Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

APPLICATIONS

- Home Automation and Security
- Automatic Meter Reading
- Factory Automation and Motor Control
- Medical Patient Monitoring
- Voice Applications
- Replacement for legacy wired UART
- Energy Management
- Remote Keyless Entry w/ Acknowledgement
- Toys
- PC peripherals

KEY FEATURES

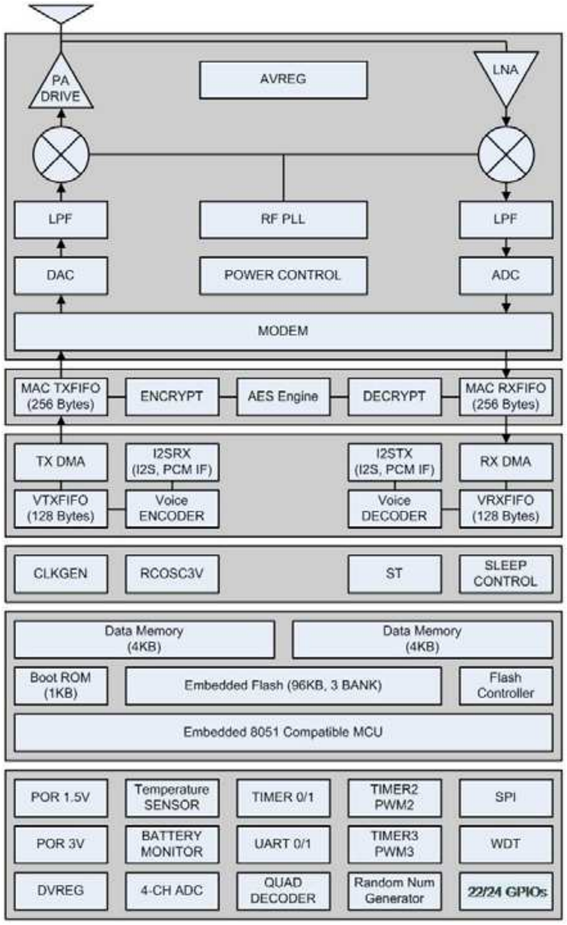
- **Embedded** 8051 Compatible **Microprocessor** with 96KB Embedded Flash Memory for Program Space plus 8KB of Data Memory
- **Scalable Data Rate**: 250kbps for ZigBee, 500kbps and 1Mbps for custom applications.
- **Voice Codec** Support: μ-law/a-law/ADPCM
- High RF **RX Sensitivity**: –98dBm @1.5V
- High RF **TX Power**: +8dBm @1.5V
- 4 Level **Power Management** Scheme with Deep Sleep Mode (0.3μA)
- **Single Voltage** operation: 1.9 to 3.3V using an internal regulator (1.5V core)
- Software **Tools and Libraries** for the Development of Custom Applications

DESCRIPTION:

ZIC2410 is a true single-chip solution, compliant with ZigBee specifications and IEEE802.15.4, a complete wireless solution for all ZigBee applications. The ***ZIC2410*** consists of an RF transceiver with baseband modem, a hardwired MAC and an embedded 8051 microcontroller with internal flash memory. The device provides numerous general-purpose I/O pins, peripheral functions such as timers and UART and is one of the first devices to provide an embedded Voice CODEC. This chip is ideal for very low power applications.

The ***ZIC2410*** is available in two industry standard packages: a 48-pin QFN (7x7mm) or a 72-pin VFBGA (5x5mm) package.

CEL provides its customers with the *CEL ZigBee Stack*, software in a compiled library, as well as all the hardware & software tools required to develop custom applications. User application software can be compiled using any popular C-language compiler such as Keil.



FEATURES

RF Transceiver

- Single-chip 2.4GHz RF Transceiver
- Programmable Output Power up to +8dBm@1.5V
- High Sensitivity of –98dBm@1.5V
- Scalable Data Rate: 250Kbps for ZigBee, 500Kbps and 1Mbps for custom application
- On-chip VCO, LNA, and PA
- Low Operating Voltage of 1.5V
- Direct Sequence Spread Spectrum
- O-QPSK Modulation
- RSSI Measurement
- Compliant to IEEE802.15.4
- No External T/R Switch or Filter needed

Hardwired MAC

- Two 256-byte circular FIFOs
- FIFO management
- AES Encryption/Decryption Engine (128bit)
- CRC-16 Computation and Check

8051-Compatible Microcontroller

- 8051 Compatible (single cycle execution)
- 96KB Embedded Flash Memory
- 8KB Data Memory
- 128-byte CPU dedicated Memory
- 1KB Boot ROM
- Dual DPTR Support
- Multi-Bank Support for 96KB Program Memory (3Banks of 32KB)
- I2S/PCM Interface with two128-byte FIFOs
- μ-law/a-law/ADPCM Voice Codec
- Two High-Speed UARTs with Two 16-byte FIFOs (up to 1Mbps)
- 4 Timers/2 PWMs
- Watchdog Timer
- Sleep Timer
- Quadrature Signal Decoder
- 24 General Purpose I/Os
- Internal RC oscillator for Sleep Timer
- On-chip Power-on-Reset
- 4-channel 8-bit ADC
- SPI Master/Slave Interface
- ISP (In System Programming)
- Internal Temperature Sensor

Clock Inputs

- 16MHz Crystal for System Clock (optional 19.2MHz)
- 32.768KHz Crystal for Sleep Timer (optional)

Power

- Internal Regulator for Single Voltage Operation w/ a large input voltage range (1.9~3.3V)
- 4-Level Power Management Scheme with Deep Sleep Mode (0.3μA)
- Separate On-chip Regulators for Analog and Digital Circuitry.
- Battery Monitoring Support

Included Software

- Application Framework
- Software Tools
- IEEE and ZigBee Compliant Libraries

Package Options

- Lead-Free 48-pin QFN Package (shown below) (7mm x 7mm x 0.9mm)
- Lead-Free 72-pin VFBGA Package (5mm × 5mm x 0.9mm)

ORDERING INFORMATION

Ordering Part Number	Description	Minimum Order Quantity (MOQ)
ZIC2410QN48R	48-pin QFN Package (T/R)	Tape & Reel (2500 per reel)
ZIC2410FG72R	72-pin VFBGA Package (T/R)	Tape & Reel (2500 per reel)
ZIC2410-EDK-1	Demonstration Kit	1

Table of Contents

1 FUNCTIONAL DESCRIPTION

Figure 1 shows the block diagram of ZIC2410. The ZIC2410 consists of a 2.4GHz RF, Modem (PHY Layer), a MAC hardware engine, a Voice CODEC block, Clocks, Peripherals, and a memory and Microcontroller (MCU) block.

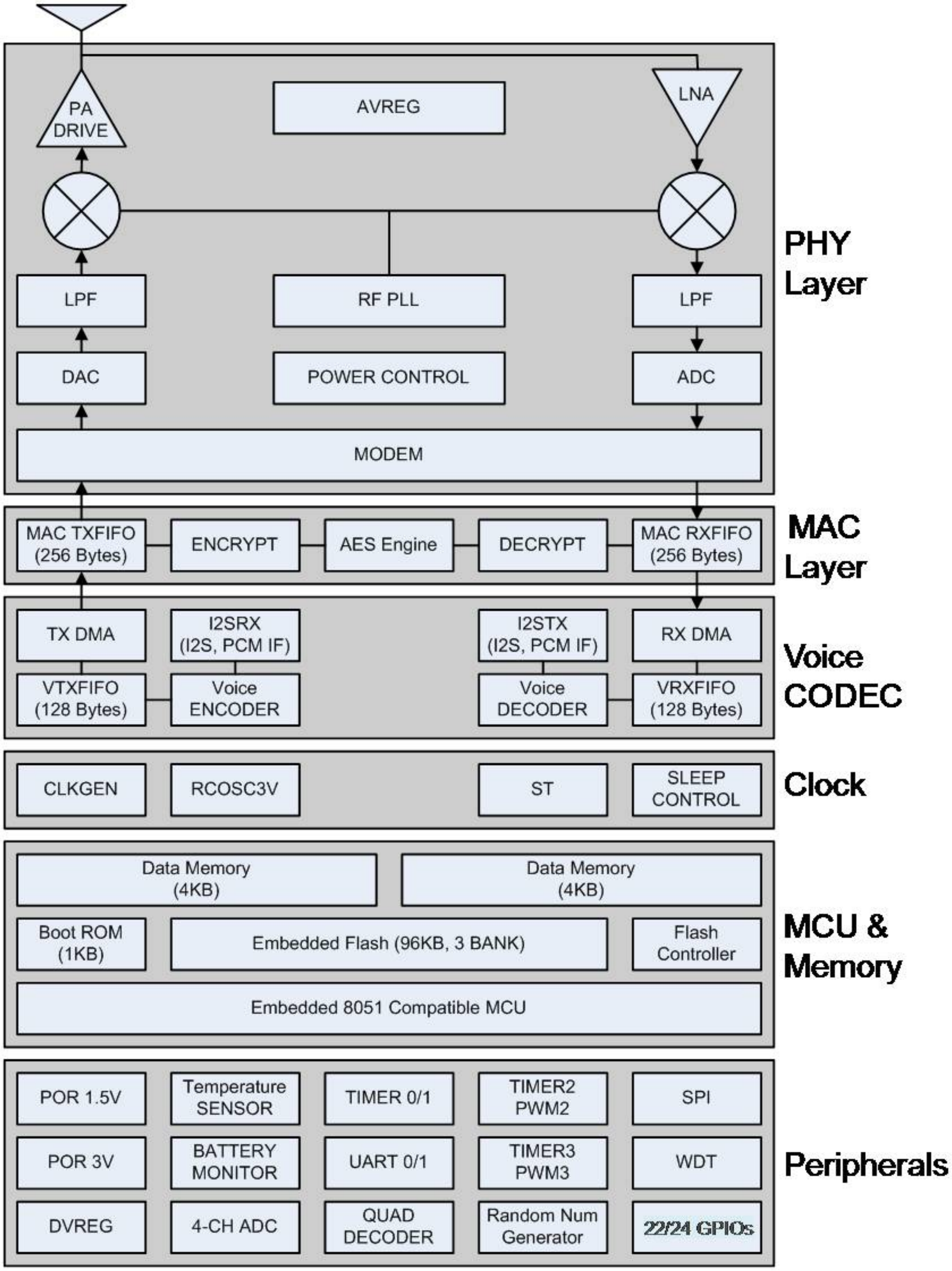


Figure 1 – Functional Block Diagram of ZIC2410

Note: The ZIC2410QN48 has 22 GPIOs; the ZIC2410FG72 has 24.

1.1 FUNCTIONAL OVERVIEW

In the receive mode, the received RF signal is amplified by the Low Noise Amplifier (LNA), down-converted to a quadrature signal and then to baseband. The baseband signal is filtered, amplified, converted to a digital signal by the ADC and transferred to a modem. The data, which is the result of signal processing such as dispreading, is transferred to the MAC block.

In transmit mode, the buffered data at the MAC is transferred to a baseband modem which, after signal processing such as spreading and pulse shaping, outputs a signal through the DAC. The Analog baseband signal is filtered by the low-pass filter, converted to RF signal by the up-conversion mixer, is amplified by PA, and finally applied to the antenna.

The MAC block provides IEEE802.15.4 compliant hardware and it is located between microprocessor and a baseband modem. MAC block includes FIFOs for transmitting/receiving packet, AES engine for security operation, CRC and related control circuit. In addition, it supports automatic CRC check and address decoding.

ZIC2410 integrates a high performance embedded microcontroller, compatible to an Intel i8051 microcontroller in an instruction level. This embedded microcontroller has 8-bit operation architecture sufficient for controller applications. The embedded microcontroller has 4-stage pipeline architecture to improve the performance over previous compatible chips making it capable of executing simple instructions during a single cycle.

The memory organization of the embedded microcontroller consists of program memory and data memory. The data memory has 2 memory areas. For more detailed explanation, refer to the data memory section (1.2.2.)

The ZIC2410 includes 22 GPIO for the QN48 packaged device and 24 GPIO for the FG72 packaged part and various peripheral circuits to aid in the development of an application circuit with an interrupt handler to control the peripherals. ZIC2410 uses 16MHz crystal oscillator for RF PLL and 8MHz clock generated from 16MHz in clock generator is used for microcontroller, MAC, and the clock of a baseband modem.

The ZIC2410 supports a voice function as follows. The data generated by an external ADC is input to the voice block via I2S interface. After the data is received via I2S it is compressed by the voice codec, and stored in Voice TXFIFO. The data in Voice TXFIFO is transferred to the MAC TXFIFO and then transmitted via PHY. In contrast, the received data in MAC RXFIFO is transferred to voice RXFIFO via DMA operation. The data in voice RXFIFO is decompressed by the internal voice codec. The decompressed data is then transferred to the external DAC via I2S interface.

1.2 MEMORY ORGANIZATION

1.2.1 PROGRAM MEMORY

The address space of the program memory is 64KB (0x0000~0XFFFF). Basically, the lower 63KB of program memory is implemented by Non-volatile memory. The upper 1KB from 0XFC00 to 0XFFFF is implemented by both Non-volatile memory and ROM. As shown in Figure 2 below, there are two types of memory in the same address space. The address space, which is implemented by Non-volatile memory, is used as general program memory and the address space, which is implemented by ROM, is used for ISP (In-System Programming).

As shown in (a) of Figure 2 below, when Power is turned on, the upper 1KB of program memory is mapped to ROM. As shown in (b) of Figure 2, if this program area (1KB) is used as non-volatile program memory, ENROM should be set to '0'. See the SFR section (1.2.4) for ENROM.

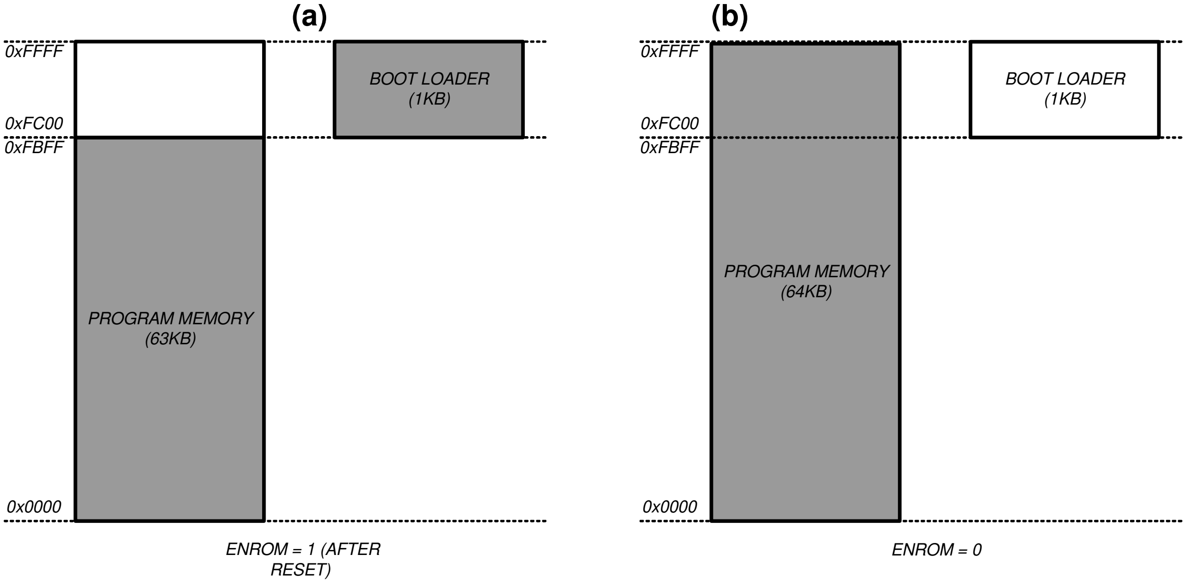


Figure 2 – Address Map of Program Memory

ZIC2410 includes non-volatile memory of 96KB. However, as described already, program memory area is 64KB. Therefore, if necessary, the upper 64KB of physical 96KB non-volatile memory is separated into two 32KB memory banks. Each bank is logically mapped to the program memory. When FBANK value is '0', lower 64KB of non-volatile memory is used as shown in (a) of Figure 3. When FBANK value is '1', lower 32 KB and upper 32KB of non-volatile memory are used as shown in (b) of Figure 3. See the SFR section (1.2.4) for FBANK.

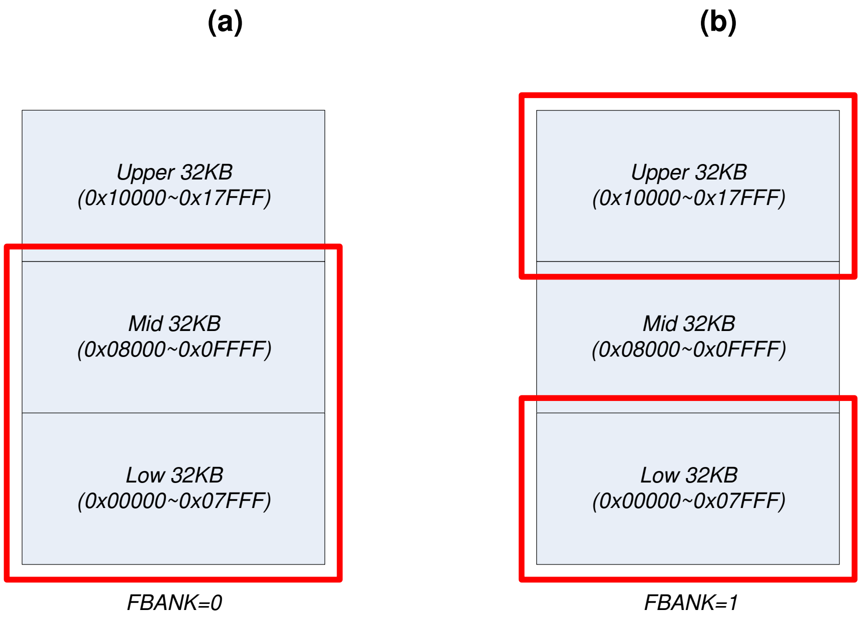


Figure 3 – Bank Selection of Program Memory

1.2.2 DATA MEMORY

ZIC2410 reserves 64 KB of data memory address space. This address space can be accessed by the MOVX command.

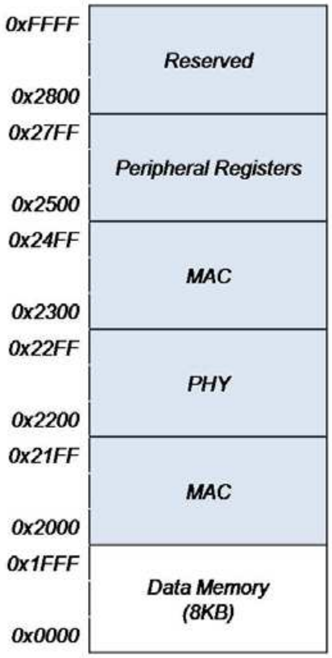


Figure 4 shows the address map of this data memory.

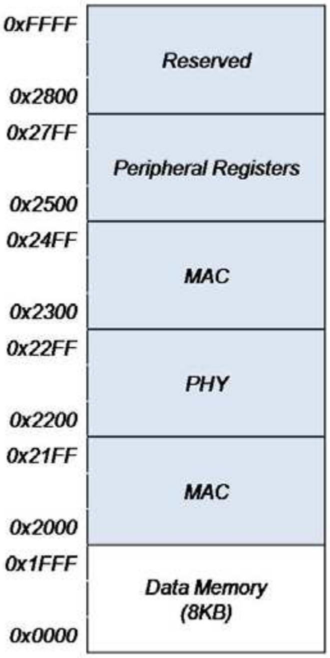


Figure 4 – Address Map of Data Memory

The data memory used in the application programs resides in the address range 0x0000-0x1FFF.

The registers and memory used in the MAC block reside in the address range 0x2000-0x21FF and 0x2300-0x24FF respectively. The registers to control or report the status of the PHY block reside in the address range 0x2200-0x22FF.

Registers related to the numberous peripheral functions of the embedded microprocessor reside in the address range of 0x2500-0x27FF.

1.2.3 GENERAL PURPOSE REGISTERS (GPR)

Figure 5 describes the address map of the General Purpose Registers (GPRs). GPRs can be addressed either directly or indirectly. As shown in the lower address space of Figure 5, a bank consists of 8 registers.

The address space above the bank area is the bit addressable area, which is used as a flag by software or by a bit operation. The address space above the bit addressable area includes registers used as a general purpose of a byte unit. For the detailed information, refer to the paragraphs following Figure 5 below.

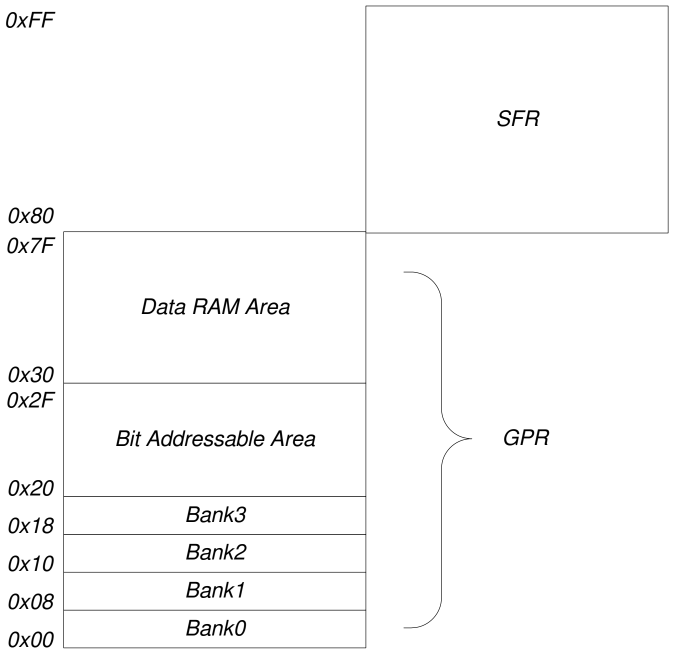


Figure 5 – GPRs Address Map

Register Bank 0-3: It is located from 0x00 to 0x1F (32 bytes). One bank consists of each 8 registers out of 32 registers. Therefore, there are total 4 banks. Each bank should be selected by software as referring the RS field in PSW register. The bank (8 registers) selected by RS value can be accessed by a name (R0-R7) by software. After reset, the default value is set to bank0.

Bit Addressable Area: The address is assigned to each bit of 16 bytes (0x20~0x2F) and registers, which is the multiple of 8, in SFR. Each bit can be accessed by the address which is assigned to these bits. 128 bits (16 bytes, 0x20~0x2F) can be accessed by direct addressing for each bit (0~127) and by a byte unit as using the address from 0x20~0x2F.

Data RAM Area: A user can use registers (0x30~0x7F) as a general purpose.

1.2.4 SPECIAL FUNCTION REGISTERS (SFR)

Generally, a register is used to store the data. MCU needs the memory to control the embedded hardware or the memory to show the hardware status. Special Function Registers (SFRs) process the functions described above. SFRs include the status or control of the I/O ports, the timer registers, the stack pointers and so on. Table 1 shows the address to all SFRs in ZIC2410.

All SFRs are accessed by a byte unit. However, when SFR address is a multiple of 8, it can be accessed by a bit unit.

Table 1 – Special Function Register (SFR) Map

Register Name	SFR Address	B7	B6	B5	B4	B3	B2	B1	B0	Initial Value
EIP	0xF8		VCEIP	SPIIP	RTCIP	T3IP	AESIP	T2IP	RFIP	0x00
B	0xF0									0x00
EIE	0xE8		VCEIE	SPIIE	RTCIE	T3IE	AESIE	T2IE	RFIE	0x00

Register Name	SFR Address	B7	B6	B5	B4	B3	B2	B1	B0	Initial Value
ACC	0xE0									0x00
EICON	0xD8					RTCIF				0x00
WDT	0xD2				WDTWE	WDTEN	WDTCLR	WDTPRE		0x0B
PSW	0xD0	CY	AC	F0	RS		OV	F1	P	0x00
WCON	0xC0						ISPMODE	ENROM		0x00
P3REN	0xBC									0xFF
P1REN	0xBA									0xFF
P0REN	0xB9									0xFF
IP	0xB8		PS1		PS0	PT1	PX1	PT0	PX0	0x00
P3OEN	0xB4									0x00
P1OEN	0xB2									0x00
P0OEN	0xB1									0x00
P3	0xB0									0x3F
TL3	0xAD									0x00
TL2	0xAC									0x00
TH3	0xAB									0x00
TH2	0xAA									0x00
T23CON	0xA9					TR3	M3	TR2	M2	0x00
IE	0xA8	EA	ES1		ES0	ET1	EX1	ET0	EX0	0x00
AUXR1	0xA2								DPS	0x00
FBANK	0xA1	RAM1	RAM0					FBANK		0x00
EXIF	0x91	T3IF	AESIF	T2IF	RFIF					0x00
P1	0x90									0xFF
TH1	0x8D									0x00
TH0	0x8C									0x00
TL1	0x8B									0x00
TL0	0x8A									0x00
TMOD	0x89	GATE1	CT1	M1		GATE0	CT0	M0		0x00
TCON	0x88	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0x00
PCON	0x87							PD	IDLE	0x00
P0SEL	0x85							ExNoEdge	P0AndSEL	0x00
P0MSK	0x84									0xFF
DPH	0x83									0x00
DPL	0x82									0x00
SP	0x81									0x07
P0	0x80									0xFF

The following section describes each SFR related to microprocessor.

Table 2 – Register Bit Conventions

Symbol	Access Mode
RW	Read/write
RO	Read Only

Table 3 – Special Function Registers

Bit	Name	Descriptions	R/W	Reset Value
WCON (WRITE CONTROL REGISTER, 0xC0) This register can control the upper 1KB of program memory.				
7:3		Reserved		0
2	**ISPMODE**	**ISP Mode Indication**: When MS [1:0], an external pin, is '3', this field is set to 1 by hardware. It notifies the MCU whether ISPMODE or not.	RO	-
1	**ENROM**	When this field is '1', the upper 1KB (0xFC00~0xFFFF) is mapped to ROM. When this field is '0', the upper 1KB (0xFC00~0xFFFF) is mapped to non-volatile memory.	R/W	1
0		Reserved		0
FBANK (PROGRAM MEMORY BANK SELECTION REGISTER, 0xA1)				
7:1		Reserved		0x00
0	**FBANK**	Program Memory Bank Select. 0: Bank0 (Default) 1: Bank1 2: Not Used 3: Not Used	R/W	0
ACCUMULATOR (0xE0) This register is marked as A or ACC and it is related to all the operations.				
7:0	**A**	Accumulator	R/W	0x00
B REGISTER (0xF0) This register is used for a special purpose when multiplication and division are processed. For other instructions, it can be used as a general-purpose register. After multiplication is processed, this register contains the MSB data and 'A register' contains LSB data for the multiplication result. In division operation, this register stores the value before division (dividend) and the remainder after division. At this time, before division, the divisor should be stored in 'A register' and result value (quotient) is stored in it after division**.**				
7:0	**B**	B register. Used in MUL/DIV instructions.	R/W	0x00
PROGRAM STATUS WORD (PSW, 0xD0) This register stores the status of the program. The explanation for each bit is as follows.				
7	**CY**	Carry flag	R/W	0
6	**AC**	Auxiliary carry flag	R/W	0
5	**F0**	Flag0. User-defined	R/W	0
4:3	**RS**	Register bank select. 0: Bank0 1: Bank1 2: Bank2 3: Bank3	R/W	0
2	**OV**	Overflow flag	R/W	0
1	**F1**	Flag1. User-defined	R/W	0
0	**P**	Parity flag. Set to 1 when the value in accumulator has odd number of '1' bits.	R/W	0
STACK POINTER (0x81) When PUSH and CALL commands are executed, some data (like the parameters by function call) are stored in stack to inform the values. In the embedded MCU, the data memory area which can be used for a general purpose (0x08~0x7F) is used as a stack area. This register value is increased before the data is stored and the register value is decreased after the data is read when the data of stack is disappeared by POP and RET command. The default value is 0x07.				
7:0	**SP**	Stack Pointer	R/W	0x07

Bit	Name	Descriptions	R/W	Reset Value
DATA POINTER (DPH: 0x83, DPL: 0x82) Data pointer consists of a high byte (DPH) and a low byte (DPL) to support 16-bit address. It can be accessed by 16-bit register or by two 8-bit registers respectively.				
7:0	**DPH**	Data pointer, high byte	R/W	0x00
7:0	**DPL**	Data pointer, low byte	R/W	0x00
AUXR1 (AUXILIARY CONTROL REGISTER, 0xA2) This register is used to implement Dual DPTR functions. Physically, DPTR consists of DPTR0 and DPTR1. However, DPTR0 and DPTR1 can be accessed depending on the DPS value of AUXR1 respectively. In other words, they cannot be accessed at the same time.				
7:1		Reserved		0x00
0	**DPS**	**Dual DPTR Select**: Used to select either DPTR0 or DPTR1. When DSP is '0', DPTR0 is selected. When DSP is '1', DPTR1 is selected.	R/W	0
P3 (0xB0) This port register can be used as other functions besides general purpose I/O.				
7	**P3.7**	This port register is used as a general purpose I/O port (12mA Drive).	R/W	0
	/PWM3	When Timer3 is operated as a PWM mode, it outputs PWM wave (PWM3) of Timer3.		
	/CTS1	When port register is used as UART1, it is used as a CTS signal (CTS1) of UART1.		
	/SPICSN	When used as a Master mode, SPI Slave Select signal is outputted. When used as a Slave mode, this port register receives SPI Slave Select signal. This signal activate in low		
6	**P3.6**	This port register is used as a general purpose I/O port (12mA Drive)	R/W	0
	/PWM2	When Timer2 is operated as a PWM mode, it outputs PWM wave (PWM2) of Timer2.		
	/RTS1	When port register is used as UART1, it is used as a RTS signal (RTS1) of UART1.		
	/SPICLK	When used as a Master mode, SPI clock is outputted. When used as a Slave mode, this port register receives SPI clock.		
5	**P3.5**	This port register is used as a general purpose I/O port.	R/W	1
	/T1	When Timer1 is operated as a COUNTER mode, it is operated as a counter input signal (T1) of Timer1.		
	/CTS0	When port register is used as UART0, it is used as a CTS signal (CTS0) of UART0.		
	/SPIDO	In a Master mode or a Slave mode, this port register is used for outputting SPI data.		
	/QUADYB	When port register is used as QUAD function, it is used as the input signal of YB value.		
4	**P3.4**	This port register is used as a general purpose I/O port.	R/W	1
	/T0	When Timer0 is operated as a COUNTER mode, it is operated as a counter input signal (T0) of Timer0.		
	/RTS0	When port register is used as UART0, it is used as a RTS signal (RTS0) of UART0.		

Bit	**Name**	**Descriptions**	**R/W**	**Reset Value**
	/SPIDI	In a Master mode or a Slave mode, this port register is used for receiving SPI data.		
	/QUADYA	When port register is used as QUAD function, it is used as the input signal of YA value.		
3	**P3.3**	This port register is used as a general purpose I/O port.	R/W	1
	/INT1	When port register is used as an input signal, it can receive an external interrupt (INT1).		
2	**P3.2**	This port register is used as a general purpose I/O port.	R/W	1
	/INT0	When port register is used as an input signal, it can receive an external interrupt (INT0).		
1	**P3.1**	This port register is used as a general purpose I/O port.	R/W	1
	/TXD0	When port register is used as UART0, it is used as a UART0 data output (TXD0).		
	/QUADXB	When port register is used as QUAD function, it is used as the input signal of XB value.		
0	**P3.0**	This port register is used as a general purpose I/O port.	R/W	1
	/RXD0	When port register is used as UART0, it is used as a UART0 data input (RXD0).		
	/QUADXA	When port register is used as QUAD function, it is used as the input signal of XA value.		
P1 (0x90) This port register can be used as other functions besides general purpose I/O.				
7	**P1.7**	This port register is used as a general purpose I/O port.	R/W	1
	/P0AND	When P0AndSel value in P0SEL register is set to ‘1’, P1.7 outputs the result of bit-wise AND operation of (P0 OR P0MSK).		
	/TRSW	It can be used as TRSW (RF TX/RX Indication signal) signal by setting the PHY register.		
6	**P1.6**	This port register is used as a general purpose I/O port.	R/W	1
	/TRSWB	It can be used as TRSWB (TRSW Inversion) signal by setting the PHY register.		
5	**P1.5**	This port register is used as a general purpose I/O port.	R/W	1
4	**P1.4**	This port register is used as a general purpose I/O port.	R/W	1
	/QUADZB	When this port register is used as QUAD function, it is used as the input signal of ZB value.		
	/RTXTALI	This port register is used for connecting to the external crystal (32.768KHz), which is used in the Sleep Timer, by setting the PHY register.		
3	**P1.3**	This port register is used as a general purpose I/O port.	R/W	1
	/QUADZA	When this port register is used as QUAD function, it is used as the input signal of ZA value.		

Bit	**Name**	**Descriptions**	**R/W**	**Reset Value**
	/RTXTALO	This port register is used for connecting to the external crystal (32.768KHz), which is used in the Sleep Timer, by setting the PHY register.		
	/RTCLKO	This port register is used to output the internal RCOSC by setting the PHY register.		
2	**P1.2**	This port register is used as a general purpose I/O port.	R/W	1
1	**P1.1**	This port register is used as a general purpose I/O port.	R/W	1
	/TXD1	When this port register is used as UART1, it is used as UART1 data output (TXD1).		
0	**P1.0**	This port register is used as a general purpose I/O port.	R/W	1
	/RXD1	When this port register is used as UART1, it is used as UART1 data input (RXD1).		
P0 (0x80) This port register can be used as other functions besides general purpose I/O.				
7	**P0.7**	This port register is used as a general purpose I/O port.	R/W	1
	/I2STXMCLK	When this port register is used as I2S, it is operated as TX Master clock of I2S interface.		
6	**P0.6**	This port register is used as a general purpose I/O port.	R/W	1
	/I2STXBCLK	When this port register is used as I2S, it is operated as TX Bit clock of I2S interface.		
5	**P0.5**	This port register is used as a general purpose I/O port.	R/W	1
	/I2STXLRCK	When this port register is used as I2S, it is operated as TX LR clock of I2S interface.		
4	**P0.4**	This port register is used as a general purpose I/O port.	R/W	1
	/I2STXDO	When this port register is used as I2S, it is operated as TX data output of I2S interface.		
3	**P0.3**	This port register is used as a general purpose I/O port.	R/W	1
	/I2SRXMCLK	When this port register is used as I2S, it is operated as RX Master clock of I2S interface.		
2	**P0.2**	This port register is used as a general purpose I/O port.	R/W	1
	/I2SRXBCLK	When this port register is used as I2S, it is operated as RX Bit clock of I2S interface.		
1	**P0.1**	This port register is used as a general purpose I/O port.	R/W	1
	/I2SRXLRCK	When this port register is used as I2S, it is operated as the RX LR clock of the I2S interface.		
0	**P0.0**	This port register is used as a general purpose I/O port.	R/W	1
	/I2SRXDI	When this port register is used as I2S, it is operated as the RX data input of the I2S interface.		
P0OEN/P1OEN/P3OEN (0xB1, 0xB2, 0xB4) P0OEN, P1OEN and P3OEN enable the output of port0, 1 and 3. When each bit is cleared to ‘0’, the output of the corresponding port is enabled. For example, when 4th bit of P1OEN is set to low, the output of port1.3 is enabled.				
7		Reserved		0

Bit	Name	Descriptions	R/W	Reset Value
7:0	**P3OEN**	It controls the TX buffer function for each pin in Port3. When each bit field is set to '0', the TX buffer of the corresponding pin outputs the value.	R/W	0x00
6:0	**P1OEN**	It controls the TX buffer function for each pin in Port1. When each bit field is set to '0', the TX buffer of the corresponding pin outputs the value. P1.7 only acts as output.	R/W	0x00
7:0	**P0OEN**	It controls the TX buffer function for each pin in Port0. When each bit field is set to '0', the TX buffer of the corresponding pin outputs the value.	R/W	0x00
P0REN/P1REN/P3REN (0xB9, 0xBA, 0xBC) P0REN, P1REN, P3REN enable Pull-up of port 0, 1 and 3. When each bit area is cleared to '0', the Pull-up of the corresponding port is enabled.				
7		Reserved		1
7:0	**P3REN**	It controls the Pull-up function for each pin in Port3. When each bit field is set to '0', the Pull-up function of the corresponding pin is operated.	R/W	0xFF
6:0	**P1REN**	It controls the Pull-up function for each pin in Port1. When each bit field is set to '0', the Pull-up function of the corresponding pin is operated. **P1.7 doesn't have a control field because it is operated as an output.*	R/W	0x7F
7:0	**P0REN**	It controls the Pull-up function for each pin in Port0. When each bit field is set to '0', the Pull-up function of the corresponding pin is operated.	R/W	0xFF
P0MSK (P0 INPUT MASK REGISTER, 0x84)				
7:0	**P0MSK**	This register is used for masking the input of P0 pin (Refer to P0AndSel in P0SEL register).	R/W	0xFF
P0SEL (P0 INPUT SELECTION REGISTER, 0x85)				
7:2		Reserved		0
1	**ExNoEdge**	Controls the wake up of the MCU by an external interrupt when in the power-down mode. When this field is '0', the MCU wakes up when INT0 or INT1 signal is high (This is the normal case in the MCU.) When this field is '1', the MCU is woken up by the wakeup signal of the SleepTimer. Remote control function can be implemented by the interrupt service routine of the MCU when the WAKEUP signal occurs by adjusting the RTDLY value in the Sleep Timer while either INT0 or INT1 is low.	R/W	0
0	**P0AndSel**	When this field is set to '1', P0 and P0MSK are ORed per bit. The bits of the result value are to be ANed and then output to P1.7. This function is used to implement remote control function.	R/W	0

1.3 RESET

The ZIC2410 should be reset to be operated. There are three kinds of reset sources. The first one is to use an external reset pin (RESET#). When applying a low signal to this pin for more than 1ms, ZIC2410 is reset. Second, ZIC2410 can be reset by an internal POR when it is powered up as using the internal Power-On-Reset (POR) block. Third, as a reset by the watchdog timer, a reset signal is generated when the internal counter of watchdog timer reaches a pre-set value.

Table 4 – Power-On-Reset Specifications

Parameter	MIN	TYP	MAX	UNIT
1.5V POR Release		1.18		V
1.5V POR Hysteresis		0.11		V

NOTE

Reference circuit of ZIC2410 is as follows. When the ZIC2410 is operated below minimum operating voltage, a reset error will occur because of the unstable voltage. It is recommended to use an external reset IC to improve stability in low voltage conditions.

[Application Circuit by adjusting RESET-IC]

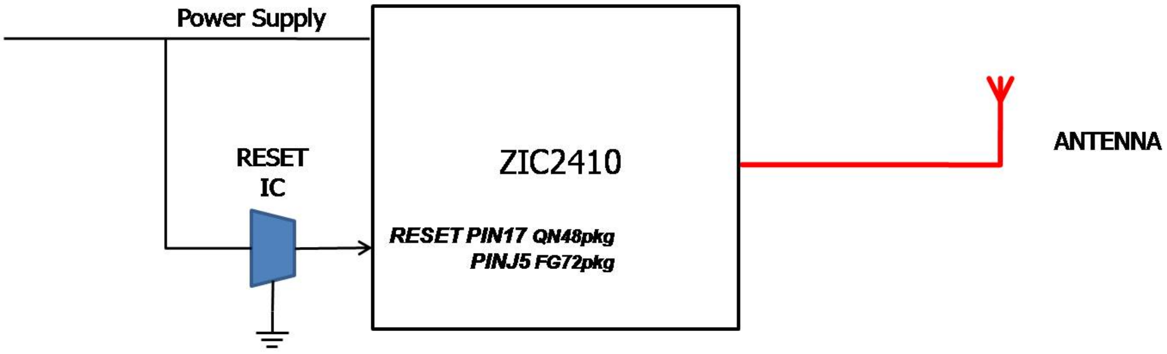


Figure 6 – Reset Circuit

[Reset Circuit by adjusting ELM7527NB]

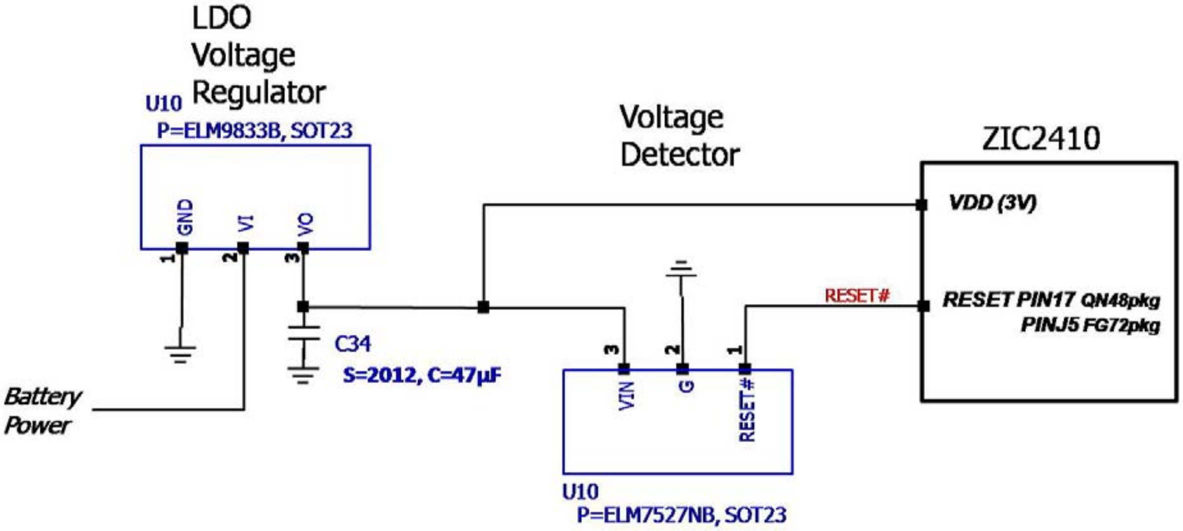


Figure 7 – Reset Circuit Using ELM7527NB

Checking the RESET-IC Circuit

1. In the application circuit of ZIC2410, please connect RESET# PIN to Pull-up register and should not connect it to capacitor.
2. When applying RESET-IC, detection voltage should be set over 1.9V.
3. The interval (T_reset) until from the time which reset signal by Reset IC has been adjusted to the time which the voltage of VDD (3.0) is dropped to 1.6V should be longer than 1ms.
4. T_reset time is adjusted when modifying capacitor value connected to VDD (3.0).

[RESET Timing]

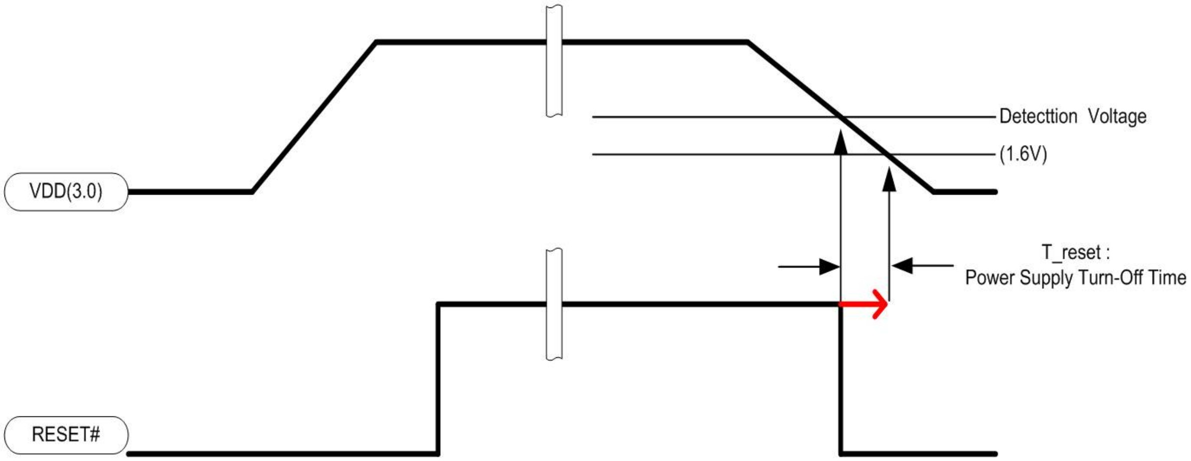


Figure 8 – Reset Timing Diagram

1.4 CLOCK SOURCE

The ZIC2410 can use either a 16MHz or a 19.2MHz crystal as the system clock source. An external 32.768 KHz crystal or the internal clock generated from internal the RCOSC is used for the Sleep Timer clock.

For the internal 8051 MCU Clock in the ZIC2410, either 8MHz or 16MHz can be used. When selecting the 8051 MCU Clock (8MHz, 16MHz), the CLKDIV0 register should be set as follows.

Please note the crystal oscillator input (XOSCI) can also be driven by a CMOS clock source.

CLKDIV0 **(OPERATING FREQUENCY CONTROL REGISTER, 0x22C3)**

Table 5 – Clock Registers

Bit	Name	Descriptions	R/W	Reset Value
7:0	CLKDIV0	This register is used to control the clock of the internal 8051 MCU. When this register is set to 0xFF, the clock is set to 8MHz; when set to 0x00, the clock is set to 16MHz. All other values except 0xFF and 0x00 are reserved.	R/W	0xFF

1.5 INTERRUPT SCHEMES

The program interrupt functions of the embedded MCU are similar to other microprocessors. When an interrupt occurs, the interrupt service routine at the corresponding vector address is executed. When the interrupt service routine process is completed, the program is resumed from the point of time at which the interrupt occurred. Interrupts can be initiated from the internal operation of the embedded microprocessor (e.g. the overflow of the timer count) or from an external signal.

The ZIC2410 has 13 interrupt sources. Table 6 describes the detailed information for each of the interrupt sources. The 'Interrupt Address' indicates the address where the interrupt service routine is located. The 'Interrupt Flag' is the bit that notifies the MCU that the corresponding interrupt has occurred. 'Interrupt Enable' is the bit which decides whether each interrupt has been enabled. 'Interrupt Priority' is the bit which decides the priority of the interrupt. The 'Interrupt Number' is the interrupt priority fixed by the hardware. That is, when two or more interrupts having the same 'Interrupt Priority' value, occur simultaneously, the lower 'Interrupt Number' is processed first.

Table 6 – Interrupt Descriptions

Interrupt Number	Interrupt Type	Interrupt Address	Interrupt Flag	Interrupt Enable	Interrupt Priority
0	External Interrupt0	0003H	TCON.IE0	IE.EX0	IP.PX0
1	Timer0 Interrupt	000BH	TCON.TF0	IE.ET0	IP.PT0
2	External Interrupt1	0013H	TCON.IE1	IE.EX1	IP.PX1
3	Timer1 Interrupt	001BH	TCON.TF1	IE.ET1	IP.PT1
4	UART0 Interrupt (TX) UART0 Interrupt (RX)	0023H	***Note 1***	IE.ES0	IP.PS0
7	UART1Interrupt (TX) UART1 Interrupt (RX)	003BH	***Note 1***	IE.ES1	IP.PS1
8	PHY Interrupt	0043H	EXIF.PHYIF	EIE.RFIE	EIP.RFIP
9	Timer2 Interrupt	004BH	EXIF.T2IF	EIE.T2IE	EIP.T2IP
10	AES Interrupt	0053H	EXIF.AESIF	EIE.AESIE	EIP.AESIP
11	Timer3 Interrupt	005BH	EXIF.T3IF	EIE.T3IE	EIP.T3IP
12	Sleep Timer Interrupt	0063H	EICON.RTCIF	EIE.RTCIE	EIP.RTCIP
13	SPI Interrupt	0068H	***Note 2***	EIE.SPIIE	EIP.SPIIP
14	Voice Interrupt	0073H	***Note 3***	EIE.VCEIE	EIP.VCEIP

Note 1: *In the case of a UART Interrupt, bit [0] of the IIR register (0x2502, 0x2512) in the UART block is used as a flag. Also, the Tx, Rx, Timeout, Line Status and Modem Status interrupts can be distinguished by bit [3:1] value. For more detailed information, refer to the UART0/1 description in Section 1.7.6.*

Note 2: *In the case of an SPI interrupt, there is another interrupt enable bit in the SPI register besides EIE.SPIIE. In order to enable an SPI interrupt, both SPIE in the SPCR (0x2540) register and EIE.SPIIE should be set to '1. SPIF in the SPSR (0x2541) register acts as an interrupt flag.*

Note 3: *In case of a Voice interrupt, there are interrupt enable registers and interrupt flag registers in the voice block. The interrupt enable register are VTFINTENA (0x2770), VRFINTENA (0x2771) and VDMINTENA (0x2772). The interrupt flag register are VTFINTVAL (0x2776), VRFINTVAL (0x2777), and VDMINTVAL (0x2778). There are 24 interrupt sources. When both an interrupt enable signal and an interrupt flag signal are set to '1,' voice interrupt is enabled.*

Table 7 – INTERRUPT Registers

Bit	Name	Descriptions	R/W	Reset Value
IE (INTERRUPT ENABLE REGISTER, 0xA8) The EA bit in the IE register is the global interrupt enable signal for all interrupts. In addition, each interrupt is masked by each interrupt enable bit. Therefore, in order to use an interrupt, both EA and the specific interrupt enable bit should be set to '1'. When the bit for each interrupt is '0', that interrupt is disabled. When the bit for each interrupt is '1', that interrupt is enabled.				
7	**EA**	Global interrupt enable 0: No interrupt will be acknowledged. 1: Each interrupt source is individually enabled or disabled by setting its corresponding enable bit.	R/W	0
6	**ES1**	UART1 interrupt enable 1: interrupt enabled.	R/W	0
5		Reserved		0
4	**ES0**	UART0 interrupt enable 1: interrupt enabled.	R/W	0
3	**ET1**	Timer1 interrupt enable 1: interrupt enabled.	R/W	0
2	**EX1**	External interrupt1 enable 1: interrupt enabled.	R/W	0
1	**ET0**	Timer0 interrupt enable 1: interrupt enabled.	R/W	0
0	**EX0**	External interrupt0 enable 1: interrupt enabled.	R/W	0
IP (INTERRUPT PRIORITY REGISTER, 0xB8) If a bit corresponding to each interrupt is '0', the corresponding interrupt has lower priority and if a bit is '1', the corresponding interrupt has higher priority.				
7		Reserved		0
6	**PS1**	UART1 interrupt priority 1: UART1 interrupt has higher priority.	R/W	0
5		Reserved		0
4	**PS0**	UART 0 interrupt priority 1: UART0 interrupt has higher priority.	R/W	0
3	**PT1**	Timer1 interrupt priority 1: Timer1 interrupt has higher priority.	R/W	0
2	**PX1**	External interrupt1 interrupt priority 1: External interrupt1interrupt has higher priority.	R/W	0
1	**PT0**	Timer0 interrupt priority 1: Timer0 interrupt has higher priority.	R/W	0
0	**PX0**	External interrupt0 interrupt priority 1: External interrupt0 interrupt has higher priority.	R/W	0
EIE (EXTENDED INTERRUPT ENABLE REGISTER, 0xE8) If a bit is '0', corresponding interrupt is disabled and if a bit is '1', corresponding interrupt is enabled. Refer to the following table.				
7		Reserved	R/W	0
6	**VCEIE**	Voice Interrupt Enable. 0: Interrupt disabled 1: Interrupt enabled	R/W	0
5	**SPIIE**	SPI Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled	R/W	0
4	**RTCIE**	Sleep Timer Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled	R/W	0
3	**T3IE**	Timer3 Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled	R/W	0
2	**AESIE**	AES Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled	R/W	0

Bit	Name	Descriptions	R/W	Reset Value
1	**T2IE**	Timer2 Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled	R/W	0
0	**RFIE**	RF Interrupt Enable 0: Interrupt disabled 1: Interrupt enabled	R/W	0
EIP (EXTENDED INTERRUPT PRIORITY REGISTER, 0xF8) If a bit is '0', the corresponding interrupt has lower priority. If a bit is '1', the corresponding interrupt has higher priority.				
7		Reserved		0
6	**VCEIP**	Voice Interrupt Priority 1: Voice interrupt has higher priority. 0: Voice interrupt has lower priority.	R/W	0
5	**SPIIP**	SPI Interrupt Priority 1:SPI interrupt has higher priority. 0:SPI interrupt has lower priority.	R/W	0
4	**RTCIP**	Sleep Timer Interrupt Priority 1: Sleep Timer interrupt has higher priority. 0: Sleep Timer interrupt has lower priority.	R/W	0
3	**T3IP**	Timer3 Interrupt Priority 1: Timer3 interrupt has higher priority. 0: Timer3 interrupt has lower priority.	R/W	0
2	**AESIP**	AES Interrupt Priority 1: AES interrupt has higher priority. 0: AES interrupt has lower priority.	R/W	0
1	**T2IP**	Timer2 Interrupt Priority 1: Timer2 interrupt has higher priority. 0: Timer2 interrupt has lower priority.	R/W	0
0	**RFIP**	RF Interrupt Priority 1: RF interrupt has higher priority. 0: RF interrupt has lower priority.	R/W	0
EXIF (EXTENDED INTERRUPT FLAG REGISTER, 0x91) This register stores the interrupt state corresponding to each bit. When the interrupt corresponding to a bit is triggered, the flag is set to '1'.				
7	**T3IF**	Timer3 Interrupt Flag. 1: Interrupt pending	R/W	0
6	**AESIF**	AES Interrupt Flag. 1: Interrupt pending	R/W	0
5	**T2IF**	Timer2 Interrupt Flag. 1: Interrupt pending	R/W	0
4	**RFIF**	RF Interrupt Flag. 1: Interrupt pending	R/W	0
3:0		Reserved		0
EICON (EXTENDED INTERRUPT CONTROL REGISTER, 0xD8)				
7		Reserved		0
6:4		Reserved		0
3	**RTCIF**	Sleep Timer Interrupt Flag. 1: Interrupt pending	R/W	0
2:0		Reserved		0

1.6 POWER MANAGEMENT

There are three Power-Down modes in the ZIC2410. Each mode can be set by PDMODE [1:0] bits in PDCON (0x22F1) register and Power-Down mode can be started by setting PDSTART bit to 1. Each mode has a different current consumption and different wake-up sources. Table 8 describes the three Power-Down modes.

Table 8 – Power Down Modes

PDMODE [1:0]	Description	Wake-Up Source	Regulator for Digital block	Current
0	No power-down	-	-	-
1	PM1 mode	Hardware Reset, Sleep Timer interrupt, External interrupt	ON	25μA
2	PM2 mode	Hardware Reset, Sleep Timer interrupt, External interrupt	OFF (After wake-up, register configuration is required)	<2μA
3	PM3 mode	Hardware Reset, External interrupt	OFF (After wake-up, register configuration is required)	0.3μA

The following describes the time it takes from Power-Down mode to system operation for each of the wake-up sources.

① Hardware Reset Wake Up

Hardware Reset Wake Up time in PM1, PM2 and PM3 is around 1001μsec. For more detailed information, refer to the Figure 35.

② Sleep Timer Interrupt Wake Up

The following shows the timing of the Sleep Timer Interrupt Wake Up. As shown in Figure 9 below, the time of Power Down mode is set by register RTINT and register RTDLY should be set at greater than or equal to '0x11' in order to stabilize the crystal. In the case of PM1 and PM2,the minimum time until the system is operating after going into the Power Down mode, is around 534μsec (RTINT:0x01, RTDLY:0x11).

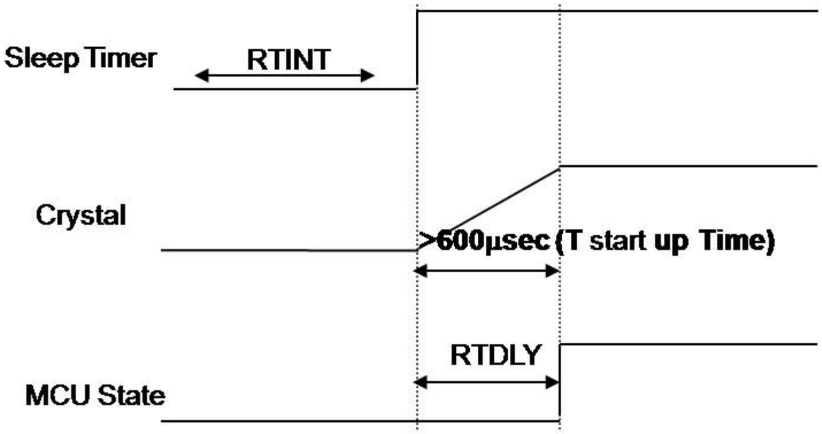


Figure 9 – Sleep Timer Interrupt: Wake Up Times

Based on the CEL' reference circuit.

③ External interrupt Wake Up

The following shows the time of External Interrupt Wake Up. The time, until system is operated, is different based on the releasing time of external interrupt. For example, external interrupt can be released before RTDLY minimum time or after RTDLY minimum time. By considering these two causes, it is recommended to set RTDLY to over 600μsec at least. In addition, Register RTDLY should be set over '0x11' at least to stabilize crystal.

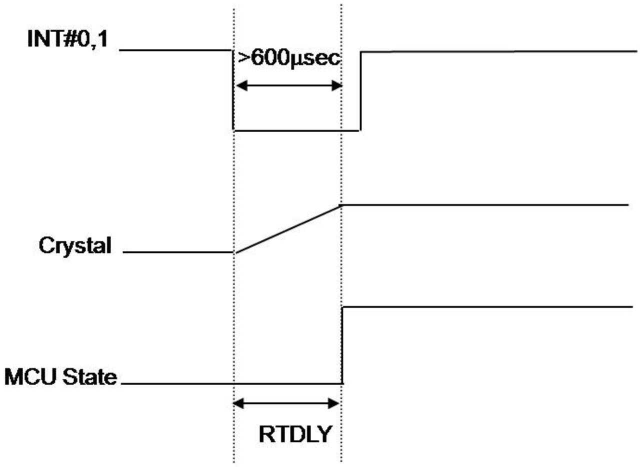


Figure 10 – External Timer Interrupt: Wake Up Times

Based on the CEL' reference circuit.

The following table describes the status of voltage regulator, oscillator, and sleep timer in normal mode (PM0) and each Power-Down mode.

Table 9 – Status in Power-Down Modes

Power Mode	AVREG	DVREG	Main OSC	Sleep Timer
PM0	ON	ON	ON	ON
PM1	OFF	ON	OFF	ON
PM2	OFF	OFF	OFF	ON
PM3	OFF	OFF	OFF	OFF

When exiting from a Power-Down mode initiated by a Sleep Timer interrupt, RTDLY (0x22F4) register specifies the delay time for oscillator stabilization. If the delay time is too short, the oscillator can become unstable and cause a problem of fetching a wrong instruction command in the MCU.

In addition, there are two Power-Down modes that can be only used in the MCU. One is PD (Power-Down) mode and the other is IDLE mode. PD (Power-Down) mode of MCU is enabled by setting PD in PCON register to '1'. In PD (Power-Down) mode, all the clocks of MCU are stopped and current consumption is minimized. When interrupt, which is allowed for wake-up, occurs, it exits from PD mode. After exiting, first, the corresponding interrupt service routine is executed. And then, the next instruction after the instruction for setting PD to '1' is executed.

In IDLE mode, clocks of all the blocks in the MCU except the peripherals are stopped. The current consumption is 2.7mA. When an interrupt occurs (except a timer interrupt or an external interrupt) the IDLE bit is cleared and the device exits from the IDLE mode. The required interrupt service routine is then executed and the next instruction (after the instruction setting IDLE to '1') is executed.

Table 10 – Power Control Registers

Bit	Name	Descriptions	R/W	Reset Value
PCON (POWER CONTROL REGISTER, 0x87)				
7:2		Reserved		0
1	**PD**	Power-down Mode. When this field is set to ‘1’, all the clocks in MCU are stopped.	R/W	0
0	**IDLE**	Idle Mode. When this field is set to ‘1’, all the clocks in MCU except peripherals are stopped. Only peripherals operate normally.	R/W	0

When ZIC2410 goes into Power-Down mode by setting PDSTART field of PDM register, PD bit of PCON register should also be set. To go into PD (Power-Down) mode, PDMODE field should be set as 1, 2, or 3. After that, PD bit of PCON register should be set to 1 by the following instruction that set PDMODE. For more detailed information, please refer to the Figure 11.

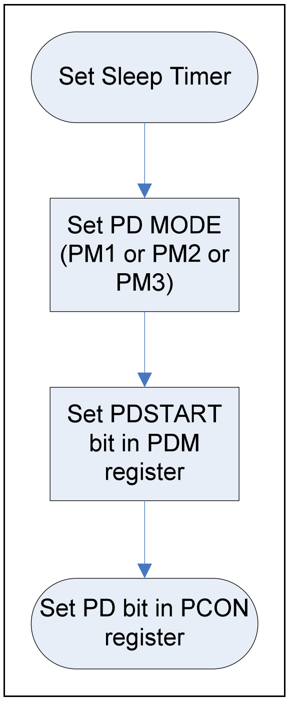

Set Sleep Timer
Set PD MODE (PM1 or PM2 or PM3)
Set PDSTART bit in PDM register
Set PD bit in PCON register

Figure 11 – Power-Down mode setting procedure