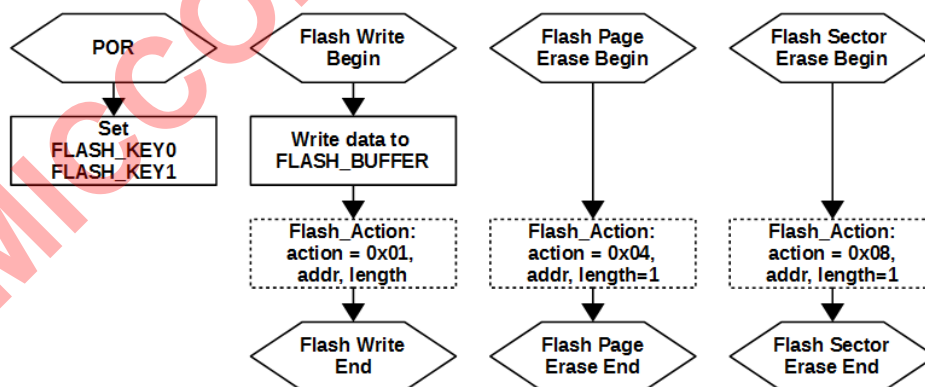


Figure 27.2 Flash IAP flow chart

## 28. Encryption and Authentication

For Bluetooth Low Energy application, it uses AES-128 link layer encryption block with Counter MODE CBC MAC defined in IETF RFC 3610. A3107M0 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.

### 28.1 AES

AES (Advanced Encryption Standard) is a symmetric block cipher on 128-bits data blocks, it consists 10 encryption rounds during encryption process. Figure 28.1 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 block diagram of the AES with 128 bit data is shown below in Figure 28.1.

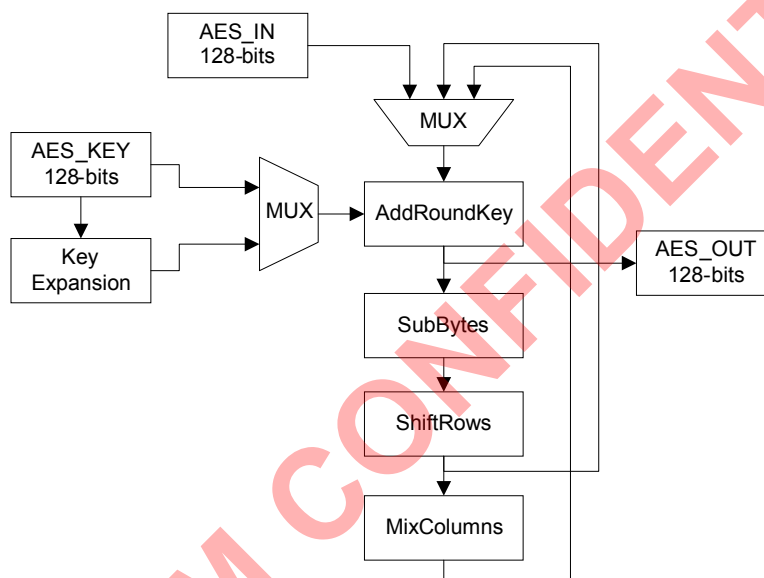


Figure 28.1 Structure of the AES-128 Core

#### 28.1.1 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.

#### 28.1.2 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.

#### 28.1.3 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.

#### 28.1.4 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.

### 28.1.5 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 28.2.

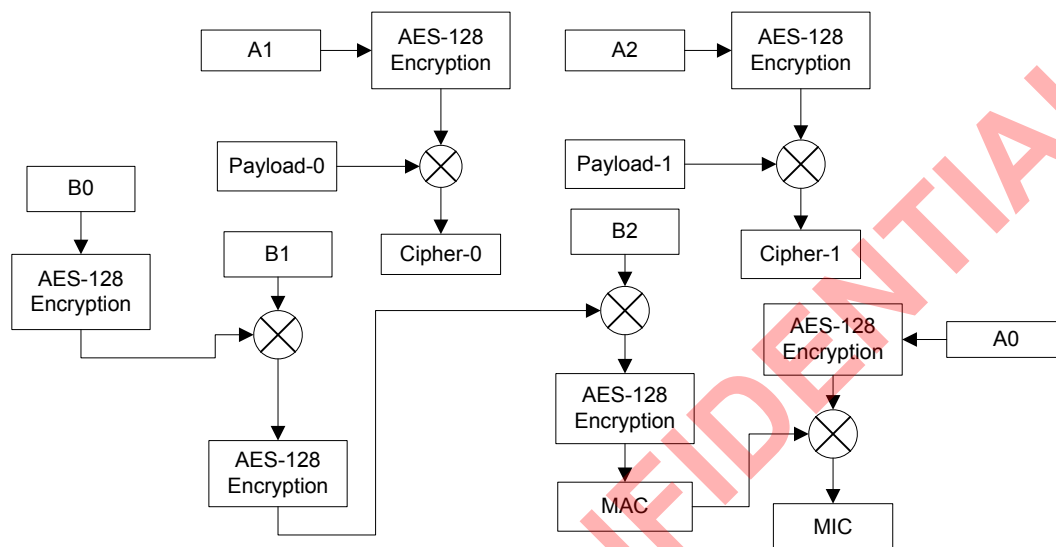


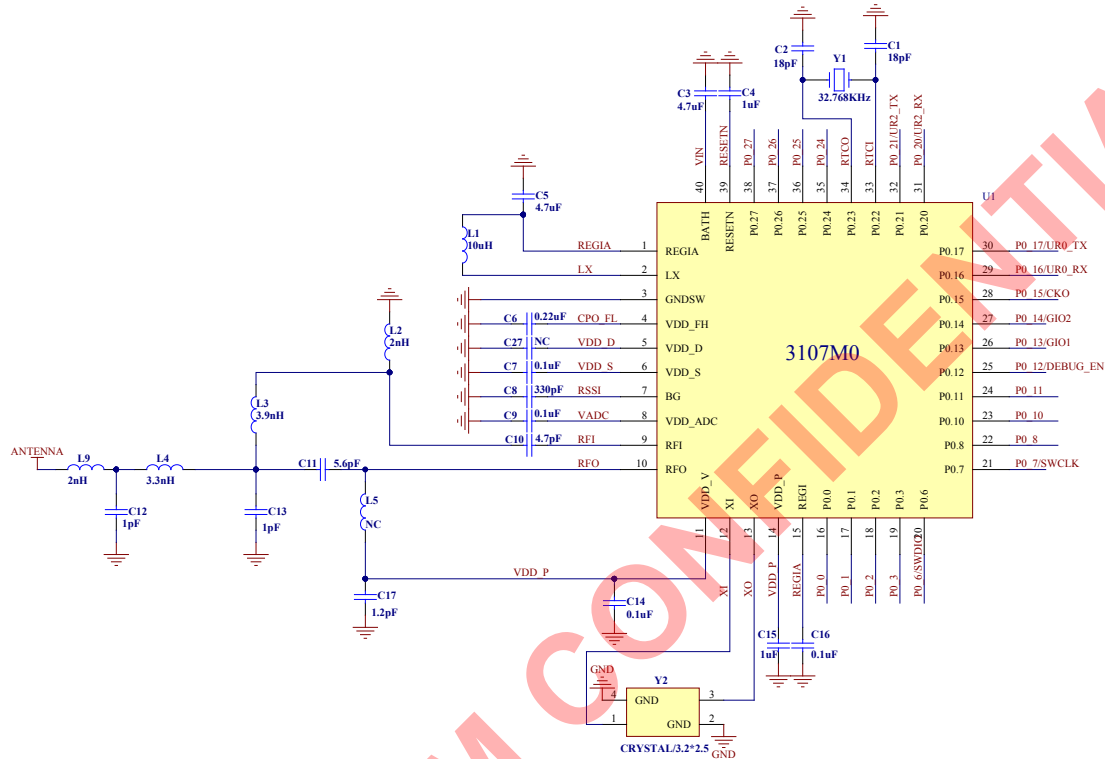
Figure 28.2 CCM Encryption Procedure

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.

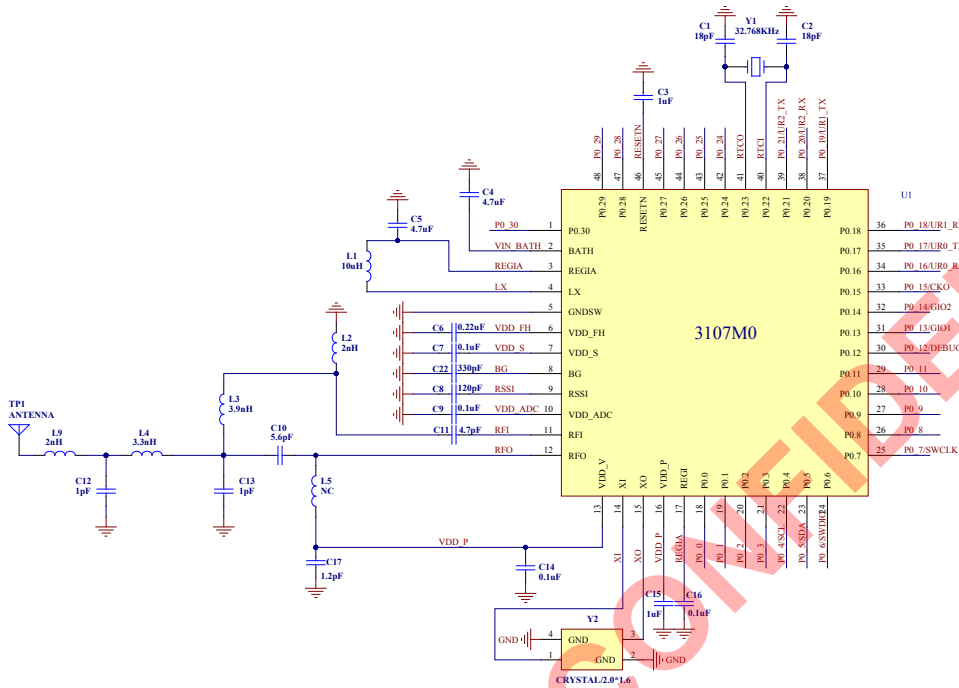
## 29. Application circuit

**Below are AMICCOM's ref. design circuits. For more details, please contact AMICCOM's FAE for more details.**

### A3107M0 QFN40 5x5 Package



A3107M0 QFN48 6x6 Package



### 30. Abbreviations

ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AGC	Automatic Gain Control
AIF	Auto IF
BER	Bit Error Rate
BLE	Bluetooth Low Energy
BW	Bandwidth
CD	Carrier Detect
CPU	Central processing unit
CRC	Cyclic Redundancy Check
DC	Direct Current
DC-DC	Direct Current- Direct Current Regulator
FC	Frequency Compensation
FEC	Forward Error Correction
FIFO	First in First out
FSK	Frequency Shift Keying
GPIO	General-purpose input/output
I <sup>2</sup> C	Inter-Integrated Circuit
ICE	In Circuit Emulator
ID	Identifier
IF	Intermediate Frequency
I/O	Input/output
IRQ	Interrupt request
ISM	Industrial, Scientific and Medical
KB	1024 bytes
Kbps	Kilobits per second
LDO	Low Drop Out Regulator
LNA	Low Noise Amplifier
LO	Local Oscillator
LSB	Least-significant bit/byte
MCU	Micro Controller Unit
MISO	Master in, slave out
MOSI	Master out, slave in
PA	Power Amplifier
PFD	Phase Frequency Detector for PLL
PLL	Phase Lock Loop
PM1	Power Mode1
PM2	Power Mode2
PM3	Power Mode3
POR	Power on Reset
PWM	Pulse width modulation
RF	Radio frequency
RSSI	Received Signal Strength Indicator
RTC	Real-time clock
RX	Receiver
RXLO	Receiver Local Oscillator
SCK	Serial clock
SPI	Serial to Parallel Interface
SRAM	Static random-access memory
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

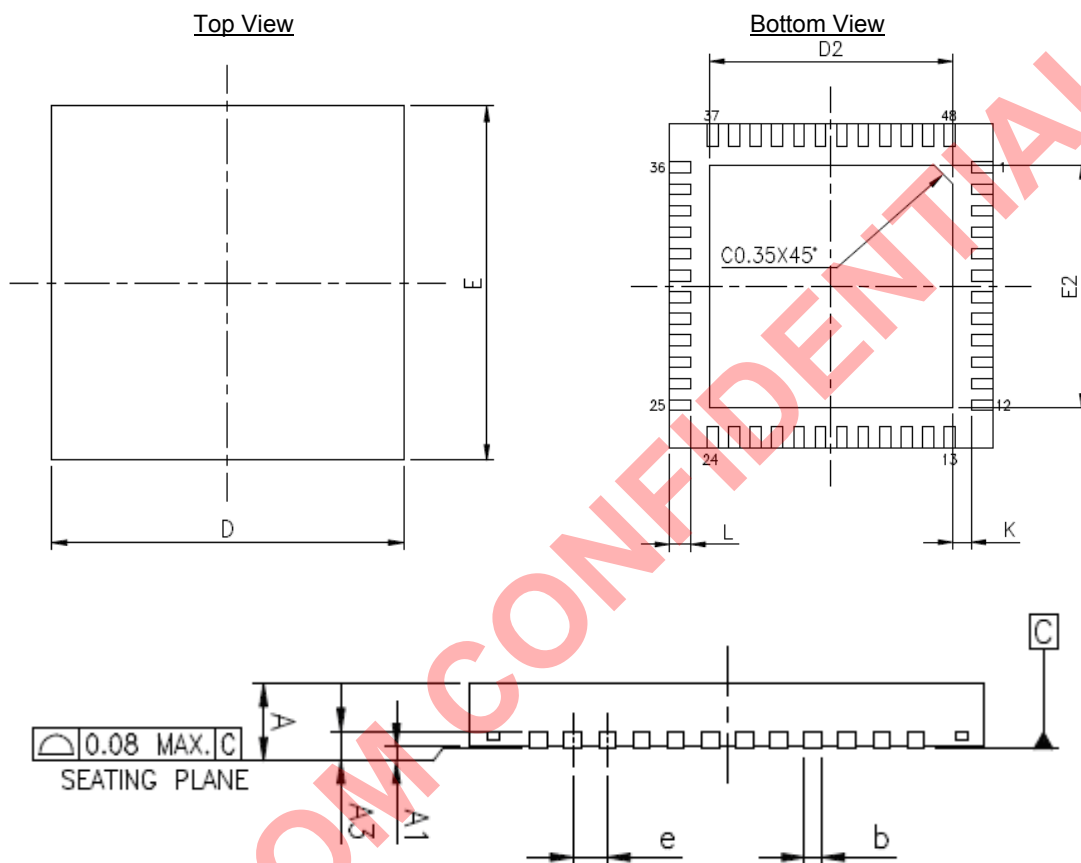


### 31. Ordering Information

Part No.	Package	Units Per Reel / Tray
A31U07F810GAQ6C/Q	QFN48L with 31I/O, DC/DC Pb Free, Tape & Reel, -40°C ~ 85°C	3K
A31U07F810GAQ6C	QFN48L with 31I/O, DC/DC Pb Free, Tray, -40°C ~ 85°C	490EA
A31U07F8102AQ5A /Q	QFN40L with 23I/O, DC/DC Pb Free, Tape & Reel, -40°C ~ 85°C	3K
A31U07F8102AQ5A	QFN40L with 23I/O, DC/DC Pb Free, Tray, -40°C ~ 85°C	490EA

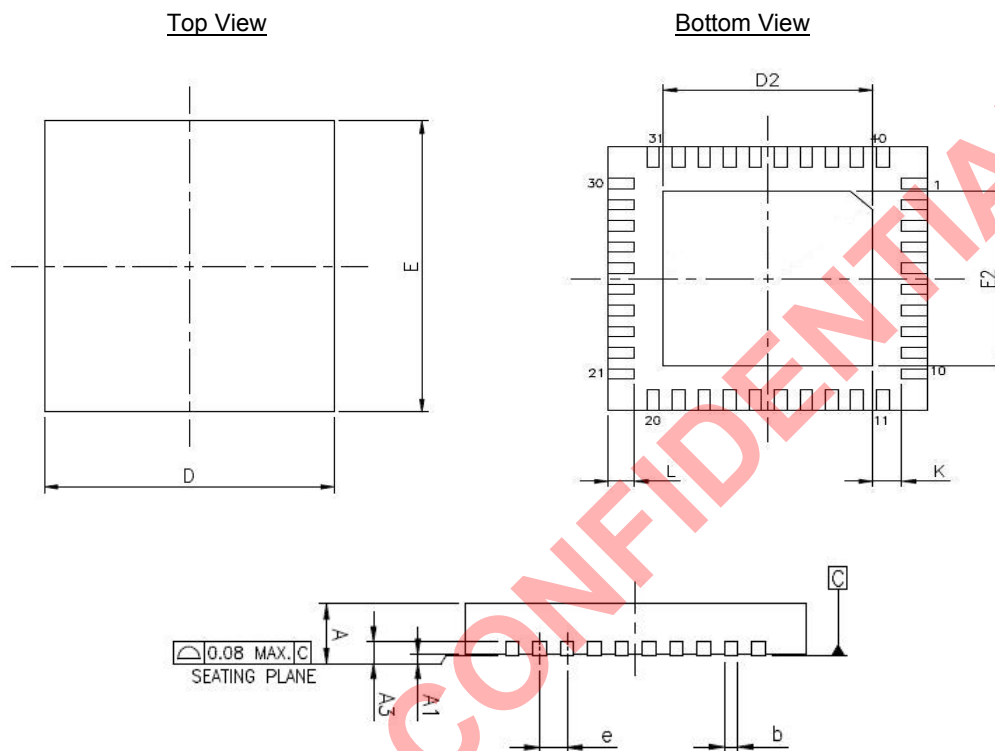
### 32. Package Information

#### 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 <sub>1</sub>	0	0.001	0.002	0.00	0.02	0.05
A <sub>3</sub>	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 <sub>2</sub>	0.146	0.177	0.179	3.70	4.50	4.55
E	0.240			6.1BSC		
E <sub>2</sub>	0.146	0.177	0.179	3.70	4.50	4.55
$\boxed{e}$	0.016BSC			0.4BSC		
L	0.013	0.016	0.020	0.32	0.4	0.48
k	0.008			0.2		

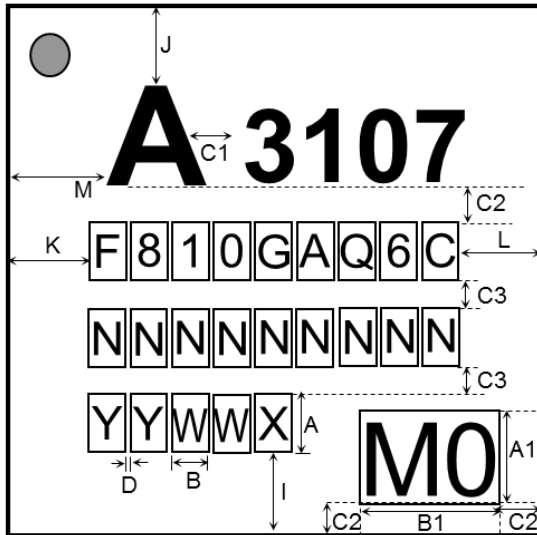
### QFN5\*5 40L Outline Dimensions



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 <sub>1</sub>	0.000	0.001	0.002	0.00	0.02	0.05
A <sub>3</sub>	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 <sub>2</sub>	0.126	-	0.138	3.20	-	3.50
E	0.194	-	0.200	4.924	-	5.1
E <sub>2</sub>	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		

### 33. Top Marking Information

- Part No. : A31U07F810GAQ6C
- 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

YYWW

: DATECODE

X

: PKG HOUSE ID

NNNNNNNNNN

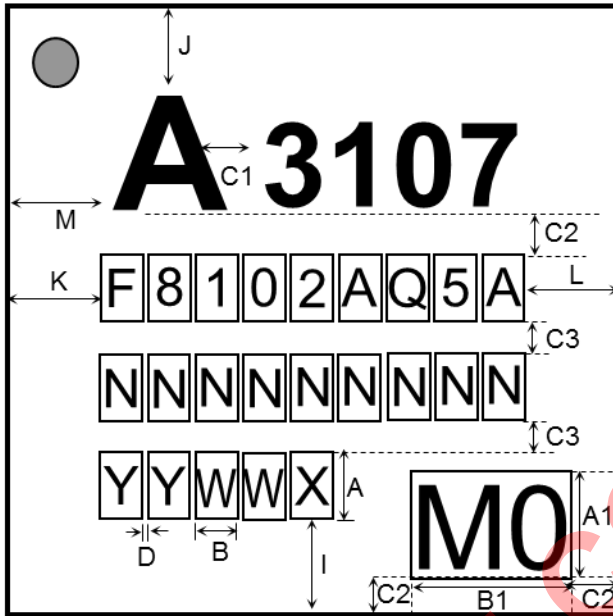
: LOT NO.  
(max. 9 characters)

I=J  
 K=L

0.90  
 A  
 0.78

0.75  
 3107  
 1.70

- Part No. : A31U07F8102AQ5A
- 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

YYWW

: DATECODE

X

: PKG HOUSE ID

NNNNNNNNNN

: LOT NO.  
 (max. 9 characters)

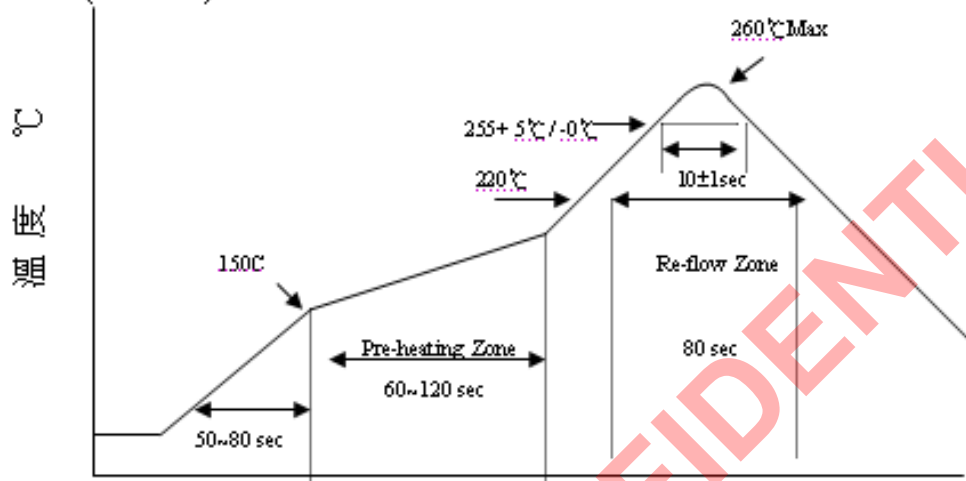
I=J  
 K=L

0.80  
 A  
 0.68

0.65  
 3107  
 1.6

### 34. Reflow Profile

LEAD FREE (GREEN) PROFILE :

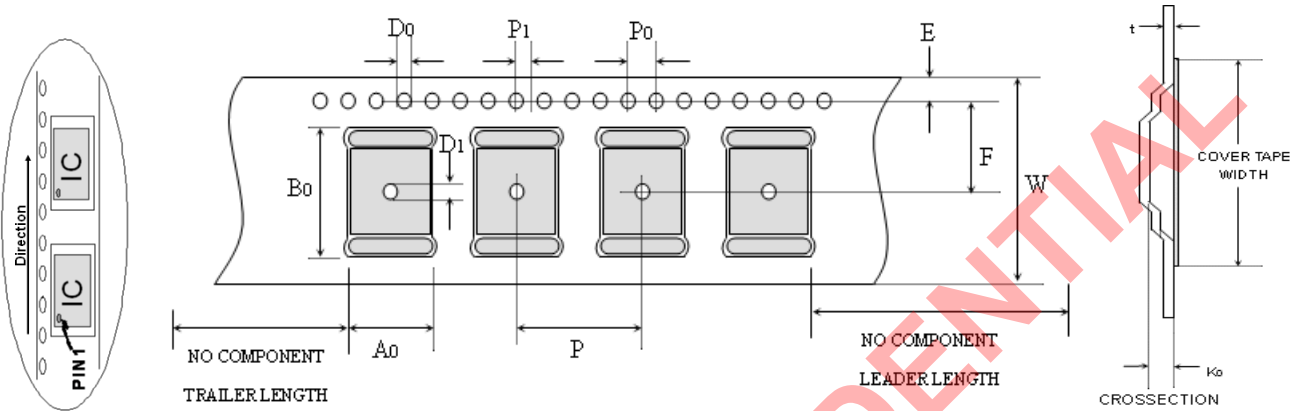


Actual Measurement Graph



### 35. Tape Reel Information

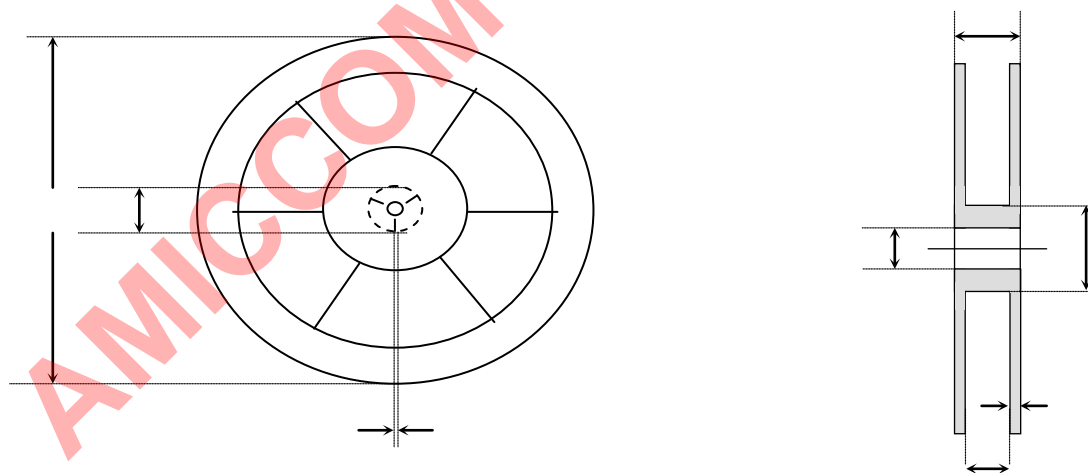
#### Cover / Carrier Tape Dimension



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

### 36. Product Status

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

RF ICs AMICCOM



#### Headquarter

10F., No.18, Taiyuan St., Zhubei City, Hsinchu  
County 30265, Taiwan, R.O.C.  
Tel : 886-3-5601717

#### Web Site

<http://www.amiccom.com.tw>

