SN8F5708

USER'S MANUAL

Preliminary Version 0.1

SN8F5708
SN8F57081
SN8F5707
SN8F5705

SONiX 8-Bit Micro-Controller

SONIX reserves the right to make change without further notice to any products herein to improve reliability, function or design. SONIX does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. SONIX products are not designed, intended, or authorized for us as components in systems intended, for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SONIX product could create a situation where personal injury or death may occur. Should Buyer purchase or use SONIX products for any such unintended or unauthorized application. Buyer shall indemnify and hold SONIX and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that SONIX was negligent regarding the design or manufacture of the part.

AMENDENT HISTORY

Version	Date	Description
VER 0.1	Apr. 2015	First issue.
VER 0.2	May 2015	1. Remove PDIP48 package type. 2. Add QFN46 package type and modify QFN32 5x5 package type to QFN32 4x4.

Table of Content

1 PRODUCT OVERVIEW

SN8F5708 8-bit micro-controller is a new series 8051 based production applied advanced semiconductor technology to implement flash ROM architecture. Under flash ROM platform, SN8F5708 builds in in-system-programming (ISP) function extending to EEPROM emulation and On Chip Debug Support (OCDS). It offers 6-set individual programmable PWMs, 3-type serial interfaces, 2-Set OP Amp, 2-Set Comparator, high performance 12+4-channel 12-bit ADC and flexible operating modes. Powerful functionality, high reliability and low power consumption can apply to AC power application and battery level application easily.

1.1 FEATURES

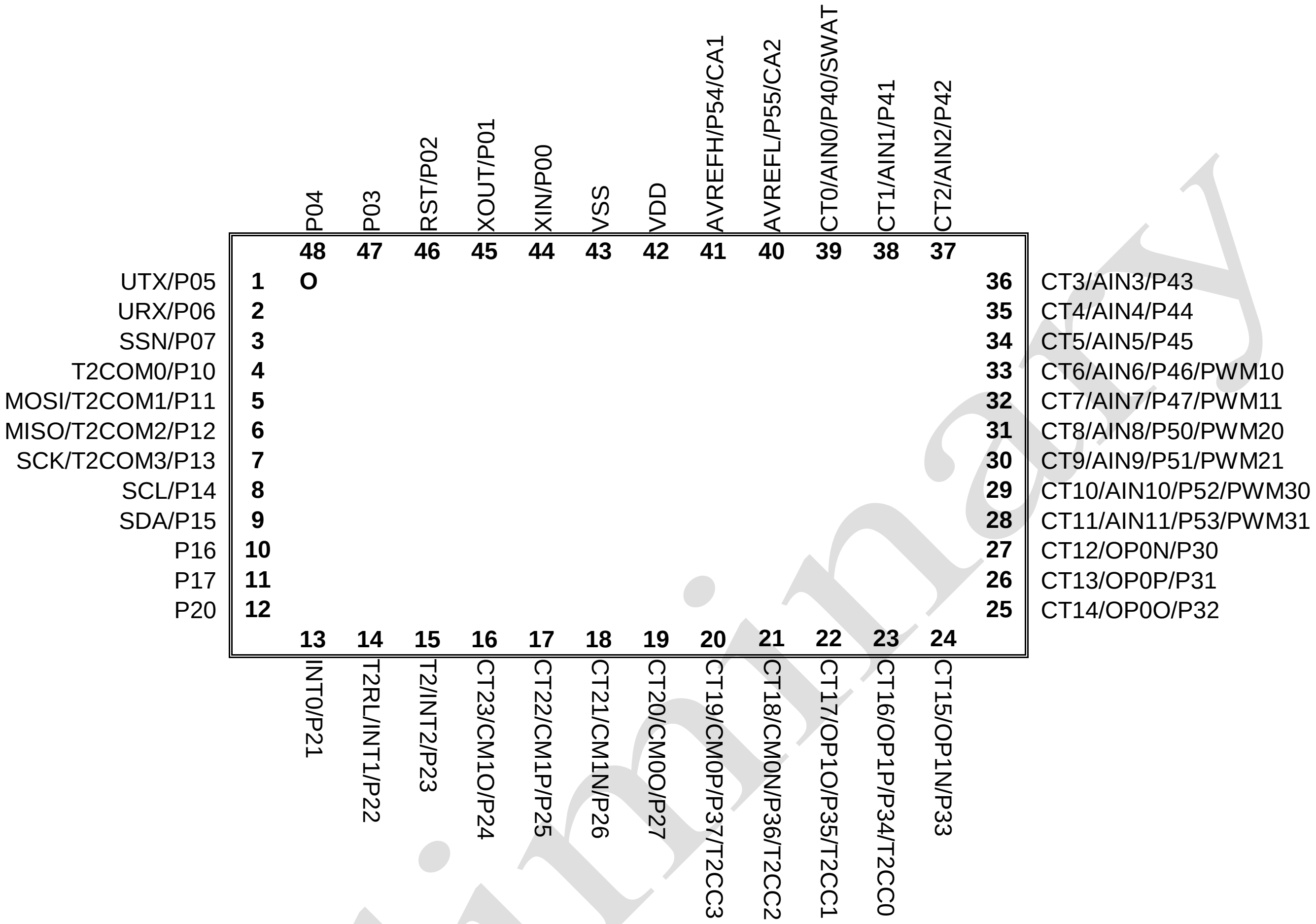
- **Memory configuration**
  Flash ROM size: 16K x 8 bits. Including EEROM emulation. (In system programming)
  Internal RAM size: 256 x 8 bits.
  Extension RAM size: 1024 x 8 bits.
- **8-bit stack pointer**
- **19 interrupt sources**
  16 internal interrupts: T0, T1, T2, UART, SPI, I2C, ADC, PW1, PW2, PW3, CMP0, CMP1, T2 COM0/1/2/3
  3 external interrupts: INT0, INT1, INT2
- **Multi-interrupt vector structure.**
  Each of interrupt sources has a unique interrupt vector.
- **Four Priority Levels Interrupt Control.**
- **I/O pin configuration**
  Bi-directional: P0, P1, P2, P3, P4, P5.
  Wakeup: P0, P1 level change.
  Pull-up resisters: P0, P1, P2, P3, P4, P5.
  External interrupt: INT0, INT1, INT2
  ADC input pin: AIN0~AIN11.
- **Fcpu (Instruction cycle)**
  Fcpu = Fhosc/1, Fhosc/2, Fhosc/4, Fhosc/8, Fhosc/16, Fhosc/32, Fhosc/64, Fhosc/128
- **On chip watchdog timer and clock source**
- **1.8V/2.4V/3.3V 3-level LVD with trim.**
- **8051 instructions**
  Keil-C Compatible.
  32-bit division, 16-bit multiplication.
- **On Chip Debug Support (OCDS)**
  One-wire interface.
  Keil uVision IDE Compatible.
- **Tree 16-bit timers. (T0, T1, T2).**
  T0: Timer0.
  T1: Timer1.
  T2: Timer2/Capture/Compare.
- **3-Set 8~16-bit duty/cycle programmable PWM generator. 2-ch output of each PWM generator with 1-ch deadband.**
- **Serial Interface: SPI, I2C, UART**
- **12+4 channel 12-bit SAR ADC with 3-level Int. Ref.**
  Twelve external ADC input
  One internal battery measurement
  Two OP output terminal
  Internal AD reference voltage (VDD, 4V, 3V, 2V).
- **2-Set OP Amp**
- **2-Set Comparator with 3-level Int. Ref.**
- **Four system clocks**
  External high clock: Crystal type up to 16MHz
  External clock input: Digital clock input up to 32MHz.
  Internal high clock: RC type 32MHz
  Internal low clock: RC type 32KHz
- **Three operating modes**
  Normal mode: Both high and low clock active
  Stop mode: Both high and low clock stop
  Idle mode: Periodical wakeup by timer
- **Package (Chip form support)**
  LQFP 48 pin
  QFN 48 pin
  QFN 46 pin
  LQFP 44 pin
  LQFP 32 pin
  QFN 32 pin

☞ **Features Selection Table**

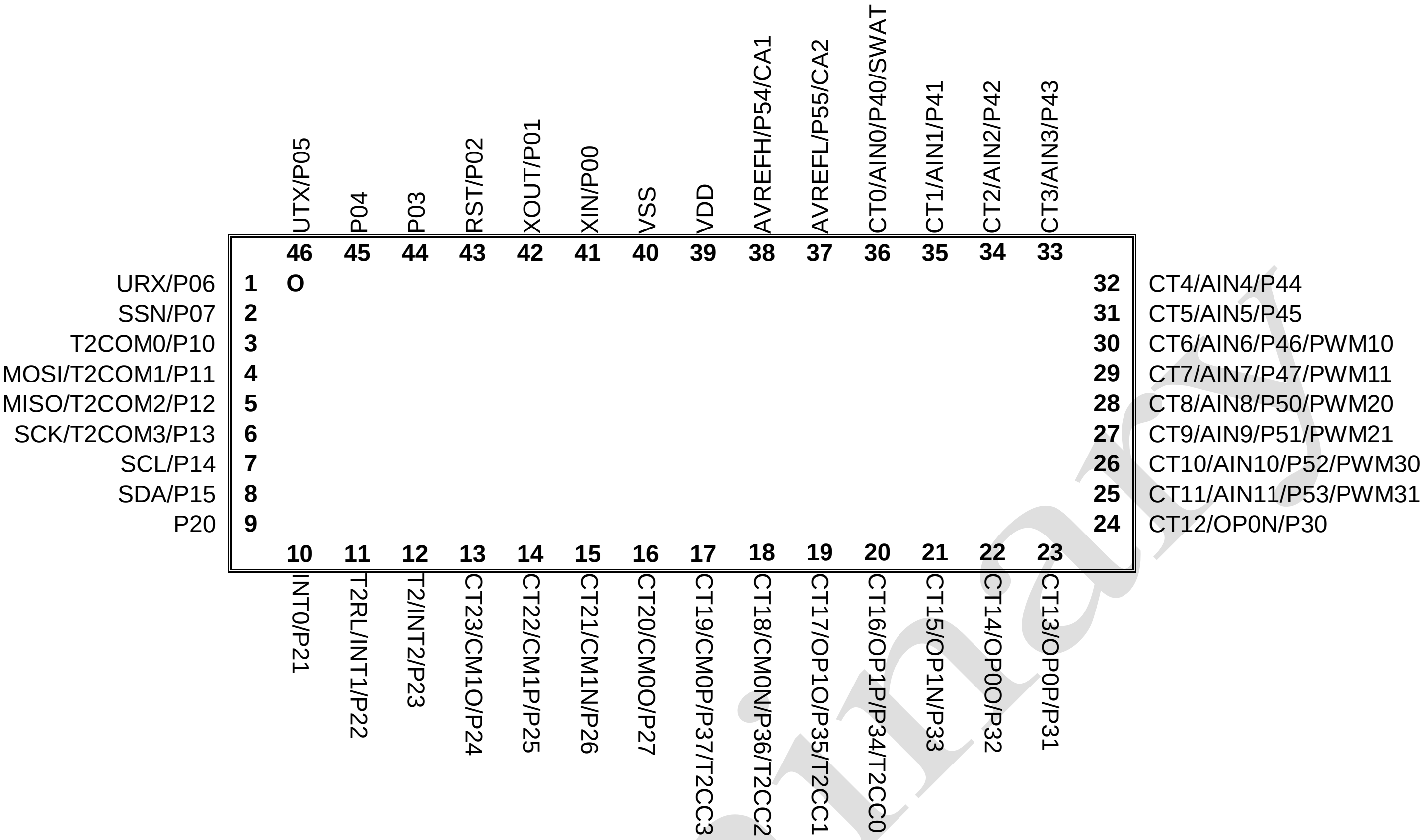
CHIP	ROM	RAM	Stack	Timer	I/O	PWM	I2C	SPI	UART	ADC CH	OP	CMP	Ext. INT	ISP/ OCDS	Operating Voltage	Package
SN8F5708	16K*8	128*8 1024*8	8bit	16-bit*3	46	6-ch	V	V	V	12+4	2	2	3	V	1.8V~5.5V	LQFP48 QFN48
SN8F57081	16K*8	128*8 1024*8	8bit	16-bit*3	44	6-ch	V	V	V	12+4	2	2	3	V	1.8V~5.5V	QFN46
SN8F5707	16K*8	128*8 1024*8	8bit	16-bit*3	42	6-ch	V	V	V	12+4	2	2	3	V	1.8V~5.5V	LQFP44
SN8F5705	16K*8	128*8 1024*8	8bit	16-bit*3	30	6-ch	V	V	V	7+4	2	2	1	V	1.8V~5.5V	LQFP32 QFN32

1.2 PIN ASSIGNMENT

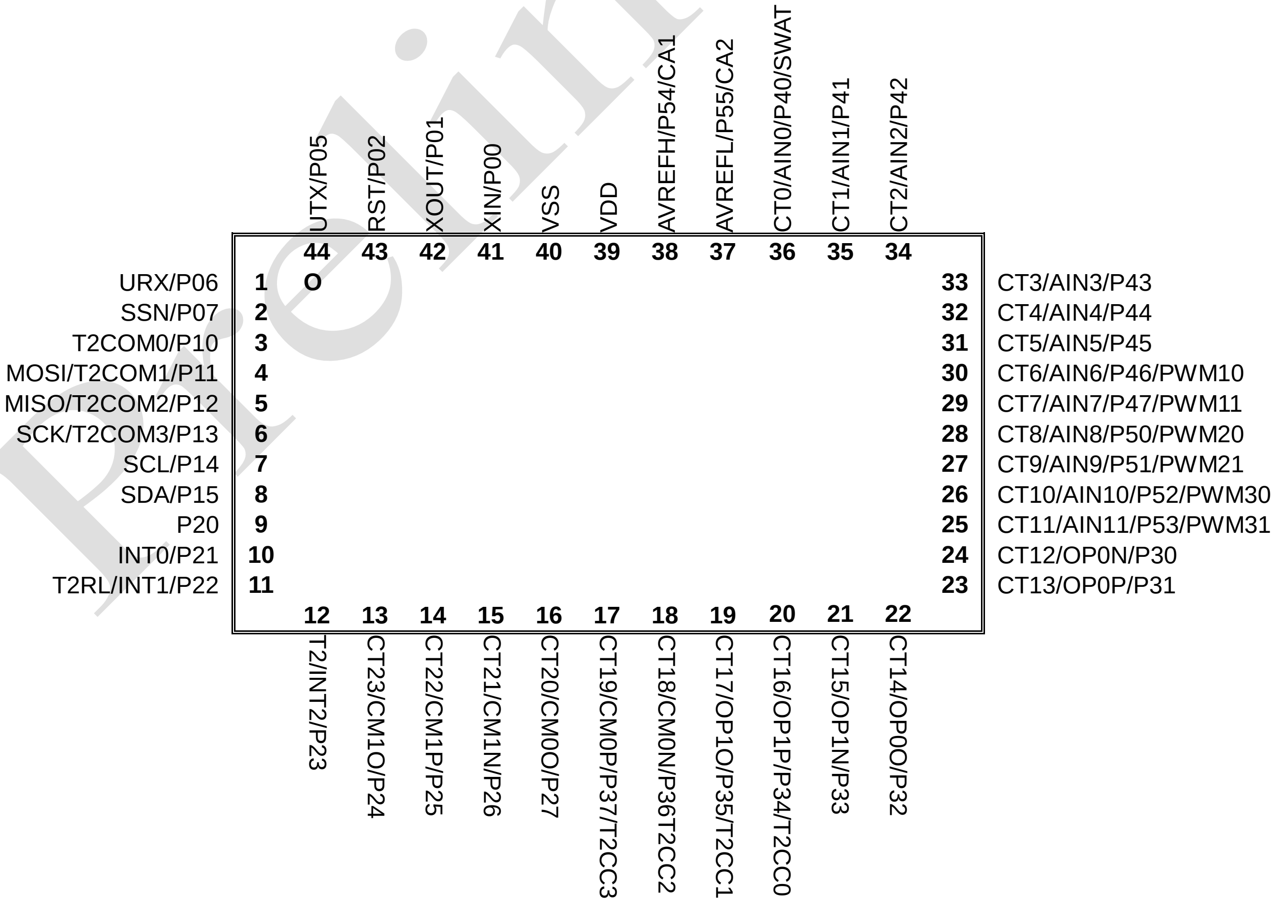
SN8F5708F/J (LQFP48 pin/QFN48 pin):



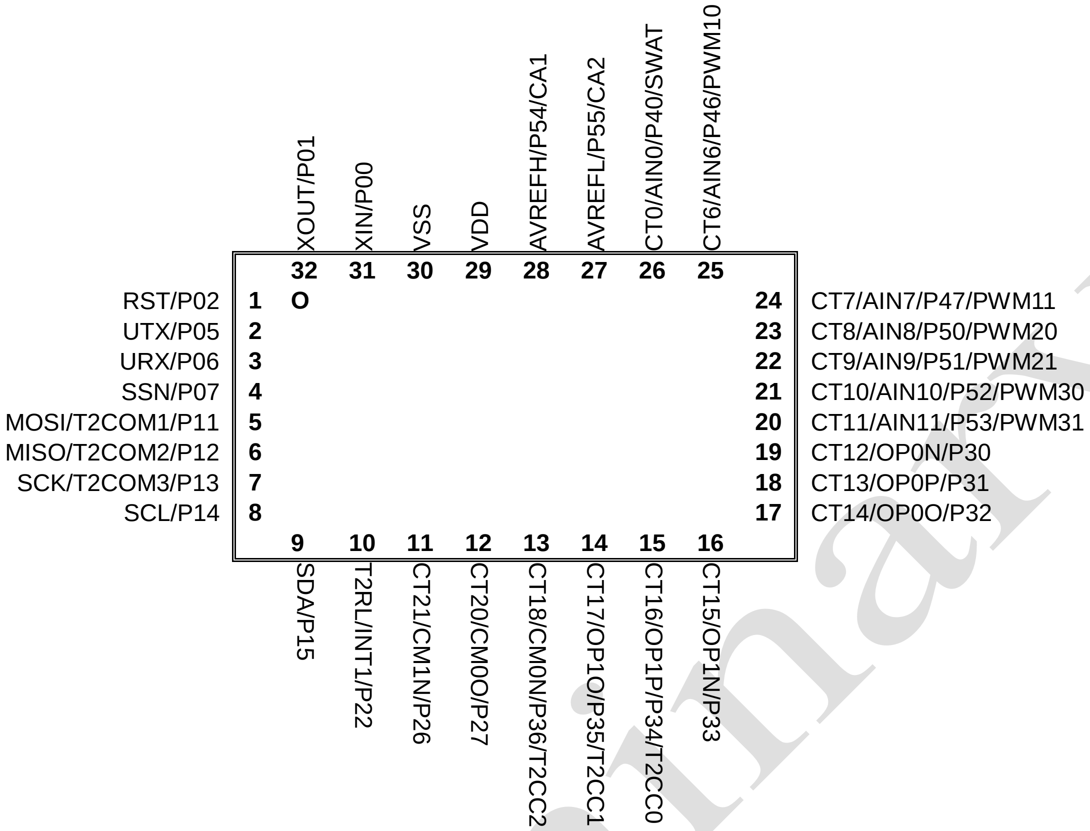
SN8F57081J (QFN46 pin):



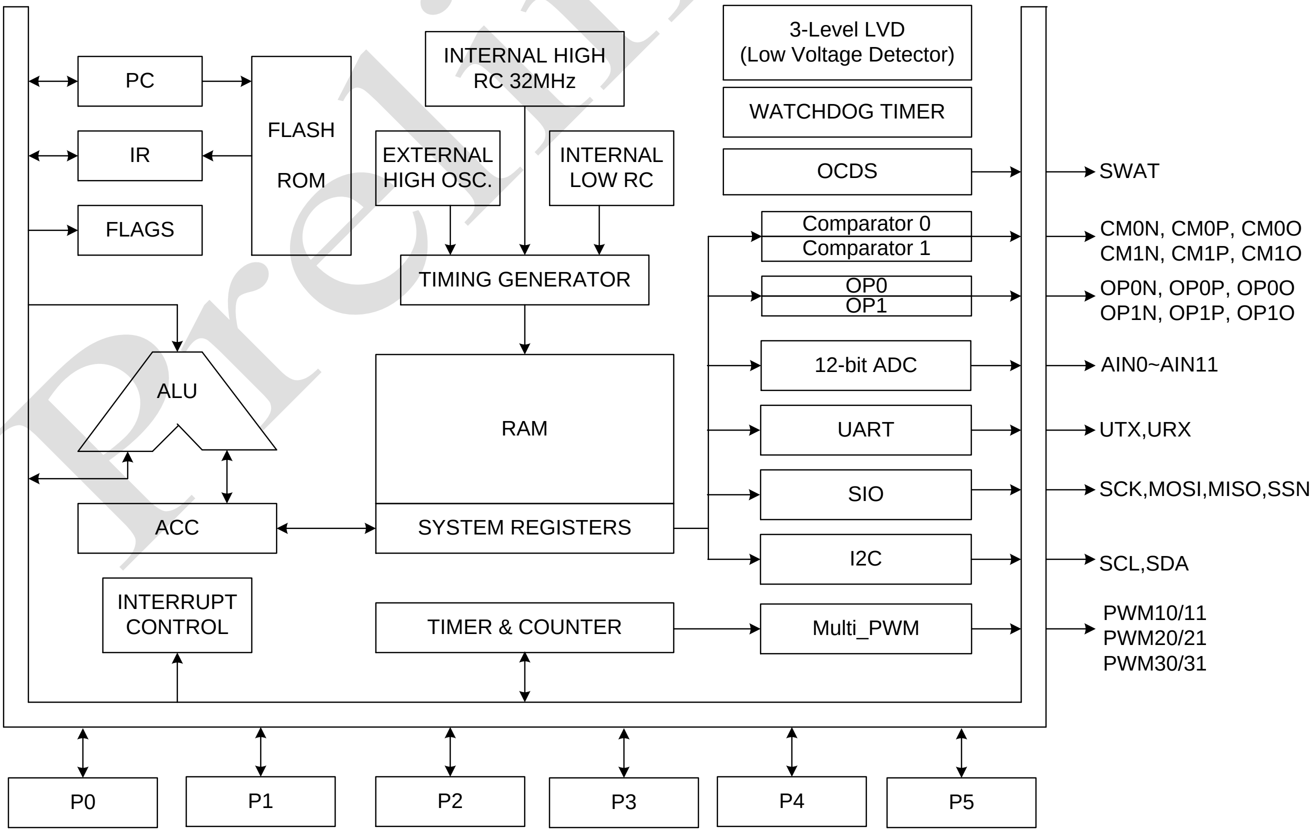
SN8F5707F (LQFP44):



SN8F5705F/J (LQFP32 pin/QFN32 pin):



1.3 SYSTEM BLOCK DIAGRAM



1.4 PIN DESCRIPTIONS

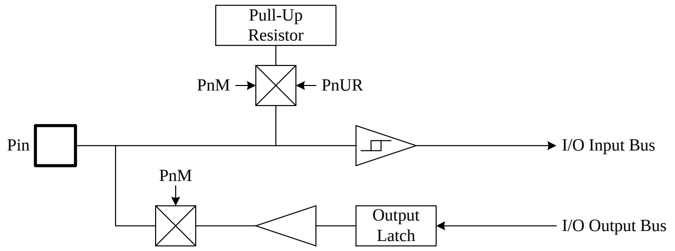
PIN NAME	TYPE	DESCRIPTION
VDD, VSS	P	Power supply input pins for digital and analog circuit.
P0.0/XIN	I/O	P0.0: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. XIN: Oscillator input pin while external crystal enable.
P0.1/XOUT	I/O	P0.1: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. XOUT: Oscillator output pin while external crystal enable.
P0.2/RST	I/O	P0.2: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. RST: System external reset input pin. Schmitt trigger structure, active “low”, normal stay to “high”.
P0.3	I/O	P0.3: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
P0.4	I/O	P0.4: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
P0.5/UTX	I/O	P0.5: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. UTX: UART transmit output pin.
P0.6/URX	I/O	P0.6: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. URX: UART receive input pin.
P0.7/SSN	I/O	P0.7: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. SSN: Slave selection.
P1.0/T2COM0	I/O	P1.0: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. T2COM0: T2 Compare0 output
P1.1/MOSI/ T2COM1	I/O	P1.1: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. MOSI: Master out slave in. T2COM1: T2 Compare1 output
P1.2/MISO/ T2COM2	I/O	P1.2: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. MISO: Master in slave out. T2COM2: T2 Compare2 output
P1.3/SCK/ T2COM3	I/O	P1.3: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. SCK: SIO clock pin. T2COM3: T2 Compare3 output
P1.4/SCL	I/O	P1.4: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. SCL: MSP master mode clock pin.
P1.5/SDA	I/O	P1.5: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up. SDA: MSP master mode data pin.
P1.6	I/O	P1.6: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
P1.7	I/O	P1.7: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. Level change wake-up.
P2.0	I/O	P2.0: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
P2.1/INT0	I/O	P2.1: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. INT0: External interrupt 0 input pin.
P2.2/INT1/ T2RL	I/O	P2.2: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. INT1: External interrupt 1 input pin. T2RL: T2 reload trigger input.
P2.3/INT2/T2	I/O	P2.3: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. INT2: External interrupt 2 input pin. T2: T2 event counter input pin.
P2.4/CM1O/ CT23	I/O	P2.4: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters. CM1O: The output pin of comparator.

		CT23: Capacitive touch channel 23 input pin
P2.5/CM1P/ CT22	I/O	P2.5: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		CM1P: The positive input pin of comparator
		CT22: Capacitive touch channel 22 input pin.
P2.6/CM1N/ CT21	I/O	P2.6: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		CM1N: The negative input pin of comparator
		CT21: Capacitive touch channel 21 input pin.
P2.7/CM0O/ CT20	I/O	P2.7: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		CM0O: The output pin of comparator.
		CT20: Capacitive touch channel 20 input pin.
P3.0/OP0N/ CT12	I/O	P3.0: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		OP0N: The negative input pin of OP
		CT12: Capacitive touch channel 12 input pin.
P3.1/OP0P/ CT13	I/O	P3.1: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		OP0P: The positive input pin of OP.
		CT13: Capacitive touch channel 13 input pin.
P3.2/OP0O/ CT14	I/O	P3.2: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		OP0O: The output pin of OP.
		CT4: Capacitive touch channel 14 input pin.
P3.3/OP1N/ CT15	I/O	P3.3: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		OP1N: The negative input pin of OP
		CT15: Capacitive touch channel 15 input pin.
P3.4/OP1P/ T2CC0/CT16	I/O	P3.4: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		OP1P: The positive input pin of OP
		T2CC0: T2 Capture0 input.
		CT16: Capacitive touch channel 16 input pin.
P3.5/OP1O/ T2CC1/CT17	I/O	P3.5: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		OP1O: The output pin of OP.
		T2CC1: T2 Capture1 input.
		CT17: Capacitive touch channel 17 input pin.
P3.6/CM0N/ T2CC2/CT18	I/O	P3.6: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		CM0N: The negative input pin of comparator
		T2CC2: T2 Capture2 input.
		CT18: Capacitive touch channel 18 input pin.
P3.7/CM0P/ T2CC3/CT19	I/O	P3.7: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		CM0P: The positive input pin of comparator
		T2CC3: T2 Capture3 input.
		CT19: Capacitive touch channel 19 input pin.
P4.0/AIN0/ SWAT/CT0	I/O	P4.0: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resistor.
		AIN0: ADC channel 0 input pin.
		SWAT: Single Wire Asynchronous Interface pin.
		CT0: Capacitive touch channel 0 input pin.
P4.1/AIN1/CT1	I/O	P4.1: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AIN1: ADC channel 1 input pin.
		CT1: Capacitive touch channel 1 input pin.
P4.2/AIN2/CT2	I/O	P4.2: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AIN2: ADC channel 2 input pin.
		CT2: Capacitive touch channel 2 input pin.
P4.3/AIN3/CT3	I/O	P4.3: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AIN3: ADC channel 3 input pin.
		CT3: Capacitive touch channel 3 input pin.
P4.4/AIN4/CT4	I/O	P4.4: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AIN4: ADC channel 4 input pin.
		CT4: Capacitive touch channel 4 input pin.
P4.5/AIN5/CT5	I/O	P4.5: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AIN5: ADC channel 5 input pin.
		CT5: Capacitive touch channel 5 input pin.
P4.6/AIN6/ PWM10/CT6	I/O	P4.6: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AIN6: ADC channel 6 input pin.
		PWM10: PW1 programmable PWM output pin.
		CT6: Capacitive touch channel 6 input pin.
P4.7/AIN7/ PWM11/CT7	I/O	P4.7: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AIN7: ADC channel 7 input pin.
		PWM11: PW1 programmable PWM output pin.

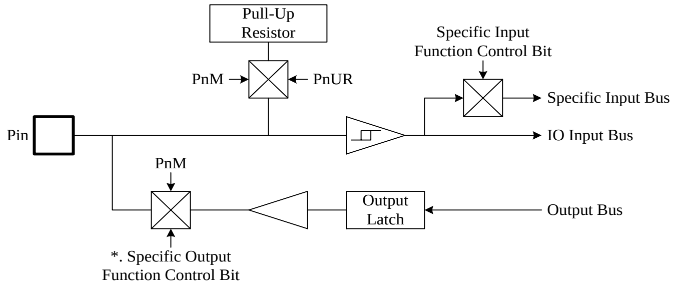
		CT7: Capacitive touch channel 7 input pin.
P5.0/AIN8/ PWM20/CT8	I/O	P5.0: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AIN8: ADC channel 8 input pin.
		PWM20: PW2 programmable PWM output pin.
		CT8: Capacitive touch channel 8 input pin.
P5.1/AIN9/ PWM21/CT9	I/O	P5.1: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AIN9: ADC channel 9 input pin.
		PWM21: PW2 programmable PWM output pin.
		CT9: Capacitive touch channel 9 input pin.
P5.2/AIN10/ PWM30/CT10	I/O	P5.2: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AIN10: ADC channel 10 input pin.
		PWM30: PW3 programmable PWM output pin.
		CT10: Capacitive touch channel 10 input pin.
P5.3/AIN11/ PWM31/CT11	I/O	P5.3: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AIN11: ADC channel 11 input pin.
		PWM31: PW3 programmable PWM output pin.
		CT11: Capacitive touch channel 11 input pin.
P5.4/AVREFH/ CA1	I/O	P5.4: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AVREFH: ADC external high reference voltage input.
		CA1: CA1: Capacitive ADC Sensing pin 1.
P5.5/AVREFL/ CA2	I/O	P5.5: Bi-direction pin. Schmitt trigger structure as input mode. Built-in pull-up resisters.
		AVREFL: ADC external high reference voltage input.
		CA2: CA2: Capacitive ADC Sensing pin 2.

1.5 PIN CIRCUIT DIAGRAMS

- **Normal Bi-direction I/O Pin.**

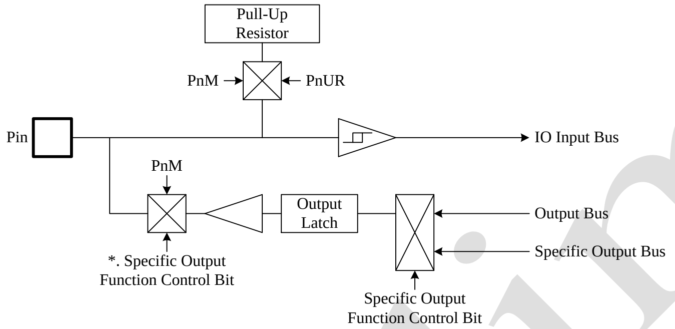


- **Bi-direction I/O Pin Shared with Specific Digital Input Function, e.g. INT0, Event counter, UART…**



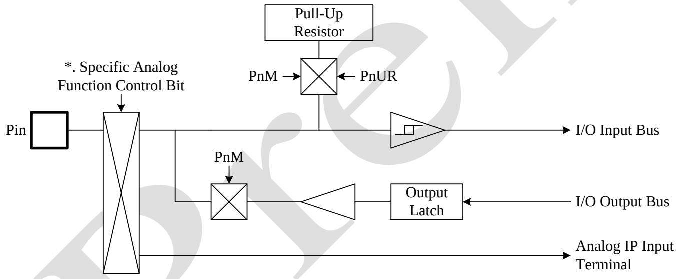
***. Some specific functions switch I/O direction directly, not through PnM register.**

- **Bi-direction I/O Pin Shared with Specific Digital Output Function, e.g. PWM, UART…**



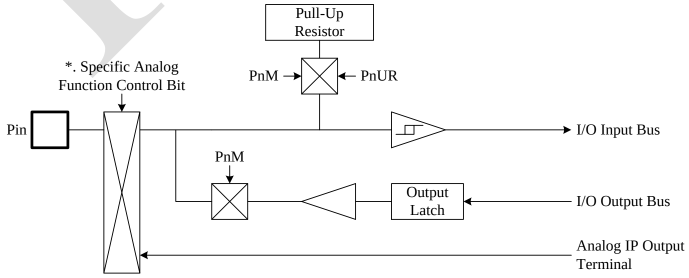
***. Some specific functions switch I/O direction directly, not through PnM register.**

- **Bi-direction I/O Pin Shared with Specific Analog Input Function, e.g. XIN, ADC…**



***. Some specific functions switch I/O direction directly, not through PnM register.**

- **Bi-direction I/O Pin Shared with Specific Analog Output Function, e.g. XOUT…**



***. Some specific functions switch I/O direction directly, not through PnM register.**

2 CENTRAL PROCESSOR UNIT (CPU)

2.1 PROGRAM MEMORY (ROM)

☞ ***16K BYTE ROM***

Address	***ROM***	Comment
0000H	*Reset vector*	Reset vector
0001H 0002H	*General purpose area*	User program
0003H	***INT0 Interrupt vector***	**Interrupt vector**
000BH	***T0 Interrupt vector***	
0013H	***INT1 Interrupt vector***	
001BH	***T1 Interrupt vector***	
0023H	***UART Interrupt vector***	
002BH	***T2 Interrupt vector***	
0043H	***I2C Interrupt vector***	
004BH	***SPI Interrupt vector***	
0053H	***T2 COM0 Interrupt vector***	
005BH	***T2 COM1 Interrupt vector***	
0063H	***T2 COM2 Interrupt vector***	
006BH	***T2 COM3 Interrupt vector***	
0083H	***PWM1 Interrupt vector***	
008BH	***PWM2 Interrupt vector***	
0093H	***PWM3 Interrupt vector***	
009BH	***ADC Interrupt vector***	
00A3H	***Comparator 0 Interrupt vector***	
00ABH	***Comparator 1 Interrupt vector***	
00EBH	***INT2 Interrupt vector***	
00ECH	*General purpose area*	User program End of user program
3FF6H 3FF7H . 3FFDH 3FFEH 3FFFH	*Reserved*	

The ROM includes Reset vector, Interrupt vector, General purpose area and Reserved area. The Reset vector is program beginning address. The Interrupt vector is the head of interrupt service routine when any interrupt occurring. The General purpose area is main program area including main loop, sub-routines and data table.

- **0x0000 Reset Vector: Program counter points to 0x0000 after any reset events (power on reset, reset pin reset, watchdog reset, LVD reset…).**
- **0x0001~0x0002: General purpose area to process system reset operation.**
- **0x0003~0x00EB: Multi interrupt vector area. Each of interrupt events has a unique interrupt vector.**
- **0x00EC~0x3FE0: General purpose area for user program and ISP (EEPROM function).**
- **0x3FE0~0x3FF6: General purpose area for user program. Do not execute ISP.**
- **0x3FF6~0x3FFF: Reserved area. Do not execute ISP.**
- **ROM security rule is all address ROM data protected and outputs 0x0000.**

2.1.1 RESET VECTOR (0000H)

A one-byte vector address area is used to execute system reset.

- **Power On Reset (POR=1).**
- **Watchdog Reset (WDT=1).**
- **External Reset (RST=1).**

After power on reset, external reset or watchdog timer overflow reset, then the chip will restart the program from address 0000h and all system registers will be set as default values. It is easy to know reset status from POR, WDT and RST flags of PFLAG register. The following example shows the way to define the reset vector in the program memory.

- **Example: Defining Reset Vector**

```
            ORG      0             ; 0000H
            JMP      START         ; Jump to user program address.
            …

            ORG      0ECH
START:                             ; 00ECH, The head of user program.
            …                      ; User program
            …


            END                    ; End of program
```

> ✸ ***Note: The head of user program should skip interrupt vector area to avoid program execution error.***

2.1.2 INTERRUPT VECTOR (0003H~00EBH)

A 19-byte vector address area is used to execute interrupt request. If any interrupt service executes, the program counter (PC) value is stored in stack buffer and jump to 0003h~00EBh of program memory to execute the vectored interrupt. This interrupt is multi-vector and each of interrupts points to unique vector. Users have to define the interrupt vector. The following example shows the way to define the interrupt vector in the program memory.

Interrupt Vector

	ROM
0003H	***INT0 Interrupt vector***
000BH	***T0 Interrupt vector***
0013H	***INT1 Interrupt vector***
001BH	***T1 Interrupt vector***
0023H	***UART Interrupt vector***
002BH	***T2 Interrupt vector***
0043H	***I2C Interrupt vector***
004BH	***SPI Interrupt vector***
0053H	***T2 COM0 Interrupt vector***
005BH	***T2 COM1 Interrupt vector***
0063H	***T2 COM2 Interrupt vector***
006BH	***T2 COM3 Interrupt vector***
0083H	***PWM1 Interrupt vector***
008BH	***PWM2 Interrupt vector***
0093H	***PWM3 Interrupt vector***
009BH	***ADC Interrupt vector***
00A3H	***Comparator 0 Interrupt vector***
00ABH	***Comparator 1 Interrupt vector***
00EBH	***INT2 Interrupt vector***

When one interrupt request occurs, and the program counter points to the correlative vector to execute interrupt service routine. If INT0 interrupt occurs, the program counter points to ORG 3. If T0 interrupt occurs, the program counter points to ORG 0BH. In normal condition, several interrupt requests happen at the same time. So the priority of interrupt sources is very important, or the system doesn't know which interrupt is processed first. The interrupt priority is decided by natural priority and priority level (The detail will be shown in the Section 6.3).

- **Example: Defining Interrupt Vector. The interrupt service routine is following user program.**

```
                ORG         0               ; 0000H
                JMP         START           ; Jump to user program address.
                …
                ORG         0X0003          ; Jump to interrupt service routine address.
                JMP         ISR_INT0
                ORG         0X000B
                JMP         ISR_T0
                ORG         0X0013
                JMP         ISR_INT1
                …
                …
                ORG         0X0023
                JMP         ISR_UART
                ORG         0X00EB
                JMP         ISR_INT2
                …
                ORG         0X00ECH
START:                                      ; 00ECH, The head of user program.
                …                           ; User program.
                …
                …
                JMP         START           ; End of user program.
```

```
                  …
ISR_ INT0:                                           ; The head of interrupt service routine.
                  PUSH       ACC                     ; Save ACC to stack buffer.
                  PUSH       PSW                     ; Save PSW to stack buffer.
                  …
                  POP        PSW                     ; Load PSW from stack buffer.
                  POP        ACC                     ; Load ACC from stack buffer.
                  RETI                               ; End of interrupt service routine.
ISR_T0:                                              ;
                  PUSH       ACC                     ; Save ACC to stack buffer.
                  PUSH       PSW                     ; Save PSW to stack buffer.
                  …
                  POP        PSW                     ; Load PSW from stack buffer.
                  POP        ACC                     ; Load ACC from stack buffer.
                  RETI                               ; End of interrupt service routine.
                  …
                  …
                  …
                  …
                  …
ISR_INT2                                             ;
                  PUSH       ACC                     ; Save ACC to stack buffer.
                  PUSH       PSW                     ; Save PSW to stack buffer.
                  …
                  POP        PSW                     ; Load PSW from stack buffer.
                  POP        ACC                     ; Load ACC from stack buffer.
                  RETI                               ; End of interrupt service routine.


                  END                                ; End of program.
```

2.2 DATA MEMORY (RAM)

☞ ***256 X 8-bit RAM (Internal Data Memory)***

Address	RAM Location		
000h 01Fh	***Work Register Area***		RAM Bank 0 00h~7Fh of Bank 0 is direct and indirect access RAM
020h 02Fh	***Bit Addressable Area***		
030h … 07Fh	***General Purpose Area***		
080h … 0FFh	***General Purpose Area (Indirect Access)***	***Special Function Register (Direct Access)***	080h~0FFh of Bank 0 store special function registers. End of Bank 0

The 256-byte data RAM in internal data memory is a standard 8051 RAM access configuration. The upper 128-byte RAM is general purpose RAM and can configure by direct addressing access and indirect addressing access. The lower 128-byte can be indirect access RAM in general purpose or direct access RAM in special function register (SFR).

- **0x0000~0x007F: General purpose RAM contains work register area and bit addressable area. In this area, direct or indirect addressing can be used.**
- **0x0000~0x001F: Work register area includes 4-bank. Each bank has 8 work registers (R0~R7) which is selected by RS0/RS1 in PSW register.**
- **0x0020~0x002F: Bit addressable area. It is accessible via address 0x0000~0x007F.**
- **0x0080~0x00FF: General purpose area in indirect addressing access or special function register in direct addressing access**

☞ ***1024 X 8-bit SRAM (Extension Data Memory)***

Address	RAM Location	
0000h 03FFh	***General Purpose Area***	Extension RAM Bank 0 End of Bank 0

SN8F5708 support additional 1024-byte RAM in extension data memory.

2.2.1 BIT ADDRESSABLE AREA

In the bit addressable area, user can read or write any single bit in this range by using the unique address for that bit. SN8F5708 supports 16bytes bit addressable RAM area giving 128 addressable bits. Each bit has individual address in the range from 00H to 7FH. Thus, the bit can be addressed directly. Bit 0 of the byte 20H has bit address 00H and Bit 7 of the byte 20H has bit address 07H. Bit 0 of the byte 2FH has bit address 78H and Bit 7 of the byte 2FH has bit address 7FH. When set "SETB 42H", it means the bit2 of the byte 28H is set.

Bit addressable area table

Bit Addressable Area	Byte Address	Bite 0	Bite 1	Bite 2	Bite 3	Bite 4	Bite 5	Bite 6	Bite 7
	0x20	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
	0x21	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
	0x22	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
	0x23	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
	0x24	0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
	0x25	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
	0x26	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
	0x27	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
	0x28	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47
	0x29	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
	0x2A	0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57
	0x2B	0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F
	0x2C	0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67
	0x2D	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F
	0x2E	0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77
	0x2F	0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F

2.2.2 SYSTEM REGISTER

2.2.2.1 SYSTEM REGISTER TABLE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80	P0	SP	DPL	DPH	DPL1	DPH1	WDTR	PCON	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PEDGE
90	P1	P1W	DPS	DPC	PECMD	PEROML	PEROMH	PERAM	S0CON	S0BUF	IEN2	OPM	CMP0M	CMP1M	P2CON	P3CON
A0	P2	PW2M	PW2YL	PW2YH	PW2BL	PW2BH	PW2DL	PW2DH	IEN0	IP0	S0RELL	PW1M	PW1YL	PW1YH	PW1BL	PW1BH
B0	P3	PW3M	PW3YL	PW3YH	PW3BL	PW3BH	PW3DL	PW3DH	IEN1	IP1	S0RELH	PW1DH	PW1DL	PW1A	PW2A	IRCON2
C0	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3	T2CON	-	CRCL	CRCH	TL2	TH2	CMPT	PW3A
D0	PSW	IEN4	ADM	ADB	ADR	VREFH	P4CON	P5CON	S0CON2	ADT	I2CDAT	I2CADR	I2CCON	I2CSTA	SMBSEL	SMBDST
E0	ACC	SPSTA	SPCON	SPDAT	P1OC	CLKSEL	CLKCMD	TCON0	P4	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
F0	B	P0UR	P1UR	P2UR	P3UR	P4UR	P5UR	SRST	P5	P0M	P1M	P2M	P3M	P4M	P5M	PFLAG

2.2.2.2 SYSTEM REGISTER DESCRIPTION

SP = Stack pointer register.
PFLAG = Special flag register.
WDTR = Watchdog timer clear register.
PnM = Port n input/output mode register.
Pn = Port n data buffer.
PnUR = Port n pull-up resister control register.
TCON = Timer/ counter control register.
TCON0 = T0/T1 clock control register.
TMOD = T0, T1 mode register.
TL0, TH0 = T0 counting register.
TL1, TH1 = T1 counting register.
TL2, TH2 = T2 counting register.
DPH, DPL = Data pointer register.
DPH1, DPL1 = Data pointer 1 register.
DPS = Data pointer register select register.
DPC = Data pointer control register.
PSW = System flag register.
ACC = Accumulator register.
MD0~MD5 = Multiplication/ division control register.
B = Multiplication/ division data buffer.
ARCON = Arithmetic control register.
VREFH = ADC reference voltage control register.
ADM = ADC mode register.
ADR = ADC resolution select register.
ADT = ADC offset calibration register.
ADB = ADC data buffer.
SPSTA = SPI status register.
SPCON = SPI control register.
SPDAT = SPI data buffer.
CLKSEL = Clock switch select register.
CLKCMD = Clock switch control Register.
CMP0M = Comparator 0 control register.
CMP1M = Comparator 1 control register.
CMPT = Comparator 0 / 1 model register.

PCON = System mode register.
CKCON = System control register.
PEDGE = P2.1~P2.3 edge direction register.
IP0, IP1 = Interrupt priority register.
P1W = P1 wake-up control register.
CCEN = Compare /capture enable register.
CRCH, CRCL = Compare/reload/capture register.
CCHn, CCLn = Compare /capture register.
IEN0~IEN4 = Interrupt enable register
IRCON, IRCON2 = Interrupt request control register.
PWnM = PWMn mode control register.
PWnY = PW1 cycle control buffer.
PWnB = PWM B point dead band control buffer.
PWnD = PWM duty control buffer.
PWnA = PWM A point dead band control buffer.
OPM = OP AMP register.
I2CDAT = I2C data buffer.
I2CADR = Own I2C slave address.
I2CCON = I2C interface operation control register.
I2CSTA = I2C Status Code.
SMBSEL = SMBus mode control register.
SMBDST = SMBus internal timeout register.
P4CON, P5CON = P4, P5 configuration register.
P1OC = Open drain control register.
SRST = Software reset register.
PECMD = ISP command register.
PEROM = ISP ROM address
PERAM = ISP RAM mapping address
S0CON = UART control register.
S0BUF = UART data buffer.
S0RELH, S0RELL = UART reload register.
S0CON2 = UART baud rate control register.
P2CON = P2configuration control register.
P3CON = P3 configuration control register.

2.2.2.3 BIT DEFINITION of SYSTEM REGISTER

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Remarks
080H	P07	P06	P05	P04	P03	P02	P01	P00	R/W	P0
081H	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	R/W	SP
082H	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0	R/W	DPL
083H	DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0	R/W	DPH
084H	DPL17	DPL16	DPL15	DPL14	DPL13	DPL12	DPL11	DPL10	R/W	DPL1
085H	DPH17	DPH16	DPH15	DPH14	DPH13	DPH12	DPH11	DPH10	R/W	DPH1
086H	WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0	W	WDTR
087H	SMOD	-	-	-	P2SEL	GF0	STOP	IDLE	R/W	PCON
088H	TF1	TR1	TF0	TR0	IE1	-	IE0	-	R/W	TCON
089H	T1GATE	T1CT	T1M1	T1M0	T0GATE	T0CT	T0M1	T0M0	R/W	TMOD
08AH	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00	R/W	TL0
08BH	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10	R/W	TL1
08CH	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00	R/W	TH0
08DH	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10	R/W	TH1
08EH	PSYN	PWSC2	PWSC1	PWSC0	ESYN	EWSC2	EWSC1	EWSC0	R/W	CKCON
08FH	-	-	EX2G1	EX2G0	EX1G1	EX1G0	EX0G1	EX0G0	R/W	PEDGE
090H	P17	P16	P15	P14	P13	P12	P11	P10	R/W	P1
091H	P17W	P16W	P15W	P14W	P13W	P12W	P11W	P10W	R/W	P1W
092H	-	-	-	-	-	-	-	DPSEL0	R/W	DPS
093H	-	-	-	-	NDPS	ATMS	ATMD	ATME	R/W	DPC
094H	PECMD7	PECMD6	PECMD5	PECMD4	PECMD3	PECMD2	PECMD1	PECMD0	W	PECMD
095H	PEROM7	PEROM6	PEROM5	-	PECMD11	PECMD10	PECMD9	PECMD8	R/W	PEROML
096H	PEROM15	PEROM14	PEROM13	PEROM12	PEROM11	PEROM10	PEROM9	PEROM8	R/W	PEROMH
097H	PERAM7	PERAM6	PERAM5	PERAM4	PERAM3	PERAM2	PERAM1	PERAM0	R/W	PERAM
098H	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0	R/W	S0CON
099H	S0BUF7	S0BUF6	S0BUF5	S0BUF4	S0BUF3	S0BUF2	S0BUF1	S0BUF0	R/W	S0BUF
09AH	-	-	ECMP1	ECMP0	EADC	EPWM3	EPWM2	-	R/W	IEN2
09BH	PW3CM1	PW3CM0	PW2CM1	PW2CM0	PW1CM1	PW1CM0	OP1EN	OP0EN	R/W	OPM
09CH	CM0EN	-	CM0S1	CM0S0	CM0OEN	CM0OUT	CM0G1	CM0G0	R/W	CMP0M
09DH	CM1EN	-	CM1S1	CM1S0	CM1OEN	CM1OUT	CM1G1	CM1G0	R/W	CMP1M
09EH	P2CON7	P2CON6	P2CON5	P2CON4	-	-	-	-	R/W	P2CON
09FH	P3CON7	P3CON6	P3CON5	P3CON4	P3CON3	P3CON2	P3CON1	P3CON0	R/W	P3CON
0A0H	P27	P26	P25	P24	P23	P22	P21	P20	R/W	P2
0A1H	PW2rate2	PW2rate1	PW2rate0	PWNV21	PWNV20	PWCH21	PWCH20	PW2PO	R/W	PW2M
0A2H	PW2Y7	PW2Y6	PW2Y5	PW2Y4	PW2Y3	PW2Y2	PW2Y1	PW2Y0	R/W	PW2YL
0A3H	PW2Y15	PW2Y14	PW2Y13	PW2Y12	PW2Y11	PW2Y10	PW2Y9	PW2Y8	R/W	PW2YH
0A4H	PW2B7	PW2B6	PW2B5	PW2B4	PW2B3	PW2B2	PW2B1	PW2B0	R/W	PW2BL
0A5H	PW2B15	PW2B14	PW2B13	PW2B12	PW2B11	PW2B10	PW2B9	PW2B8	R/W	PW2BH
0A6H	PW2D7	PW2D6	PW2D5	PW2D4	PW2D3	PW2D2	PW2D1	PW2D0	R/W	PW2DL
0A7H	PW2D15	PW2D14	PW2D13	PW2D12	PW2D11	PW2D10	PW2D9	PW2D8	R/W	PW2DH
0A8H	EAL	-	ET2	ES0	ET1	EX1	ET0	EX0	R/W	IEN0
0A9H	-	-	IP05	IP04	IP03	IP02	IP01	IP00	R/W	IP0
0AAH	S0RELL7	S0RELL6	S0RELL5	S0RELL4	S0RELL3	S0RELL2	S0RELL1	S0RELL0	R/W	S0RELL
0ABH	PW1rate2	PW1rate1	PW1rate0	PWNV11	PWNV10	PWCH11	PWCH10	PW1PO	R/W	PW1M
0ACH	PW1Y7	PW1Y6	PW1Y5	PW1Y4	PW1Y3	PW1Y2	PW1Y1	PW1Y0	R/W	PW1YL
0ADH	PW1Y15	PW1Y14	PW1Y13	PW1Y12	PW1Y11	PW1Y10	PW1Y9	PW1Y8	R/W	PW1YH
0AEH	PW1B7	PW1B6	PW1B5	PW1B4	PW1B3	PW1B2	PW1B1	PW1B0	R/W	PW1BL
0AFH	PW1B15	PW1B14	PW1B13	PW1B12	PW1B11	PW1B10	PW1B9	PW1B8	R/W	PW1BH
0B0H	P37	P36	P35	P34	P33	P32	P31	P30	R/W	P3
0B1H	PW3rate2	PW3rate1	PW3rate0	PWNV31	PWNV30	PWCH31	PWCH30	PW3PO	R/W	PW3M
0B2H	PW3Y7	PW3Y6	PW3Y5	PW3Y4	PW3Y3	PW3Y2	PW3Y1	PW3Y0	R/W	PW3YL
0B3H	PW3Y15	PW3Y14	PW3Y13	PW3Y12	PW3Y11	PW3Y10	PW3Y9	PW3Y8	R/W	PW3YH
0B4H	PW3R7	PW3R6	PW3R5	PW3R4	PW3R3	PW3R2	PW3R1	PW3R0	R/W	PW3BL
0B5H	PW3R15	PW3R14	PW3R13	PW3R12	PW3R11	PW3R10	PW3R9	PW3R8	R/W	PW3BH
0B6H	PW3D7	PW3D6	PW3D5	PW3D4	PW3D3	PW3D2	PW3D1	PW3D0	R/W	PW3DL
0B7H	PW3D15	PW3D14	PW3D13	PW3D12	PW3D11	PW3D10	PW3D9	PW3D8	R/W	PW3DH
0B8H	ET2RL	-	ET2C3	ET2C2	ET2C1	ET2C0	ESPI	EI2C	R/W	IEN1
0B9H	-	-	IP15	IP14	IP13	IP12	IP11	IP10	R/W	IP1
0BAH	S0RELH7	S0RELH6	S0RELH5	S0RELH4	S0RELH3	S0RELH2	S0RELH1	S0RELH0	R/W	S0RELH
0BBH	PW1D7	PW1D6	PW1D5	PW1D4	PW1D3	PW1D2	PW1D1	PW1D0	R/W	PW1DL
0BCH	PW1D15	PW1D14	PW1D13	PW1D12	PW1D11	PW1D10	PW1D9	PW1D8	R/W	PW1DH
0BDH	PW1A7	PW1A6	PW1A5	PW1A4	PW1A3	PW1A2	PW1A1	PW1A0	R/W	PW1A
0BEH	PW2A7	PW2A6	PW2A5	PW2A4	PW2A3	PW2A2	PW2A1	PW2A0	R/W	PW2A
0BFH	-	-	-	CMP1F	CMP0F	ADCF	PWM3F	PWM2F	R/W	IRCON2
0C0H	TF2RL	TF2	TF2C3	TF2C2	TF2C1	TF2C0	-	-	R/W	IRCON
0C1H	COCAH3	COCAL3	COCAH2	COCAL2	COCAH1	COCAL1	COCAH0	COCAL0	R/W	CCEN
0C2H	CCL17	CCL16	CCL15	CCL14	CCL13	CCL12	CCL11	CCL10	R/W	CCL1

0C3H	CCH17	CCH16	CCH15	CCH14	CCH13	CCH12	CCH11	CCH10	R/W	CCH1
0C4H	CCL27	CCL26	CCL25	CCL24	CCL23	CCL22	CCL21	CCL20	R/W	CCL2
0C5H	CCH27	CCH26	CCH25	CCH24	CCH23	CCH22	CCH21	CCH20	R/W	CCH2
0C6H	CCL37	CCL36	CCL35	CCL34	CCL33	CCL32	CCL31	CCL30	R/W	CCL3
0C7H	CCH37	CCH36	CCH35	CCH34	CCH33	CCH32	CCH31	CCH30	R/W	CCH3
0C8H	T2PS	I3FR	GF1	T2R1	T2R0	T2CM	T2I1	T2I0	R/W	T2CON
0C9H	-	-	-	-	-	-	-	-	-	-
0CAH	CRCL7	CRCL6	CRCL5	CRCL4	CRCL3	CRCL2	CRCL1	CRCL0	R/W	CRCL
0CBH	CRCH7	CRCH6	CRCH5	CRCH4	CRCH3	CRCH2	CRCH1	CRCH0	R/W	CRCH
0CCH	TL27	TL26	TL25	TL24	TL23	TL22	TL21	TL20	R/W	TL2
0CDH	TH27	TH26	TH25	TH24	TH23	TH22	TH21	TH20	R/W	TH2
0CEH	-	-	-	-	CM1T1	CM1T0	CM0T1	CM0T0	R/W	CMPT
0CFH	PW3A7	PW3A6	PW3A5	PW3A4	PW3A3	PW3A2	PW3A1	PW3A0	R/W	PW3A
0D0H	CY	AC	F0	RS1	RS0	OV	F1	P	R/W	PSW
0D1H	EPWM1	EX2	-	-	PWM1F	IE2	-	-	R/W	IEN4
0D2H	ADENB	ADS	EOC	CHS4	CHS3	CHS2	CHS1	CHS0	R/W	ADM
0D3H	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	R/W	ADB
0D4H	TCHEN	GCHS	ADCKS1	ADCKS0	ADB3	ADB2	ADB1	ADB0	R/W	ADR
0D5H	EVHENB	EVLENB	-	ADPWS	-	VHS2	VHS1	VHS0	R/W	VREFH
0D6H	P4CON7	P4CON6	P4CON5	P4CON4	P4CON3	P4CON2	P4CON1	P4CON0	R/W	P4CON
0D7H	-	-	P5CON5	P5CON4	P5CON3	P5CON2	P5CON1	P5CON0	R/W	P5CON
0D8H	BD	-	-	-	-	-	-	-	R/W	S0CON2
0D9H	-	-	-	-	-	-	-	-	-	-
0DAH	I2CDAT7	I2CDAT6	I2CDAT5	I2CDAT4	I2CDAT3	I2CDAT2	I2CDAT1	I2CDAT0	R/W	I2CDAT
0DBH	I2CADR7	I2CADR6	I2CADR5	I2CADR4	I2CADR3	I2CADR2	I2CADR1	GC	W	I2CADR
0DCH	CR2	ENS1	STA	STO	SI	AA	CR1	CR0	R/W	I2CCON
0DDH	I2CSTA7	I2CSTA6	I2CSTA5	I2CSTA4	I2CSTA3	-	-	-	R	I2CSTA
0DEH	SMBEXE	-	-	-	-	SMBTOP2	SMBTOP1	SMBTOP0	R/W	SMBSEL
0DFH	SMBDST7	SMBDST6	SMBDST5	SMBDST4	SMBDST3	SMBDST2	SMBDST1	SMBDST0	R/W	SMBDST
0E0H	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0	R/W	ACC
0E1H	SPIF	WCOL	SSERR	MODF	-	-	-	-	R	SPSTA
0E2H	SPR2	SPEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0	R/W	SPCON
0E3H	SPDAT7	SPDAT6	SPDAT5	SPDAT4	SPDAT3	SPDAT2	SPDAT1	SPDAT0	R/W	SPDAT
0E4H	PW3EN	PW2EN	PW1EN	P06OC	P05OC	P13OC	P12OC	P11OC	R/W	P1OC
0E5H	-	-	-	-	-	CLKRATE2	CLKRATE1	CLKRATE0	R/W	CLKSEL
0E6H	CLKCMD7	CLKCMD6	CLKCMD5	CLKCMD4	CLKCMD3	CLKCMD2	CLKCMD1	CLKCMD0	W	CLKCMD
0E7H	T0EXT	T0RATE2	T0RATE1	T0RATE0	-	T1RATE2	T1RATE1	T1RATE0	R/W	TCON0
0E8H	P47	P46	P45	P44	P43	P42	P41	P40	R/W	P4
0E9H	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00	R/W	MD0
0EAH	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10	R/W	MD1
0EBH	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20	R/W	MD2
0ECH	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30	R/W	MD3
0EDH	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40	R/W	MD4
0EEH	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50	R/W	MD5
0EFH	MDEF	MDOV	SLR	SC4	SC3	SC2	SC1	SC0	R/W	ARCON
0F0H	B7	B6	B5	B4	B3	B2	B1	B0	R/W	B
0F1H	P07R	P06R	P05R	P04R	P03R	P02R	P01R	P00R	R/W	P0UR
0F2H	P17R	P16R	P15R	P14R	P13R	P12R	P11R	P10R	R/W	P1UR
0F3H	P27R	P26R	P25R	P24R	P23R	P22R	P21R	P20R	R/W	P2UR
0F4H	P37R	P36R	P35R	P34R	P33R	P32R	P31R	P30R	R/W	P3UR
0F5H	P47R	P46R	P45R	P44R	P43R	P42R	P41R	P40R	R/W	P4UR
0F6H	-	-	P55R	P54R	P53R	P52R	P51R	P50R	R/W	P5UR
0F7H	-	-	-	-	-	-	-	SRSTREQ	R/W	SRST
0F8H	-	-	P55	P54	P53	P52	P51	P50	R/W	P5
0F9H	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M	R/W	P0M
0FAH	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M	R/W	P1M
0FBH	P27M	P26M	P25M	P24M	P23M	P22M	P21M	P20M	R/W	P2M
0FCH	P37M	P36M	P35M	P34M	P33M	P32M	P31M	P30M	R/W	P3M
0FDH	P47M	P46M	P45M	P44M	P43M	P42M	P41M	P40M	R/W	P4M
0FEH	-	-	P55M	P54M	P53M	P52M	P51M	P50M	R/W	P5M
0FFH	POR	WDT	RST	-	-	LVD24	LVD33	-	R	PFLAG

2.2.3 ACCUMULATOR

The ACC is an 8-bit data register responsible for transferring or manipulating data between ALU and data memory. If the result of operating is overflow (OV) or there is carry (C or AC) and parity (P) occurrence, then these flags will be set to PSW register.

➢ **Example: Read and write ACC value.**

```
; Read ACC data and store in BUF data memory

                    MOV         BUF, A

; Write a immediate data into ACC

                    MOV         A, #0FH

; Write ACC data from BUF data memory

                    MOV         A, BUF
```

2.2.4 B REGISTER

The B register is the 8-bit buffer. There are two major functions of this register.

- Can be used as general working registers
- Can be used during multiplying and division instructions.

0F0H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B	B7	B6	B5	B4	B3	B2	B1	B0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After Reset	0	0	0	0	0	0	0	0

Bit [7:0] **B:** The B register is used during multiplying and division instructions. It can also be used as a scratch-pad register to hold temporary data.

2.2.5 PROGRAM FLAG

The PFLAG register contains the arithmetic status of ALU operation, system reset status and LVD detecting status. POR, WDT, and RST bits indicate system reset status including power on reset, LVD reset, reset by external pin active and watchdog reset. LVD24, LVD33 bits indicate LVD detecting power voltage status.

0FFH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFLAG	POR	WDT	RST	-	-	LVD24	LVD33	
Read/Write	R	R	R	-	-	R	R	-
After Reset	-	-	-	-	-	0	0	-

Bit 7 **POR:** Power on reset and LVD brown-out reset indicator.
0 = Non-active.
1 = Reset active. LVD announces reset flag.

Bit 6 **WDT:** Watchdog reset indicator.
0 = Non-active.
1 = Reset active. Watchdog announces reset flag.

Bit 5 **RST:** External reset indicator.
0 = Non-active.
1 = Reset active. External reset announces reset flag.

Bit 2 **LVD24:** LVD 24 indicator.
0 = Vdd > 2.4V.

1 = Vdd ≦ 2.4V.

Bit 1 **LVD33:** LVD 33 indicator.
0 = Vdd > 3.3V.
1 = Vdd ≦ 3.3V.

2.2.6 SYSTEM FLAG REGISTER

The system flag register contains the arithmetic status of ALU operation. CY, AC, OV, P bits indicate the result status of ALU operation. RS[1:0] is register bank select control bit. F0 and F1 are general purpose flag for user.

0D0H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PSW	CY	AC	F0	RS1	RS0	OV	F1	P
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
After Reset	0	0	0	0	0	0	0	0

Bit 7 **CY:** Carry flag.
1 = Addition with carry, subtraction without borrowing, rotation with shifting out logic “1”, comparison result ≥ 0.
0 = Addition without carry, subtraction with borrowing signal, rotation with shifting out logic “0”, comparison result < 0.

Bit 6 **AC:** Auxiliary carry flag.
1 = If there is a carry-out from 3rd bit of Accumulator in BCD operations.
0 = If there is no a carry-out from 3rd bit of Accumulator in BCD operations.

Bit 5 **F0:** General purpose flag 0. General purpose flag available for user.

Bit [4:3] **RS[1:0]:** Register bank select control bit, used to select working register bank.

RS1	RS0	Select Register Bank	Location
0	0	Bank0	(00H – 07H)
0	1	Bank1	(08H – 0FH)
1	0	Bank2	(10H – 17H)
1	1	Bank3	(18H – 1FH)

Bit 2 **OV:** Overflow flag.
0 = Non-overflow in Accumulator during arithmetic Operations.
1 = overflow in Accumulator during arithmetic Operations.

Bit 1 **F1:** General purpose flag 1. General purpose flag available for user.

Bit 0 **P:** Parity flag. Reflects the number of ‘1’s in the Accumulator.
1 = Accumulator contains an odd number of ‘1’s.
0 = if Accumulator contains an even number of ‘1’s.

2.2.7 SOFTWARE RESET FLAG

The software reset control by SRSTREQ bit. In the writing condition, both write “0” to SRSTREQ or single write “1” to SRSTREQ will have no effect. When write “1” twice to SRSTREQ, system will generate an internal software reset. In the reading condition, SRSTREQ bit contain the information of reset source. If SRSTREQ=0, the source of last reset sequence was not a software reset (hardware, watchdog or debugger reset). If SRSTREQ=1, the source of last reset sequence was a software reset (caused by double writing ‘1’ value to the “SRSTREQ” bit)

0F7H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SRST	-	-	-	-	-	-	-	SRSTREQ
Read/Write	-	-	-	-	-	-	-	R/W
After Reset	-	-	-	-	-	-	-	0

Bit 0 **SRSTREQ:** Software reset request.
Write:
0 = No effect.
1 = Internal software reset. (Double writing “1”)
Read:
0 = Source of last reset sequence was not a software reset.
1 = Source of last reset sequence was a software reset.

2.2.8 SYSTEM CONTROL REGISTER

The contents of CKCON register defines the number of internally generated wait states that occur during read/write accesses to external data and program memory. It also controls the type of write access to either of the memory spaces.

08EH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CKCON	PSYN	PWSC2	PWSC1	PWSC0	ESYN	EWSC2	EWSC1	EWSC0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After Reset	0	1	1	1	0	0	0	1

Bit 7 **PSYN:** Program memory synchronous/asynchronous write control.
0 = “Synchronous” type write access is performed.
1 = “Asynchronous” type write access is performed.

Bit [6:4] **PWSC[2:0]:** Program memory wait state control.

CKCON register PWSC[2:0]	Program memory wait cycle
000	1
001	2
010	3
011	4
100	5
101	6
110	7
111	8

Bit 3 **ESYN:** External data memory synchronous/asynchronous write control
0 = “Synchronous” type write access is performed.
1 = “Asynchronous” type write access is performed.

Bit [2:0] **EWSC[2:0]:** External data memory stretch cycle control.

CKCON register EWSC[2:0]	External RAM read wait cycle	External RAM write wait cycle	
		ESYN = 0	ESYN = 1
000	1	1	3
001	2	2	4
010	3	3	5
011	4	4	6
100	5	5	7

101	6	6	8
110	7	7	9
111	8	8	10

087H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	SMOD				P2SEL	GF0	STOP	IDLE
Read/Write	R/W	-	-	-	R/W	R/W	R/W	R/W
After Reset	0	-	-	-	1	0	0	0

Bit 7 **SMOD:** UART baud rate select (baud rate doubler).

Bit 3 **P2SEL:** High-order address byte configuration bit. Chooses the higher byte of address (“MEMADDR[15:8]”) during MOVX @Ri operations
0 = The “MEMADDR [15:8]” = “P2REG”. The “P2REG” is the contents of Port2 output register.
1 = The “MEMADDR [15:8]” = 0X00.

Bit 2 **GF0:** General Purpose Flag.

Bit 1 **STOP:** Stop mode control.
Setting this bit activates the Stop Mode. This bit is always read as 0.

Bit 0 **IDLE:** Idle mode control
Setting this bit activates the Idle Mode. This bit is always read as 0.

2.2.9 DATA POINTER REGISTERS

Data Pointer Registers can be accessed through DPL and DPH. The actual Data Pointer is selected by DPSEL register. These registers are intended to hold 16-bit address in the indirect addressing mode used by MOVX (move external memory), MOVC (move program memory) or JMP (computed branch) instructions. They may be manipulated as 16-bit register or as two separate 8-bit registers. DPH holds higher byte and DPL holds lower byte of indirect address. It is generally used to access external code or data space (e.g. MOVC A,@A+DPTR or MOV A,@DPTR respectively).

The Data Pointer 1 Register can be accessed through DPL1 and DPH1. Those SFR locations refer always to the DPTR1, regardless of the actual Data Pointer selection by DPS register. This 16-bit register is used by all DPTR-related instructions when the LSB of the DPS register is set to 1, otherwise the DPTR is taken from DPH and DPL.

Data pointer register select by DPSEL0 bit. When DPSEL0=0, Data Pointer Register (DPH/DPL) is selected. When DPSEL0=1, Data Pointer 1 Register (DPH1/DPL1) is selected.

SN8F5708 contains an optional DPTR-related arithmetic unit. It provides auto-increment / auto-decrement by 1 or 2, and auto-switching between active DPTRs. Those functions are controlled by the “DPC” register. There are separate “DPC” register for each DPTR, to provide high flexibility in data transfers. The “DPC” SFR address 0x93 points to the “WINDOW” where the actual “DPC” is selected using the “DPS” register, same as for the DPTR.

082H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPL	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After Reset	0	0	0	0	0	0	0	0

083H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPH	DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After Reset	0	0	0	0	0	0	0	0

Data Pointer Register

084H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPL1	DPL17	DPL16	DPL15	DPL14	DPL13	DPL12	DPL11	DPL10
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After Reset	0	0	0	0	0	0	0	0

085H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPH1	DPH17	DPH16	DPH15	DPH14	DPH13	DPH12	DPH11	DPH10
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After Reset	0	0	0	0	0	0	0	0

The Data Pointer 1 Register

092H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPS	-	-	-	-	-	-	-	DPSEL0
Read/Write	-	-	-	-	-	-	-	R/W
After Reset	-	-	-	-	-	-	-	0

Bit 0 **DPSEL0:** Data Pointer Register select. The contents of this field specifies the number of active data pointer register.

0 = Data Pointer Register (DPH/DPL) is selected.

1 = Data Pointer 1 Register (DPH1/DPL1) is selected.

093H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPC	-	-	-	-	NDPS	ATMS	ATMD	ATME
Read/Write	-	-	-	-	R/W	R/W	R/W	R/W
After Reset	-	-	-	-	0	0	0	0

Bit 3 **NDPS:** Next Data Pointer Selection. The contents of this field is loaded to the “DPS” register after each

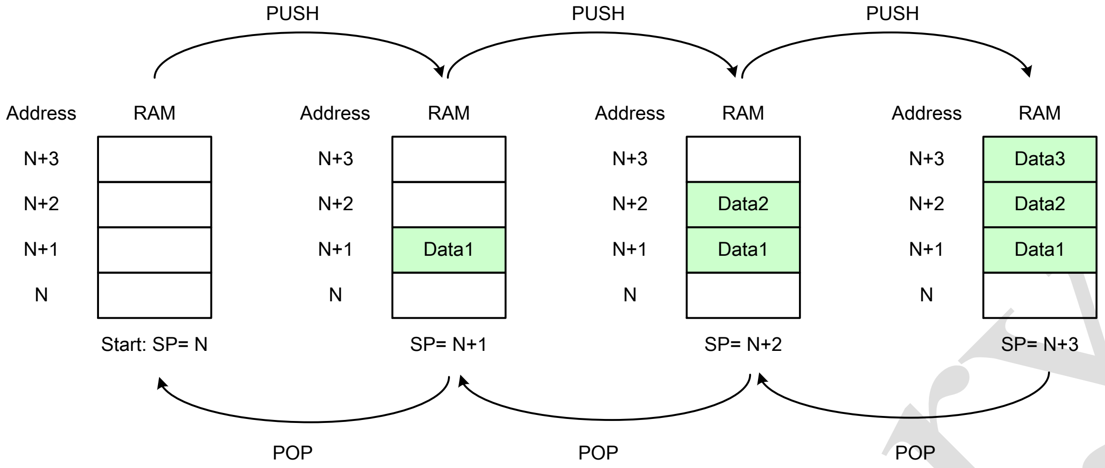
MOVX @DPTR instruction. Note that this feature is always enabled, therefore for each of the “DPS” registers this field has to contain a different value pointing to itself so that the auto-switching does not occur with default (reset) values.

Bit 2 **ATMS:** Auto-modification size.
When zero, the current DPTR is automatically modified by 1 after each MOVX @DPTR instruction when DPC.0=1.
When one, the current DPTR is automatically modified by 2 after each MOVX @DPTR instruction when DPC.0=1.

Bit 1 **ATMD:** Auto-modification direction.
When zero, the current DPTR is automatically incremented after each MOVX @DPTR instruction when DPC.0=1.
When one, the current DPTR is automatically decremented after each MOVX @DPTR instruction when DPC.0=1.

Bit 0 **ATME:** Auto-modification enable.
When set, enables auto-modification of the current DPTR after each MOVX @DPTR instruction.

2.3 STACK OPERATION



- The stack is a section of RAM used by the CPU to store information temporarily. This information could be data or an address.
- The register used to access the stack is called the SP (stack pointer) register. Stack pointer (SP) is an 8-bit register at address 81H.
- It contains the address of the data item currently on top of the stack.
- Stack operations include pushing data on the stack and popping data off the stack.
- Pushing increments SP before writing the data.
- Popping from the stack reads the data and decrements the SP.
- Depending on the initial value of the SP, stack can have different sizes. (E.g., MOV SP, #60H).
- The default value of SP (after system reset) is 07H.
- This result in the first stack write operation to store data in location 08H which means that register bank 1 (and possible 2 and 3) are not available. User may initialize the SP to avoid this.
- The interrupt and PUSH/POP/CALL/RET(I) functions are the trigger source of stack operation.

2.3.1 STACK POINTER REGISTER

081H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SP	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After Reset	0	0	0	0	0	1	1	1

Bit [7:0] **SP[7:0]:** This register points to the top of stack in internal data memory space. It is used to store the return address of program before executing interrupt routine or subprograms. The SP is incremented before executing PUSH or CALL instruction and it is decremented after executing POP or RET(I) instruction (it always points the top of stack).

2.3.2 PUSH/POP OPERATING

- “PUSH” operation: PUSH operation saves directly addressed data onto into stack buffer (RAM). (e.g., PUSH ACC; PUSH 5AH).

- As system push data onto the stack, the stack pointer (SP) is incremented by one. When PUSH is executed, the contents of the register are saved on the stack and SP is incremented by 1.

- “POP” operation: POP operation reloads directly addressed data from stack buffer (RAM). (e.g., POP ACC; POP 5AH).

Popping the contents of the stack back into a given register is the opposite process of pushing. With every pop, the top byte of the stack is copied to the register specified by the instruction and the stack pointer is decremented once.

2.3.3 INTERRUPT AND CALL OPERATING

- When CALL subroutine and exit subroutine after executing RET instruction, stack operation executes, too.

- When system jumps to interrupt vector and return to main routine after executing RETI instruction, stack operation executes.

	Interrupt N	LCALL addr16	ACALL addr11	RET/RETI
Stack	Pushing	Pushing	Pushing	Popping

Operation	SP = SP + 1
	(SP) = PC[7-0]
	SP = SP + 1
	(SP) = PC[15-8]
	PC = INT_VEC_0xN
		PC = PC + 3
		SP = SP + 1
		(SP) = PC[7-0]
		SP = SP + 1
		(SP) = PC[15-8]
		PC = addr16
			PC = PC + 2
			SP = SP + 1
			(SP) = PC[7-0]
			SP = SP + 1
			(SP) = PC[15-8]
			PC10-0 = addr11
				PC[15-8] = (SP)
				SP = SP - 1
				PC[7-0] = (SP)
				SP = SP – 1

Final SP	SP+2	SP+2	SP+2	SP-2
Final PC	INT_VEC_0xN	addr16	Addr11	(SP), (SP-1)

2.4 CODE OPTION TABLE

The code option is the system hardware configurations including oscillator type, noise filter option, watchdog timer operation, LVD option, reset pin option and Flash ROM security control. The code option items are as following table:

Code Option	Content	Function Description
High_Clk	IHRC_32M	High speed internal 32MHz RC. XIN/XOUT pins are bi-direction GPIO mode.
	IHRC_32M_RTC	High speed internal 32MHz RC. XIN/XOUT pins are connected to external 32768Hz crystal.
	Ext_CLK	XIN pin connect external clock (1M ~32M), XOUT pin is bi-direction GPIO mode.
	12M X'tal	High speed crystal /resonator (e.g. 12MHz) for external high clock oscillator.
	4M X'tal	Standard crystal /resonator (e.g. 4M) for external high clock oscillator.
Noise_Filter	Enable	Enable Noise Filter.
	Disable	Disable Noise Filter.
WDT_CLK	Flosc/4	Watchdog timer clock source Flosc/4.
	Flosc/8	Watchdog timer clock source Flosc/8.
	Flosc/16	Watchdog timer clock source Flosc/16.
	Flosc/32	Watchdog timer clock source Flosc/32.
Watch_Dog	Always_On	Watchdog timer is always on enable even in power down and green mode.
	Enable	Enable watchdog timer. Watchdog timer stops in power down mode and green mode.
	Disable	Disable Watchdog function.
Reset_Pin	Reset	Enable External reset pin.
	P02	Enable P0.2.
Security	Enable	Enable ROM code Security function.
	Disable	Disable ROM code Security function.
LVD	LVD_L	LVD will reset chip if VDD is below 1.8V
	LVD_M	LVD will reset chip if VDD is below 1.8V Enable LVD24 bit of PFLAG register for 2.4V low voltage indicator.
	LVD_H	LVD will reset chip if VDD is below 2.4V Enable LVD33 bit of PFLAG register for 3.3V low voltage indicator.
	LVD_MAX	LVD will reset chip if VDD is below 3.3V

2.4.1 Reset_Pin code option

The reset pin is shared with general input only pin controlled by code option.

- **Reset:** The reset pin is external reset function. When falling edge trigger occurring, the system will be reset.
- **P02:** Set reset pin to general bi-direction pin (P0.2). The external reset function is disabled and the pin is bi-direction pin.

2.4.2 Security code option

Security code option is Flash ROM protection. When enable security code option, the ROM code is secured and not dumped complete ROM contents.

2.4.3 Noise Filter code option

Noise Filter code option is a power noise filter manner to reduce noisy effect of system clock. If noise filter enable, in high noisy environment, enable noise filter, enable watchdog timer and select a good LVD level can make whole system work well and avoid error event occurrence.

3 RESET

3.1 OVERVIEW

The system would be reset in three conditions as following.

- Power on reset
- Watchdog reset
- Brown out reset
- External reset (only supports external reset pin enable situation)

When any reset condition occurs, all system registers keep initial status, program stops and program counter is cleared. After reset status released, the system boots up and program starts to execute from ORG 0. The POR, WDT and RST flags indicate system reset status. The system can depend on POR, WDT and RST status and go to different paths by program.

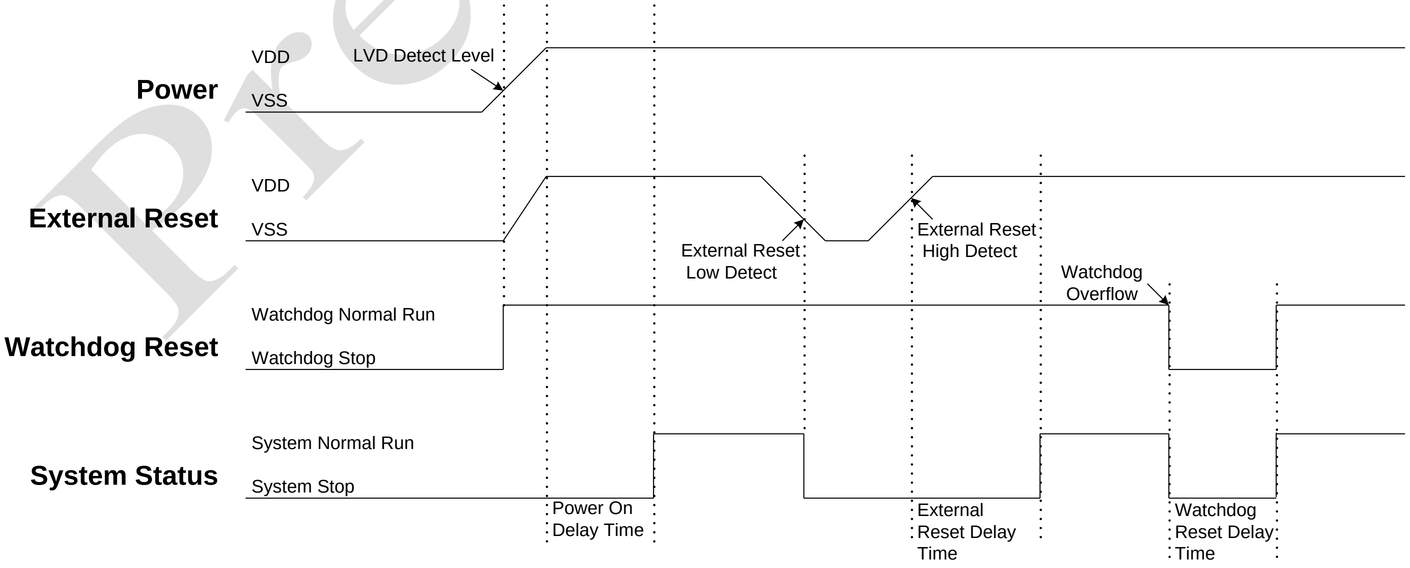
0FFH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFLAG	POR	WDT	RST	-	-	LVD24	LVD33	-
Read/Write	R	R	R	-	-	R	R	-
After reset	-	-	-	-	-	0	0	-

Bit 7 **POR:** Power on reset and LVD brown-out reset indicator.
0 = Non-active.
1 = Reset active. LVD announces reset flag.

Bit 6 **WDT:** Watchdog reset indicator.
0 = Non-active.
1 = Reset active. Watchdog announces reset flag.

Bit 5 **RST:** External reset indicator.
0 = Non-active.
1 = Reset active. External reset announces reset flag.

Finishing any reset sequence needs some time. The system provides complete procedures to make the power on reset successful. For different oscillator types, the reset time is different. That causes the VDD rise rate and start-up time of different oscillator is not fixed. RC type oscillator's start-up time is very short, but the crystal type is longer. Under client terminal application, users have to take care the power on reset time for the master terminal requirement. The reset timing diagram is as following.



3.2 POWER ON RESET

The power on reset depend no LVD operation for most power-up situations. The power supplying to system is a rising curve and needs some time to achieve the normal voltage. Power on reset sequence is as following.

- **Power-up:** System detects the power voltage up and waits for power stable.
- **External reset (only external reset pin enable):** System checks external reset pin status. If external reset pin is not high level, the system keeps reset status and waits external reset pin released.
- **System initialization:** All system registers is set as initial conditions and system is ready.
- **Oscillator warm up:** Oscillator operation is successfully and supply to system clock.
- **Program executing:** Power on sequence is finished and program executes from ORG 0.

3.3 WATCHDOG RESET

Watchdog reset is a system protection. In normal condition, system works well and clears watchdog timer by program. Under error condition, system is in unknown situation and watchdog can't be clear by program before watchdog timer overflow. Watchdog timer overflow occurs and the system is reset. After watchdog reset, the system restarts and returns normal mode. Watchdog reset sequence is as following.

- **Watchdog timer status:** System checks watchdog timer overflow status. If watchdog timer overflow occurs, the system is reset.
- **System initialization:** All system registers is set as initial conditions and system is ready.
- **Oscillator warm up:** Oscillator operation is successfully and supply to system clock.
- **Program executing:** Power on sequence is finished and program executes from ORG 0.

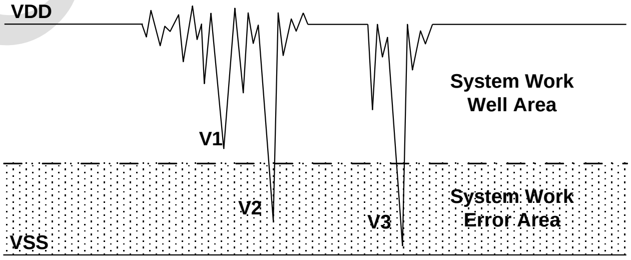
Watchdog timer application note is as following.

- Before clearing watchdog timer, check I/O status and check RAM contents can improve system error.
- Don't clear watchdog timer in interrupt vector and interrupt service routine. That can improve main routine fail.
- Clearing watchdog timer program is only at one part of the program. This way is the best structure to enhance the watchdog timer function.

✷ ***Note: Please refer to the "WATCHDOG TIMER" about watchdog timer detail information.***

3.4 BROWN OUT RESET

The brown out reset is a power dropping condition. The power drops from normal voltage to low voltage by external factors (e.g. EFT interference or external loading changed). The brown out reset would make the system not work well or executing program error.



Brown Out Reset Diagram

The power dropping might through the voltage range that's the system dead-band. The dead-band means the power range can't offer the system minimum operation power requirement. The above diagram is a typical brown out reset diagram. There is a serious noise under the VDD, and VDD voltage drops very deep. There is a dotted line to separate the system working area. The above area is the system work well area. The below area is the system work error area called dead-band. V1 doesn't touch the below area and not effect the system operation. But the V2 and V3 is under the below area and may induce the system error occurrence. Let system under dead-band includes some conditions.

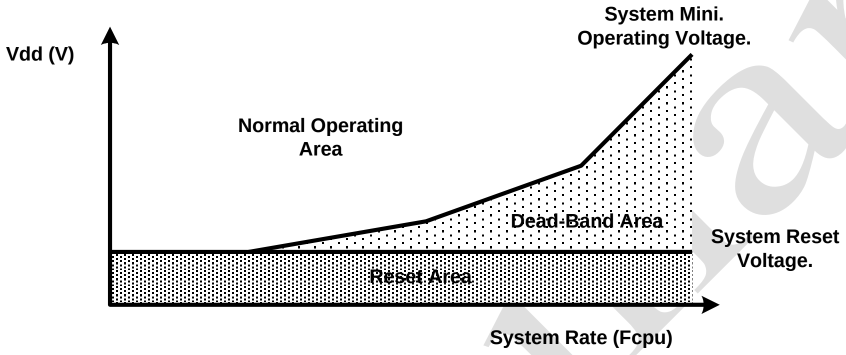
DC application:

The power source of DC application is usually using battery. When low battery condition and MCU drive any loading, the power drops and keeps in dead-band. Under the situation, the power won't drop deeper and not touch the system reset voltage. That makes the system under dead-band.

AC application:
In AC power application, the DC power is regulated from AC power source. This kind of power usually couples with AC noise that makes the DC power dirty. Or the external loading is very heavy, e.g. driving motor. The loading operating induces noise and overlaps with the DC power. VDD drops by the noise, and the system works under unstable power situation.
The power on duration and power down duration are longer in AC application. The system power on sequence protects the power on successful, but the power down situation is like DC low battery condition. When turn off the AC power, the VDD drops slowly and through the dead-band for a while.

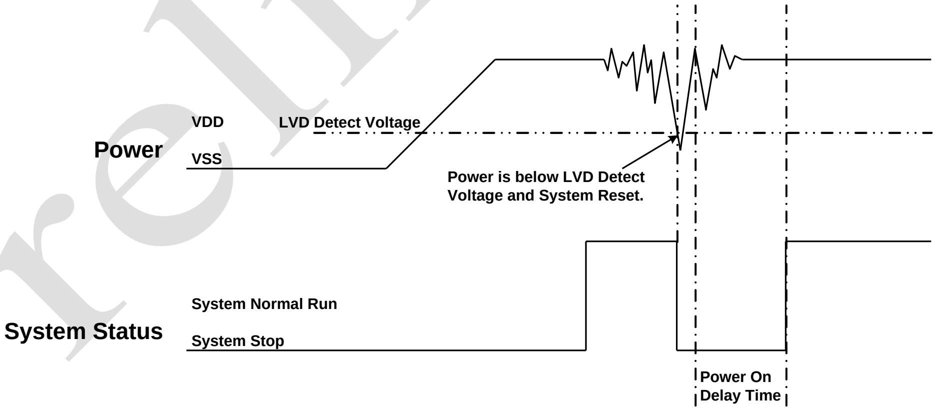
3.4.1 THE SYSTEM OPERATING VOLTAGE

To improve the brown out reset needs to know the system minimum operating voltage which is depend on the system executing rate and power level. Different system executing rates have different system minimum operating voltage. The electrical characteristic section shows the system voltage to executing rate relationship.



Normally the system operation voltage area is higher than the system reset voltage to VDD, and the reset voltage is decided by LVD detect level. The system minimum operating voltage rises when the system executing rate upper even higher than system reset voltage. The dead-band definition is the system minimum operating voltage above the system reset voltage.

3.4.2 LOW VOLTAGE DETECTOR (LVD)



The LVD (low voltage detector) is built-in Sonix 8-bit MCU to be brown out reset protection. When the VDD drops and is below LVD detect voltage, the LVD would be triggered, and the system is reset. The LVD detect level is different by each MCU. The LVD voltage level is a point of voltage and not easy to cover all dead-band range. Using LVD to improve brown out reset is depend on application requirement and environment. If the power variation is very deep, violent and trigger the LVD, the LVD can be the protection. If the power variation can touch the LVD detect level and make system work error, the LVD can't be the protection and need to other reset methods. More detail LVD information is in the electrical characteristic section.

The LVD is three levels design (1.8V/2.4V/3.3V) and controlled by LVD code option. The 1.8V LVD is always enable for power on reset and Brown Out reset. The 2.4V LVD includes LVD reset function and flag function to indicate VDD status function. The 3.3V includes flag function to indicate VDD status. LVD flag function can be an **easy low battery detector**. LVD24, LVD33 flags indicate VDD voltage level. For low battery detect application, only checking LVD24, LVD33 status to be battery status. This is a cheap and easy solution.

0FFH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFLAG	POR	WDT	RST	-	-	LVD24	LVD33	-
Read/Write	R	R	R	-	-	R	R	-
After reset	-	-	-	-	-	0	0	-

Bit 2 **LVD24:** LVD24 low voltage detect indicator.
0 = Vdd > LVD24 detect level.
1 = Vdd < LVD24 detect level.

Bit 1 **LVD33:** LVD33 low voltage detect indicator.
0 = Vdd > LVD33 detect level.
1 = Vdd < LVD33 detect level.

LVD	**LVD Code Option**		
	LVD_L	**LVD_M**	**LVD_H**
1.8V Reset	Available	Available	Available
2.4V Flag	-	Available	-
2.4V Reset	-	-	Available
3.3V Flag	-	-	Available

LVD_L
If VDD < 1.8V, system will be reset.
Disable LVD24 and LVD33 bit of PFLAG register.
LVD_M
If VDD < 1.8V, system will be reset.
Enable LVD24 bit of PFLAG register. If VDD > 2.4V, LVD24 is “0”. If VDD <= 2.4V, LVD24 flag is “1”.
Disable LVD33 bit of PFLAG register.
LVD_H
If VDD < 2.4V, system will be reset.
Enable LVD24 bit of PFLAG register. If VDD > 2.4V, LVD24 is “0”. If VDD <= 2.4V, LVD24 flag is “1”.
Enable LVD33 bit of PFLAG register. If VDD > 3.3V, LVD33 is “0”. If VDD <= 3.3V, LVD33 flag is “1”.
LVD_MAX
If VDD < 3.3V, system will be reset.

✷ ***Note:***
1. ***After any LVD reset, LVD24, LVD33 flags are cleared.***
2. ***The voltage level of LVD 2.4V or 3.3V is for design reference only. Don't use the LVD indicator as precision VDD measurement.***

3.4.3 BROWN OUT RESET IMPROVEMENT

How to improve the brown reset condition? There are some methods to improve brown out reset as following.

- **LVD reset**
- **Watchdog reset**
- **Reduce the system executing rate**
- **External reset circuit. (Zener diode reset circuit, Voltage bias reset circuit, External reset IC)**

> ✸ ***Note:***
>
> 1. ***The "Zener diode reset circuit", "Voltage bias reset circuit" and "External reset IC" can completely improve the brown out reset, DC low battery and AC slow power down conditions.***
> 2. ***For AC power application and enhance EFT performance, the system clock is 4MHz/4 (1 mips) and use external reset (" Zener diode reset circuit", "Voltage bias reset circuit", "External reset IC"). The structure can improve noise effective and get good EFT characteristic.***

Watchdog reset:
The watchdog timer is a protection to make sure the system executes well. Normally the watchdog timer would be clear at one point of program. Don't clear the watchdog timer in several addresses. The system executes normally and the watchdog won't reset system. When the system is under dead-band and the execution error, the watchdog timer can't be clear by program. The watchdog is continuously counting until overflow occurrence. The overflow signal of watchdog timer triggers the system to reset, and the system return to normal mode after reset sequence. This method also can improve brown out reset condition and make sure the system to return normal mode.
If the system reset by watchdog and the power is still in dead-band, the system reset sequence won't be successful and the system stays in reset status until the power return to normal range. Watchdog timer application note is as following.

Reduce the system executing rate:
If the system rate is fast and the dead-band exists, to reduce the system executing rate can improve the dead-band. The lower system rate is with lower minimum operating voltage. Select the power voltage that's no dead-band issue and find out the mapping system rate. Adjust the system rate to the value and the system exits the dead-band issue. This way needs to modify whole program timing to fit the application requirement.

External reset circuit:
The external reset methods also can improve brown out reset and is the complete solution. There are three external reset circuits to improve brown out reset including "Zener diode reset circuit", "Voltage bias reset circuit" and "External reset IC". These three reset structures use external reset signal and control to make sure the MCU be reset under power dropping and under dead-band. The external reset information is described in the next section.

3.5 EXTERNAL RESET

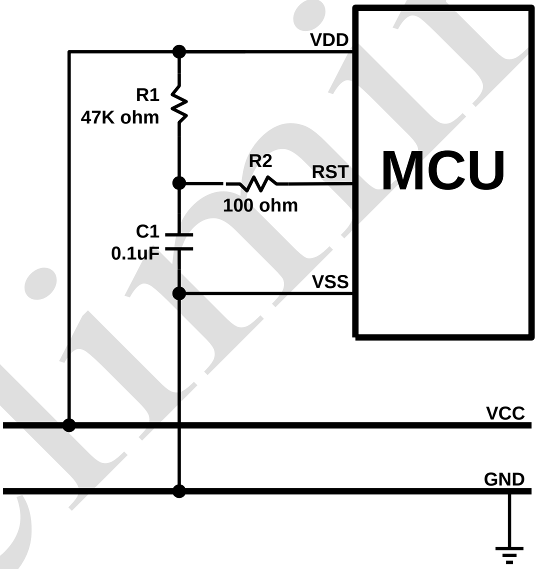
External reset function is controlled by "Reset_Pin" code option. Set the code option as "Reset" option to enable external reset function. External reset pin is Schmitt Trigger structure and low level active. The system is running when reset pin is high level voltage input. The reset pin receives the low voltage and the system is reset. The external reset operation actives in power on and normal running mode. During system power-up, the external reset pin must be high level input, or the system keeps in reset status. External reset sequence is as following.

- **External reset (only external reset pin enable):** System checks external reset pin status. If external reset pin is not high level, the system keeps reset status and waits external reset pin released.
- **System initialization:** All system registers is set as initial conditions and system is ready.
- **Oscillator warm up:** Oscillator operation is successfully and supply to system clock.
- **Program executing:** Power on sequence is finished and program executes from ORG 0.

The external reset can reset the system during power on duration, and good external reset circuit can protect the system to avoid working at unusual power condition, e.g. brown out reset in AC power application…

3.6 EXTERNAL RESET CIRCUIT

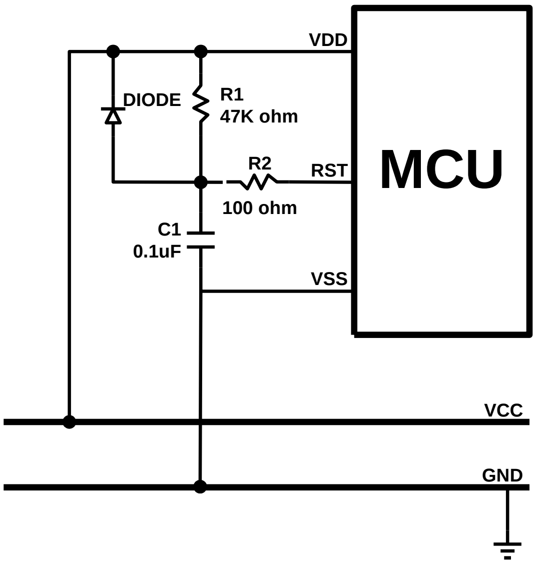
3.6.1 Simply RC Reset Circuit



This is the basic reset circuit, and only includes R1 and C1. The RC circuit operation makes a slow rising signal into reset pin as power up. The reset signal is slower than VDD power up timing, and system occurs a power on signal from the timing difference.

> ✷ ***Note: The reset circuit is no any protection against unusual power or brown out reset.***

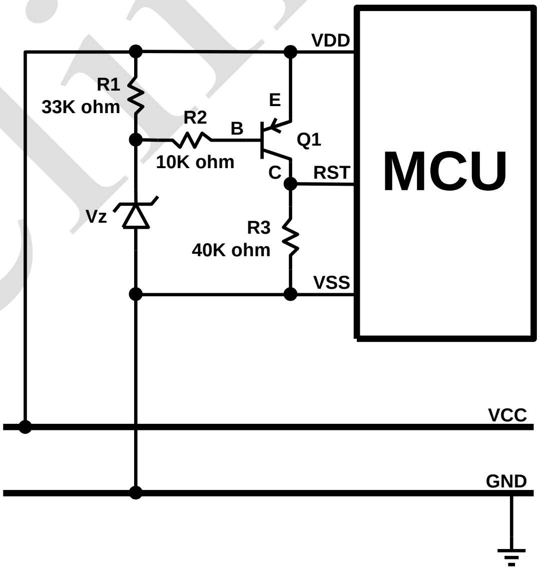
3.6.2 Diode & RC Reset Circuit



This is the better reset circuit. The R1 and C1 circuit operation is like the simply reset circuit to make a power on signal. The reset circuit has a simply protection against unusual power. The diode offers a power positive path to conduct higher power to VDD. It is can make reset pin voltage level to synchronize with VDD voltage. The structure can improve slight brown out reset condition.

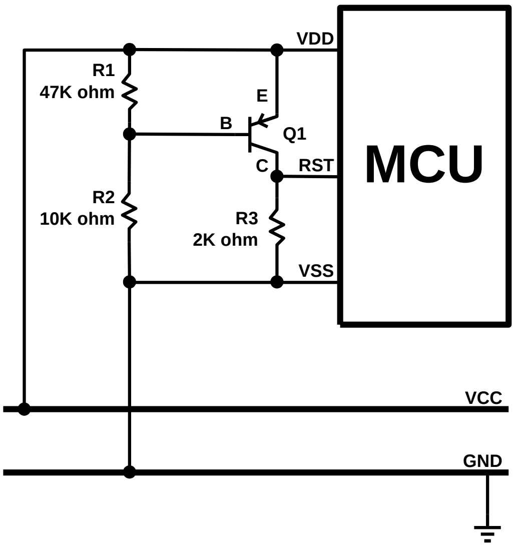
- ***Note: The R2 100 ohm resistor of "Simply reset circuit" and "Diode & RC reset circuit" is necessary to limit any current flowing into reset pin from external capacitor C in the event of reset pin breakdown due to Electrostatic Discharge (ESD) or Electrical Over-stress (EOS).***

3.6.3 Zener Diode Reset Circuit



The zener diode reset circuit is a simple low voltage detector and can **improve brown out reset condition completely**. Use zener voltage to be the active level. When VDD voltage level is above "Vz + 0.7V", the C terminal of the PNP transistor outputs high voltage and MCU operates normally. When VDD is below "Vz + 0.7V", the C terminal of the PNP transistor outputs low voltage and MCU is in reset mode. Decide the reset detect voltage by zener specification. Select the right zener voltage to conform the application.

3.6.4 Voltage Bias Reset Circuit

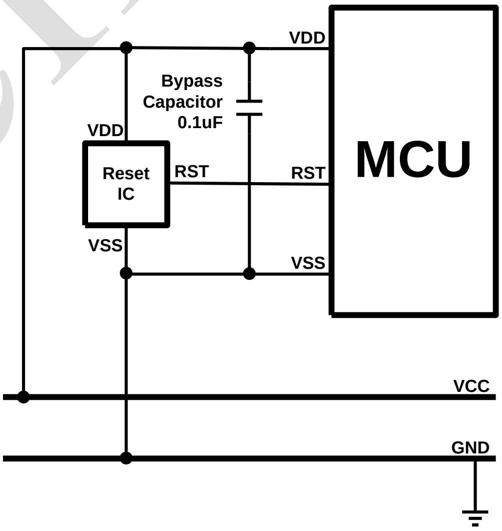


The voltage bias reset circuit is a low cost voltage detector and can **improve brown out reset condition completely**. The operating voltage is not accurate as zener diode reset circuit. Use R1, R2 bias voltage to be the active level. When VDD voltage level is above or equal to “0.7V x (R1 + R2) / R1”, the C terminal of the PNP transistor outputs high voltage and MCU operates normally. When VDD is below “0.7V x (R1 + R2) / R1”, the C terminal of the PNP transistor outputs low voltage and MCU is in reset mode.

Decide the reset detect voltage by R1, R2 resistances. Select the right R1, R2 value to conform the application. In the circuit diagram condition, the MCU’s reset pin level varies with VDD voltage variation, and the differential voltage is 0.7V. If the VDD drops and the voltage lower than reset pin detect level, the system would be reset. If want to make the reset active earlier, set the R2 > R1 and the cap between VDD and C terminal voltage is larger than 0.7V. The external reset circuit is with a stable current through R1 and R2. For power consumption issue application, e.g. DC power system, the current must be considered to whole system power consumption.

✷ ***Note: Under unstable power condition as brown out reset, “Zener diode rest circuit” and “Voltage bias reset circuit” can protects system no any error occurrence as power dropping. When power drops below the reset detect voltage, the system reset would be triggered, and then system executes reset sequence. That makes sure the system work well under unstable power situation.***

3.6.5 External Reset IC



The external reset circuit also use external reset IC to enhance MCU reset performance. This is a high cost and good effect solution. By different application and system requirement to select suitable reset IC. The reset circuit can improve all power variation.

4 SYSTEM CLOCK

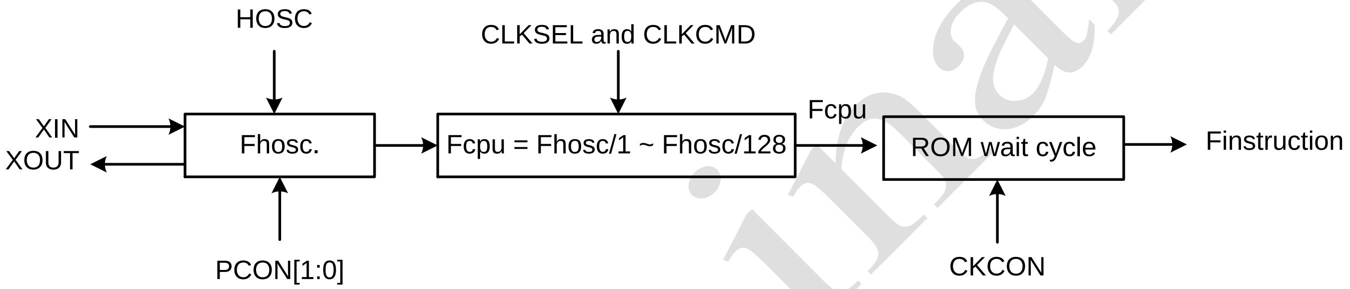
4.1 OVERVIEW

The micro-controller is a single clock system that is high-speed clock. The high-speed clock includes internal high-speed oscillator, external oscillators and external clock input mode selected by “High_CLK” code option. The high-speed clock can be system clock source through a divider to decide the system clock rate by CLKSEL and CLKCMD register.

- **High-speed oscillator**

Internal high-speed oscillator is 32MHz RC type called “**IHRC_32M**” and “**IHRC_32M_RTC**”.
External high-speed oscillator includes crystal/ceramic (4MHz, 12MHz) and External clock input mode.

- **System clock block diagram**



- HOSC: High_Clk code option.
- Fhosc: External high-speed clock / Internal high-speed RC clock.
- Fcpu: System clock rate.
- Finstruction: Instruction cycle.

4.2 FCPU

The system clock rate is called “**Fcpu**” which is divided from the system clock source and decides the system operating rate. Fcpu rate is selected by CLKSEL and CLKCMD register. The range is **Fhosc/1~Fhosc/128** under system normal mode. If the system high clock source is external 4MHz crystal, and the CLKSEL is Fhosc/4, the Fcpu frequency is 4MHz/4 = 1MHz. The instruction cycle is called “Finstruction” which is Fcpu divided by ROM wait cycle. The Finstruction maximum limitation is 8MHz.

Finstruction = Fcpu / ROM wait cycle

> ✸ ***Note: Finstruction maximum limitation is 8MHz therefore if user select Fcpu= Fhosc/1 (Fhosc=32MHz), ROM wait cycle must be set “4”.***

4.2.1 SYSTEM CLOCK RATE CONTROL REGISTER

System clock rate is control by CLKSEL and CLKCMD registers. CLKSEL decide system clock rate to Fhosc/1~Fhosc/128 and write 0x69 to CLKCMD to start clock switch procedure.

0E5H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKSEL	-	-	-	-	-	CLKRATE2	CLKRATE1	CLKRATE0
Read/Write	-	-	-	-	-	R/W	R/W	R/W
After reset	-	-	-	-	-	0	0	0

Bit[2:0] **CLKRATE[2:0]:** Fcpu clock source select bits.
000 = Fhosc/128, 001 = Fhosc/64, 010 = Fhosc/32, 011 = Fhosc/16, 100 =Fhosc/8, 101 = Fhosc/4, 110 =

Fhosc/2, 111 = Fhosc/1.

0E6H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCMD	CLKCMD7	CLKCMD6	CLKCMD5	CLKCMD4	CLKCMD3	CLKCMD2	CLKCMD1	CLKCMD0
Read/Write	W	W	W	W	W	W	W	W
After reset	0	0	0	0	0	0	0	0

Bit[7:0] **CLKCMD[7:0]:** Clock switch control.
0x69: Clock switch start.
Other: Nothing.

08EH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CKCON	PSYN	PWSC2	PWSC1	PWSC0	ESYN	EWSC2	EWSC1	EWSC0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After Reset	0	1	1	1	0	0	0	1

Bit 7 **PSYN:** Program memory synchronous/asynchronous write control.
0 = "Synchronous" type write access is performed.
1 = "Asynchronous" type write access is performed.

Bit [6:4] **PWSC[2:0]:** Program memory wait state cycle.

CKCON register PWSC[2:0]	Program memory wait cycle
000	1
001	2
010	3
011	4
100	5
101	6
110	7
111	8

Bit 3 **ESYN:** External data memory synchronous/asynchronous write control
0 = "Synchronous" type write access is performed.
1 = "Asynchronous" type write access is performed.

Bit [2:0] **EWSC[2:0]:** External data memory stretch cycle control.

CKCON register EWSC[2:0]	External RAM read wait cycle	External RAM write wait cycle	
		ESYN = 0	ESYN = 1
000	1	1	3
001	2	2	4
010	3	3	5
011	4	4	6
100	5	5	7
101	6	6	8
110	7	7	9
111	8	8	10

System clock rate and program memory wait state cycle limitation as follows.

Code Option High_CLK	CLKSEL[2:0]	CKCON[6:4]
IHRC_32M	"111" = Fhosc/1	Only support "011(/4)"~"111(/8)"
	"110" = Fhosc/2	Only support "001(/2)"~"111(/8)"
	"101~000" = Fhosc/4~Fhosc/128	Support "000(/1)"~"111(/8)"
12M X'tal (Crystal 16MHz)	"111" = Fhosc/1	Only support "001(/2)"~"111(/8)"
	"110~000" = Fhosc/2~Fhosc/128	Support "000(/1)"~"111(/8)"
4M X'tal (Crystal 8MHz)	"111~000" = Fhosc/1~Fhosc/128	Support "000(/1)"~"111(/8)"
Ext_CLK (32MHz)	"111" = Fhosc/1	Only support "011(/4)"~"111(/8)"
	"110" = Fhosc/2	Only support "001(/2)"~"111(/8)"
	"101~000" = Fhosc/4~Fhosc/128	Support "000(/1)"~"111(/8)"

4.3 NOISE FILTER

The Noise Filter controlled by “Noise_Filter” code option is a low pass filter and supports external oscillator including RC and crystal modes. The purpose is to filter high rate noise coupling on high clock signal from external oscillator.
In high noisy environment, enable “Noise_Filter” code option is the strongly recommendation to reduce noise effect.

4.4 SYSTEM HIGH-SPEED CLOCK

The system high-speed clock has internal and external two-type. The external high-speed clock includes 4MHz, 12MHz crystal/ceramic and external clock input mode. These high-speed oscillators are selected by “**High_CLK**” code option. The internal high-speed clock supports real time clock (RTC) function. Under “IHRC_32M_RTC” mode, the internal high-speed clock and external 32KHz oscillator active. The internal high-speed clock is the system clock source, and the external 32KHz oscillator is the RTC clock source to supply a accurately real time clock rate.

4.4.1 HIGH_CLK CODE OPTION

For difference clock functions, Sonix provides multi-type system high clock options controlled by “High_CLK” code option. The High_CLK code option defines the system oscillator types including IHRC_32M, IHRC_32M_RTC, Ext_CLK, 12M X’tal and 4M X’tal. These oscillator options support different bandwidth oscillator.

- **IHRC_32M:** The system high-speed clock source is internal high-speed 32MHz RC type oscillator. In the mode, XIN and XOUT pins are bi-direction GPIO mode, and not to connect any external oscillator device.
- **IHRC_32M_RTC:** The system high-speed clock source is internal high-speed 32MHz RC type oscillator. The RTC clock source is external low-speed 32768Hz crystal. The XIN and XOUT pins are defined to drive external 32768Hz crystal and disables GPIO function.
- **Ext_CLK:** The system high-speed clock source is external clock input mode. The input signal only connects to XIN pin, and the XOUT pin is bi-direction GPIO mode.
- **12M X’tal:** The system high-speed clock source is external high-speed crystal/ceramic. The oscillator bandwidth is 10MHz~16MHz.
- **4M X’tal:** The system high-speed clock source is external high-speed crystal/resonator. The oscillator bandwidth is 1MHz~10MHz.

4.4.2 INTERNAL HIGH-SPEED OSCILLATOR RC TYPE (IHRC)

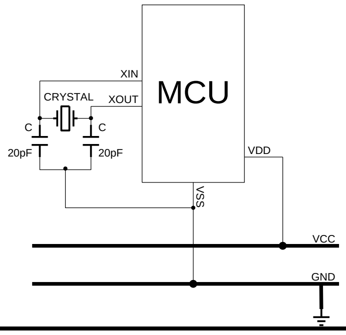
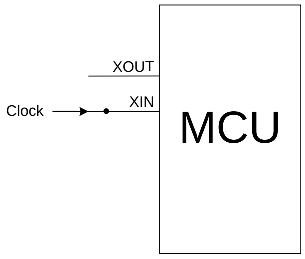
The internal high-speed oscillator is 32MHz RC type. The accuracy is ±2% under commercial condition. When the “High_CLK” code option is “IHRC_32M” or “IHRC_32M_RTC”, the internal high-speed oscillator is enabled.

- **IHRC_32M:** The system high-speed clock is internal 32MHz oscillator RC type. XIN/XOUT pins are general purpose I/O pins.
- **IHRC_32M_RTC:** The system high-speed clock is internal 32MHz oscillator RC type, and the real time clock is external 32768Hz crystal. XIN/XOUT pins connect with external 32768Hz crystal.

4.4.3 EXTERNAL HIGH-SPEED OSCILLATOR

The external high-speed oscillator includes 4MHz, 12MHz, and external clock input mode. The 4MHz and 12MHzoscillators support crystal and ceramic types connected to XIN/XOUT pins with 20pF capacitors to ground. The external clock input mode is only connected to XIN pin.

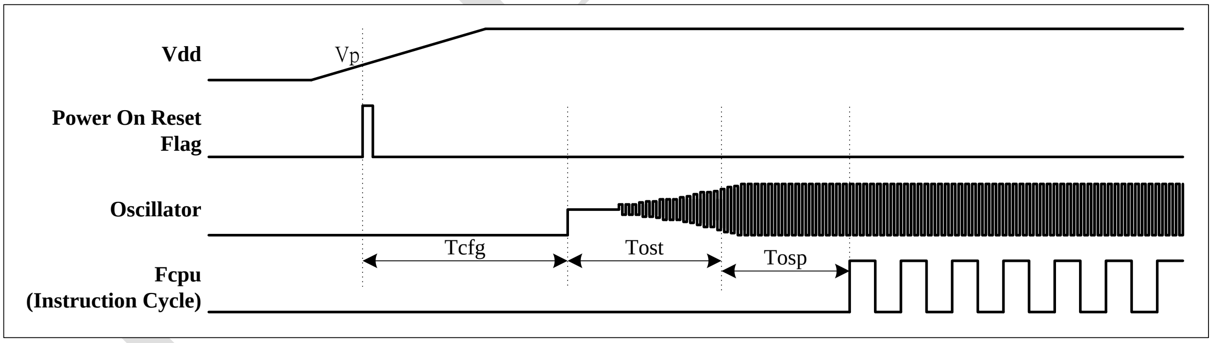
4.4.4 EXTERNAL OSCILLATOR APPLICATION CIRCUIT

CRYSTAL/CERAMIC	EXTERNAL CLOCK
	

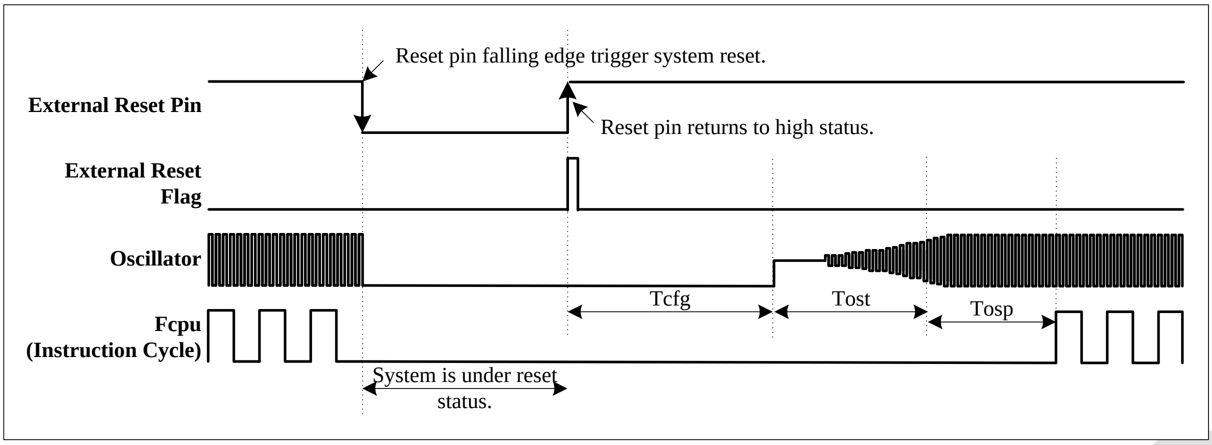
4.5 SYSTEM CLOCK TIMING

Parameter	Symbol	Description	Typical
Hardware configuration time	Tcfg	131072*F_{IHRC}	4.096ms @ F_{IHRC} = 32MHz
Oscillator start up time	Tost	The start-up time is depended on oscillator's material, factory and architecture. Normally, the low-speed oscillator's start-up time is lower than high-speed oscillator. The RC type oscillator's start-up time is faster than crystal type oscillator.	-
Oscillator warm-up time	Tosp	Oscillator warm-up time of reset condition. 2048*F_{hosc+} 5*F_{ILRC} (Power on reset, LVD reset, watchdog reset, external reset pin active.)	64ms @ F_{hosc} = 32KHz 668us @ F_{hosc} = 4MHz 284us @ F_{hosc} = 16MHz
		Oscillator warm-up time of power down mode wake-up condition. 2048*F_{hosc+}5*F_{ILRC} ……Crystal/resonator type oscillator, e.g. 32768Hz crystal, 4MHz crystal, 16MHz crystal… 64*F_{hosc+}5*F_{ILRC}……RC type oscillator, e.g. internal high-speed RC type oscillator.	X'tal: 64ms @ F_{hosc} = 32KHz 668us @ F_{hosc} = 4MHz 284us @ F_{hosc} = 16MHz RC: 172us @ F_{hosc} = 4MHz 160us @ F_{hosc} = 16MHz

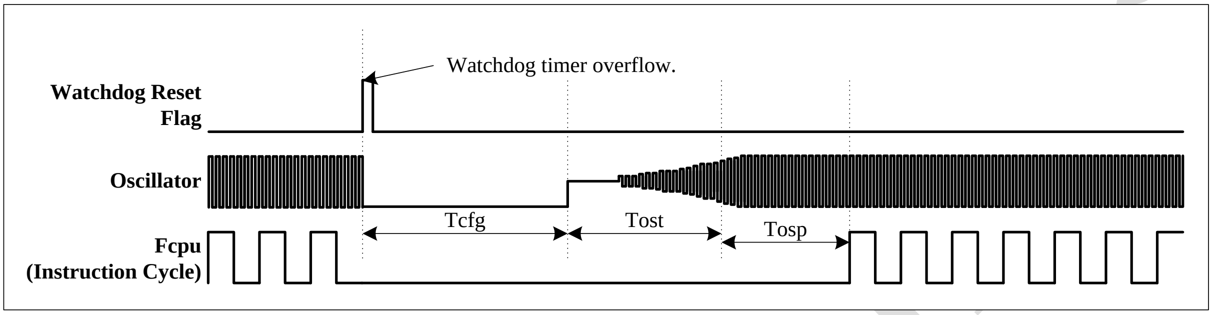
- **Power On Reset Timing**



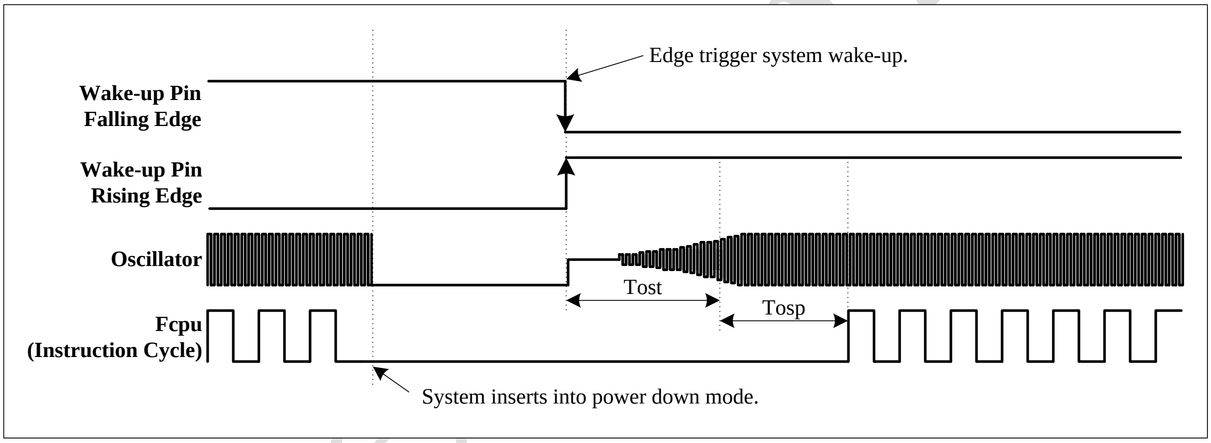
- **External Reset Pin Reset Timing**


Reset pin falling edge trigger system reset.
External Reset Pin
Reset pin returns to high status.
External Reset Flag
Oscillator
Tcfg
Tost
Tosp
Fcpu (Instruction Cycle)
System is under reset status.

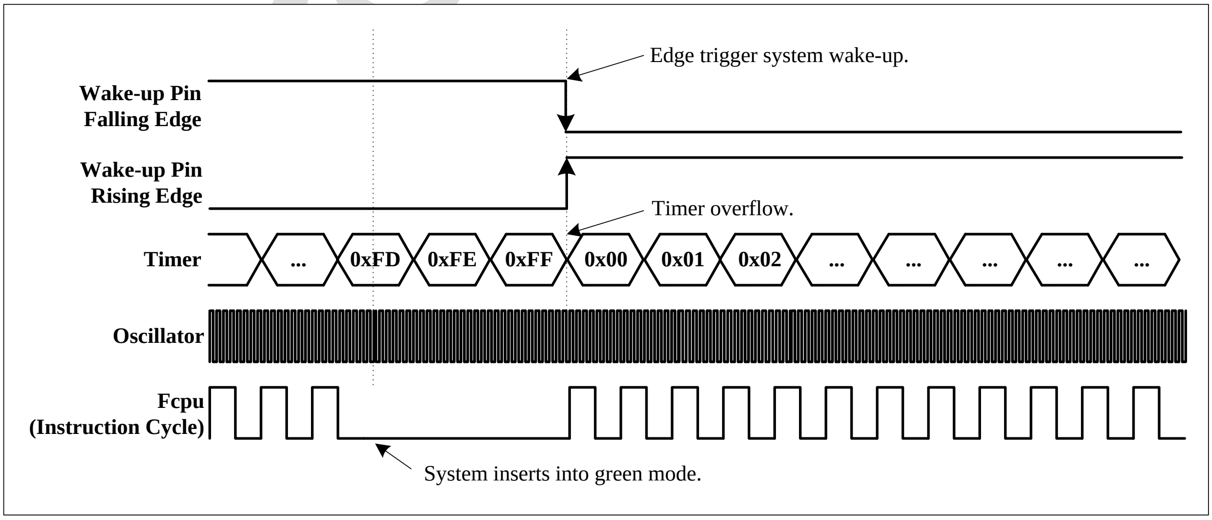
- ## Watchdog Reset Timing



- ## Power Down Mode Wake-up Timing

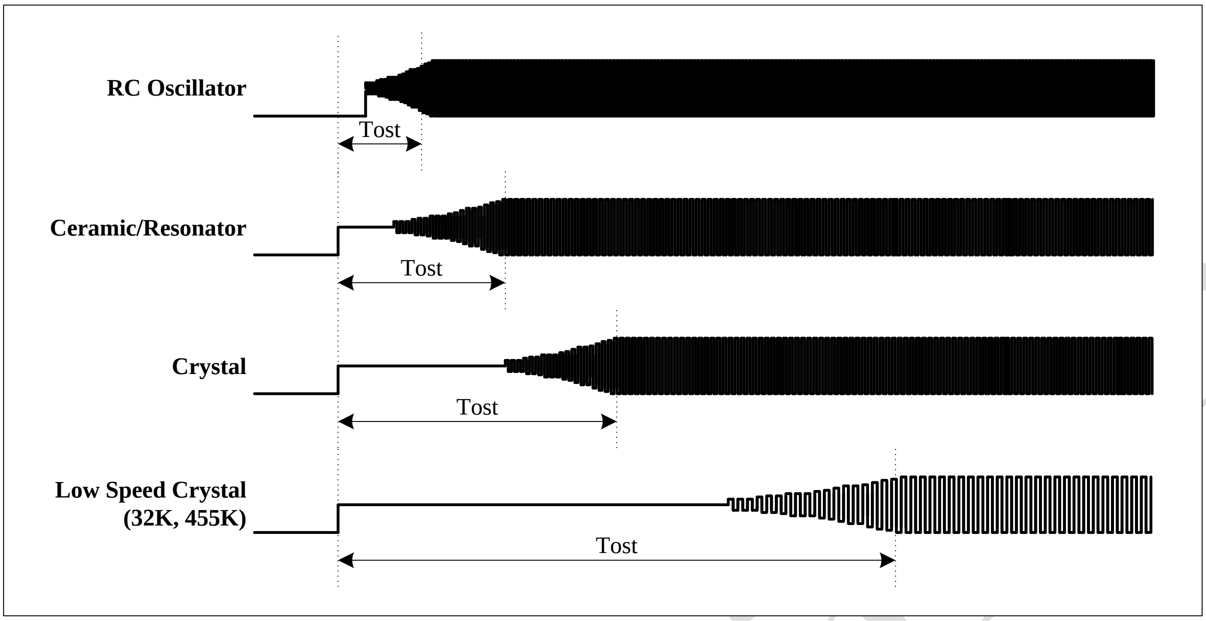


- ## Idle Mode Wake-up Timing



● Oscillator Start-up Time

The start-up time is depended on oscillator's material, factory and architecture. Normally, the low-speed oscillator's start-up time is lower than high-speed oscillator.



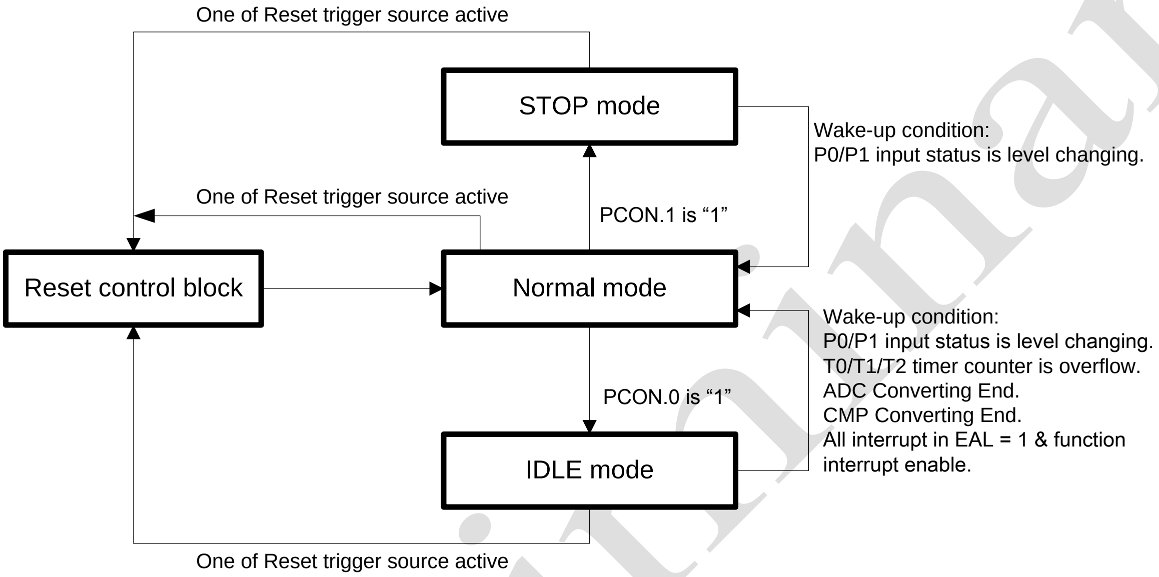
5 SYSTEM OPERATION MODE

5.1 OVERVIEW

The chip builds in three operating mode for difference clock rate and power saving reason. These modes control oscillators, op-code operation and analog peripheral devices' operation.

- Normal mode: System high-speed operating mode.
- IDLE mode: System idle mode (Green mode).
- STOP mode: System power saving mode (Sleep mode).

Operating Mode Control Block


One of Reset trigger source active
STOP mode
Wake-up condition:
P0/P1 input status is level changing.
One of Reset trigger source active
PCON.1 is "1"
Reset control block
Normal mode
Wake-up condition:
P0/P1 input status is level changing.
T0/T1/T2 timer counter is overflow.
ADC Converting End.
CMP Converting End.
All interrupt in EAL = 1 & function interrupt enable.
PCON.0 is "1"
IDLE mode
One of Reset trigger source active

Operating Mode Clock Control Table

Operating Mode	Normal Mode	IDLE Mode	STOP Mode
IHRC	**IHRC, IHRC_RTC:** Running **Ext. OSC:** Disable	**IHRC, IHRC_RTC:** Running **Ext. OSC:** Disable	Stop
ILRC	Running	Running	By Watch_Dog Code option: Enable/Disable: Stop Always On: Running
Ext. Osc.	**IHRC:** Disable **IHRC_RTC, Ext. OSC :** Running	**IHRC:** Disable **IHRC_RTC, Ext. OSC :** Running	Stop
CPU instruction	Executing	Stop	Stop
Timer 0 (Timer, Event counter)	Active By tr0	Active By tr0	Inactive
Timer 1 (Timer, Event counter,)	Active By tr1	Active By tr1	Inactive
Timer 2 (Timer, Event counter, compare, capture)	Active By t2i0 & t2i1	Active By t2i0 & t2i1	Inactive
PWM0	Active as enable	Active as enable	Inactive
PWM1	Active as enable	Active as enable	Inactive
PWM2	Active as enable	Active as enable	Inactive
SIO	Active as enable	Active as enable	Inactive
I2C	Active as enable	Active as enable	Inactive
UART	Active as enable	Active as enable	Inactive
ADC	Active as enable	Active as enable	Inactive
Comparator	Active as enable	Active as enable	Inactive
OP	Active as enable	Active as enable	Inactive
Watchdog timer	By Watch_Dog Code option	By Watch_Dog Code option	By Watch_Dog Code option
Internal interrupt	All active	All active	All inactive
External interrupt	All active	All active	All inactive
Wakeup source	-	P0, P1, Reset, T0/T1/T2, ADC, CMP. All interrupt in EAL = 1 & function interrupt enable	P0, P1, Reset

- **Ext.Osc:** External high-speed oscillator (XIN/XOUT).
- **IHRC:** Internal high-speed oscillator RC type.
- **ILRC:** Internal low-speed oscillator RC type.

5.2 NORMAL MODE

The Normal Mode is system high clock operating mode. The system clock source is from high speed oscillator. The program is executed. After power on and any reset trigger released, the system inserts into normal mode to execute program. When the system is wake-up from STOP/IDLE mode, the system also inserts into normal mode. In normal mode, the high speed oscillator actives, and the power consumption is largest of all operating modes.

- The program is executed, and full functions are controllable.
- The system rate is high speed.
- The high speed oscillator and internal low speed RC type oscillator active.
- Normal mode can be switched to other operating modes through PCON register.
- STOP/IDLE mode is wake-up to normal mode.

5.3 STOP MODE

The STOP mode is the system ideal status. No program execution and oscillator operation. Only internal regulator actives to keep all control gates status, register status and SRAM contents. The STOP mode is waked up by P0/P1 hardware level change trigger. P0 wake-up function is always enables. The STOP mode is wake-up to normal mode. Inserting STOP mode is controlled by stop bit of PCON register. When stop = 1, the system inserts into STOP Mode. After system wake-up from STOP mode, the stop bit is disabled (zero status) automatically.

- The program stops executing, and full functions are disabled.
- All oscillators including external high speed oscillator, internal high speed oscillator and internal low speed oscillator stop.
- Only internal regulator actives to keep all control gates status, register status and SRAM contents.
- The system inserts into normal mode after wake-up from STOP mode.
- The STOP mode wake-up source is P0/P1 level change trigger.

5.4 IDLE MODE

The IDLE mode is another system ideal status not like STOP mode. In STOP mode, all functions and hardware devices are disabled. But in IDLE mode, the system clock source keeps running, so the power consumption of IDLE mode is larger than STOP mode. In IDLE mode, the program isn’t executed, but the timer with wake-up function actives as enabled, and the timer clock source is the non-stop system clock. The IDLE mode has 2 wake-up sources. One is the P0/P1 level change trigger wake-up. The other one is any interrupt in EAL = 1 & function interrupt enable. That’s mean users can setup any function with interrupt enable, and the system is waked up until the interrupt issue. Inserting IDLE mode is controlled by idle bit of PCON register. When idle = 1, the system inserts into IDLE mode. After system wake-up from IDLE mode, the idle bit is disabled (zero status) automatically.

- The program stops executing, and full functions are disabled.
- Only the timer with wake-up function actives.
- The oscillator to be the system clock source keeps running, and the other oscillators operation is depend on system operation mode configuration.
- If inserting IDLE mode from normal mode, the system insets to normal mode after wake-up.
- The IDLE mode wake-up sources are P0/P1 level change trigger and any interrupt in EAL = 1 & function interrupt enable.
- If the function clock source is system clock, the functions are workable as enabled and under IDLE mode, e.g. Timer, PWM, event counter…
- ADC can work in IDLE mode. After ADC operating with interrupt enable, the system would be waked up from IDLE mode to normal mode.
- CMP can work in IDLE mode. After CMP trigger edge condition is found with interrupt enable, the system would be waked up from IDLE mode to normal mode.

5.5 WAKEUP

5.5.1 OVERVIEW

Under stop mode (sleep mode) or idle mode, program doesn't execute. The wakeup trigger can wake the system up to normal mode. The wakeup trigger sources are external trigger (P0/P1 level change) and internal trigger (any interrupt in EAL = 1 & function interrupt enable). The wakeup function builds in interrupt operation issued IRQ flag and trigger system executing interrupt service routine as system wakeup occurrence.

- Stop mode is waked up to normal mode. The wakeup trigger is only external trigger (P0/P1 level change)
- Idle mode is waked up to normal mode. The wakeup triggers are external trigger (P0/P1 level change) and internal trigger (any interrupt in EAL = 1 & function interrupt enable).

5.5.2 WAKEUP TIME

When the system is in stop mode (sleep mode), the high clock oscillator stops. When waked up from power down mode, MCU waits for 2048 external high-speed oscillator clocks + 5 internal low-speed oscillator clocks and 64 internal high-speed oscillator clocks + 5 internal low-speed oscillator clocks as the wakeup time to stable the oscillator circuit. After the wakeup time, the system goes into the normal mode.

✸ ***Note: Wakeup from green mode is no wakeup time because the clock doesn't stop in green mode.***

The value of the external high clock oscillator wakeup time is as the following.

The Wakeup time = 1/Fosc * 2048 (sec) + 1/Flosc * 5 + high clock start-up time

Example: In power down mode (sleep mode), the system is waked up. After the wakeup time, the system goes into normal mode. The wakeup time is as the following.

The wakeup time = 1/Fosc * 2048 + 1/Flosc * 5 = 0.512 ms (Fosc = 4MHz)
The total wakeup time = 0.512 ms + oscillator start-up time

The value of the internal high clock oscillator RC type wakeup time is as the following.

The Wakeup time = 1/Fosc * 64 (sec) + 1/Flosc * 5 + high clock start-up time

Example: In stop mode (sleep mode), the system is waked up. After the wakeup time, the system goes into normal mode. The wakeup time is as the following.

The wakeup time = 1/Fosc * 64 +1/Flosc * 5= 160 us (Fhosc = 16MHz)

✸ ***Note: The high clock start-up time is depended on the VDD and oscillator type of high clock.***

5.5.3 P1W WAKEUP CONTROL REGISTER

Under stop mode (sleep mode) and green mode, the I/O ports with wakeup function are able to wake the system up to normal mode. The wake-up trigger edge is level changing. When wake-up pin occurs rising edge or falling edge, the system is waked up by the trigger edge. The Port 0 and Port 1 have wakeup function. Port 0 wakeup function always enables, but the Port 1 is controlled by the P1W register.

091H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1W	P17W	P16W	P15W	P14W	P13W	P12W	P11W	P10W
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit[7:0] **P10W~P17W:** Port 1 wakeup function control bits.
0 = Disable P1n wakeup function.
1 = Enable P1n wakeup function.

6 INTERRUPT

6.1 OVERVIEW

This MCU provides 19 interrupt sources, including 3 external interrupt (INT0/INT1/INT2) and 16 internal interrupt (T0/ T1/ T2/ T2COM0/ T2COM1/ T2COM2/ T2COM3/ PW1/ PW2/ PW3/ ADC/ CMP0/ CMP1/ UART/ SPI/ I2C) with 4 priority levels. Each interrupt source includes one or more interrupt request flag, and can be enabled or disabled respectively. During all interrupt operation, the global interrupt control bit (EAL) must be enabled. Most interrupt request flags must be cleared by software. However, some interrupt request flags can be cleared by hardware automatically before executing the ISR. The summary of interrupt source, interrupt vector, priority order and control bit are shown as the table 6.1.

Table 6.1 Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Interrupt Request Flag	Bit Addressable?	Clear by hardware?	Interrupt Request Enable
Reset	0x00	Top	None	N/A	N/A	Always Enable
INT0	0x03	1	IE0 (TCON.1)	Y	Y	EX0 (IEN0.0)
PWM1	0x83	2	PWM1F (IEN4.3)			EPWM1 (IEN4.7)
I2C	0x43	3	SI (I2CCON.3)			EI2C (IEN1.0)
T0	0x0B	4	TF0 (TCON.5)	Y	Y	ET0 (IEN0.1)
PWM2	0x8B	5	PWM2F (IRCON2.0)			EPWM2 (IEN2.1)
SPI	0x4B	6	SPIF (SPSTA.7) WCOL (SPSTA.6) SSERR (SPSTA.5) MODF (SPSTA.4)			ESPI (IEN1.1)
INT1	0x13	7	IE1 (TCON.3)	Y	Y	EX1 (IEN0.2)
PWM3	0x93	8	PWM3F (IECON2.1)			EPWM3 (IEN2.2)
T2COM0	0x53	9	TF2C0 (IRCON.2)	Y	Y	ET2C0 (IEN1.2)
T1	0x1B	10	TF1 (TCON.7)	Y	Y	ET 1 (IEN0.3)
ADC	0x9B	11	ADCF (IRCON2.2)			EADC (IEN2.3)
T2COM1	0x5B	12	TF2C1 (IRCON.3)	Y	Y	ET2C1 (IEN1.3)
UART	0x23	13	TI0 (S0CON.1) RI0 (S0CON.0)	Y		ES0 (IEN0.4)
Comparator 0	0xA3	14	CMP0F (IRCON2.3)			ECMP0 (IEN2.4)
TCOM2	0x63	15	TF2C2 (IRCON.4)	Y	Y	ET2C2 (IEN1.4)
T2	0x2B	16	TF2 (IRCON.6) TF2RL (IRCON07)	Y		ET2 (IEN0.5) ET2RL (IEN1.7)
Comparator 1	0xAB	17	CMP1F (IRCON2.4)			ECMP1 (IEN2.5)
INT2	0xEB	18	IE2 (IEN4.2)		Y	EX2 (IEN4.6)
T2COM3	0x6B	19	TF2C3 (IRCON.5)	Y	Y	ET2C3 (IEN1.5)

6.2 INTERRUPT OPERATION

Interrupt operation is controlled by interrupt request and IEN bits. The interrupt request is interrupt source event indicator, no matter what interrupt function status (enable or disable). The IEN control the system interrupt execution. If IEN = 0, the system won’t jump to interrupt vector to execute interrupt routine. If IEN = 1, the system executes interrupt operation when each of interrupt request flags actives.

- **When both IEN = 1 and interrupt request = 1, the program counter points to interrupt vector and execute interrupt service routine**.

When any interrupt requests occurs, the system provides to jump to interrupt vector and execute interrupt routine. The first procedure is “PUSH” operation. The end procedure after interrupt service routine execution is “POP” operation. The “PUSH” and “POP” operations are through instruction (PUSH, POP) and executed by software.

- **"PUSH" operation: PUSH operation saves directly addressed data onto into stack buffer (RAM). (e.g., PUSH ACC; PUSH 5AH).**

- **"POP" operation: POP operation reloads directly addressed data from stack buffer (RAM). (e.g., POP 5AH; POP ACC).**

- **The SP is incremented before executing PUSH or CALL instruction and it is decremented after executing POP or RET(I) instruction (it always points the top of stack).**

6.3 INTERRUPT PRIORITY

Interrupt priority is decided by natural priority and priority level. Each interrupt source has its specific nature priority order. However, the priority order can be changed by different priority level. In the condition of all interrupt sources have the same priority level, the priority order will follow natural priority, as shown in table 6.1.

- **Nature Priority**

Interrupt group	The highest priority in groups		The 2nd priority in groups		The 3th priority in groups		The lowest priority in groups	
	Address	Interrupt Vector	Address	Interrupt Vector	Address	Interrupt Vector	Address	Interrupt Vector
Group0	0X03	INT0	0X83	PWM1			0X43	I2C
Group1	0X0B	T0	0X8B	PWM2			0X4B	SPI
Group2	0X13	INT1	0X93	PWM3			0X53	T2 COM1
Group3	0X1B	T1	0X9B	ADC			0X5B	T2 COM2
Group4	0X23	UART	0XA3	Comparator 0			0X63	T2 COM3
Group5	0X2B	T2	0XAB	Comparator 1	0XEB	INT2	0X6B	T2 COM4

The interrupt vectors can be assigned to 6 groups (group 0~group 5), as the table above. Inside the group, the priority is follow by nature priority, which is fixed by hardware and it can't be modified. The first column has the highest nature priority and the 4th column has the lowest nature priority.

If INT0, I2C and PWM1 interrupt requests happen at the same time, the system processing interrupt sequence is INT0, PWM1 and then I2C. The system processes INT0 interrupt service routine first, and then processes PWM1 interrupt routine…Until finishing processing all interrupt requests.

- **Priority Level**

Level	Priority within the same priority level	IP1.x bit	IP0.x bit
Level 0	Lowest	0	0
Level 1		0	1
Level 2		1	0
Level 3	Highest	1	1

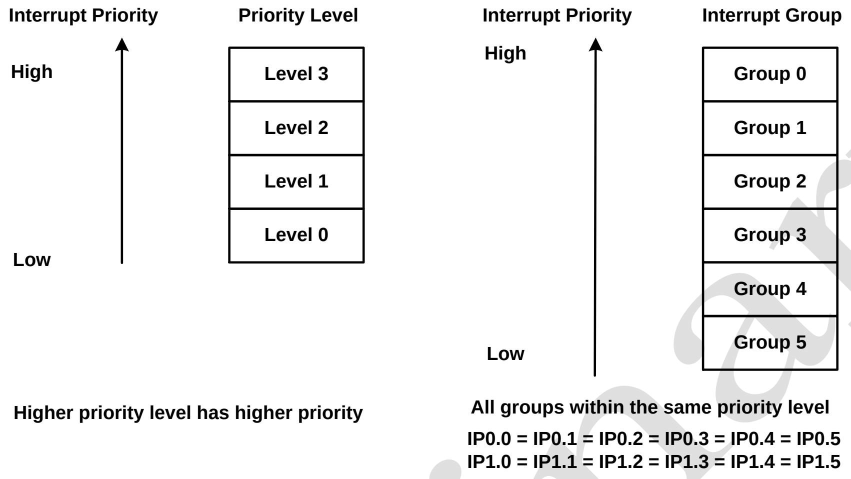
Between the groups, group 0 has the highest nature priority and group 5 has the lowest. The interrupt priority between groups can be modified by priority level. Priority level can be set to 4 levels, from level 0~level 3, and it is defined by IP0, IP1 register (as the table above). Level 0 is the lowest priority and level 3 is the highest. Higher priority level ISR has higher priority and it can interrupt the lower priority level ISR. As the same priority level, the higher nature priority ISR has higher priority and it won't interrupt the lower nature priority ISR. After the previous ISR is finished, the next ISR will execute.

0A9H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IP0	-	-	IP05	IP04	IP03	IP02	IP01	IP00
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	R/W
After reset	-	-	0	0	0	0	0	0

Bit [5:0] **IP0[5:0]:** Interrupt priority. Each bit together with corresponding bit from IP1 register specifies the priority level of the respective interrupt priority group.

0B9H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IP1	-	-	IP15	IP14	IP13	IP12	IP11	IP10
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	R/W
After reset	-	-	0	0	0	0	0	0

Bit [5:0] **IP1[5:0]:** Interrupt priority. Each bit together with corresponding bit from IP0 register specifies the priority level of the respective interrupt priority group.



Higher priority level has higher priority

All groups within the same priority level
IP0.0 = IP0.1 = IP0.2 = IP0.3 = IP0.4 = IP0.5
IP1.0 = IP1.1 = IP1.2 = IP1.3 = IP1.4 = IP1.5

When more than one interrupt request occur, the highest priority request must be executed first. Choose the highest priority request according natural priority and priority level. The steps are as the following:

1. Choose the groups which have the highest priority level between all groups.
2. Choose the group which is the highest nature priority between the groups with the highest priority level.
3. Choose the ISR which has the highest nature priority inside the group with the highest priority.

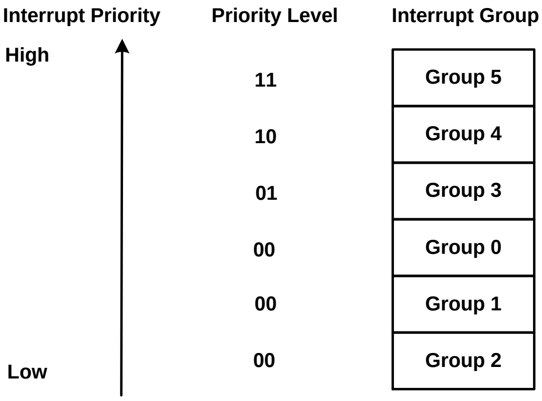
	ROM	**Nature Priority**
0003H	***INT0 Interrupt vector***	**1**
0083H	***PWM1 Interrupt vector***	**2**
0043H	***I2C Interrupt vector***	**3**
000BH	***T0 Interrupt vector***	**4**
008BH	***PWM2 Interrupt vector***	**5**
004BH	***SPI Interrupt vector***	**6**
0013H	***INT1 Interrupt vector***	**7**
0093H	***PWM3 Interrupt vector***	**8**
0053H	***T2 COM0 Interrupt vector***	**9**
001BH	***T1 Interrupt vector***	**10**
009BH	***ADC Interrupt vector***	**11**
005BH	***T2 COM1 Interrupt vector***	**12**
0023H	***UART Interrupt vector***	**13**
00A3H	***Comparator 0 Interrupt vector***	**14**
0063H	***T2 COM2 Interrupt vector***	**15**
002BH	***T2 Interrupt vector***	**16**
00ABH	***Comparator 1 Interrupt vector***	**17**
00EBH	***INT2 Interrupt vector***	**18**
006BH	***T2 COM3 Interrupt vector***	**19**

After system reset, IP1 = IP0 = 0x00, ORG 0x03 is nature priority 1. ORG 0x83 is nature priority 2…In the case, the interrupt nature priority is as above.

Example: Priority level setting.

```
MOV     IP0, #00101000B  ; Set group0~group5 in different priority level.
MOV     IP1, #00110000B  ;
```

As the example, group5 has the highest priority level and group0~group2 have the lowest priority level. It means the interrupt vector in group5 has the highest interrupt priority, the 2nd interrupt priority in group4 and the 3rd interrupt priority in group3. Group0~ group2 have the same priority level thus the nature priority rule will be followed. Therefore, interrupt priority will be group5> group4> group3> group0> group1> group2.



Interrupt Priority	Priority Level	Interrupt Group
High	11	Group 5
	10	Group 4
	01	Group 3
	00	Group 0
	00	Group 1
Low	00	Group 2

Priority Level:

GP5>GP4>GP3=GP2=GP1=GP0

Interrupt Priority:

GP5>GP4>GP3>GP0>GP1>GP2

6.4 IEN INTERRUPT ENABLE REGISTER

IEN is the interrupt request control register including 16 internal interrupts, 3 external interrupts enable control bits. One of the register to be set "1" is to enable the interrupt request function. Once of the interrupt occur, the stack is incremented and program jump to ORG 0x03~0xEB to execute interrupt service routines. The program exits the interrupt service routine when the returning interrupt service routine instruction (RETI) is executed.

0A8H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN0	EAL	-	ET2	ES0	ET1	EX1	ET0	EX0
Read/Write	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	-	0	0	0	0	0	0

Bit 7 **EAL:** Enable all interrupt control bit.
0 = Disable all interrupt function.
1 = Enable all interrupt function.

Bit 5 **ET2:** T2 timer interrupt control bit
0 = Disable T2 interrupt function.
1 = Enable T2 interrupt function.

Bit 4 **ES0:** UART interrupt control bit.
0 = Disable UART interrupt function.
1 = Enable UART interrupt function.

Bit 3 **ET1:** T1 timer interrupt control bit.
0 = Disable T1 interrupt function.
1 = Enable T1 interrupt function.

Bit 2 **EX1:** External P2.2 interrupt (INT1) control bit.
0 = Disable INT1 interrupt function.
1 = Enable INT1 interrupt function.

Bit 1 **ET0:** T0 timer interrupt control bit.
0 = Disable T0 interrupt function.
1 = Enable T0 interrupt function.

Bit 0 **EX0:** External P2.1 interrupt (INT0) control bit.
0 = Disable INT0 interrupt function.
1 = Enable INT0 interrupt function.

0B8H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN1	ET2RL	-	ET2C3	ET2C2	ET2C1	ET2C0	ESPI	EI2C
Read/Write	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	-	0	0	0	0	0	0

Bit 7 **ET2RL:** T2 Timer external reload interrupt control bit.
0 = Disable T2 external reload interrupt function.
1 = Enable T2 external reload interrupt function.

Bit 5 **ET2C3:** T2 Timer COM3 interrupt control bit.
0 = Disable T2COM3 interrupt function.
1 = Enable T2COM3 interrupt function.

Bit 4 **ET2C2:** T2 Timer COM2 interrupt control bit.
0 = Disable T2COM2 interrupt function.
1 = Enable T2COM2 interrupt function.

Bit 3 **ET2C1:** T2 Timer COM1 interrupt control bit.
0 = Disable T2COM1 interrupt function.
1 = Enable T2COM1 interrupt function.

Bit 2 **ET2C0:** T2 Timer COM0 interrupt control bit.
0 = Disable T2COM0 interrupt function.
1 = Enable T2COM0 interrupt function.

Bit 1 **ESPI:** SPI interrupt control bit
0 = Disable SPI interrupt function.
1 = Enable SPI interrupt function.

Bit 0 **EI2C:** I2C interrupt control bit.
0 = Disable I2C interrupt function.
1 = Enable I2C interrupt function.

09AH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN2	-	-	ECMP1	ECMP0	EADC	EPWM3	EPWM2	-
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	-
After reset	-	-	0	0	0	0	0	-

Bit 5 **ECMP1:** Comparator 1 interrupt control bit.
0 = Disable CMP1 interrupt function.
1 = Enable CMP1 interrupt function.

Bit 4 **ECMP0:** Comparator 0 interrupt control bit.
0 = Disable CMP0 interrupt function.
1 = Enable CMP0 interrupt function.

Bit 3 **EADC:** ADC interrupt control bit.
0 = Disable ADC interrupt function.
1 = Enable ADC interrupt function.

Bit 2 **EPWM3:** PWM3 interrupt control bit.
0 = Disable PWM3 interrupt function.
1 = Enable PWM3 interrupt function.

Bit 1 **EPWM2:** PWM2 interrupt control bit
0 = Disable PWM2 interrupt function.
1 = Enable PWM2 interrupt function.

0D1H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN4	EPWM1	EX2	-	-	PWM1F	IE2	-	-
Read/Write	R/W	R/W	-	-	R/W	R/W	-	-
After reset	0	0	-	-	0	0	-	-

Bit 7 **EPWM1:** PWM1 interrupt control bit.
0 = Disable PWM1 interrupt function.
1 = Enable PWM1 interrupt function.

Bit 6 **EX2:** External P2.3 interrupt (INT2) control bit.
0 = Disable INT2 interrupt function.
1 = Enable INT2 interrupt function.

6.5 INTERRUPT REQUEST REGISTER

The request register includes all interrupt request indication flags. Each one of the interrupt requests occurs, the bit of the request register would be set “1”. Most request value needs to be clear by programming after detecting the flag. In the interrupt vector of program, users know the any interrupt requests occurring by the register and do the routine corresponding of the interrupt request. However, some interrupt request can be cleared by hardware automatically before executing ISR. In this condition, users can ignore the steps of check request flag.

0C0H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRCON	TF2RL	TF2	TF2C3	TF2C2	TF2C1	TF2C0	-	-
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	-	-
After reset	0	0	0	0	0	0	-	-

Bit 7 **TF2RL:** T2 timer external reload interrupt request flag.
0 = None TF2RL interrupt request
1 = TF2RL interrupt request.

Bit 6 **TF2:** T2 timer interrupt request flag.
0 = None T2 interrupt request.
1 = T2 interrupt request.

Bit 5 **TF2C3:** T2 Timer COM3 interrupt request flag.
0 = None T2COM3 interrupt request.
1 = T2COM3 interrupt request.

Bit 4 **TF2C2:** T2 Timer COM2 interrupt request flag.
0 = None T2COM2 interrupt request.
1 = T2COM2 interrupt request.

Bit 3 **TF2C1:** T2 Timer COM1 interrupt request flag.
0 = None T2COM1 interrupt request.
1 = T2COM1 interrupt request.

Bit 2 **TF2C0:** T2 Timer COM0 interrupt request flag.
0 = None T2COM0 interrupt request.
1 = T2COM0 interrupt request.

0BFH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRCON2	-	-	-	CMP1F	CMP0F	ADCF	PWM3F	PWM2F
Read/Write	-	-	-	R/W	R/W	R/W	R/W	R/W
After reset	-	-	-	0	0	0	0	0

Bit 4 **CMP1F:** Comparator 1 interrupt request flag.
0 = None CMP1 interrupt request
1 = CMP1 interrupt request.

Bit 3 **CMP0F:** Comparator 0 interrupt request flag.
0 = None CMP0 interrupt request.
1 = CMP0 interrupt request.

Bit 2 **ADCF:** ADC interrupt request flag.
0 = None ADC interrupt request.
1 = ADC interrupt request.

Bit 1 **PWM3F:** PWM3 interrupt request flag.
0 = None PWM3 interrupt request
1 = PWM3 interrupt request.

Bit 0 **PWM2F:** PWM2 interrupt request flag.
0 = None PWM2 interrupt request
1 = PWM2 interrupt request.

088H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
Read/Write	R/W	R/W	R/W	R/W	R/W	-	R/W	-
After reset	0	0	0	0	0	-	0	-

Bit 7 **TF1:** T1 timer external reload interrupt request flag.
0 = None T1 interrupt request
1 = T1 interrupt request.

Bit 6 **TR1:** T1 timer enable control bit.
0 = Disable T1 timer function.
1 = Enable T1 timer function.

Bit 5 **TF0:** T0 timer external reload interrupt request flag.
0 = None T0 interrupt request
1 = T0 interrupt request.

Bit 4 **TR0:** T0 timer enable control bit.
0 = Disable T0 timer function.
1 = Enable T0 timer function.

Bit 3 **IE1:** External P2.2 interrupt (INT1) request flag
0 = None INT1 interrupt request.
1 = INT1 interrupt request.

Bit 1 **IE0:** External P2.1 interrupt (INT0) request flag
0 = None INT0 interrupt request.
1 = INT0 interrupt request.

098H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:6] **SM[1:0]:** UART mode select.
00 = Mode 0, Synchronous shift register.
01 = Mode1, 8-bit UART with variable Baud Rate
10 = Mode2, 9-bit UART with fixed Baud Rate
11 = Mode3, 9-bit UART with variable Baud Rate

Bit 5 **SM20:** Multiprocessor communication enable.
0 = Disable multiprocessor communication
1 = Enable multiprocessor communication.

Bit 4 **REN0:** UART receive control bit.
0 = Disable UART receive function.

1 = Enable UART receive function.

Bit 3 **TB80:** 9th transmission bit.
TB80 is the 9th transmission bit in Mode2 and Mode3. The state of this bit corresponds with the state of the 9th transmitted bit (e.g. parity check or multiprocessor communication). It is controlled by software

Bit 2 **RB80:** 9th receive bit.
RB80 is the 9th bit received in Mode2 and Mode3. It reflects the state of the 9th received bit. In Mode 1, if multiprocessor communication is enabled (sm20 = 0), this bit is the stop bit that was received. In Mode 0 this bit is not used

Bit 1 **TI0:** UART transmit interrupt request flag. It indicates completion of a serial transmission at UART. It is set by hardware at the end of bit 8 in mode 0 or at the beginning of a stop bit in other modes. It must be cleared by software.
0 = None UART transmit interrupt request.
1 = UART transmit interrupt request.

Bit 0 **RI0:** UART receive interrupt request flag. It is set by hardware after completion of a serial reception at UART. It is set by hardware at the end of bit 8 in mode 0 or in the middle of a stop bit in other modes. It must be cleared by software.
0 = None UART receive interrupt request.
1 = UART receive interrupt request.

0E1H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPSTA	SPIF	WCOL	SSERR	MODF	-	-	-	-
Read/Write	R	R	R	R	-	-	-	-
After reset	0	0	0	0	-	-	-	-

Bit 7 **SPIF:** Serial Peripheral Data Transfer Flag.
Set by hardware upon data transfer completion. Cleared by reading the SPSTA register with the SPIF bit set, and then reading the SPDAT register.

Bit 6 **WCOL:** Write Collision Flag.
Set by hardware upon write collision to SPDAT. Cleared by an access to SPSAT register and an access to SPDAT register.

Bit 5 **SSERR:** Synchronous Serial Slave Error Flag.
Set by hardware when “ssn” input is deasserted before the end of receive sequence. Cleared by disabling the SPI module (clearing SPEN bit in SPCON register).

Bit4 **MODF:** Mode Fault Flag.
Set by hardware when the “ssn” pin level is in conflict with actual mode of the SPI_MS controller (configured as master while externally selected as slave). Cleared by hardware when the “ssn” pin is at appropriate level and the SPCON register be writed any value.

0DCH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CCON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

The I2CCON register controls the operation of I2C interface. The CPU can read from and write to this 8-bit, directly addressable SFR. Two bits of this register are affected by the I2C hardware: the SI bit is set when serial interrupt is requested, and the STO bit is cleared when STOP condition is present on the I2C bus.

Bit 6 **ENS1:** I2C enable bit.
0 = I2C function is enabled
1 = I2C function is disabled

Bit 5 **STA:** START Flag.
0 = No START condition is transmitted.
1 = A START condition is transmitted if the bus is free.

Bit 4 **STO:** STOP Flag.
0 = No STOP condition is transmitted.
1 = A STOP condition is transmitted to the I2C bus in master mode.

Bit 3 **SI:** Serial Interrupt Flag.
The SI is set by hardware when one of 25 out of 26 possible I2C states is entered. The only state that does not set the SI is state F8h, which indicates that no relevant state information is available. The SI flag must be cleared by software. In order to clear the SI bit, '0' must be written to this bit. Writing a '1' to SI bit does not change value of the SI.

Bit 2 **AA:** Assert Acknowledge Flag.
0 = An "not acknowledge" will be returned when:
- a data byte has been received while I2C was in master receiver mode
- a data byte has been received while I2C was in slave receiver mode

1 = An "acknowledge" will be returned when:
- the "own slave address" has been received
- the general call address has been received while GC bit in I2CADDR register was set
- a data byte has been received while I2C was in master receiver mode
- a data byte has been received while I2C was in slave receiver mode

0D1H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN4	EPWM1	EX2	-	-	PWM1F	IE2	-	-
Read/Write	R/W	R/W	-	-	R/W	R/W	-	-
After reset	0	0	-	-	0	0	-	-

Bit 7 **EPWM1:** PWM1 interrupt control bit.
0 = Disable PWM1 interrupt function.
1 = Enable PWM1 interrupt function.

Bit 6 **EX2:** External P2.3 interrupt (INT2) control bit.
0 = Disable INT2 interrupt function.
1 = Enable INT2 interrupt function.

Bit 3 **PWM1F:** PWM1 interrupt request flag.
0 = None PWM1 interrupt request
1 = PWM1 interrupt request.

Bit 2 **IE2:** External P2.3 interrupt (INT2) request flag
0 = None INT2 interrupt request.
1 = INT2 interrupt request.

6.6 EAL GLOBAL INTERRUPT OPERATION

EAL is the global interrupt control bit. All interrupts start work after the EAL = 1 It is necessary for interrupt service request. One of the interrupt requests occurs, and the program counter (PC) points to the interrupt vector (ORG 0X03~0XEB).

0A8H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN0	EAL	-	ET2	ES0	ET1	EX1	ET0	EX0
Read/Write	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	-	0	0	0	0	0	0

Bit 7 **EAL:** Enable all interrupt control bit.
0 = Disable all interrupt function.
1 = Enable all interrupt function.

> ✷ ***Note: The EAL bit must enable during all interrupt operation.***

6.7 EXTERNAL INTERRUPT OPERATION (INT0~INT2)

Sonix provides 3 sets external interrupt sources in the micro-controller. INT0, INT1 and INT2 are external interrupt trigger sources and build in edge trigger configuration function. When the external edge trigger occurs, the external interrupt request flag will be set to "1" when the external interrupt control bit enabled. If the external interrupt control bit is disabled, the external interrupt request flag won't active when external edge trigger occurrence. When external interrupt control bit is enabled and external interrupt edge trigger is occurring, the program counter will jump to the interrupt vector (ORG 0x0003/0x0013/0x00EB) and execute interrupt service routine.

08FH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEDGE	-	-	EX2G1	EX2G0	EX1G1	EX1G0	EX0G1	EX0G0
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	R/W
After reset	-	-	1	0	1	0	1	0

Bit [1:0] **EX0G [1:0]:** External interrupt 0 trigger edge control register.
00 = Reserved.
01 = Rising edge trigger.
10 = Falling edge trigger (default).
11 = Both rising and falling edge trigger (Level change trigger).

Bit [3:2] **EX1G [1:0]:** External interrupt 1 trigger edge control register.
00 = Reserved.
01 = Rising edge trigger.
10 = Falling edge trigger (default).
11 = Both rising and falling edge trigger (Level change trigger).

Bit [5:4] **EX2G [1:0]:** External interrupt 2 trigger edge control register.
00 = Reserved.
01 = Rising edge trigger.
10 = Falling edge trigger (default).
11 = Both rising and falling edge trigger (Level change trigger).

Example: Setup INT0 interrupt request and bi-direction edge trigger.

```
            MOV      A, #03H
            MOV      PEDGE, A        ; Set INT0 interrupt trigger as bi-direction edge.

            SETB     EX0             ; Enable INT0 interrupt service
            CLR      IE0             ; Clear INT0 interrupt request flag
            SETB     EAL             ; Enable EAL
```

Example: INT0 interrupt service routine.

```
            ORG      0X0003          ; Jump to interrupt service routine address.
            JMP      ISR_INT0
ISR_ INT0:                           ; The head of interrupt service routine.
            PUSH     ACC             ; Save ACC to stack buffer.
            PUSH     PSW             ; Save PSW to stack buffer.
            …
            POP      PSW             ; Load PSW from stack buffer.
            POP      ACC             ; Load ACC from stack buffer.
            RETI                     ; End of interrupt service routine.
```

✸ ***Note:***

1. ***Before use INT0~INT2 function, user must enable INT interrupt service and EAL bit.***
2. ***Maximum external signal trigger rate is Fcpu/2.***

6.8 TIMER0/TIMER1 INTERRUPT OPERATION

When the T0/T1 counter occurs overflow, the TF0/TF1 will be set to “1” no matter the ET0/ET1 is enable or disable. If the ET0/ET1 = 0, the trigger event will make the TF0/TF1 to be “1” but the system will not enter interrupt vector. If the ET0/ET1 = 1, the trigger event will make the TF0/TF1 to be “1” but TF0 can be cleared by hardware automatically before the system enter interrupt vector. In the T0/T1 interrupt enable, TF0/TF1 doesn’t need to clear by software. Users need to care for the operation under multi-interrupt situation.

Example: T0 interrupt request setup.

```
                CLR      ET0            ; Disable T0 interrupt service
                CLR      TR0            ; Disable T0 timer
                MOV      A, #00H        ;
                MOV      TMOD, A        ; Set T0 Mode 0
                MOV      A, #00H        ; Set TH0/TL0 initial value = 00H
                MOV      TH0, A         ;
                MOV      TL0, A         ;

                SETB     ET0            ; Enable T0 interrupt service
                CLR      TF0            ; Clear T0 interrupt request flag
                SETB     TR0            ; Enable T0 timer

                SETB     EAL            ; Enable EAL
```

Example: T0 interrupt service routine.

```
                ORG      0X000B         ; Jump to interrupt service routine address.
                JMP      ISR_T0
ISR_ T0:                                ; The head of interrupt service routine.
                PUSH     ACC            ; Save ACC to stack buffer.
                PUSH     PSW            ; Save PSW to stack buffer.
                …
                POP      PSW            ; Load PSW from stack buffer.
                POP      ACC            ; Load ACC from stack buffer.
                RETI                    ; End of interrupt service routine.
```

6.9 T2 INTERRUPT OPERATION

When the T2 counter overflows, the TF2 will be set to “1” no matter the ET2 is enable or disable. If the ET2=1 and the trigger event TF2 is set to be “1”. As the result, the system will execute the interrupt vector. If the ET2 = 0, the trigger event TF2 is still set to be “1”. Moreover, the system won’t execute interrupt vector. TF2 flag must be clear by software. Users need to be cautious with the operation under multi-interrupt situation.

Example: T2 interrupt request setup.

```
                CLR         ET2             ; Disable T2 interrupt service
                MOV         A, #00H         ; Set TH2/TL2 initial value = 00H
                MOV         TH2, A          ;
                MOV         TL2, A          ;

                SETB        ET2             ; Enable T2 interrupt service
                CLR         TF2             ; Clear T2 interrupt request flag
                MOV         T2CON, #01H     ; Enable T2 timer

                SETB        EAL             ; Enable EAL
```

Example: T2 interrupt service routine.

```
                ORG         0X002B          ; Jump to interrupt service routine address.
                JMP         ISR_T2
ISR_ T2:                                    ; The head of interrupt service routine.
                PUSH        ACC             ; Save ACC to stack buffer.
                PUSH        PSW             ; Save PSW to stack buffer.
                CLR         TF2             ; Clear T2 interrupt request flag
                …
                POP         PSW             ; Load PSW from stack buffer.
                POP         ACC             ; Load ACC from stack buffer.
                RETI                        ; End of interrupt service routine.
```

6.10 UART INTERRUPT OPERATION

When the UART transmitter successfully, the TI0/RI0 will be set to “1” no matter the ES0 is enable or disable. If the ES0 and the trigger event TI0/RI0 is set to be “1”. As the result, the system will execute the interrupt vector. If the ES0 = 0, the trigger event TI0/RI0 is still set to be “1”. Moreover, the system won’t execute interrupt vector. Users need to be cautious with the operation under multi-interrupt situation.

Example: UART receive and transmit interrupt request setup.

```
                CLR       TI0              ; Clear UART transmit interrupt request flag
                CLR       RI0              ; Clear UART receive interrupt request flag
                SETB      REN0             ; Enable UART transmit function

                SETB      ES0              ; Enable UART receive and transmit interrupt service
                SETB      EAL              ; Enable EAL
```

Example: UART receive and transmit interrupt service routine.

```
                ORG       0X0023           ; Jump to interrupt service routine address.
                JMP       ISR_ UART
ISR_ UART:                                 ; The head of interrupt service routine.
                PUSH      ACC              ; Save ACC to stack buffer.
                PUSH      PSW              ; Save PSW to stack buffer.

                JB        RI0, ISR_ RX     ; Identify the interrupt event

ISR_ TX:
                CLR       TI0              ; Clear UART transmit interrupt request flag
                …
                JMP       ISR_ END

ISR_ RX:
                CLR       RI0              ; Clear UART receive interrupt request flag
                …

ISR_ END:
                POP       PSW              ; Load PSW from stack buffer.
                POP       ACC              ; Load ACC from stack buffer.
                RETI                       ; End of interrupt service routine.
```

6.11 SPI INTERRUPT OPERATION

When the SPI converting successfully, the SPIF will be set to "1" no matter the ESPI is enable or disable. If the ESPI and the trigger event SPIF is set to be "1". As the result, the system will execute the interrupt vector. If the ESPI = 0, the trigger event SPIF is still set to be "1". Moreover, the system won't execute interrupt vector. Users need to be cautious with the operation under multi-interrupt situation.

Example: SPI interrupt request setup.

```
                SETB      ESPI            ; Enable UART SPI interrupt service
                SETB      EAL             ; Enable EAL
```

Example: SPI interrupt service routine.

```
                ORG       0X004B          ; Jump to interrupt service routine address.
                JMP       ISR_ SPI
ISR_ SPI:                                 ; The head of interrupt service routine.
                PUSH      ACC             ; Save ACC to stack buffer.
                PUSH      PSW             ; Save PSW to stack buffer.

                MOV       A, SPSTA        Clear SPI interrupt request flag


                POP       PSW             ; Load PSW from stack buffer.
                POP       ACC             ; Load ACC from stack buffer.
                RETI                      ; End of interrupt service routine.
```

6.12 PWM INTERRUPT OPERATION (PW1~PW3)

Sonix provides three PWM (PW1/PW2/PW3) with interrupt function in the micro-controller. When PW1/PW2/PW3 timer overflow occurrence, PW1~PW3 overflow condition is conformed and PWM1F/PWM2F/PWM3F set as “1” no matter the EPWM1/EPWM2/EPWM3 is enable or disable. If the EPWM1/EPWM2/EPWM3 and the trigger event PWM1F/PWM2F/ PWM3F is set to be “1”. As the result, the system will execute the interrupt vector. Moreover, the system won’t execute interrupt vector. Users need to be cautious with the operation under multi-interrupt situation.

Example: PW1 interrupt request setup.

```
                MOV     IEN4, #80H      ; Enable PW1 interrupt service and clear PW1interrupt
                                        ; request flag
                MOV     P1OC,#20H       ; Enable PW1
                SETB    EAL             ; Enable EAL
```

Example: PW1 interrupt service routine.

```
                ORG     0X0083          ; Jump to interrupt service routine address.
                JMP     ISR_ PW1
ISR_ PW1:                               ; The head of interrupt service routine.
                PUSH    ACC             ; Save ACC to stack buffer.
                PUSH    PSW             ; Save PSW to stack buffer.

                MOV     A, IEN4         Clear PW1 interrupt request flag
                ANL     A, #F7H
                MOV     IEN4, A
                …
                POP     PSW             ; Load PSW from stack buffer.
                POP     ACC             ; Load ACC from stack buffer.
                RETI                    ; End of interrupt service routine.
```

6.13 ADC INTERRUPT OPERATION

When the ADC converting successfully, the ADCF will be set to “1” no matter the EADC is enable or disable. If the EADC and the trigger event ADCF is set to be “1”. As the result, the system will execute the interrupt vector. If the EADC = 0, the trigger event ADCF is still set to be “1”. Moreover, the system won’t execute interrupt vector even when the EADC is set to be “1”. Users need to be cautious with the operation under multi-interrupt situation.

Example: ADC interrupt request setup.

```
                …
                MOV         IEN2,#08H           ; Enable ADC interrupt service and
                SETB        EAL                 ; Enable EAL
```

Example: ADC interrupt service routine.

```
                ORG         0X009B              ; Jump to interrupt service routine address.
                JMP         ISR_ ADC
ISR_ ADC:                                       ; The head of interrupt service routine.
                PUSH        ACC                 ; Save ACC to stack buffer.
                PUSH        PSW                 ; Save PSW to stack buffer.

                MOV         A, IRCON2           Clear ADC interrupt request flag
                ANL         A, #0FBH
                MOV         IRCON2, A
                …
                POP         PSW                 ; Load PSW from stack buffer.
                POP         ACC                 ; Load ACC from stack buffer.
                RETI                            ; End of interrupt service routine.
```

6.14 COMPARATOR INTERRUPT OPERATION

Sonix provides two comparator with interrupt function in the micro-controller. The comparator interrupt trigger edge direction depends on CM0G/CM1G. When the comparator output status transition occurs, the comparator interrupt request flag will be set to “1” no matter the comparator interrupt control bit status. The comparator interrupt flag doesn’t active only when comparator control bit is disabled. When comparator interrupt control bit is enabled and comparator interrupt edge trigger is occurring, the program counter will jump to the interrupt vector (ORG A3H/ABH) and execute interrupt service routine.

Example: Comparator interrupt request setup.

```
                    …

                    MOV         IEN2, #10H        ; Enable Comparator0 interrupt service
                    SETB        EAL               ; Enable EAL
```

Example: Comparator interrupt service routine.

```
                    ORG         0X00A3               ; Jump to interrupt service routine address.
                    JMP         ISR_ CMP0
ISR_ CMP0:                                           ; The head of interrupt service routine.
                    PUSH        ACC                  ; Save ACC to stack buffer.
                    PUSH        PSW                  ; Save PSW to stack buffer.

                    MOV         A, IRCON2            Clear comparator 0 interrupt request flag
                    ANL         A, #11110111B
                    MOV         IRCON2, A
                    …
                    POP         PSW                  ; Load PSW from stack buffer.
                    POP         ACC                  ; Load ACC from stack buffer.
                    RETI                             ; End of interrupt service routine.
```

6.15 MULTI-INTERRUPT OPERATION

Under certain condition, the software designer uses more than one interrupt requests. Processing multi-interrupt request requires setting the priority of the interrupt requests. The request flags of interrupts are controlled by the interrupt event. Nevertheless, the request flag “1” doesn’t mean the system will execute the interrupt vector. In addition, which means the request flags can be set “1” by the events without enable the interrupt. Once the event occurs, the request flag will be logic “1”. The request flag and its trigger event relationship is as the below table.

Interrupt vector	*Interrupt Name*	*Trigger condition*
0x03	IE0	INT0 control by PEDGE.
0x0B	TF0	Timer 0 overflow
0x13	IE1	INT1 control by PEDGE.
0x1B	TF1	Timer 1 overflow
0x23	TI0/RI0	UART transmission end
0x2B	TF2/TF2RL	Timer 2 overflow or external reload signal
0x43	SI	I2C transmitter successfully.
0x4B	SPIF	SPI transmitter successfully
0x53	TF2C0	Timer 2 counter equal CRC.
0x5B	TF2C1	Timer 2 counter equal to CC1.
0x63	TF2C2	Timer 2 counter equal to CC2.
0x6B	TF2C3	Timer 2 counter equal to CC3.
0x83	PWM1F	PWM1 overflow
0x8B	PWM2F	PWM2 overflow
0x93	PWM3F	PWM3 overflow
0x9B	ADCF	ADC transfer end
0xA3	CMP0F	Comparator 0 transfer end
0xAB	CMP1F	Comparator 1 transfer end
0xEB	IE2	INT2 control by PEDGE.

For multi-interrupt conditions, two things need to be taking care of. One is that it is multi-vector and each of interrupts points to unique vector. Two is users have to define the interrupt vector. The following example shows the way to define the interrupt vector in the program memory.

- **Example: Check the interrupt request under multi-interrupt operation .**

```
                 ORG       0X0003         ; Jump to interrupt service routine address.
                 JMP       ISR_INT0
                 ORG       0X000B
                 JMP       ISR_T0
                 ORG       0X0013
                 JMP       ISR_INT1
                 …
                 …
                 ORG       0X0023
                 JMP       ISR_UART
                 ORG       0X00EB
                 JMP       ISR_INT2
ISR_ INT0:                                ; The head of interrupt service routine.

                 RETI                     ; End of interrupt service routine.
ISR_T0:                                   ;

                 RETI                     ; End of interrupt service routine.
                 …
                 …
ISR_INT2                                  ;

                 RETI                     ; End of interrupt service routine.
```

7 IO PORT

7.1 OVERVIEW

The micro-controller builds in 46 pin I/O. Most of the I/O pins are mixed with analog pins and special function pins. The I/O shared pin list is as following.

PIN NAME	TYPE	Name	TYPE	Shared Pin Control Condition
P0.0	I/O	XIN	AC	High_CLK code option = IHRC_32M_RTC, 4M, 12M, Ext_CLK
P0.1	I/O	XOUT	AC	High_CLK code option = IHRC_32M_RTC, 4M, 12M
P0.2	I/O	RST	DC	Reset_Pin code option = Reset
P0.5	I/O	UTX	DC	S0BUF register write operation.
P0.6	I/O	URX	DC	REN0=1
P0.7	I/O	SSN	DC	SPEN=1
P1.0	I/O	T2COM0	DC	COCAH0, COCAL0=10
P1.1	I/O	MOSI	DC	SPEN=1
		T2COM1	DC	COCAH1, COCAL1=10
P1.2	I/O	MISO	DC	SPEN=1
		T2COM2	DC	COCAH2, COCAL2=10
P1.3	I/O	SCK	DC	SPEN=1
		T2COM3	DC	COCAH3, COCAL3=10
P1.4	I/O	SCL	DC	ENS1=1
P1.5	I/O	SDA	DC	ENS1=1
P2.1	I/O	INT0	DC	EX0=1
P2.2	I/O	INT1	DC	EX1=1
		T2RL	DC	T2R[1:0]=10/11
P2.3	I/O	INT2	DC	EX2=1
		T2	DC	T2I[1:0]=01/11
P2.4	I/O	CM1O	AC	CM1OEN=1
		CT23	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=10111b
P2.5	I/O	CM1P	AC	CM1EN=1
		CT22	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=10110b
P2.6	I/O	CM1N	AC	CM1EN=1
		CT21	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=10101b
P2.7	I/O	CM0O	AC	CM0OEN=1
		CT20	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=10100b
P3.0	I/O	OP0N	AC	OP0EN=1
		CT12	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=01100b
P3.1	I/O	OP0P	AC	OP0EN=1
		CT13	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=01101b
P3.2	I/O	OP0O	AC	OP0EN=1
		CT14	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=01110b
P3.3	I/O	OP1N	AC	OP1EN=1
		CT15	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=01111b
P3.4	I/O	OP1P	AC	OP1EN=1
		T2CC0	DC	COCAH0 COCAL0=01/11
		CT16	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=10000b
P3.5	I/O	OP1O	AC	OP1EN=1
		CT17	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=10001b
		T2CC1	DC	COCAH1, COCAL1=01/11
P3.6	I/O	CM0N	AC	CM0EN=1
		CT18	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=10010b
		T2CC2	DC	COCAH2, COCAL=01/11
P3.7	I/O	CM0P	AC	CM0EN=1
		CT19	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=10011b
		T2CC3	DC	COCAH3, COCAL3=01/11

P4.0	I/O	AIN0	AC	ADENB=1,GCHS=1,CHS[4:0]=00000b
		CT0	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=00000b
		SWAT	DC	OCDS mode.
P4 [5:1]	I/O	AIN[5:1]	AC	ADENB=1,GCHS=1,CHS[4:0]=00001b~00101b
		CT[5:1]	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=00001b~00101b
P4.6	I/O	AIN6	AC	ADENB=1,GCHS=1,CHS[4:0]=00110b
		CT6	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=00110b
		PWM10	DC	PW1EN=1, PWCH10=1
P4.7	I/O	AIN7	AC	ADENB=1,GCHS=1,CHS[4:0]=00111b
		CT7	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=00111b
		PWM11	DC	PW1EN=1, PWCH11=1
P5.0	I/O	AIN8	AC	ADENB=1,GCHS=1,CHS[4:0]=01000b
		CT8	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=01000b
		PWM20	DC	PW2EN=1, PWCH20=1
P5.1	I/O	AIN9	AC	ADENB=1,GCHS=1,CHS[4:0]=01001b
		CT9	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=01001b
		PWM21	DC	PW2EN=1, PWCH21=1
P5.2	I/O	AIN10	AC	ADENB=1,GCHS=1,CHS[4:0]=01010b
		CT10	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=01010b
		PWM30	DC	PW3EN=1, PWCH30=1
P5.3	I/O	AIN11	AC	ADENB=1,GCHS=1,CHS[4:0]=01011b
		CT11	AC	ADENB=1,GCHS=1, TCHEN=1, CHS[4:0]=01011b
		PWM31	DC	PW3EN=1, PWCH31=1
P5.4	I/O	AVREFH	AC	EVHENB=1
		CA1	AC	TCHEN=1
P5.5	I/O	AVREFL	AC	EVLENB=1
		CA2	AC	TCHEN=1

7.2 I/O PORT MODE

The port direction is programmed by PnM register. When the bit of PnM register is "0", the pin is input mode. When the bit of PnM register is "1", the pin is output mode.

0F9H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0FAH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0FBH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2M	P27M	P26M	P25M	P24M	P23M	P22M	P21M	P20M
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0FCH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3M	P37M	P36M	P35M	P34M	P33M	P32M	P31M	P30M
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0FDH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4M	P47M	P46M	P45M	P44M	P43M	P42M	P41M	P40M
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0FEH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P5M	-	-	P55M	P54M	P53M	P52M	P51M	P50M
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	R/W
After reset	-	-	0	0	0	0	0	0

Bit [7:0] **PnM[7:0]:** Pn mode control bits. (n = 0~5).
0 = Pn is input mode.
1 = Pn is output mode.

➢ **Example: I/O mode selecting**

```
CLR          P0M            ; Set all ports to be input mode.
CLR          P1M
CLR          P2M

MOV          A, #0FFH       ; Set all ports to be output mode.
MOV          P0M, A
MOV          P1M,A
MOV          P2M, A

MOV          A, #00H        ; Set P4.0 to be input mode.
MOV          P4M, A

MOV          A, #01H        ; Set P4.0 to be output mode.
MOV          P4M, A
```

7.3 I/O PULL UP REGISTER

The I/O pins build in internal pull-up resistors and only support I/O input mode. The port internal pull-up resistor is programmed by PnUR register. When the bit of PnUR register is “0”, the I/O pin’s pull-up is disabled. When the bit of PnUR register is “1”, the I/O pin’s pull-up is enabled.

0F1H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0UR	P07R	P06R	P05R	P04R	P03R	P02R	P01R	P00R
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0F2H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1UR	P17R	P16R	P15R	P14R	P13R	P12R	P11R	P10R
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0F3H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2UR	P27R	P26R	P25R	P24R	P23R	P22R	P21R	P20R
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0F4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3UR	P37R	P36R	P35R	P34R	P33R	P32R	P31R	P30R
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0F5H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4UR	P47R	P46R	P45R	P44R	P43R	P42R	P41R	P40R
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0F6H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P5UR	-	-	P55R	P54R	P53R	P52R	P51R	P50R
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	R/W
After reset	-	-	0	0	0	0	0	0

- **Example: I/O Pull up Register**

```
MOV          A, #0FFH          ; Enable Port0, 1, 2 Pull-up register,
MOV          P0UR, A           ;
MOV          P1UR,A
MOV          P2UR, A
```

7.4 I/O DATA REGISTER

080H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	P07	P06	P05	P04	P03	P02	P01	P00
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

090H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1	P17	P16	P15	P14	P13	P12	P11	P10
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

0A0H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2M	P27	P26	P25	P24	P23	P22	P21	P20
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

0B0H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3M	P37	P36	P35	P34	P33	P32	P31	P30
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

0E8H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4	P47	P46	P45	P44	P43	P42	P41	P40
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

0F8H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P5	-	-	P55	P54	P53	P52	P51	P50
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	R/W
After reset	-	-	1	1	1	1	1	1

- **Example: Read data from input port.**

```
MOV        A, P0        ; Read data from Port 0
MOV        A, P1        ; Read data from Port 1
MOV        A, P2        ; Read data from Port 2
```

- **Example: Write data to output port.**

```
MOV        A, #0FFH     ; Write data FFH to all Port.
MOV        P0, A
MOV        P1, A
MOV        P2, A
```

- **Example: Write one bit data to output port.**

```
SETB       P40          ; Set P4.0 and P5.3 to be “1”.
SETB       P53

CLR        P4.0         ; Set P4.0 and P5.3 to be “0”.
CLR        P5.3
```

7.5 ADC/CMP/OP SHARE PIN

The Port 2, Port 3, Port4 and Port 5 are shared with ADC/OP/CMP input function and no Schmitt trigger structure. These pins of Port 2, Port 3, Port4 and Port 5 are digital I/O pins. Connect an analog signal to COMS digital input pin, especially the analog signal level is about 1/2 VDD will cause extra current leakage. In the stop mode, the above leakage current will be a big problem. Unfortunately, if users connect more than one analog input signal to Port 2, Port 3, Port4 and Port 5 will encounter above current leakage situation. P2CON is Port2 Configuration register. P3CON is Port3 Configuration register. P4CON is Port4 Configuration register. P5CON is Port5 Configuration register.Write “1” into P2CON.n, P3CON.n, P4CON.n or P5CON.n will configure related Port 2, Port 3, Port4 or Port 5 pin will be set as input mode and disable pull-up resistor

09EH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2CON	P2CON7	P2CON6	P2CON5	P2CON4	-	-	-	-
Read/Write	R/W	R/W	R/W	R/W	-	-	-	-
After reset	0	0	0	0	-	-	-	-

Bit [7:4] **P2CON [7:4]:** P2 configuration control bits.
0 = P2.n can be analog input pin (OP/CMP input pin) or digital I/O pin.
1 = P2.n is pure analog input pin and can’t be a digital GPIO pin.

09FH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3CON	P3CON7	P3CON6	P3CON5	P3CON4	P3CON3	P3CON2	P3CON1	P3CON0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **P3CON [7:0]:** P3 configuration control bits.
0 = P3.n can be analog input pin (OP/CMP input pin) or digital I/O pin.
1 = P3.n is pure analog input pin and can’t be a digital GPIO pin.

0D6H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4CON	P4CON7	P4CON6	P4CON5	P4CON4	P4CON3	P4CON2	P4CON1	P4CON0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

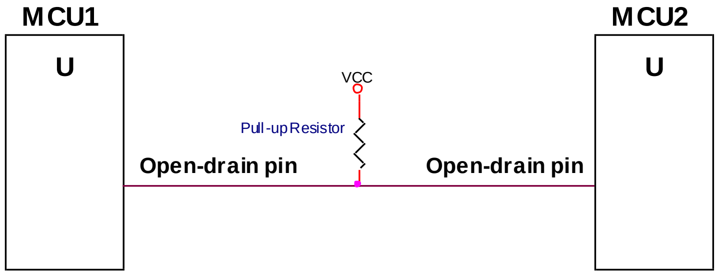
Bit [7:0] **P4CON [7:0]:** P4 configuration control bits.
0 = P4.n can be analog input pin (ADC input pin) or digital I/O pin.
1 = P4.n is pure analog input pin and can’t be a digital GPIO pin.

0D7H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P5CON	-	-	P5CON5	P5CON4	P5CON3	P5CON2	P5CON1	P5CON0
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	R/W
After reset	-	-	0	0	0	0	0	0

Bit [5:0] **P5CON [5:0]:** P5 configuration control bits.
0 = P5.n can be analog input pin (ADC input pin) or digital I/O pin.
1 = P5.n is pure analog input pin and can’t be a digital GPIO pin.

7.6 OPEN-DRAIN REGISTER

P0.5, P0.6, P1.1~P1.3 built in open-drain function. These pins must be set as output mode when enable open-drain function. Open-drain external circuit is as following.



The pull-up resistor is necessary. Open-drain output high is driven by pull-up resistor. Output low is sunken by MCU's pin.

0E4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1OC	PW3EN	PW2EN	PW1EN	P06OC	P05OC	P13OC	P12OC	P11OC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [4:0] **P11OC/P12OC/P13OC/P05OC/P06OC:** Open drain control bit.
0 = Disable open-drain structure.
1 = Enable open-drain structure.

8 Multiplication Division Unit (MDU)

8.1 OVERVIEW

The MDU (Multiplication Division Unit) is an on-chip arithmetic co-processor which enables the MCU to perform additional extended arithmetic operations. This unit provides 32-bit division, 16-bit multiplication, shift and normalize operations. All operations are unsigned integer operations.

The MDU is handled by seven registers, which are memory mapped as Special Function Registers. The arithmetic unit allows to perform operations concurrently to and independent of the CPU's activity.

Operands and results are stored in "MD0" "MD1" "MD2" "MD3" "MD4" "MD5" registers. The MDU is controlled by the "ARCON" register. Any calculation of the MDU overwrites its operands.

8.2 MULTIPLICATION/DIVISION COMTROL REGISTERS

0E9H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MD0	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0EAH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MD1	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0EBH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MD2	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0ECH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MD3	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0EDH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MD4	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0EEH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MD5	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Multiplication/Division Registers – MD0, MD1, MD2, MD3, MD4, MD5

The MD0, MD1, MD2, MD3, MD4, MD5 are registers used in the MDU operation.

0EFH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ARCON	MDEF	MDOV	SLR	SC4	SC3	SC2	SC1	SC0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

The ARCON (Arithmetic Control Register) register controls the operation of MDU and informs about its current state.

Bit 7 **MDEF:** MDU Error flag MDEF.
Indicates an improperly performed operation (when one of the arithmetic operations has been restarted or interrupted by a new operation).

The “MDEF” error flag indicates an improperly performed operation (when one of the arithmetic operations has been restarted or interrupted by a new operation). The error flag mechanism is automatically enabled with the first write operation to “MD0” and disabled with the final read instruction from “MD3” (multiplication or shift/normalize) or “MD5” (division) in phase three.
The error flag is set when:

- There is a write access to “MDx” registers (any of “MD0” “MD1” “MD2” “MD3” “MD4” “MD5” and “ARCON”) during phase two of MDU operation (restart or calculations interrupting)
- There is a read access to one of “MDx” registers during phase two of MDU operation when the error flag mechanism is enabled. In such condition error flag is set but the calculation is not interrupted.

The error flag is reset only after read access to “ARCON” register. The error flag is read only.

Bit 6 **MDOV:** MDU Overflow flag MDOV.
Overflow occurrence in the MDU operation.

The “MDOV” overflow flag is set when one of the following conditions occurs:

- Division by zero
- Multiplication with a result greater than 0000 FFFFh
- Start of normalizing if the most significant bit of “MD3” is set (“MD3.7” = ‘1’)

Any operation of the MDU that does not match the above conditions clears the overflow flag. Note that the overflow flag is exclusively controlled by hardware. It cannot be written.

Bit 5 **SLR:** Shift direction.
0 = Shift left operation.
1 = Shift right operation.

Bit [4:0] **SC [4:0]:** Shift counter.
When set to all ‘0’s, normalize operation is selected. After normalization, the “SC [4:0]” contains the number of normalizing shifts performed.
When at least one of these bits is set “1” shift operation is selected. The number of shifts performed is determined by the number written to “SC [4:0]”, where “SC.4” is the MSB.

8.3 MDU OPERATION DESCRIPTION

The operation of the MDU consists of three phases:
(a) Loading the MDx Registers
The type of calculation the MDU has to perform is selected in accordance with the order in which the “MDx” registers are written.

Operating	32bit /16bit	16bit / 16bit	16bit x16bit	Shifting/ Normalizing
First write	MD0 Dividend_L	MD0 Dividend_L	MD0 Multiplicand_L	MD0 LSB
	MD1 Dividend_ML	MD1 Dividend_H	MD4 Multiplier_L	MD1
	MD2 Dividend_MH			MD2
	MD3 Dividend_H			MD3 MSB
	MD4 Divisor_L	MD4 Divisor_L	MD1 Multiplicand_H	
Last write	MD5 Divisor_H	MD5 Divisor_H	MD5 Multiplier_H	ARCON

- 32 bit / 16 bit: Dividend [31:0] = MD3, MD2, MD1, MD0; Divisor [15:0] = MD5, MD4
- 16 bit / 16 bit: Dividend [15:0] = MD1, MD0; Divisor [15:0] = MD5, MD4
- 16 bit x 16 bit: Multiplicand [15:0] = MD1, MD0; Multiplier [15:0] = MD5, MD4

Writing to “MD0” is the first transfer to be done in order to start any operation. Next write operations must be done as shown in “Table. MDU Registers Write Sequence” to determine appropriate MDU operation. Last writing finally starts selected operation.
The SFR Control detects some of the above sequences and passes control to the Arithmetic Operation. When a write access occurs to “MD2” or “MD3” between write accesses to “MD0” and finally to “MD5”, then “32 bit/16 bit division” will be selected. In other case when a write access occurs to “MD4” or “MD1” before final writing to “MD5”, then “16bit/16 bit division or multiplication” will be selected. Writing to “MD4” selects “16 bit/16 bit division” and writing to “MD1” selects “multiplication”. After reset, the MDU is set for “multiplication” as a default operation. Any write access to “ARVON” forces “shift or normalize” operation.

(b) Executing Calculation

During executing operation, the MDU works on its own in parallel to the CPU.

Operation	Number of clock cycles	
Division 32bit / 16bit	17 clock cycles	
Division 16bit / 16bit	9 clock cycles	
Multiplication	11 clock cycles	
Shift	Min 3 clock cycles (SC[4:0]=01H)	Max 18 clock cycles(SC[4:0]=1FH)
Normalize	Min 4 clock cycles (SC[4:0]<=01H)	Max 19 clock cycles (SC[4:0]<=1FH)

(c) Reading the Result from the MDx Registers

Operating	32bit /16bit	16bit / 16bit	16bit x16bit	Shifting/ Normalizing
First read	MD0 Quotient _L	MD0 Quotient_L	MD0 Product_L	MD0 LSB
	MD1 Quotient _ML	MD1 Quotient_H	MD1 Product _ML	MD1
	MD2 Quotient _MH		MD2 Product _MH	MD2
	MD3 Quotient _H			
	MD4 Remainder_L	MD4 Remainder_L		
Last read	MD5 Remainder_H	MD5 Remainder_H	MD3 Product _H	MD3 MSB

- 32 bit / 16 bit: Quotient [31:0] = MD3, MD2, MD1, MD0; Remainder [15:0] = MD5, MD4
- 16 bit / 16 bit: Quotient [15:0] = MD1, MD0; Remainder [15:0] = MD5, MD4
- 16 bit x 16 bit: Product [31:0] = MD3, MD3, MD1, MD0

The Read-out sequence of the first “MDx” registers is not critical but the last read (from “MD5” - division and “MD3” - multiplication, shift or normalize) determines the end of a whole calculation (end of phase three).

(d) Normalizing

All leading zeroes of 32-bit integer variable stored in “MD0” “MD1” “MD2” ”MD3” registers (the latter contains the most significant byte) are removed by shift left operations. The whole operation is completed when the MSB (most significant bit) of “MD3” register contains a ’1’. After normalizing, bits “SC [4(MSB):0(LSB)]” contain the number of shift left operations, which were done.

(e) Shifting

In shift operation, 32-bit integer variable stored in “MD0” “MD1” “MD2” ”MD3” registers (the latter contains the most significant byte) is shifted left or right by a specified number of bits. The “SLR” bit (“ARCON.5”) defines the shift direction, and bits “SC [4:0]” specify the shift count (which must not be 0). During shift operation, zeroes come into the left end of “MD3” for shifting right or right end of the “MD0” for shifting left.

9 TIMERS

9.1 WATCHDOG TIMER

The watchdog timer (WDT) is a binary up counter designed for monitoring program execution. If the program goes into the unknown status by noise interference, watchdog timer overflow signal raises and resets MCU. Watchdog timer clock source is internal low-speed oscillator 32KHz RC type and through programmable pre-scaler controlled by WDT_CLK code option.

Watchdog timer interval time = 1024 * 1/ (Internal Low-Speed oscillator frequency/WDT Pre-scalar) …sec
= 1024 / (32KHz/WDT Pre-scaler) …sec

Internal low-speed oscillator	WDT pre-scaler	Watchdog interval time
Flosc=32KHz	Flosc/1	1024/(32000/1)=32ms
	Flosc/2	1024/(32000/2)=64ms
	Flosc/4	1024/(32000/4)=128ms
	Flosc/8	1024/(32000/8)=256ms

The watchdog timer has three operating options controlled “WatchDog” code option.

- **Disable:** Disable watchdog timer function.
- **Enable:** Enable watchdog timer function. Watchdog timer actives in normal mode and slow mode. In power down mode and green mode, the watchdog timer stops.
- **Always_On:** Enable watchdog timer function. The watchdog timer actives and not stop in power down mode and green mode.

> ✸ ***Note:*** **In high noisy environment, the “Always_On” option of watchdog operations is the strongly recommendation to make the system reset under error situations and re-start again.**

Watchdog clear is controlled by WDTR register. Moving **0x5A** data into WDTR is to reset watchdog timer.

086H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTR	WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0
Read/Write	W	W	W	W	W	W	W	W
After reset	0	0	0	0	0	0	0	0

Example: An operation of watchdog timer is as following. To clear the watchdog timer counter in the top of the main routine of the program.

```
Main:
            MOV           A, #5AH          ; Clear the watchdog timer.
            MOV           WDTR, A
            …
            CALL          SUB1
            CALL          SUB2
            …
            JMP           MAIN
```

Watchdog timer application note is as following.

- Before clearing watchdog timer, check I/O status and check RAM contents can improve system error.
- Don’t clear watchdog timer in interrupt vector and interrupt service routine. That can improve main routine fail.
- Clearing watchdog timer program is only at one part of the program. This way is the best structure to enhance the watchdog timer function.

➢ **Example: An operation of watchdog timer is as following. To clear the watchdog timer counter in the top of the main routine of the program.**

```
Main:
                …                                   ; Check I/O.
                …                                   ; Check RAM
Err:            JMP SUBERR                          ; I/O or RAM error. Program jump here and don’t
                                                    ; clear watchdog. Wait watchdog timer overflow to reset IC.

Correct:                                            ; I/O and RAM are correct. Clear watchdog timer and
                                                    ; execute program.
                MOV             A, #5AH             ; Clear the watchdog timer.
                B0MOV           WDTR, A
                …
                CALL            SUB1
                CALL            SUB2
                …
                …
                …
                JMP             MAIN
```

9.2 T0 16-BIT TIMER

9.2.1 OVERVIEW

The T0 timer is a 16-bit binary up timer with 4 operating mode. If T0 timer occurs an overflow, it will continue counting and issue a time-out signal to indicate T0 time out event. T0 clock source includes internal and external types. Internal clock source is from instruction cycle (Fcpu) or internal oscillator clocks (Fhosc), and external clock source is from external oscillator (i.g., 12M X'tal, 4M X'tal, 32K X'tal, ERC, ExtCLK). When clock source is from instruction cycle (Fcpu), timer 0 is incremented every 12 clock cycles, which means that it counts up after every 12 periods of the clock signal. When clock source is from internal oscillator clocks (Fhosc) or external oscillator, both clock sources are through the T0 prescaler to adjust different periods. All clock sources can be gated by external signal from P2.1/INT0 if T0GATE is enabled. When INT0 is low, the T0 timer is stopped. The T0 builds in idle mode wake-up function when interrupt function is enabled. When T0 timer overflow occurs under idle mode, the system will be waked-up to last operating mode. The main purposes of the T0 timer are as following.

- **13-bit programmable up counting timer (Mode0):** Generate time-out at specific time intervals based on the selected clock frequency.
- **16-bit programmable up counting timer (Mode1):** Generate time-out at specific time intervals based on the selected clock frequency.
- **8-bit programmable up counting timer with auto-reload (Mode2):** Generate time-out at specific time intervals based on the selected clock frequency.
- **Tw0 8-bit programmable up counting timer (Mode3):** Generates interrupts at specific time intervals based on the selected clock frequency.
- **Interrupt function:** T0 timer function supports interrupt function. When T0 timer occurs overflow, the TF0 actives and the system points program counter to interrupt vector to do interrupt sequence.
- **Idle mode function:** T0 keeps running in idle mode and can wake-up from idle mode as TR0 = 1 when interrupt function is enabled. System will be wake-up after T0 timer overflow occurrence.

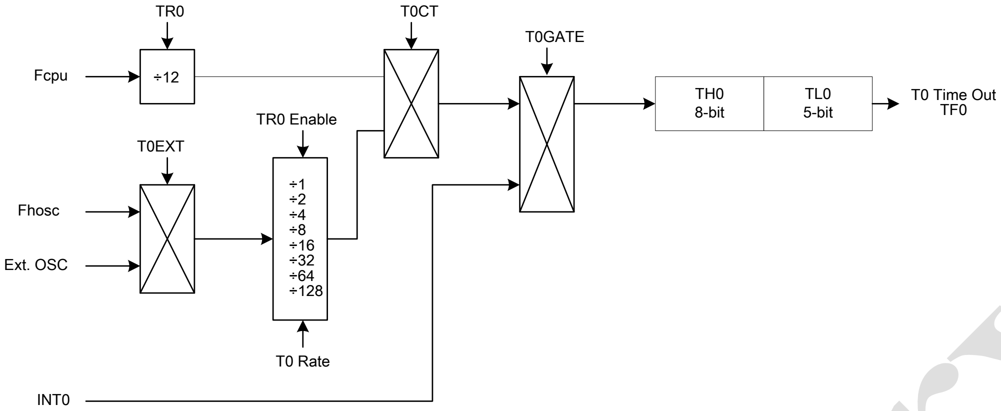
9.2.2 T0 TIMER OPERATION

T0 timer is controlled by TR0 bit. When TR0=0, T0 timer stops. When TR0=1, T0 timer starts to count. Before enabling T0 timer, setup T0 timer's configurations to select timer function modes, e.g. clock source, timer mode, interrupt function…T0 timer counter is controlled by TH0 and TL0 register.TL0/TH0 increases "1" by timer clock source. When T0 overflow event occurs, TF0 flag is set as "1" to indicate overflow. The overflow condition is T0 counter count from full scale to zero scale. TF0 can be cleared by program or cleared by hardware automatically (if interrupt function enabled). If T0 timer interrupt function is enabled (ET0=1), the system will execute interrupt procedure. The interrupt procedure is system program counter points to interrupt vector (ORG 000BH) and executes interrupt service routine after T0 overflow occurrence. TF0 is cleared by hardware automatically in interrupt procedure. T0 timer builds in idle mode wake-up function when interrupt function is enabled. T0 timer keeps counting in idle mode. If interrupt function is enabled, when T0 timer overflow occurs, the system will be waked-up from idle mode to normal mode, and execute T0 interrupt operation. If interrupt function is disabled, the idle mode wake-up function is disabled.

T0 provides different clock sources to implement different applications and configurations. T0 clock source includes Fcpu (instruction cycle), Fhosc (high speed oscillator) and external oscillator (i.g., 12M X'tal, 4M X'tal, 32K X'tal, ERC, ExtCLK) controlled by T0CT and T0EXT bits. If T0CT=0, T0 clock source is Fcpu/12. If T0CT=1, T0 clock source is Fhosc (T0EXT=0) or external oscillator (T0EXT=1) through T0rate[2:0] pre-scalar to decide Fhosc/1~Fhosc/128 or Ext. OSC/1~ Ext. OSC /128. T0GATE bit controls the clock source is gating by external signal from P2.1/INT0. When INT0 is low, the T0 is stopped.

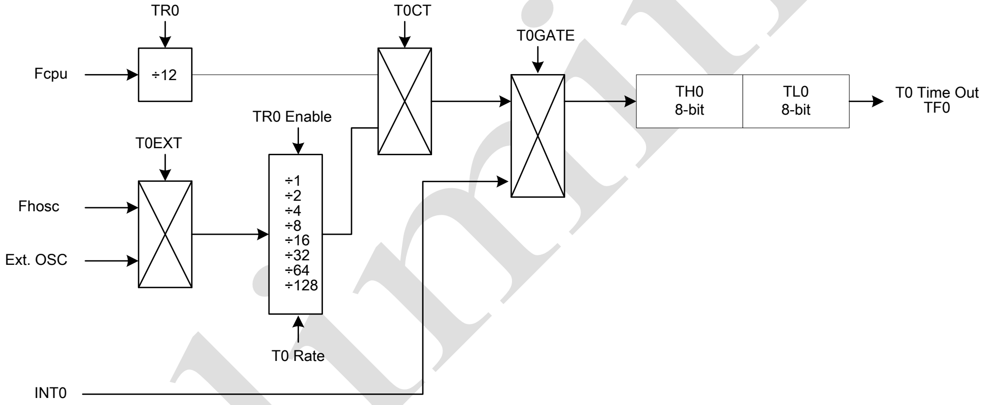
T0 timer can operate as 4 operating mode that is controlled by T0M[1:0].

- **Mode0: 13-bit Timer (T0M[1:0]=00)**



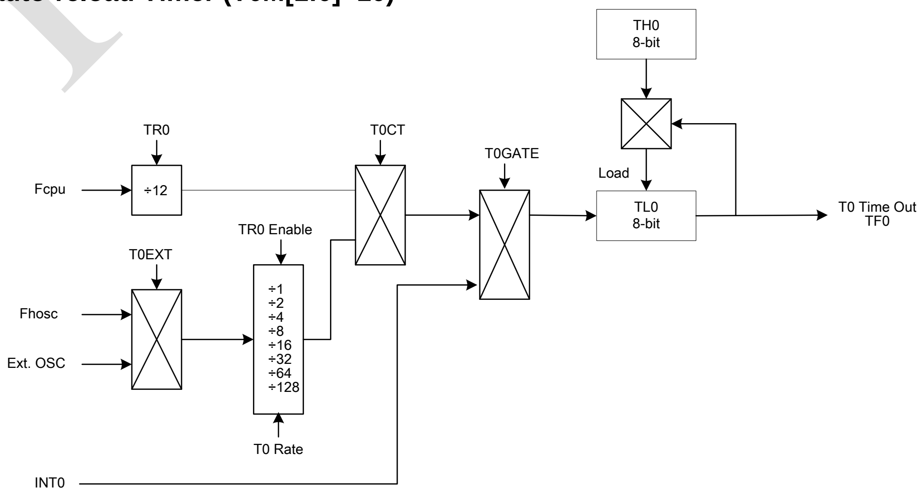
In mode 0, T0 operate as 13-bit timer. 13-bit timer counter is controlled by TH0 and TL0 register. TH0 holds the higher 8-bit and TL0 holds the lower 5-bit. The upper 3-bit of TL0 is useless. The overflow condition is T0 counter count from full scale (0x1FFF) to zero scale (0x0000).When T0 overflow event occurs, TF0 flag is set as "1" to indicate overflow. In mode 0, T0 counter doesn't build in auto-reload function. Set the T0 interval time through setting TH0/ TL0 by program and have to set again when T0 timer overflows, or T0 timer counts from 0x0000 to full scale (0X1FFF Not keep the correct interval time.

- **Mode1: 16-bit Timer (T0M[1:0]=01)**



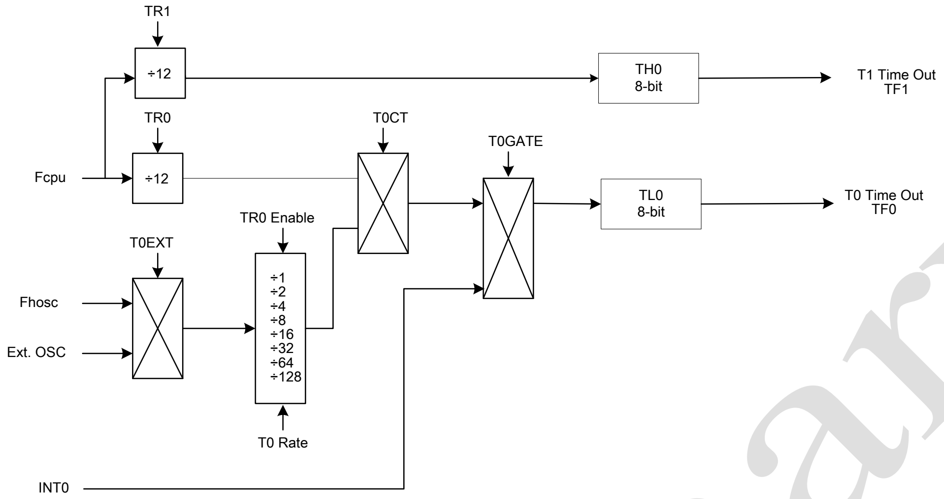
In mode 1, T0 operate as 16-bit timer. 16-bit timer counter is controlled by TH0 and TL0 register. TH0 holds the higher 8-bit and TL0 holds the lower 8-bit. The overflow condition is T0 counter count from full scale (0xFFFF) to zero scale (0x0000).When T0 overflow event occurs, TF0 flag is set as "1" to indicate overflow. In mode 1, T0 counter doesn't build in auto-reload function. Set the T0 interval time through setting TH0/ TL0 by program and have to set again when T0 timer overflows, or T0 timer counts from 0x0000 to full scale (0XFFFF). Not keep the correct interval time.

- **Mode2: 8-bit auto-reload Timer (T0M[1:0]=10)**



In mode 2, T0 operate as 8-bit timer. 8-bit timer counter is controlled by TL0 register. The overflow condition is TL0 counter count from full scale (0xFF) to zero scale (0x00). When T0 overflow event occurs, TF0 flag is set as ”1” to indicate overflow. In mode 2, T0 counter build in auto-reload function. It is to flash TL0 during T0 counting, to set the new value to TH0 (reload buffer), and the new value will be loaded from TH0 to TL0 after T0 overflow occurrence automatically. The auto-reload function is no any control interface and always actives as T0 enables.

- **Mode3: Two 8 bit Timers (T0M[1:0]=11)**



In mode 3, T0 operate as two 8-bit timers. Two 8-bit timer counters are controlled by TL0 and TH0 register respectively. The timer TL0 is controlled by T0 control bit (TR0/T0GATE/T0CT/T0EXT/T0Rate/TF0). TL0 clock source could be Fcpu , Fhosc or external oscillator. The timer TH0 is enabled by TR1 and the clock source is only from Fcpu/12. TH0 can sets the Timer1 overflow (TF1) and controls the Timer1 interrupt. In mode 3, TL0/TH0 counter doesn’t build in auto-reload function. Set the interval time through setting TH0/ TL0 by program and have to set again when TL0/TH0 timer overflows, or TL0/TH0 timer counts from 0x00 to full scale (0XFF). Not keep the correct interval time. When T0 timer is in the mode3, timer1 is inactive. However, Timer1 still can be used as baud rate generator.

9.2.3 T0 TIMER CONTROL REGISTER

088H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
Read/Write	R/W	R/W	R/W	R/W	R/W	-	R/W	-
After reset	0	0	0	0	0	-	0	-

Bit 7 **TF1:** T1 timer external reload interrupt request flag.
0 = None T1 interrupt request
1 = T1 interrupt request.

Bit 6 **TR1:** T1 timer enable control bit.
0 = Disable T1 timer function.
1 = Enable T1 timer function.

Bit 5 **TF0:** T0 timer external reload interrupt request flag.
0 = None T0 interrupt request
1 = T0 interrupt request.

Bit 4 **TR0:** T0 timer enable control bit.
0 = Disable T0 timer function.
1 = Enable T0 timer function.

Bit 3 **IE1:** External P2.2 interrupt (INT1) request flag
0 = None INT1 interrupt request.
1 = INT1 interrupt request.

Bit 1 **IE0:** External P2.1 interrupt (INT0) request flag
0 = None INT0 interrupt request.

1 = INT0 interrupt request.

089H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMOD	T1GATE	T1CT	T1M1	T1M0	T0GATE	T0CT	T0M1	T0M0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit 7 **T1GATE:** Timer 1 gate control bit
0 = Disable T1 gate control.
1 = Enable T1 gate control. External input pin (pin INT1) can gating T1 function. When INT1 is low, T1 timer function is disabled.

Bit 6 **T1CT:** Timer 1 clock source select bit.
0 = Clock source from Fcpu/12.
1 = Clock source from Fhosc.

Bit[5:4] **T1M[1:0]**: Timer 1 mode select.

T1M1	T1M0	Mode	Function
0	0	Mode 0	13-bit Timer
0	1	Mode 1	16-bit Timer.
1	0	Mode 2	8-bit auto-reload Timer.
1	1	Mode 3	Timer 1 inactive

Bit 3 **T0GATE:** Timer 0 gate control bit
0 = Disable T0 gate control.
1 = Enable T0 gate control. External input pin (pin INT0) can gating T0 function. When INT0 is low, T0 timer function is disabled.

Bit 2 **T0CT:** Timer 0 clock source select bit.
0 = Clock source is from Fcpu/12.
1 = Clock source is from Fhosc or Ext. OSC.

Bit[1:0] **T0M[1:0]**: Timer 0 mode select.

T0M1	T0M0	Mode	Function
0	0	Mode 0	13-bit Timer
0	1	Mode 1	16-bit Timer.
1	0	Mode 2	8-bit auto-reload Timer.
1	1	Mode 3	Two 8 bit Timers

08CH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TH0	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **TH0[7:0]**: Timer 0 high byte.
TH0 is high byte of 16-bit T0.

08AH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TL0	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **TL0[7:0]**: Timer 0 high byte.
TL0 is high byte of 16-bit T0.

0E7H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON0	T0EXT	T0RATE2	T0RATE1	T0RATE0	-	T1RATE2	T1RATE1	T1RATE0
Read/Write	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
After reset	0	0	0	0	-	0	0	0

Bit 7 **T0EXT:** Timer0 T0 pin clock source control bit.
0 = Clock source is from Fhsoc.
1 = Clock source is from Ext. OSC. (i.g., 12M X'tal, 4M X'tal, 32K X'tal, ExtCLK).

Bit [6:4] **T0RATE[2:0]:** T0 timer clock source select bits.
T0EXT=0 -> 000 = Fhosc/128, 001 = Fhosc /64, 010 = Fhosc /32, 011 = Fhosc /16, 100 = Fhosc /8,
101 = Fhosc /4, 110 = Fhosc u/2,111 = Fhosc /1.
T0EXT=1 -> 000 = Ext. OSC /128, 001 = Ext. OSC /64, 010 = Ext. OSC /32, 011 = Ext. OSC /16,
100 = Ext. OSC /8, 101 = Ext. OSC /4, 110 = Fhosc/2,111 = Ext. OSC /1.

Bit [2:0] **T1RATE[2:0]:** T1 timer clock source select bits.
000 = Fhosc/128, 001 = Fhosc /64, 010 = Fhosc /32, 011 = Fhosc /16, 100 = Fhosc /8, 101 = Fhosc /4,
110 = Fhosc u/2,111 = Fhosc /1.

9.3 T1 16-BIT TIMER

9.3.1 OVERVIEW

The T1 timer is a 16-bit binary up timer with 4 operating mode. If T1 timer occurs an overflow (from 0xFFFF to 0x0000), it will continue counting and issue a time-out signal to indicate T1 time out event. T1 clock source only includes internal types. Internal clock source is from instruction cycle (Fcpu) or internal oscillator clocks (Fhosc). When clock source is from instruction cycle (Fcpu), timer 0 is incremented every 12 clock cycles, which means that it counts up after every 12 periods of the clock signal. When clock source is from internal oscillator clocks (Fhosc), the clock source is through the T1 prescaler to adjust different periods. All clock sources can be gated by external signal from P2.2/INT1 if T1GATE is enabled. When INT1 is low, the T1 timer is stopped. The T1 builds in idle mode wake-up function when interrupt function is enabled. When T1 timer overflow occurs under idle mode, the system will be waked-up to last operating mode. The main purposes of the T1 timer are as following.

- **13-bit programmable up counting timer (Mode0):** Generate time-out at specific time intervals based on the selected clock frequency.
- **16-bit programmable up counting timer (Mode1):** Generate time-out at specific time intervals based on the selected clock frequency.
- **8-bit programmable up counting timer with auto-reload (Mode2):** Generate time-out at specific time intervals based on the selected clock frequency.
- **T1 timer is disabled (Mode3):** Timer1 is stopped.
- **Interrupt function:** T1 timer function supports interrupt function. When T1 timer occurs overflow, the TF1 actives and the system points program counter to interrupt vector to do interrupt sequence.
- **Idle mode function:** T1 keeps running in idle mode and can wake-up from idle mode as TR1 = 1 when interrupt function is enabled. System will be wake-up after T1 timer overflow occurrence.

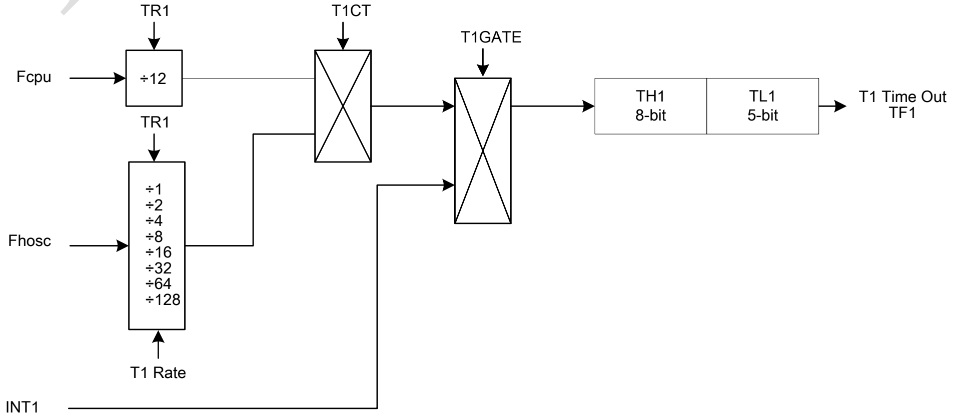
9.3.2 T1 TIMER OPERATION

T1 timer is controlled by TR1 bit. When TR1=0, T1 timer stops. When TR1=1, T1 timer starts to count. Before enabling T1 timer, setup T1 timer's configurations to select timer function modes, e.g. clock source, timer mode, interrupt function…T1 timer counter is controlled by TH1 and TL1 register.TL1/TH1 increases "1" by timer clock source. When T1 overflow event occurs, TF1 flag is set as "1" to indicate overflow. The overflow condition is T1 counter count from full scale to zero scale. TF1 can be cleared by program or cleared by hardware automatically (if interrupt function enabled). If T1 timer interrupt function is enabled (ET1=1), the system will execute interrupt procedure. The interrupt procedure is system program counter points to interrupt vector (ORG 001BH) and executes interrupt service routine after T1 overflow occurrence. TF1 is cleared by hardware automatically in interrupt procedure. T1 timer builds in idle mode wake-up function when interrupt function is enabled. T1 timer keeps counting in idle mode. If interrupt function is enabled, when T1 timer overflow occurs, the system will be waked-up from idle mode to normal mode, and execute T1 interrupt operation. If interrupt function is disabled, the idle mode wake-up function is disabled.

T1 provides different clock sources to implement different applications and configurations. T1 clock source includes Fcpu (instruction cycle) and Fhosc (high speed oscillator) controlled by T1CT1 bits. If T1CT=0, T1 clock source is Fcpu/12. If T1CT=1, T1 clock source is Fhosc1 through T1rate[2:0] pre-scalar to decide Fhosc/1~Fhosc/128. T1GATE bit controls the clock source is gating by external signal from P2.2/INT1. When INT1 is low, the T1 is stopped.

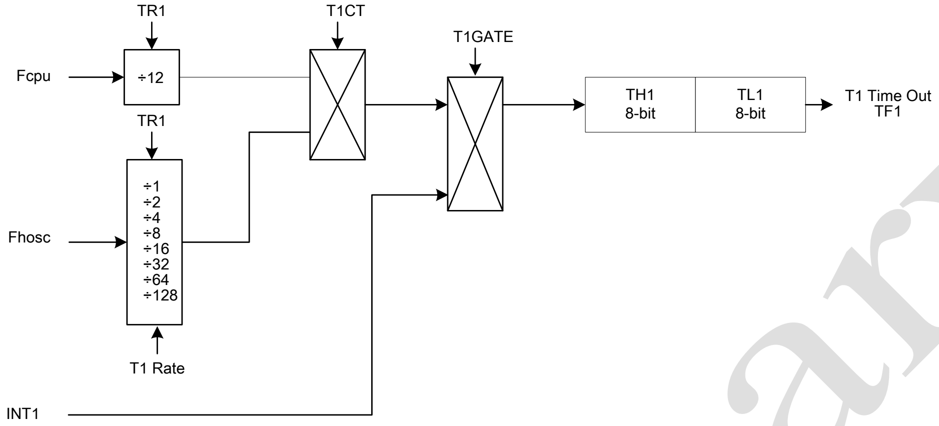
T1 timer can operate as 4 operating mode that is controlled by T1M[1:0].

- **Mode1: 13-bit Timer (T1M[1:0]=00)**



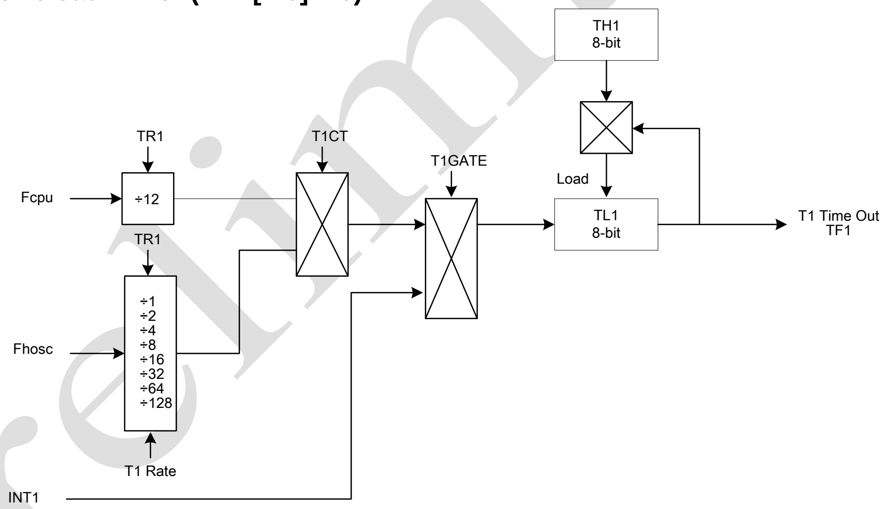
In mode 0, T1 operate as 13-bit timer. 13-bit timer counter is controlled by TH1 and TL1 register. TH1 holds the higher 8-bit and TL1 holds the lower 5-bit. The upper 3-bit of TL1 is useless. The overflow condition is T1 counter count from full scale (0x1FFF) to zero scale (0x0000).When T1 overflow event occurs, TF1 flag is set as ”1” to indicate overflow. In mode 0, T1 counter doesn’t build in auto-reload function. Set the T1 interval time through setting TH1/ TL1 by program and have to set again when T1 timer overflows, or T1 timer counts from 0x0000 to full scale (0X1FFF Not keep the correct interval time.

- **Mode1: 16-bit Timer (T1M[1:0]=01)**



In mode 1, T1 operate as 16-bit timer. 16-bit timer counter is controlled by TH1 and TL1 register. TH1 holds the higher 8-bit and TL1 holds the lower 8-bit. The overflow condition is T1 counter count from full scale (0xFFFF) to zero scale (0x0000).When T1 overflow event occurs, TF1 flag is set as ”1” to indicate overflow. In mode 1, T1 counter doesn’t build in auto-reload function. Set the T1 interval time through setting TH1/ TL1 by program and have to set again when T1 timer overflows, or T1 timer counts from 0x0000 to full scale (0XFFFF). Not keep the correct interval time.

- **Mode2: 8-bit auto-reload Timer (T1M[1:0]=10)**



In mode 2, T1 operate as 8-bit timer. 8-bit timer counter is controlled by TL1 register. The overflow condition is TL1 counter count from full scale (0xFF) to zero scale (0x00). When T1 overflow event occurs, TF1 flag is set as ”1” to indicate overflow. In mode 2, T1 counter build in auto-reload function. It is to flash TL1 during T1 counting, to set the new value to TH1 (reload buffer), and the new value will be loaded from TH1 to TL1 after T1 overflow occurrence automatically. The auto-reload function is no any control interface and always actives as T1 enables.

- **Mode3: Timer 1 inactive.**

In mode 2, T1 timer function is disable.

9.3.3 T1 TIMER CONTROL REGISTER

088H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
Read/Write	R/W	R/W	R/W	R/W	R/W	-	R/W	-
After reset	0	0	0	0	0	-	0	-

Bit 7 **TF1:** T1 timer external reload interrupt request flag.
0 = None T1 interrupt request
1 = T1 interrupt request.

Bit 6 **TR1:** T1 timer enable control bit.
0 = Disable T1 timer function.
1 = Enable T1 timer function.

Bit 5 **TF0:** T0 timer external reload interrupt request flag.
0 = None T0 interrupt request
1 = T0 interrupt request.

Bit 4 **TR0:** T0 timer enable control bit.
0 = Disable T0 timer function.
1 = Enable T0 timer function.

Bit 3 **IE1:** External P2.2 interrupt (INT1) request flag
0 = None INT1 interrupt request.
1 = INT1 interrupt request.

Bit 1 **IE0:** External P2.1 interrupt (INT0) request flag
0 = None INT0 interrupt request.
1 = INT0 interrupt request.

089H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMOD	T1GATE	T1CT	T1M1	T1M0	T0GATE	T0CT	T0M1	T0M0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit 7 **T1GATE:** Timer 1 gate control bit
0 = Disable T1 gate control.
1 = Enable T1 gate control. External input pin (pin INT1) can gating T1 function. When INT1 is low, T1 timer function is disabled.

Bit 6 **T1CT:** Timer 1 clock source select bit.
0 = Clock source from Fcpu/12.
1 = Clock source from Fhosc.

Bit[5:4] **T1M[1:0]**: Timer 1 mode select.

T1M1	T1M0	Mode	Function
0	0	Mode 0	13-bit Timer
0	1	Mode 1	16-bit Timer.
1	0	Mode 2	8-bit auto-reload Timer.
1	1	Mode 3	Timer 1 inactive

Bit 3 **T0GATE:** Timer 0 gate control bit
0 = Disable T0 gate control.
1 = Enable T0 gate control. External input pin (pin INT0) can gating T0 function. When INT0 is low, T0 timer function is disabled.

Bit 2 **T0CT:** Timer 0 clock source select bit.
0 = Clock source is from Fcpu/12.
1 = Clock source is from Fhosc or Ext. OSC.

Bit[1:0] **T0M[1:0]**: Timer 0 mode select.

T0M1	T0M0	Mode	Function
0	0	Mode 0	13-bit Timer
0	1	Mode 1	16-bit Timer.
1	0	Mode 2	8-bit auto-reload Timer.
1	1	Mode 3	Two 8 bit Timers

08DH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TH1	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **TH1[7:0]**: Timer 1 high byte.
TH1 is high byte of 16-bit T1.

08BH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TL1	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **TL1[7:0]**: Timer 1 high byte.
TL1 is high byte of 16-bit T1.

0E7H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON0	T0EXT	T0RATE2	T0RATE1	T0RATE0	-	T1RATE2	T1RATE1	T1RATE0
Read/Write	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
After reset	0	0	0	0	-	0	0	0

Bit 7 **T0EXT:** Timer0 T0 pin clock source control bit.
0 = Clock source is from Fhsoc.
1 = Clock source is from Ext. OSC. (i.g., 12M X'tal, 4M X'tal, 32K X'tal, ExtCLK).

Bit [6:4] **T0RATE[2:0]:** T0 timer clock source select bits.
T0EXT=0 -> 000 = Fhosc/128, 001 = Fhosc /64, 010 = Fhosc /32, 011 = Fhosc /16, 100 = Fhosc /8,
101 = Fhosc /4, 110 = Fhosc u/2,111 = Fhosc /1.

T0EXT=1 -> 000 = Ext. OSC /128, 001 = Ext. OSC /64, 010 = Ext. OSC /32, 011 = Ext. OSC /16,
100 = Ext. OSC /8, 101 = Ext. OSC /4, 110 = Fhosc/2,111 = Ext. OSC /1.

Bit [2:0] **T1RATE[2:0]:** T1 timer clock source select bits.
000 = Fhosc/128, 001 = Fhosc /64, 010 = Fhosc /32, 011 = Fhosc /16, 100 = Fhosc /8, 101 = Fhosc /4,
110 = Fhosc u/2,111 = Fhosc /1.

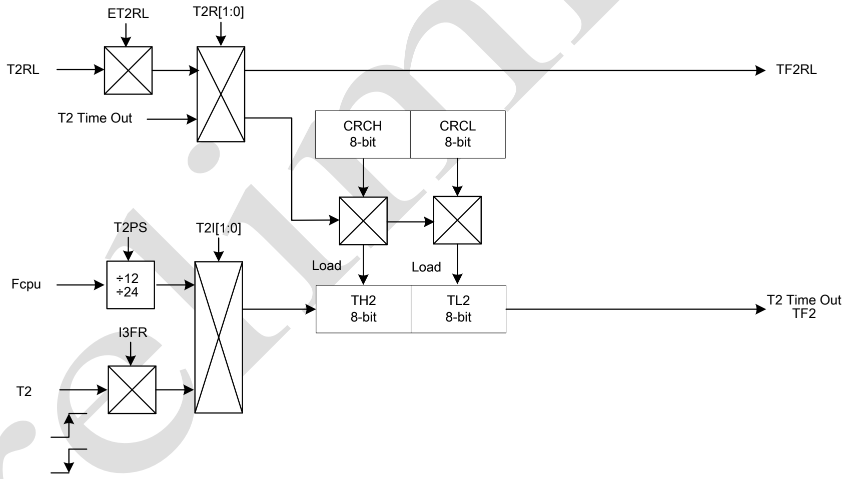
9.4 T2 16-BIT TIMER

9.4.1 OVERVIEW

The T2 timer is a 16-bit binary up timer. If T2 timer occurs an overflow (from 0xFFFF to 0x0000), it will continue counting and issue a time-out signal to indicate T2 time out event. T2 clock source includes internal and external types. Internal clock source is from instruction cycle (Fcpu) and external clock source is from P2.3/T2 GPIO pins controlled by T2I[1:0]. The external clock source is sampled at every rising edge of the clock. When clock source is from instruction cycle (Fcpu), timer 2 is incremented every 12 or 24 clock cycles depending on T2 prescaler, which means that it counts up after every 12 or 24 periods of the clock signal. When clock source is from external clock, T2 is incremented when rising edge is detected. Internal clock source can be gated by external signal from P2.3/T2 if T2I[1:0]=11. When “T2” is low, the T2 timer is stopped. The T2 builds in idle mode wake-up function when interrupt function is enabled. When T2 timer overflow occurs under idle mode, the system will be waked-up to last operating mode. The main purposes of the T2 timer are as following.

- **16-bit programmable up counting timer:** Generate time-out at specific time intervals based on the selected clock frequency.
- **Interrupt function:** T2 timer function supports interrupt function. When T2 timer occurs overflow, the TF2 actives and the system points program counter to interrupt vector to do interrupt sequence.
- **Compare function:** Compare the counter value of T2 timer with the value stored in specific 16-bit registers. The compare unit consists of four registers: CC1/ CC2/ CC3/ CRC. The comparators outputs drive the outputs of P1.0~P1.3.
- **Capture function:** Captured the actual timer contents and saved into 16-bit register upon an external event or a software write operation. The capture unit consists of four registers: CC1/ CC2/ CC3/ CRC.
- **Idle mode function:** T2 keeps running in idle mode and can wake-up from idle mode as T2 timer and interrupt function are enabled. System will be wake-up when TF2 actives after T2 timer overflow occurrence.

9.4.2 T2 TIMER OPERATION



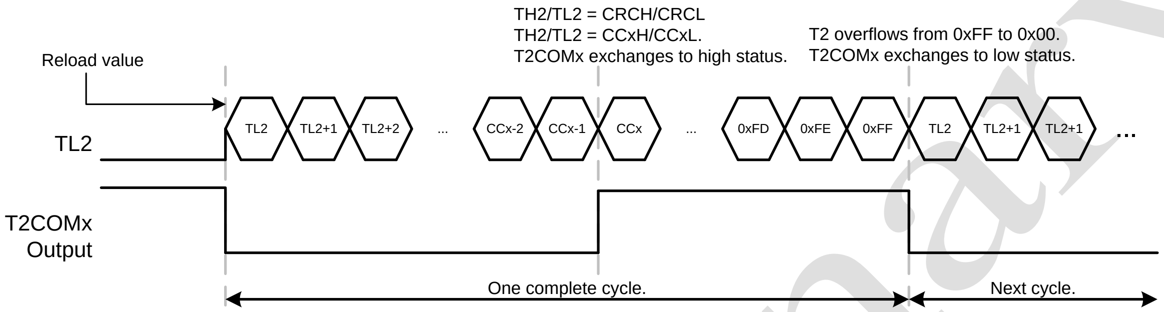
T2 timer is controlled by T2I[1:0]. When T2I[1:0]=00H, T2 timer stops. When T2I[1:0]=01H~11H, T2 timer starts to count. T2 can be configured for timer or counter operation. When T2I[1:0]=01H, the Timer 2 is incremented every 12 or 24 clock cycles depending on the 2:1 prescaler. The prescaler mode is controlled by T2PS bit. When T2I[1:0]=10H, the Timer 2 is incremented when external signal “T2” changes its value from 1 to 0. The “t2” input is sampled at every rising edge of the clock. The Timer 2 is incremented in the cycle following the one in which the transition was detected. The maximum count rate is ½ of the clock frequency. When T2I[1:0]=11H, the Timer 2 is incremented every 12 or 24 clock cycles (depending on “T2PS” flag) but additionally it is gated by external signal from P2.3/T2. When “T2”=0, the Timer 2 is stopped. Before enabling T2 timer, setup T2 timer’s configurations to select timer function modes, e.g. clock source, timer mode, interrupt function…TL2/TH2 increases “1” by timer clock source. When T2 overflow event occurs, TF2 flag is set as ”1” to indicate overflow. The overflow condition is T2 counter count from full scale (0XFFFF) to zero scale (0x00). TF2 can be cleared by program. T2 doesn’t support TF2 cleared by hardware automatically no matter interrupt function is enabled or not. T2 timer supports 2 reload mode. Reload signal can be generated by Timer 2 overflow (auto reload) or generated by negative transition at the corresponding input pin P22/T2RL (external reload) which is controlled by T2R[1:0]. If T2 timer interrupt function is enabled (ET2=1), the system will execute interrupt procedure. The interrupt procedure is system program counter points to interrupt vector (ORG 002BH) and executes interrupt service routine after T2 overflow occurrence. Clear TF2 by program is necessary in interrupt procedure. T2 timer builds in idle mode wake-up function when interrupt function is enabled. T2 timer keeps counting in idle mode. If interrupt function is enabled, when T2 timer overflow occurs, the system will be waked-up from idle mode to normal mode, and execute T2 interrupt operation. If interrupt function is disabled, the idle mode wake-up function is disabled.

9.4.3 T2 COMPARE MODE OPERATION

The compare function executes by four compare registers: CC1/ CC2/ CC3/ CRC. Each compare function can be enabled by CCEN register respectively (COCAHn, COCALn=10). In the compare mode, the value stored in register is compared with the contents of the Timer2 (TH2 and TL2). The comparators outputs drive the outputs of P1.0~P1.3 (T2COM0~T2COM3). It includes 2 compare mode controlled by T2CM.

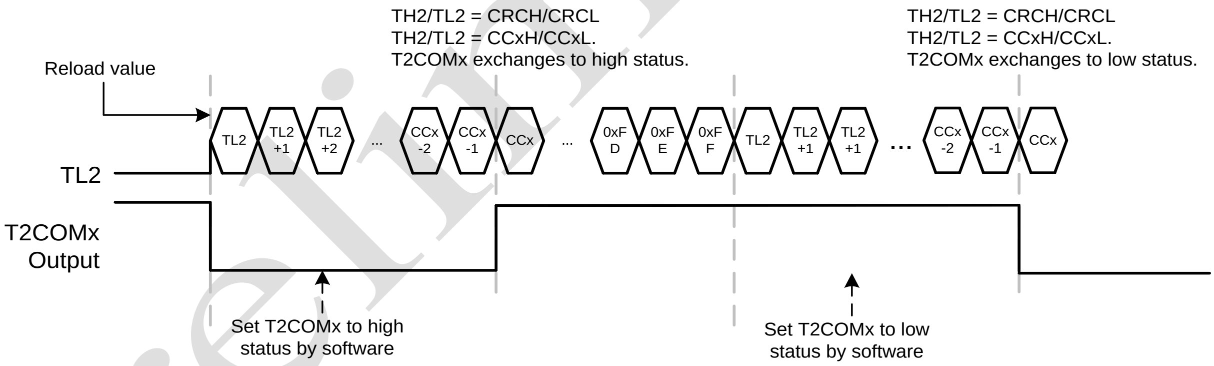
- **Compare Mode0**

When T2CM=0, compare mode0 is enabled. As the T2 counter value is equal to compare register, the related pin will change from low to high and back to low status until T2 timer overflow.



- **Compare Mode1**

When T2CM=1, compare mode1 is enabled. In this mode, the IO pin output status is controlled by software. IO output status can be set by users and stored in the shadow register at first. Until timer 2 counter value is equal to compare register, the specific value is output to IO pin. IO status will not change when T2 timer overflow in this mode.



9.4.4 T2 CAPTURE MODE OPERATION

The capture function executes by four capture registers: CC1/ CC2/ CC3/ CRC. Each capture function can be enabled and select capture mode by CCEN (COCAHn, COCALn=01 or 11) register respectively. In the capture mode, the T2 counter value can be load to capture registers.

- **Capture Mode0**

When COCAHn, COCALn=01H, capture mode0 is enable. The capture event is trigger by the rising or falling edge (control by I3FR) of T2CC0 pin (P3.4) and the rising edge of T2CC1~T2CC3 (rising edge). When the event is triggered, the T2 counter value can be load to related capture registers.

- **Capture Mode1**

When COCAHn, COCALn=11H, capture mode1 is enable. The capture event is trigger by write to low byte of capture register (CRCL/CCnL). When the event is triggered, the T2 counter value can be load to related capture registers.

9.4.5 T2 TIMER CONTROL REGISTER

0C8H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CON	T2PS	I3FR	GF1	T2R1	T2R0	T2CM	T2I1	T2I0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit 7 **T2PS:** T2 timer pre-scalar select bit.
0 = Clock source from Fcpu/12.
1 = Clock source from Fcpu/24.

Bit6 **I3FR:** Active edge selection for compare and capture mode
In compare mode,
0 = The COM0 interrupt would be generated when the content of Timer2 become not equal to the CRC register.
1 = The COM0 interrupt would be generated when the content of Timer2 become equal to the CRC register.
In capture mode 0:
0 = The timer 2 content would be latched into CRC register by T2CC0 is falling edge.
1 = The timer 2 content would be latched into CRC register by T2CC0 is rising edge.

Bit5 **GF2:** General Purpose Flag.

Bit [4:3] **T2R[1:0]:** Timer 2 reload mode selection:
0X = reload disabled
10 = Mode 0
11 = Mode 1

Bit 2 **T2CM:** Timer 2 compare mode selection
0 = Mode 0
1 = Mode 1

Bit [1:0] **T2I[1:0]:** Timer 2 input selection:
00 = timer 2 stopped
01 = input frequency Fcpu /12 or Fcpu /24
10 = timer 2 is incremented by falling edge detection at pin “T2”
11 = input frequency Fcpu /12 or Fcpu /24 gated by external pin “T2”

0CDH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TH2	TH27	TH26	TH25	TH24	TH23	TH22	TH21	TH20
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **TH2[7:0]**: Timer 2 high byte.
TH2 is high byte of 16-bit T2.

0CCH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TL2	TL27	TL26	TL25	TL24	TL23	TL22	TL21	TL20
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **TL2[7:0]**: Timer 2 high byte.
TL2 is high byte of 16-bit T2.

0C1H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCEN	COCAH3	COCAL3	COCAH2	COCAL2	COCAH1	COCAL1	COCAH0	COCAL0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:6] **COCAH3, COCAL3**: compare/capture mode for CC3 register.

COCAH3	**COCAL3**	**Function**

0	0	Compare/capture disabled
0	1	Capture on rising edge at pin T2CC3
1	0	Compare enabled
1	1	Capture on write operation into register CCL3

Bit [5:4] **COCAH2, COCAL2**: compare/capture mode for CC2 register.

COCAH3	COCAL3	Function
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin T2CC2
1	0	Compare enabled
1	1	Capture on write operation into register CCL2

Bit [3:2] **COCAH1, COCAL1**: compare/capture mode for CC1 register.

COCAH1	COCAL1	Function
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin T2CC1
1	0	Compare enabled
1	1	Capture on write operation into register CCL1

Bit [1:0] **COCAH0, COCAL0**: compare/capture mode for CRC register.

COCAH1	COCAL1	Function
0	0	Compare/capture disabled
0	1	Capture on rising edge at pin T2CC0
1	0	Compare enabled
1	1	Capture on write operation into register CRCL

0C7H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCH3	CCH37	CCH36	CCH35	CCH34	CCH33	CCH32	CCH31	CCH30
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0C6H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCL3	CCL37	CCL36	CCL35	CCL34	CCL33	CCL32	CCL31	CCL30
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0C5H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCH2	CCH27	CCH26	CCH25	CCH24	CCH23	CCH22	CCH21	CCH20
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0C4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCL2	CCL27	CCL26	CCL25	CCL24	CCL23	CCL22	CCL21	CCL20
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0C3H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCH1	CCH17	CCH16	CCH15	CCH14	CCH13	CCH12	CCH11	CCH10
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0C2H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CCL1	CCL17	CCL16	CCL15	CCL14	CCL13	CCL12	CCL11	CCL10
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

CCHn/ CCLn: 16-bit Compare/Capture (CCn) Registers.
CCHn holds the high byte and CCLn holds the low byte of CCn Register.

0CBH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCH	CRCH7	CRCH6	CRCH5	CRCH4	CRCH3	CRCH2	CRCH1	CRCH0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0CAH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCL	CRCL7	CRCL6	CRCL5	CRCL4	CRCL3	CRCL2	CRCL1	CRCL0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

CRCH/ CRCL: 16-bit Compare/Reload/Capture (CRC) Registers.
CRCH holds the high byte and CRCL holds the low byte of CRC Register.

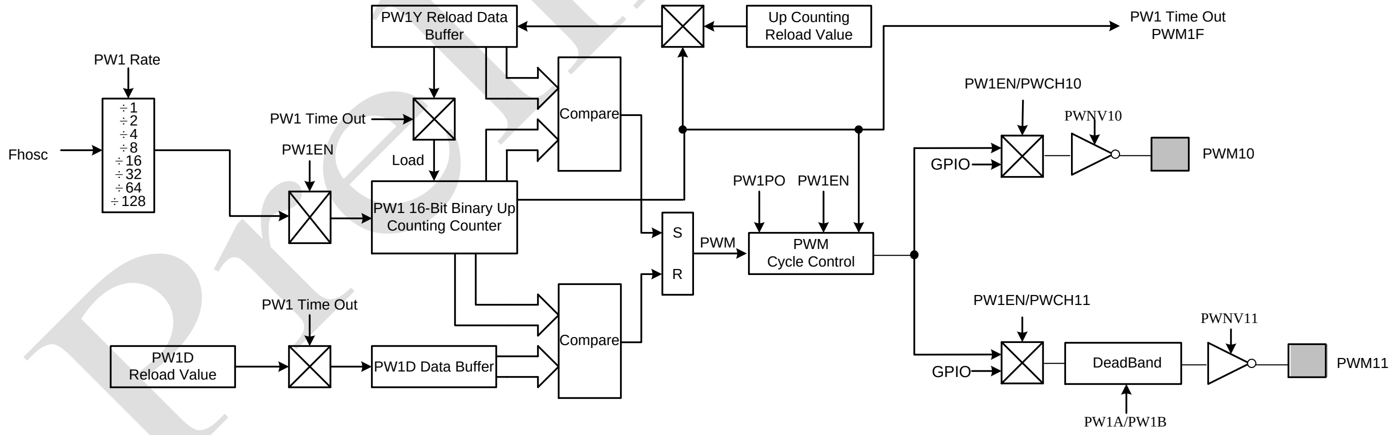
10 16-BIT PULSE WIDTH MODULATION (PWM)

10.1 PW1 16-BIT PWM FUNCTION

10.1.1 OVERVIEW

The PW1 timer is a 16-bit binary up 2-channel PWM, One Pulse PWM function functions. PW1 doesn't support normal timer function. If PW1 timer occurs an overflow (from 0x00 to PWnY-1), it will continue counting and issue a time-out signal to indicate PW1 time out event. PW1 builds in PWM function. The PWM is duty/cycle programmable controlled by PW1Y and PW1D registers. It is easy to implement buzzer, PWM and IR carry signal. PW1 PWM function also supports one pulse output signal that means only output one cycle PWM signal, not continuous. The PWM has two programmable channels shared with GPIO pins and controlled by PWCH[1:0] bit. The output operation must be through enabled each bit/channel of PWCH[1:0] bits. The enabled PWM channel exchanges from GPIO to PWM output. When the PWCH[1:0] bits disables, the PWM channel returns to GPIO mode and last status. PW1 counter supports auto-reload function which always enabled. The PW1 build in IDLE Mode wake-up function if interrupt enable. The main purposes of the PW1 timer are as following.

- **16-bit programmable up counting timer:** Generate time-out at specific time intervals based on the selected clock frequency.
- **Interrupt function:** PW1 timer function supports interrupt function. When PW1 timer occurs overflow, the PWM1F actives and the system points program counter to interrupt vector to do interrupt sequence.
- **Duty/cycle programmable PWM:** The PWM is duty/cycle programmable controlled by PW1Y and PW1D registers.
- **One Pulse PWM:** The one pulse PWM is controlled by PW1PO bit. When PW1PO=0, PW1 is normal timer mode or PWM function mode. When PW1PO=1, PW1 is one pulse PWM function. When PW1EN=1, one pulse PWM outputs and the PWM1F is issued as PW1 counter overflow, PW1EN bit is cleared automatically, the PWM channel returns to GPIO mode and last status.
- **2-Channel PWM output:** The 2-channel PWM output are controlled by PWCH[1:0] bits.



10.1.2 PW1 CLOCK SOURCE

PW1 clock source is Fhosc through PW1rate[2:0] pre-scaler.

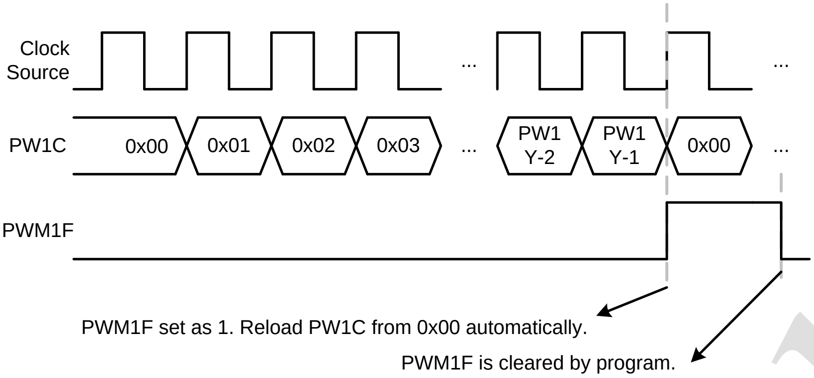
PW1 clock source is Fhosc, PW1rate[2:0] bits:

000=Fhosc/128, 001=Fhosc/64, 010=Fhosc/32, 011=Fhosc/16, 100=Fhosc/8, 101=Fhosc/4, 110=Fhosc/2, 111=Fhosc/1.

10.1.3 PW1 TIMER OPERATION

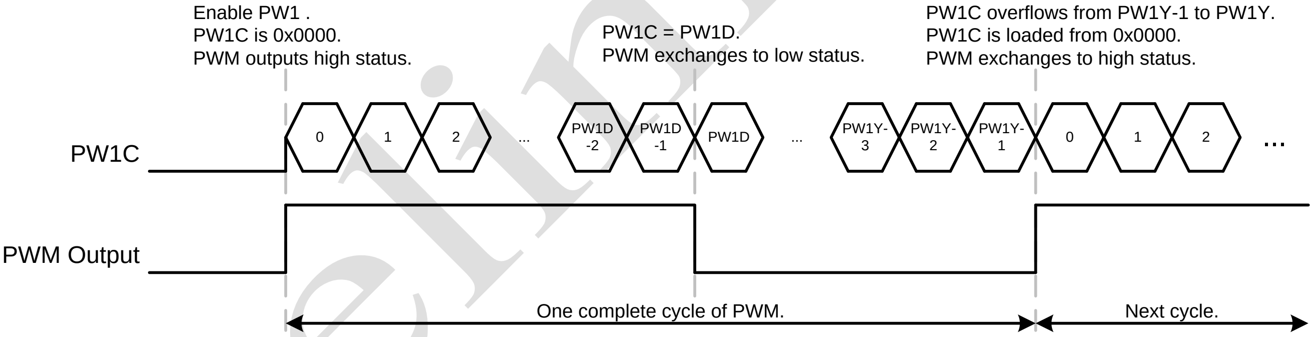
PW1 timer is controlled by PW1EN bit. When PW1EN = 1, PW1 timer starts to count. One count period is one clock source rate. PW1C is PW1 counter and up counting when PW1EN =1. When PW1C counts from 0x0000 to PW1Y-1, PW1 overflow condition is conformed and PWM1F set as “1”. PW1 builds in auto-reload function and always enabled. When PW1 timer overflow occurs, the PW1C counter buffer will be reloaded 0x0000 automatically. PW1 is double buffer design. If the PW1Y/PW1D/PW1A/PW1B is changed by program, the new value will be loaded at next overflow occurrence, or the PW1 interval time is error. If PW1 interrupt function is enabled (EPWM1 =1), the program counter is pointed to interrupt vector to execute interrupt service routine after PW1 timer overflow occurrence.

PW1 interval time = (PW1Y) /($F_{\text{PW1 clock Source}}$/PW1Rate)

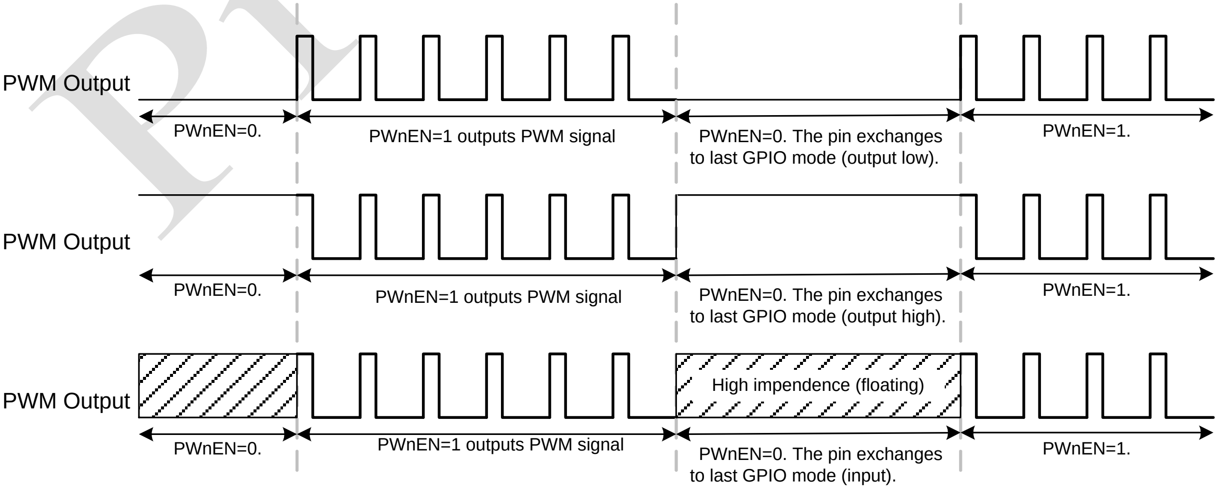

Clock Source
...
...
PW1C
0x00
0x01
0x02
0x03
...
PW1 Y-2
PW1 Y-1
0x00
...
PWM1F
PWM1F set as 1. Reload PW1C from 0x00 automatically.
PWM1F is cleared by program.

10.1.4 PW1 PWM FUNCTION

PW1 timer builds in PWM function controlled by PWnEN and PWCH[1:0] bits. PWM10, PWM11 are output pins. The PWM10, PWM11 output pins are shared with GPIO pin controlled by PWCH[1:0] bits. When output PWM function, we must be set PWnEN =1. When PWM output signal synchronize finishes, the PWM channel exchanges from GPIO to PWM output. When PWnEN = 0, the PWM channel returns to GPIO mode and last status. PWM signal is generated from the result of PW1Y and PW1D comparison. When PW1C counts from 0x0000, the PWM outputs high status which is the PWM initial status. PW1C is loaded new data from PW1Y register to decide PWM cycle and resolution. PW1C keeps counting, and the system compares PW1C and PW1D. When PW1C=PW1D, the PWM output status exchanges to low PW1C keeps counting. When PW1 timer overflow occurs (PWnY-1 to 0x0000), and one cycle of PWM signal finishes. PW1C is reloaded from 0x0000 automatically, and PWM output status exchanges to high for next cycle. PW1D decides the high duty duration, and PW1Y decides the resolution and cycle of PWM. PW1D can’t be larger than PW1Y, or the PWM signal is error.

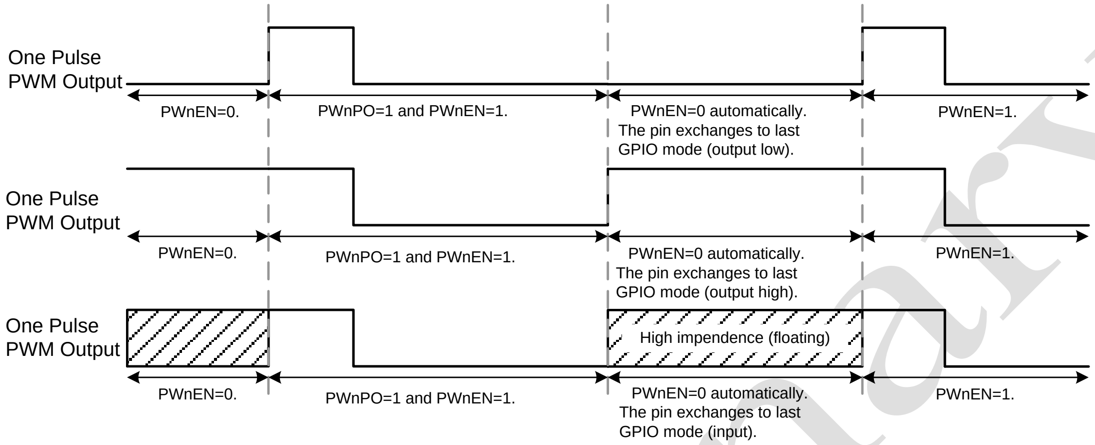

Enable PW1 .
PW1C is 0x0000.
PWM outputs high status.
PW1C = PW1D.
PWM exchanges to low status.
PW1C overflows from PW1Y-1 to PW1Y.
PW1C is loaded from 0x0000.
PWM exchanges to high status.
PW1C
0
1
2
...
PW1D -2
PW1D -1
PW1D
...
PW1Y-3
PW1Y-2
PW1Y-1
0
1
2
...
PWM Output
One complete cycle of PWM.
Next cycle.

- **PWM resolution = PW1Y**
- **PWM duty = (PW1D): (PW1Y-PW1D)**


PWM Output
PWnEN=0.
PWnEN=1 outputs PWM signal
PWnEN=0. The pin exchanges to last GPIO mode (output low).
PWnEN=1.
PWM Output
PWnEN=0.
PWnEN=1 outputs PWM signal
PWnEN=0. The pin exchanges to last GPIO mode (output high).
PWnEN=1.
PWM Output
High impendence (floating)
PWnEN=0.
PWnEN=1 outputs PWM signal
PWnEN=0. The pin exchanges to last GPIO mode (input).
PWnEN=1.

10.1.5 ONE PULSE PWM FUNCTION

When PW1PO = 0, PW1 is PWM function mode. When PW1PO = 1 and PWnEN=1, PW1 will output one pulse PWM function and the PWM1F is issued as PW1 counter overflow. PWnEN bit is cleared automatically and pulse output pin returns to idle status. To output next pulse is to set PWnEN bit by program again. One pulse PWM channels selected by PWCH[1:0] bits. PWM10, PWM11 are output pins. The PWM10, PWM11 output pins are shared with GPIO pin controlled by PWCH[1:0] bits. When output one pulse PWM function, we must be set PW1PO=PWnEN=1. When one pulse PWM output signal synchronize finishes, the PWM channel exchanges from GPIO to PWM output. When one pulse PWM output finishes, PWnEN = 0, the PWM channel returns to GPIO mode and last status.

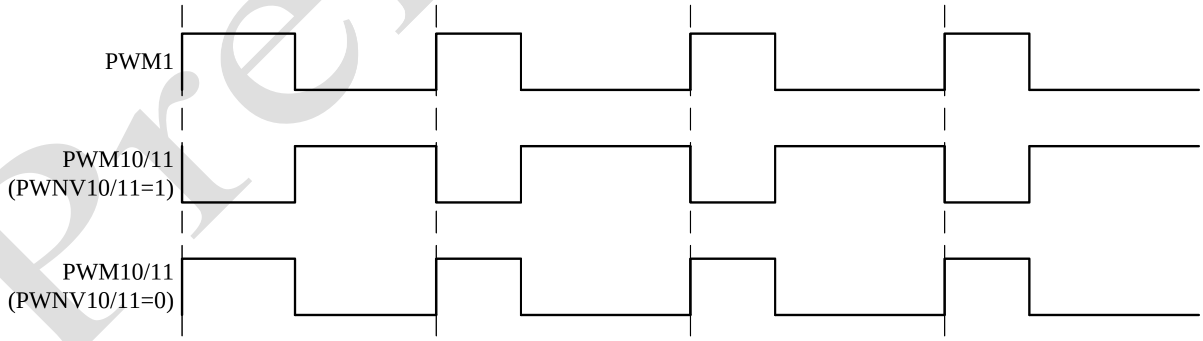

One Pulse PWM Output
PWnEN=0.
PWnPO=1 and PWnEN=1.
PWnEN=0 automatically. The pin exchanges to last GPIO mode (output low).
PWnEN=1.
One Pulse PWM Output
PWnEN=0.
PWnPO=1 and PWnEN=1.
PWnEN=0 automatically. The pin exchanges to last GPIO mode (output high).
PWnEN=1.
One Pulse PWM Output
High impendence (floating)
PWnEN=0.
PWnPO=1 and PWnEN=1.
PWnEN=0 automatically. The pin exchanges to last GPIO mode (input).
PWnEN=1.

10.1.6 2-CHANNEL PWM FUNCTION

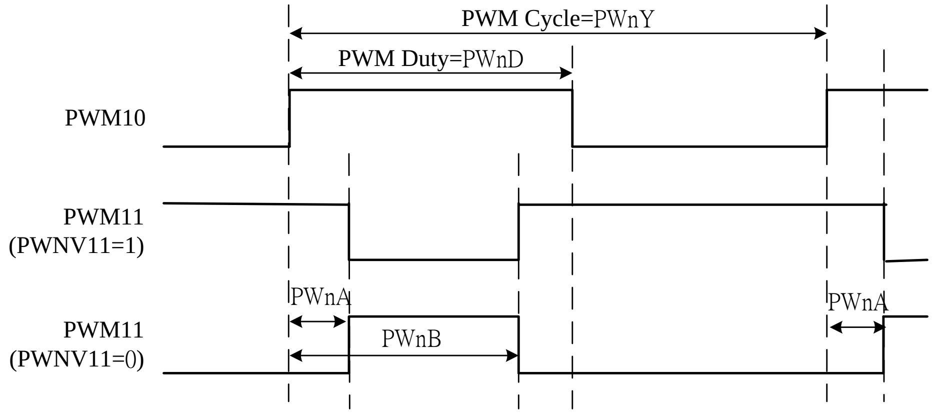
The GPIO mode of 2-channel PWM output pins can be the idle status of PWM signal. PWM high idle status is GPIO output high mode. PWM low idle status is GPIO output low mode. PWM high impendence idle status is GPIO input mode. Select a right “PWM” idle status is very important for loading control as PWM disable. The PWM signal is generated from internal PWM processor and outputs to external pin through PWCH[1:0] bits channel selections. The PWM signal of internal source and external pins are the same. The channel selections only switch PWM channels and not process the phase of PWM signal.

10.1.7 INVERSE PWM OUTPUT AND DEAD BAND FUNCTION

The PWM builds in inverse output function. The PWM has one inverse PWM signal as PWNV =1. When PWNV = 1, the PW1 outputs the inverse PWM signal of PWM1. When PWNV = 0, the PW1 outputs the non-inverse PWM signal of PWM1.The inverse PWM output waveform is below diagram.



The PWM11 builds in “Dead Band” function. The PWM signal has a delay time to normal PWM signal.



The PWM dead band occurs in PWM high pulse width, and the dead band period is programmable from PW1A and PW1D-PW1B registers. The dead band period is symmetrical at left-right terminal of PWM pulse width or not. If the bead band period is longer than PWM duty, the PWM is no output.

> ✸ ***Note: When PWM output in 1channel or 2 channel, PW1Y/PW1D/PW1B registers are not equal to "0".***

10.1.8 PW1 CONTROL REGISTERS

0ABH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1M	PW1rate2	PW1rate1	PW1rate0	PWNV11	PWNV10	PWCH11	PWCH10	PW1PO
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:5] **PW1RATE [2:0]:** PW1 timer clock source select bits.
000 = Fhosc/128, 001 = Fhosc/64, 010 = Fhosc/32, 011 = Fhosc/16, 100 = Fhosc/8, 101 = Fhosc/4, 110 = Fhosc/2, 111 = Fhosc/1.

Bit 4 **PWNV11:** PWM11 inverse output control bit.
0 = PWM11 outputs PWM1 non-inverse signal.
1 = PWM11 outputs PWM1 inverse signal.

Bit 3 **PWNV10:** PWM10 inverse output control bit.
0 = PWM10 outputs PWM1 non-inverse signal.
1 = PWM10 outputs PWM1 inverse signal.

Bit 2 **PWCH11:** PWM11 control bit.
0 = PWM11 pin GPIO mode.
1 = PWM11 output.

Bit 1 **PWCH10:** PWM10 control bit.
0 = PWM10 pin GPIO mode.
1 = PWM10 output.

Bit 0 **PW1PO:** PW1 one pulse output function control bit.
0 = Disable PW1 one pulse output function.
1 = Enable PW1 one pulse output function.

0E4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1OC	PW3EN	PW2EN	PW1EN	P06OC	P05OC	P13OC	P12OC	P11OC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:5] **PW1EN~PW3EN:** PWM enable control bit.
0 = Disable PWM output function, and PWM10/11/20/21/30/31 is GPIO mode.
1 = Enable PWM output function, and PWM10/11/20/21/30/31 outputs PWM signal.
If PWnEN =1 and PWCH= 0, PWM doesn't output and still GPIO mode.

0ADH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1YH	PW1Y15	PW1Y14	PW1Y13	PW1Y12	PW1Y11	PW1Y10	PW1Y9	PW1Y8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0ACH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1YL	PW1Y7	PW1Y6	PW1Y5	PW1Y4	PW1Y3	PW1Y2	PW1Y1	PW1Y0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PW1Y [15:0]: PW1 cycle control buffer.

0AFH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1BH	PW1B15	PW1B14	PW1B13	PW1B12	PW1B11	PW1B10	PW1B9	PW1B8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0AEH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1BL	PW1B7	PW1B6	PW1B5	PW1B4	PW1B3	PW1B2	PW1B1	PW1B0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PW1B [15:0]: PWM B dead band control buffer.

0BCH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1DH	PW1D15	PW1D14	PW1D13	PW1D12	PW1D11	PW1D10	PW1D9	PW1D8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0BBH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1DL	PW1D7	PW1D6	PW1D5	PW1D4	PW1D3	PW1D2	PW1D1	PW1D0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PW1D [15:0]: PWM duty control buffer.

0BDH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1A	PW1A7	PW1A6	PW1A5	PW1A4	PW1A3	PW1A2	PW1A1	PW1A0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PW1A [7:0]: PWM A dead band control buffer.

09BH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPM	PW3CM1	PW3CM0	PW2CM1	PW2CM0	PW1CM1	PW1CM0	OP1EN	OP0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit 2 **PW1CM0:** PWM1 output and CMP0 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 3 **PW1CM1:** PWM1 output and CMP1 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 4 **PW2CM0:** PWM2 output and CMP0 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 5 **PW2CM1:** PWM2 output and CMP1 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 6 **PW3CM0:** PWM3 output and CMP0 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 7 **PW3CM1:** PWM3 output and CMP1 trigger synchronous control bit.
0 = Disable.
1 = Enable.

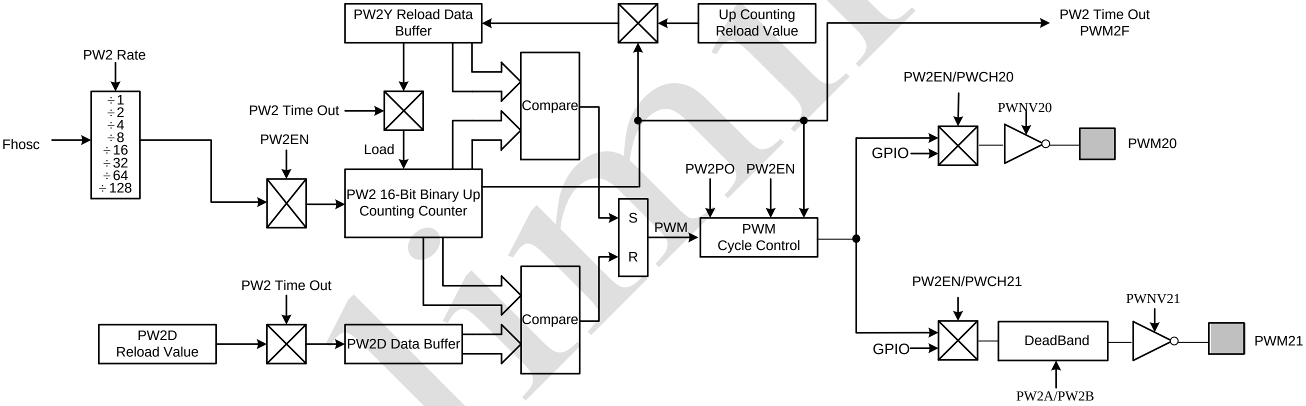
PWnEN	PWnCM1	PWnCM0	PWM Output Control Condition	Remark
0	-	-	No PWM	
1	0	0	Control by PWnEN bit	
1	0	1	Control by CMP0 Trigger	
1	1	0	Control by CMP1 Trigger	
1	1	1	Control by CMP0, CMP1 Trigger	If PWM output and stop collision, The "stop" condition is the highest prority.

10.2 PW2 16-BIT PWM FUNCTION

10.2.1 OVERVIEW

The PW2 timer is a 16-bit binary up 2-channel PWM, One Pulse PWM function functions. PW2 doesn't support normal timer function. If PW2 timer occurs an overflow (from 0x00 to PWnY-1), it will continue counting and issue a time-out signal to indicate PW2 time out event. PW2 builds in PWM function. The PWM is duty/cycle programmable controlled by PW2Y and PW2D registers. It is easy to implement buzzer, PWM and IR carry signal. PW2 PWM function also supports one pulse output signal that means only output one cycle PWM signal, not continuous. The PWM has two programmable channels shared with GPIO pins and controlled by PWCH[3:2] bit. The output operation must be through enabled each bit/channel of PWCH[3:2] bits. The enabled PWM channel exchanges from GPIO to PWM output. When the PWCH[3:2] bits disables, the PWM channel returns to GPIO mode and last status. PW2 counter supports auto-reload function which always enabled. The PW2 build in IDLE Mode wake-up function if interrupt enable. The main purposes of the PW2 timer are as following.

- **16-bit programmable up counting timer:** Generate time-out at specific time intervals based on the selected clock frequency.
- **Interrupt function:** PW2 timer function supports interrupt function. When PW2 timer occurs overflow, the PWM2F actives and the system points program counter to interrupt vector to do interrupt sequence.
- **Duty/cycle programmable PWM:** The PWM is duty/cycle programmable controlled by PW2Y and PW2D registers.
- **One Pulse PWM:** The one pulse PWM is controlled by PW2PO bit. When PW2PO=0, PW2 is normal timer mode or PWM function mode. When PW2PO=1, PW2 is one pulse PWM function. When PW2EN =1, one pulse PWM outputs and the PWM2F is issued as PW2 counter overflow, PW2EN bit is cleared automatically, the PWM channel returns to GPIO mode and last status.
- **2-Channel PWM output:** The 2-channel PWM output are controlled by PWCH[3:2] bits.



10.2.2 PW2 CLOCK SOURCE

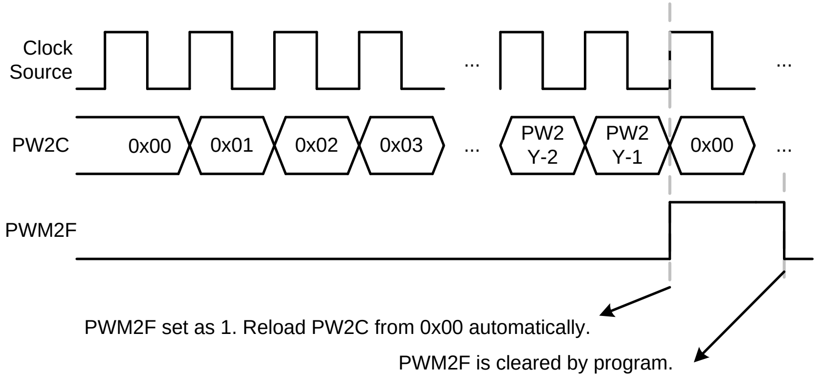
PW2 clock source is Fhosc through PW2rate[2:0] pre-scaler.

PW2 clock source is Fhosc, PW2rate[2:0] bits:
000=Fhosc/128, 001=Fhosc/64, 010=Fhosc/32, 011=Fhosc/16, 100=Fhosc/8, 101=Fhosc/4, 110=Fhosc/2, 111=Fhosc/1.

10.2.3 PW2 Timer Operation

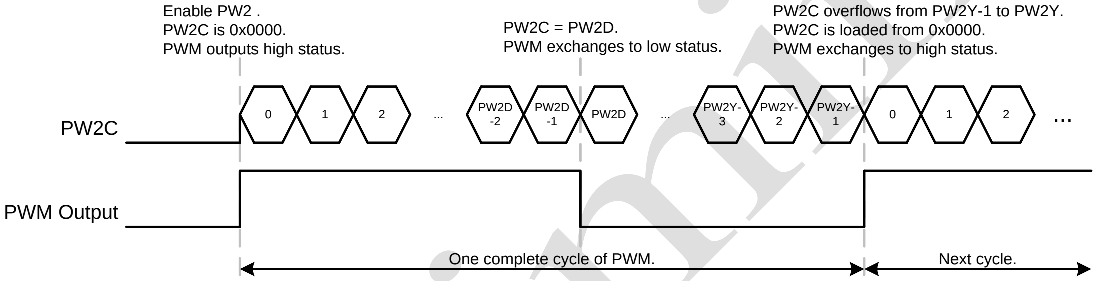
PW2 timer is controlled by PW2EN bit. When PW2EN =1, PW2 timer starts to count. One count period is one clock source rate. PW2C is PW2 counter and up counting when PW2EN =1. When PW2C counts from 0x0000 to PW2Y-1, PW2 overflow condition is conformed and PWM2F set as "1". PW2 builds in auto-reload function and always enabled. When PW2 timer overflow occurs, the PW2C counter buffer will be reloaded from 0x0000 automatically. PW2 is double buffer design. If the PW2Y/PW2D/PW2A/PW2B is changed by program, the new value will be loaded at next overflow occurrence, or the PW2 interval time is error. If PW2 interrupt function is enabled (EPWM2 =1), the program counter is pointed to interrupt vector to execute interrupt service routine after PW2 timer overflow occurrence.

PW2 interval time = (PW2Y) /(F$_{\text{PW2 clock Source}}$/PW2Rate)

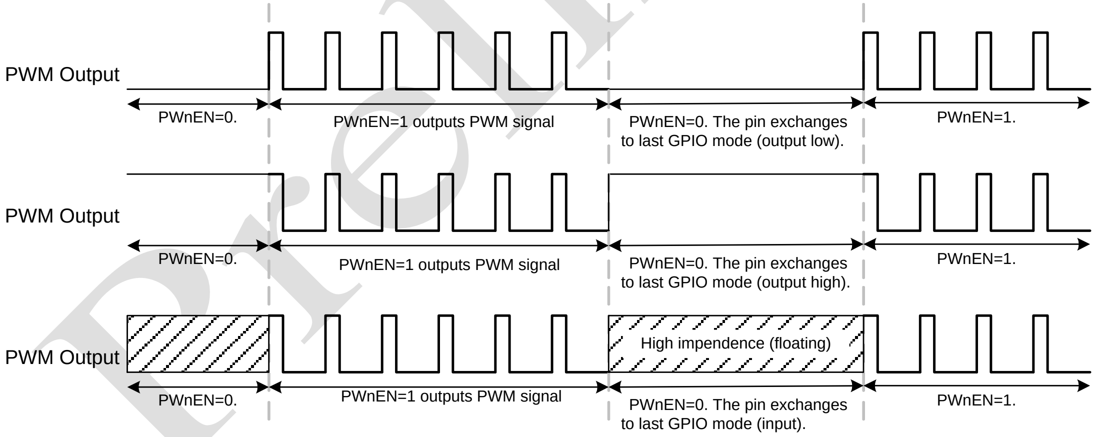

Clock Source
PW2C
0x00
0x01
0x02
0x03
...
PW2 Y-2
PW2 Y-1
0x00
...
PWM2F
PWM2F set as 1. Reload PW2C from 0x00 automatically.
PWM2F is cleared by program.

10.2.4 PW2 PWM Function

PW2 timer builds in PWM function controlled by PWnEN and PWCH[3:2] bits. PWM20, PWM21 are output pins. The PWM20, PWM21 output pins are shared with GPIO pin controlled by PWCH[3:2] bits. When output PWM function, we must be set PWnEN =1. When PWM output signal synchronize finishes, the PWM channel exchanges from GPIO to PWM output. When PWnEN = 0, the PWM channel returns to GPIO mode and last status. PWM signal is generated from the result of PW2Y and PW2D comparison. When PW2C counts from 0x0000, the PWM outputs high status which is the PWM initial status. PW2C is loaded new data from PW2Y register to decide PWM cycle and resolution. PW2C keeps counting, and the system compares PW2C and PW2D. When PW2C=PW2D, the PWM output status exchanges to low PW2C keeps counting. When PW2 timer overflow occurs (PWnY-1 to 0x0000), and one cycle of PWM signal finishes. PW2C is reloaded from 0x0000 automatically, and PWM output status exchanges to high for next cycle. PW2D decides the high duty duration, and PW2Y decides the resolution and cycle of PWM. PW2D can't be larger than PW2Y, or the PWM signal is error.

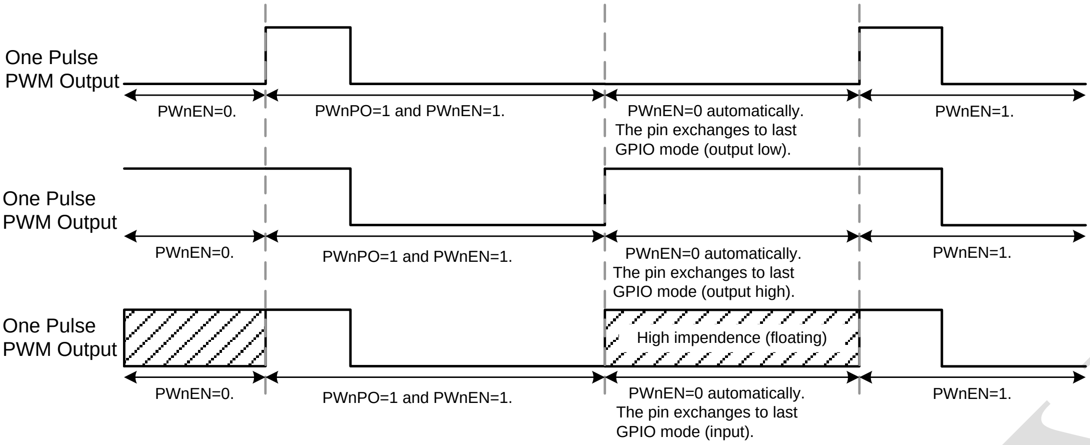

Enable PW2 .
PW2C is 0x0000.
PWM outputs high status.
PW2C = PW2D.
PWM exchanges to low status.
PW2C overflows from PW2Y-1 to PW2Y.
PW2C is loaded from 0x0000.
PWM exchanges to high status.
PW2C
0
1
2
...
PW2D -2
PW2D -1
PW2D
...
PW2Y-3
PW2Y-2
PW2Y-1
0
1
2
...
PWM Output
One complete cycle of PWM.
Next cycle.

- **PWM resolution = PW2Y**
- **PWM duty = (PW2D): (PW2Y-PW2D)**


PWM Output
PWnEN=0.
PWnEN=1 outputs PWM signal
PWnEN=0. The pin exchanges to last GPIO mode (output low).
PWnEN=1.
PWM Output
PWnEN=0.
PWnEN=1 outputs PWM signal
PWnEN=0. The pin exchanges to last GPIO mode (output high).
PWnEN=1.
PWM Output
High impendence (floating)
PWnEN=0.
PWnEN=1 outputs PWM signal
PWnEN=0. The pin exchanges to last GPIO mode (input).
PWnEN=1.

10.2.5 One Pulse PWM Function

When PW2PO = 0, PW2 is PWM function mode. When PW2PO = 1 and PWnEN=1, PW2 will output one pulse PWM function and the PWM2F is issued as PW2 counter overflow. PWnEN bit is cleared automatically and pulse output pin returns to idle status. To output next pulse is to set PWnEN bit by program again. One pulse PWM channels selected by PWCH[3:2] bits. PWM20, PWM21 are output pins. The PWM20, PWM21 output pins are shared with GPIO pin controlled by PWCH[3:2] bits. When output one pulse PWM function, we must be set PW2PO=PWnEN=1. When one pulse PWM output signal synchronize finishes, the PWM channel exchanges from GPIO to PWM output. When one pulse PWM output finishes, PWnEN = 0, the PWM channel returns to GPIO mode and last status.

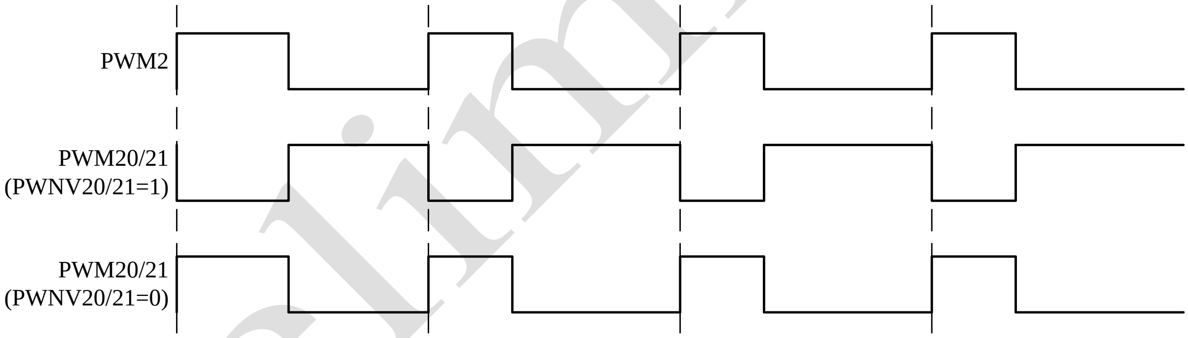

One Pulse PWM Output
PWnEN=0.
PWnPO=1 and PWnEN=1.
PWnEN=0 automatically. The pin exchanges to last GPIO mode (output low).
PWnEN=1.
One Pulse PWM Output
PWnEN=0.
PWnPO=1 and PWnEN=1.
PWnEN=0 automatically. The pin exchanges to last GPIO mode (output high).
PWnEN=1.
One Pulse PWM Output
High impendence (floating)
PWnEN=0.
PWnPO=1 and PWnEN=1.
PWnEN=0 automatically. The pin exchanges to last GPIO mode (input).
PWnEN=1.

10.2.6 2-Channel PWM Output

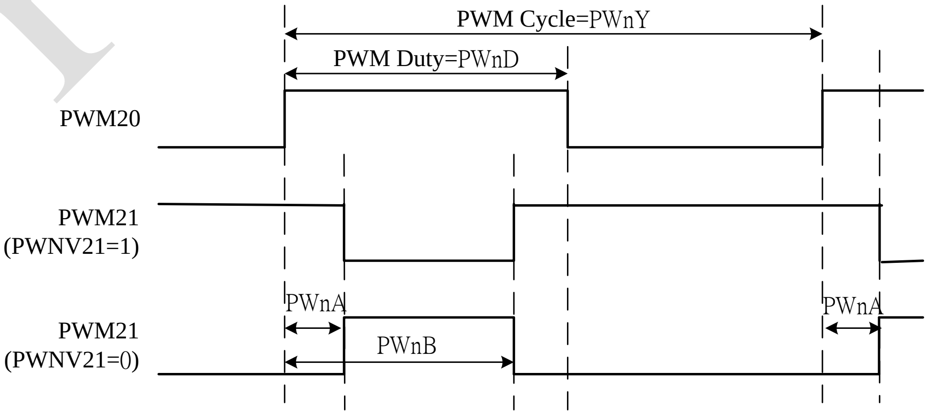
The GPIO mode of 2-channel PWM output pins can be the idle status of PWM signal. PWM high idle status is GPIO output high mode. PWM low idle status is GPIO output low mode. PWM high impendence idle status is GPIO input mode. Select a right "PWM" idle status is very important for loading control as PWM disable. The PWM signal is generated from internal PWM processor and outputs to external pin through PWCH[3:2] bits channel selections. The PWM signal of internal source and external pins are the same. The channel selections only switch PWM channels and not process the phase of PWM signal.

10.2.7 Inverse PWM Output and Dead Band Function

The PWM builds in inverse output function. The PWM has one inverse PWM signal as PWNV =1. When PWNV = 1, the PW2 outputs the inverse PWM signal of PWM2. When PWNV = 0, the PW2 outputs the non-inverse PWM signal of PWM2.The inverse PWM output waveform is below diagram.



The PWM21 builds in "Dead Band" function. The PWM signal has a delay time to normal PWM signal.



The PWM dead band occurs in PWM high pulse width, and the dead band period is programmable from PW2A and

PW2D-PW2B registers. The dead band period is symmetrical at left-right terminal of PWM pulse width or not. If the bead band period is longer than PWM duty, the PWM is no output.

> ✷ ***Note: When PWM output in 1channel or 2 channel, PW2Y/PW2D/PW2B registers are not equal to "0".***

10.2.8 PW2 Control Registers

0A1H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW2M	PW2rate2	PW2rate1	PW2rate0	PWNV21	PWNV20	PWCH21	PWCH20	PW2PO
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:5] **PW2RATE [2:0]:** PW2 timer clock source select bits.
000 = Fhosc/128, 001 = Fhosc/64, 010 = Fhosc/32, 011 = Fhosc/16, 100 = Fhosc/8, 101 = Fhosc/4, 110 = Fhosc/2, 111 = Fhosc/1.

Bit 4 **PWNV21:** PWM21 inverse output control bit.
0 = PWM21 outputs PWM2 non-inverse signal.
1 = PWM21 outputs PWM2 inverse signal.

Bit 3 **PWNV20:** PWM20 inverse output control bit.
0 = PWM20 outputs PWM2 non-inverse signal.
1 = PWM20 outputs PWM2 inverse signal.

Bit 2 **PWCH21:** PWM21 control bit.
0 = PWM21 pin GPIO mode.
1 = PWM21 output.

Bit 1 **PWCH20:** PWM20 control bit.
0 = PWM20 pin GPIO mode.
1 = PWM20 output.

Bit 0 **PW2PO:** PW2 one pulse output function control bit.
0 = Disable PW2 one pulse output function.
1 = Enable PW2 one pulse output function.

0E4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1OC	PW3EN	PW2EN	PW1EN	P06OC	P05OC	P13OC	P12OC	P11OC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:5] **PW1EN~PW3EN:** PWM enable control bit.
0 = Disable PWM output function, and PWM10/11/20/21/30/31 is GPIO mode.
1 = Enable PWM output function, and PWM10/11/20/21/30/31 outputs PWM signal.
If PWnEN =1 and PWCH= 0, PWM doesn't output and still GPIO mode.

0A3H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW2YH	PW2Y15	PW2Y14	PW2Y13	PW2Y12	PW2Y11	PW2Y10	PW2Y9	PW2Y8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0A2H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW2YL	PW2Y7	PW2Y6	PW2Y5	PW2Y4	PW2Y3	PW2Y2	PW2Y1	PW2Y0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PW2Y [15:0]: PW2 cycle control buffer.

0A5H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW2BH	PW2B15	PW2B14	PW2B13	PW2B12	PW2B11	PW2B10	PW2B9	PW2B8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0A4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW2BL	PW2B7	PW2B6	PW2B5	PW2B4	PW2B3	PW2B2	PW2B1	PW2B0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PW2B [15:0]: PWM B dead band control buffer.

0A7H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW2DH	PW2D15	PW2D14	PW2D13	PW2D12	PW2D11	PW2D10	PW2D9	PW2D8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0A6H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW2DL	PW2D7	PW2D6	PW2D5	PW2D4	PW2D3	PW2D2	PW2D1	PW2D0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PW2D [15:0]: PWM duty control buffer.

0BEH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW2A	PW2A7	PW2A6	PW2A5	PW2A4	PW2A3	PW2A2	PW2A1	PW2A0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PW2A [7:0]: PWM A dead band control buffer.

09BH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPM	PW3CM1	PW3CM0	PW2CM1	PW2CM0	PW1CM1	PW1CM0	OP1EN	OP0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit 2 **PW1CM0:** PWM1 output and CMP0 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 3 **PW1CM1:** PWM1 output and CMP1 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 4 **PW2CM0:** PWM2 output and CMP0 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 5 **PW2CM1:** PWM2 output and CMP1 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 6 **PW3CM0:** PWM3 output and CMP0 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 7 **PW3CM1:** PWM3 output and CMP1 trigger synchronous control bit.
0 = Disable.
1 = Enable.

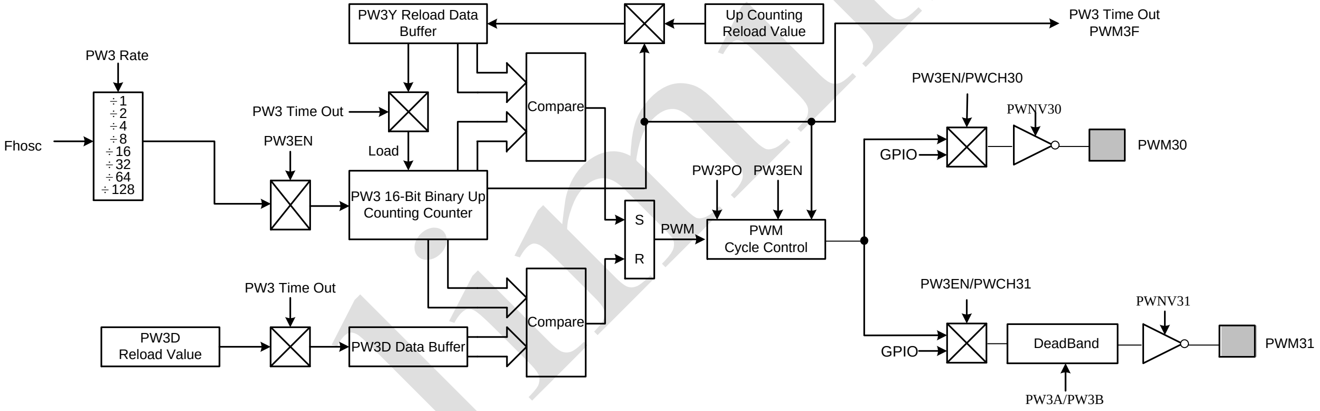
PWnEN	PWnCM1	PWnCM0	PWM Output Control Condition	Remark
0	-	-	No PWM	
1	0	0	Control by PWnEN bit	
1	0	1	Control by CMP0 Trigger	
1	1	0	Control by CMP1 Trigger	
1	1	1	Control by CMP0, CMP1 Trigger	If PWM output and stop collision, The “stop” condition is the highest prority.

10.3 PW3 16-BIT PWM FUNCTION

10.3.1 OVERVIEW

The PW3 timer is a 16-bit binary up 2-channel PWM, One Pulse PWM function functions. PW3 doesn’t support normal timer function. If PW3 timer occurs an overflow (from 0x00 to PWnY-1), it will continue counting and issue a time-out signal to indicate PW3 time out event. PW3 builds in PWM function. The PWM is duty/cycle programmable controlled by PW3Y and PW3D registers. It is easy to implement buzzer, PWM and IR carry signal. PW3 PWM function also supports one pulse output signal that means only output one cycle PWM signal, not continuous. The PWM has two programmable channels shared with GPIO pins and controlled by PWCH[5:4] bit. The output operation must be through enabled each bit/channel of PWCH[5:4] bits. The enabled PWM channel exchanges from GPIO to PWM output. When the PWCH[5:4] bits disables, the PWM channel returns to GPIO mode and last status. PW3 counter supports auto-reload function which always enabled. The PW3 build in IDLE Mode wake-up function if interrupt enable. The main purposes of the PW3 timer are as following.

- **16-bit programmable up counting timer:** Generate time-out at specific time intervals based on the selected clock frequency.
- **Interrupt function:** PW3 timer function supports interrupt function. When PW3 timer occurs overflow, the PWM3F actives and the system points program counter to interrupt vector to do interrupt sequence.
- **Duty/cycle programmable PWM:** The PWM is duty/cycle programmable controlled by PW3Y and PW3D registers.
- **One Pulse PWM:** The one pulse PWM is controlled by PW3PO bit. When PW3PO=0, PW3 is normal timer mode or PWM function mode. When PW3PO=1, PW3 is one pulse PWM function. When PW3EN =1, one pulse PWM outputs and the PWM3F is issued as PW3 counter overflow, PW3EN bit is cleared automatically, the PWM channel returns to GPIO mode and last status.
- **2-Channel PWM output:** The 2-channel PWM output are controlled by PWCH[5:4] bits.



10.3.2 PW3 Clock Source

PW3 clock source is Fhosc through PW3rate[2:0] pre-scaler.

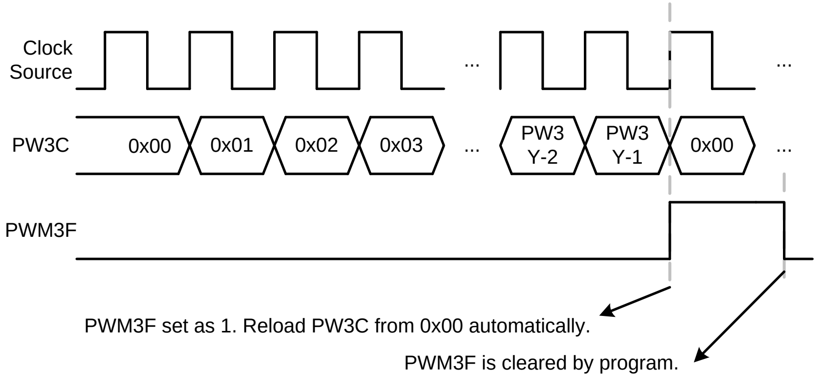
PW3 clock source is Fhosc, PW3rate[2:0] bits:

000=Fhosc/128, 001=Fhosc/64, 010=Fhosc/32, 011=Fhosc/16, 100=Fhosc/8, 101=Fhosc/4, 110=Fhosc/2, 111=Fhosc/1.

10.3.3 PW3 TIMER OPERATION

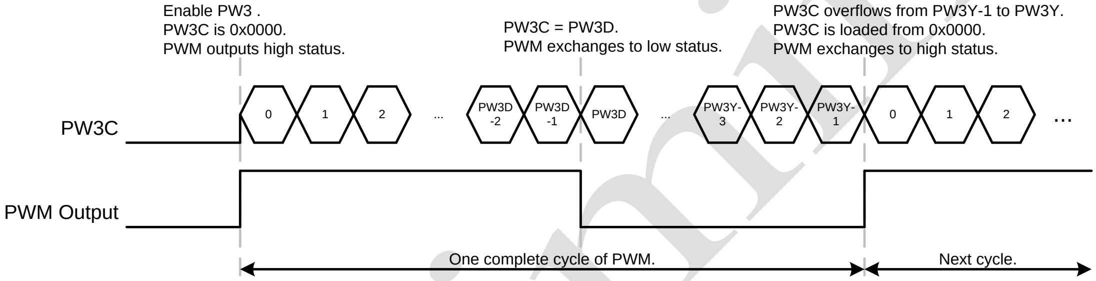
PW3 timer is controlled by PW3EN bit. When PW3EN =1, PW3 timer starts to count. One count period is one clock source rate. PW3C is PW3 counter and up counting when PW3EN =1. When PW3C counts from 0x0000 to PW3Y-1, PW3 overflow condition is conformed and PWM3F set as “1”. PW3 builds in auto-reload function and always enabled. When PW3 timer overflow occurs, the PW3C counter buffer will be reloaded from 0x0000 automatically. PW3 is double buffer design. If the PW3Y/PW3D/PW3A/PW3B is changed by program, the new value will be loaded at next overflow occurrence, or the PW3 interval time is error. If PW3 interrupt function is enabled (PW3IEN=1), the program counter is pointed to interrupt vector to execute interrupt service routine after PW3 timer overflow occurrence.

PW3 interval time = (PW3Y) /($F_{PW3\ clock\ Source}$/PW3Rate)

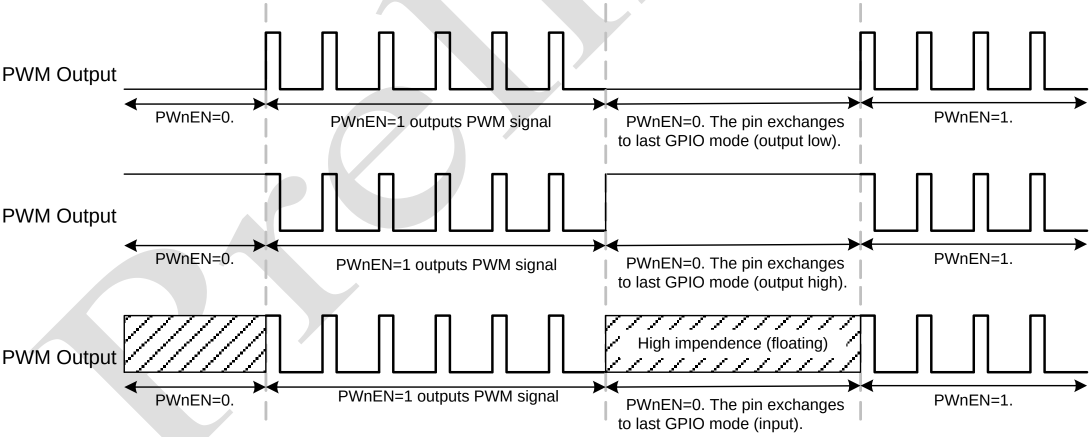

Clock Source
PW3C
0x00
0x01
0x02
0x03
...
PW3 Y-2
PW3 Y-1
0x00
...
PWM3F
PWM3F set as 1. Reload PW3C from 0x00 automatically.
PWM3F is cleared by program.

10.3.4 PW3 PWM Function

PW3 timer builds in PWM function controlled by PWnEN and PWCH[5:4] bits. PWM30, PWM31 are output pins. The PWM30, PWM31 output pins are shared with GPIO pin controlled by PWCH[5:4] bits. When output PWM function, we must be set PWnEN =1. When PWM output signal synchronize finishes, the PWM channel exchanges from GPIO to PWM output. When PWnEN = 0, the PWM channel returns to GPIO mode and last status. PWM signal is generated from the result of PW3Y and PW3D comparison. When PW3C counts from 0x0000, the PWM outputs high status which is the PWM initial status. PW3C is loaded new data from PW3Y register to decide PWM cycle and resolution. PW3C keeps counting, and the system compares PW3C and PW3D. When PW3C=PW3D, the PWM output status exchanges to low PW3C keeps counting. When PW3 timer overflow occurs (PWnY-1 to 0x0000), and one cycle of PWM signal finishes. PW3C is reloaded from 0x0000 automatically, and PWM output status exchanges to high for next cycle. PW3D decides the high duty duration, and PW3Y decides the resolution and cycle of PWM. PW3D can't be larger than PW3Y, or the PWM signal is error.

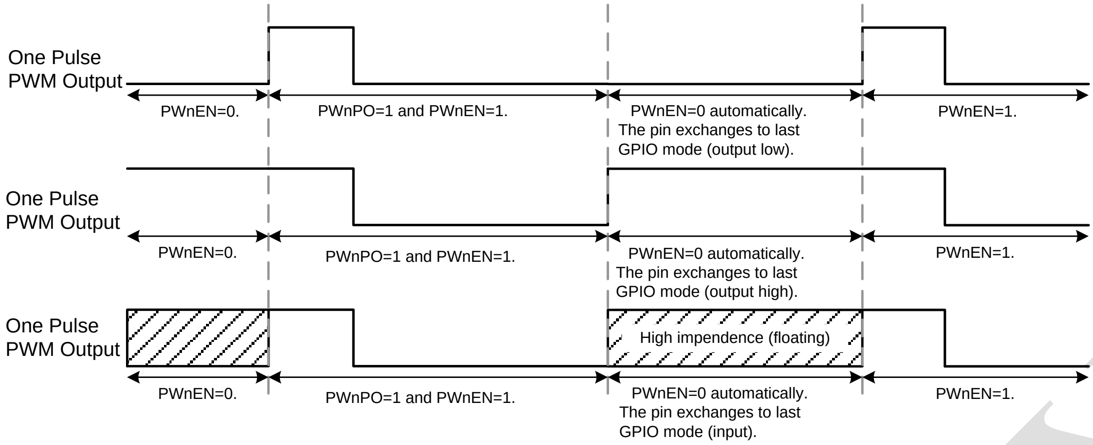

Enable PW3 .
PW3C is 0x0000.
PWM outputs high status.
PW3C = PW3D.
PWM exchanges to low status.
PW3C overflows from PW3Y-1 to PW3Y.
PW3C is loaded from 0x0000.
PWM exchanges to high status.
PW3C
0
1
2
...
PW3D -2
PW3D -1
PW3D
...
PW3Y-3
PW3Y-2
PW3Y-1
0
1
2
...
PWM Output
One complete cycle of PWM.
Next cycle.

- **PWM resolution = PW3Y**
- **PWM duty = (PW3D): (PW3Y-PW3D)**


PWM Output
PWnEN=0.
PWnEN=1 outputs PWM signal
PWnEN=0. The pin exchanges to last GPIO mode (output low).
PWnEN=1.
PWM Output
PWnEN=0.
PWnEN=1 outputs PWM signal
PWnEN=0. The pin exchanges to last GPIO mode (output high).
PWnEN=1.
PWM Output
High impendence (floating)
PWnEN=0.
PWnEN=1 outputs PWM signal
PWnEN=0. The pin exchanges to last GPIO mode (input).
PWnEN=1.

10.3.5 ONE PULSE PWM Function

When PW3PO = 0, PW3 is PWM function mode. When PW3PO = 1 and PWnEN=1, PW3 will output one pulse PWM function and the PWM3F is issued as PW3 counter overflow. PWnEN bit is cleared automatically and pulse output pin returns to idle status. To output next pulse is to set PWnEN bit by program again. One pulse PWM channels selected by PWCH[5:4] bits. PWM30, PWM31 are output pins. The PWM30, PWM31 output pins are shared with GPIO pin controlled by PWCH[5:4] bits. When output one pulse PWM function, we must be set PW3PO=PWnEN=1. When one pulse PWM output signal synchronize finishes, the PWM channel exchanges from GPIO to PWM output. When one pulse PWM output finishes, PWnEN = 0, the PWM channel returns to GPIO mode and last status.

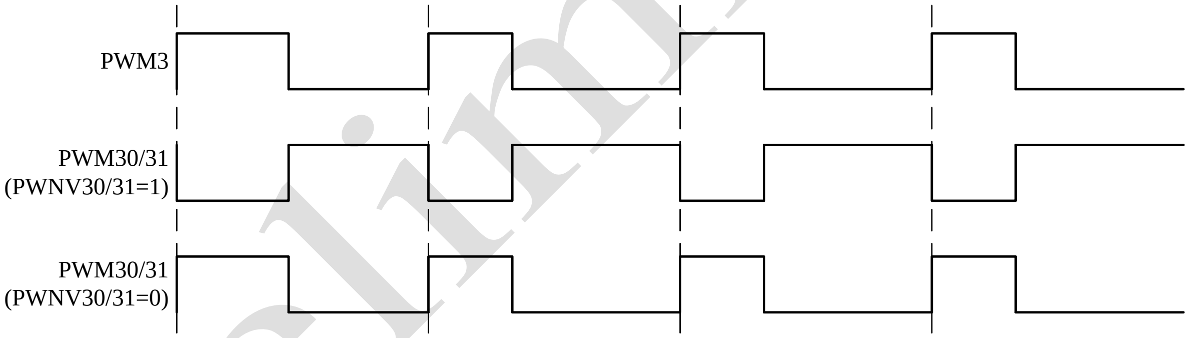

One Pulse PWM Output
PWnEN=0.
PWnPO=1 and PWnEN=1.
PWnEN=0 automatically. The pin exchanges to last GPIO mode (output low).
PWnEN=1.
One Pulse PWM Output
PWnEN=0.
PWnPO=1 and PWnEN=1.
PWnEN=0 automatically. The pin exchanges to last GPIO mode (output high).
PWnEN=1.
One Pulse PWM Output
High impendence (floating)
PWnEN=0.
PWnPO=1 and PWnEN=1.
PWnEN=0 automatically. The pin exchanges to last GPIO mode (input).
PWnEN=1.

10.3.6 2-CHANNEL PWM OUTPUT

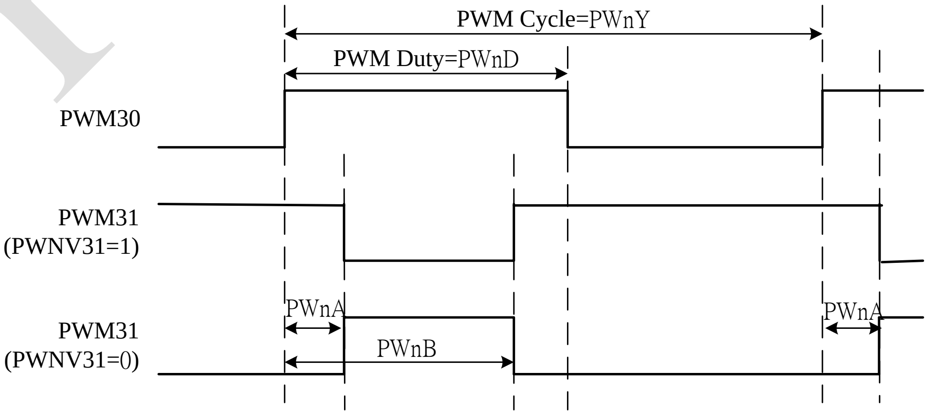
The GPIO mode of 2-channel PWM output pins can be the idle status of PWM signal. PWM high idle status is GPIO output high mode. PWM low idle status is GPIO output low mode. PWM high impendence idle status is GPIO input mode. Select a right "PWM" idle status is very important for loading control as PWM disable. The PWM signal is generated from internal PWM processor and outputs to external pin through PWCH[5:4] bits channel selections. The PWM signal of internal source and external pins are the same. The channel selections only switch PWM channels and not process the phase of PWM signal.

10.3.7 INVERSE PWM OUTOUT AND DEAD BAND FUNCTION

The PWM builds in inverse output function. The PWM has one inverse PWM signal as PWNV =1. When PWNV = 1, the PW3 outputs the inverse PWM signal of PWM3. When PWNV = 0, the PW3 outputs the non-inverse PWM signal of PWM3.The inverse PWM output waveform is below diagram.



The PWM31 builds in "Dead Band" function. The PWM signal has a delay time to normal PWM signal.



The PWM dead band occurs in PWM high pulse width, and the dead band period is programmable from PW3A and PW3D-PW3B registers. The dead band period is symmetrical at left-right terminal of PWM pulse width or not. If the bead band period is longer than PWM duty, the PWM is no output.

> ✸ ***Note: When PWM output in 1channel or 2 channel, PW3Y/PW3D/PW3B registers are not equal to "0".***

10.3.8 PW3 Control Registers

0B1H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW3M	PW3rate2	PW3rate1	PW3rate0	PWNV31	PWNV30	PWCH31	PWCH30	PW3PO
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:5] **PW3RATE [2:0]:** PW3 timer clock source select bits.
000 = Fhosc/128, 001 = Fhosc/64, 010 = Fhosc/32, 011 = Fhosc/16, 100 = Fhosc/8, 101 = Fhosc/4, 110 = Fhosc/2, 111 = Fhosc/1.

Bit 4 **PWNV31:** PWM31 inverse output control bit.
0 = PWM31 outputs PWM3 non-inverse signal.
1 = PWM31 outputs PWM3 inverse signal.

Bit 3 **PWNV30:** PWM30 inverse output control bit.
0 = PWM30 outputs PWM3 non-inverse signal.
1 = PWM30 outputs PWM3 inverse signal.

Bit 2 **PWCH31:** PWM31 control bit.
0 = PWM31 pin GPIO mode.
1 = PWM31 output.

Bit 1 **PWCH30:** PWM30 control bit.
0 = PWM30 pin GPIO mode.
1 = PWM30 output.

Bit 0 **PW3PO:** PW3 one pulse output function control bit.
0 = Disable PW3 one pulse output function.
1 = Enable PW3 one pulse output function.

0E4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1OC	PW3EN	PW2EN	PW1EN	P06OC	P05OC	P13OC	P12OC	P11OC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:5] **PW1EN~PW3EN:** PWM enable control bit.
0 = Disable PWM output function, and PWM10/11/20/21/30/31 is GPIO mode.
1 = Enable PWM output function, and PWM10/11/20/21/30/31 outputs PWM signal.
If PWnEN =1 and PWCH= 0, PWM doesn't output and still GPIO mode.

0B3H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW3YH	PW3Y15	PW3Y14	PW3Y13	PW3Y12	PW3Y11	PW3Y10	PW3Y9	PW3Y8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0B2H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW3YL	PW3Y7	PW3Y6	PW3Y5	PW3Y4	PW3Y3	PW3Y2	PW3Y1	PW3Y0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PW3Y [15:0]: PW3 cycle control buffer.

0B5H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW3BH	PW3B15	PW3B14	PW3B13	PW3B12	PW3B11	PW3B10	PW3B9	PW3B8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0B4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW3BL	PW3B7	PW3B6	PW3B5	PW3B4	PW3B3	PW3B2	PW3B1	PW3B0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

After reset	0	0	0	0	0	0	0	0

PW3B [15:0]: PWM B dead band control buffer.

0B7H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW3DH	PW3D15	PW3D14	PW3D13	PW3D12	PW3D11	PW3D10	PW3D9	PW3D8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

0B6H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW3DL	PW3D7	PW3D6	PW3D5	PW3D4	PW3D3	PW3D2	PW3D1	PW3D0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PW3D [15:0]: PWM duty control buffer.

0CFH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW3A	PW3A7	PW3A6	PW3A5	PW3A4	PW3A3	PW3A2	PW3A1	PW3A0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PW3A [7:0]: PWM A dead band control buffer.

09BH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPM	PW3CM1	PW3CM0	PW2CM1	PW2CM0	PW1CM1	PW1CM0	OP1EN	OP0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit 2 **PW1CM0:** PWM1 output and CMP0 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 3 **PW1CM1:** PWM1 output and CMP1 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 4 **PW2CM0:** PWM2 output and CMP0 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 5 **PW2CM1:** PWM2 output and CMP1 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 6 **PW3CM0:** PWM3 output and CMP0 trigger synchronous control bit.
0 = Disable.
1 = Enable.

Bit 7 **PW3CM1:** PWM3 output and CMP1 trigger synchronous control bit.
0 = Disable.
1 = Enable.

PWnEN	PWnCM1	PWnCM0	PWM Output Control Condition	Remark
0	-	-	No PWM	
1	0	0	Control by PWnEN bit	
1	0	1	Control by CMP0 Trigger	
1	1	0	Control by CMP1 Trigger	
1	1	1	Control by CMP0, CMP1 Trigger	If PWM output and stop collision, The “stop” condition is the highest prority.

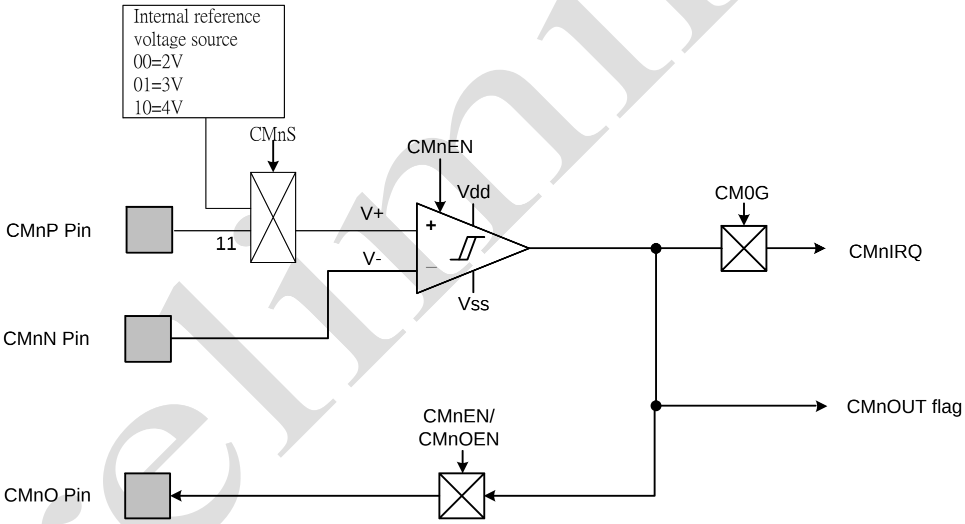
11 ANALOG COMPARATOR

11.1 OVERVIEW

The micro-controller builds in two comparator functions. When the positive input voltage is greater than the negative input voltage, the comparator output is high. When the positive input voltage is smaller than the negative input voltage, the comparator output is low. Comparator positive voltage is from internal 2V/3V/4V/CMnP (CMnS[1:0]). There is a programmable direction function to decide comparator trigger edge for indicator function. The comparator has flag indicator, interrupt function and IDLE Mode weak-up function for different application. The main purposes of comparator are as following.

- Programmable internal reference voltage connected to comparator's positive terminal.
- Comparator output function.
- Programmable trigger direction.
- Interrupt function.
- IDLE Mode wake-up function.

The Comparator pins are shared with GPIO controlled by CMnEN bit. When CMnEN=1, CMnN/CMnP pin is enabled connected to Comparator negative terminal. CMnOEN controls Comparator output connected to GPIO or not. When CMnOEN=1, Comparator output terminal is connected to GPIO pins and isolate GPIO function.



The internal reference has three steps including 2V/3V/4V controlled by CMnS[1:0] bits. Comparator pins selection table is as following.

Comparator No.	CMnEN	Comparator Negative Pin	Comparator Positive Pin				Comparator Output Pin	
			0	1	10	11	CMnOEN=0	CMnOEN=1
CMP0	CM0EN=0	All pins are GPIO mode. Comparator is disabled.						
	CM0EN=1	CM0N	2V	3V	4V	CM0P	GPIO	CM0O
CMP1	CM1EN=0	All pins are GPIO mode. Comparator is disabled.						
	CM1EN=1	CM1N	2V	3V	4V	CM1P	GPIO	CM1O

11.2 COMPARATOR OUTPUT FUNCTION

The comparator output signal is the source of comparator output function. The comparator output function includes:

- **CMnOUT output flag:** The comparator output signal is connected to CMnOUT flag directly. CMnOUT bit responses comparator status immediately. Program reads comparator status from CMnOUT bit.
- **Comparator extern pin output function:** The comparator output status can output to CMnO pin controlled by CMnOEN bit. When CMnOEN=0, the comparator output pin is GPIO mode. If CMnOEN=1, CMnO pin outputs comparator output status and isolates GPIO mode.
- **Comparator edge trigger and interrupt function:** The comparator builds in interrupt function, and the trigger edge is programmable. CMnG[1:0] bit controls comparator trigger edge. When the edge trigger condition occurs, CMPnF will be set automatically. To clear CMPnF bit must be through program. If CMnEN=1, program counter will be pointed to interrupt vector to execute interrupt service routine as CMPnF is setting. When the trigger edge direction is equal to interrupt trigger condition, the system will execute interrupt operation immediately.
- **Comparator IDLE Mode Wake-up function:** The comparator's wake-up function only support IDLE Mode (interrupt needs enable), not STOP Mode. If the trigger edge condition (comparator output status exchanging) is found, the system will be wake-up from IDLE Mode. Because in interrupt trigger condition, the CMnF is set as "1". Of course the interrupt routine is executed if the interrupt function enabled.

11.3 COMPARATOR REGISTER

09CH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMP0M	CM0EN	-	CM0S1	CM0S0	CM0OEN	CM0OUT	CM0G1	CM0G0
Read/Write	R/W	-	R/W	R/W	R/W	R	R/W	R/W
After reset	0	-	0	0	0	0	0	0

Bit 7 **CM0EN:** Comparator 0 control bit.
0 = Disable. CM0P/CM0N pins are GPIO mode.
1 = Enable. CM0P/CM0N pins are CMP input pins.

Bit [5:4] **CM0S[1:0]:** CMP0 positive input voltage control bit.

CM0S1	CM0S0	Positive input control bit.
0	0	2.0V
0	1	3.0V
1	0	4.0V
1	1	CM0P

Bit 3 **CM0OEN:** Comparator 0 output pin control bit.
0 = Disable. CM0O is GPIO mode.
1 = Enable. CM0O is comparator output pin and isolate GPIO function.

Bit 2 **CM0OUT:** Comparator 0 output flag bit.
0 = CM0P voltage is less than CM0N voltage.
1 = CM0P voltage is larger than CM0N voltage.

Bit [1:0] **CM0G[1:0]:** Comparator interrupt trigger direction control bit.
00 = Reserved.
01 = Rising edge trigger. CM0P > CM0N.
10 = Falling edge trigger. CM0P < CM0N.
11 = Both rising and falling edge trigger (Level change trigger).

09DH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMP1M	CM1EN	-	CM1S1	CM1S0	CM1OEN	CM1OUT	CM1G1	CM1G0
Read/Write	R/W	-	R/W	R/W	R/W	R	R/W	R/W
After reset	0	-	0	0	0	0	0	0

Bit 7 **CM1EN:** Comparator 1 control bit.
0 = Disable. CM1P/CM1N pins are GPIO mode.
1 = Enable. CM1P/CM1N pins are CMP input pins.

Bit [5:4] **CM1S[1:0]:** CMP1 positive input voltage control bit

CM1S1	CM1S0	Positive input control bit.
0	0	2.0V
0	1	3.0V
1	0	4.0V
1	1	CM1P

Bit 3 **CM1OEN:** Comparator 1 output pin control bit.
0 = Disable. CM1O is GPIO mode.
1 = Enable. CM1O is comparator output pin and isolate GPIO function.

Bit 2 **CM1OUT:** Comparator 1 output flag bit.
0 = CM1P voltage is less than CM1N voltage.
1 = CM1P voltage is larger than CM1N voltage.

Bit [1:0] **CM1G[1:0]:** Comparator interrupt trigger direction control bit.
00 = Reserved.
01 = Rising edge trigger. CM1P > CM1N.
10 = Falling edge trigger. CM1P < CM1N.
11 = Both rising and falling edge trigger (Level change trigger).

0CEH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMPT					CM1T1	CM1T0	CM0T1	CM0T0
Read/Write					R/W	R/W	R/W	R/W
After Reset					0	0	0	0

Bit [1:0] **CM0T[1:0]:** CMP0 with PWM trigger select bits.

CM0T1	CM0T0	PWM Synchronous Trigger Operation
0	0	CMP with PWM output is not related
0	1	CM0IRQ=1=> PWM stop
1	0	CM0P > CM0N (Rising edge trigger) => PWM output CM0P < CM0N (Falling edge trigger) => PWM stop
1	1	CM0P < CM0N(Falling edge trigger) => PWM output CM0P > CM0N (Rising edge trigger) => PWM stop

Bit [3:2] **CM1T[1:0]:** CMP1 with PWM trigger select bits.

CM1T1	CM1T0	PWM Synchronous Trigger Operation
0	0	CMP with PWM output is not related
0	1	CM1IRQ=1=> PWM stop
1	0	CM1P > CM1N (Rising edge trigger) => PWM output CM1P < CM1N (Falling edge trigger) => PWM stop
1	1	CM1P < CM1N(Falling edge trigger) => PWM output CM1P > CM1N (Rising edge trigger) => PWM stop

09FH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3CON	P3CON7	P3CON6	P3CON5	P3CON4	P3CON3	P3CON2	P3CON1	P3CON0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **P3CON [7:0]**: P3 configuration control bit.
0 = P3 can be analog input pin (OP/CMP input pin) or digital GPIO pin.
1 = P3 is pure analog input pin and can't be a digital GPIO pin.

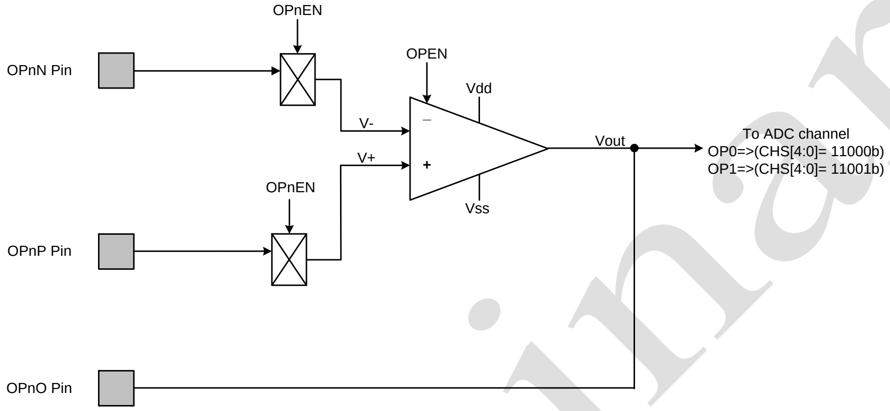
09EH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2CON	P2CON7	P2CON6	P2CON5	P2CON4	-	-	-	-
Read/Write	R/W	R/W	R/W	R/W	-	-	-	-
After reset	0	0	0	0	-	-	-	-

Bit [7:4] **P2CON [7:4]**: P4 configuration control bit.
0 = P2 can be analog input pin (OP/CMP input pin) or digital GPIO pin.
1 = P2 is pure analog input pin and can't be a digital GPIO pin.

12 OPERATIONAL AMPLIFIER

12.1 OVERVIEW

The micro-controller builds in two operational amplifiers (OP-Amp). The OP-Amp power range is Vss~Vdd. OP-Amp input signal and output voltage are within the voltage range. The OP-Amp output pin is programmable to connect with ADC input channel for voltage measurement. The OP AMP pins are shared with GPIO controlled by OP0EN/OP1EN bit. When OPEN=0, OP AMP pins are GPIO mode. When OPEN=1, GPIO pins switch to OP AMP and isolate GPIO path.



OP pins selection table is as following.

OP No.	OPnEN	OP Positive Pin	OP Negative Pin	OP Output Pin
OP0	OP0EN=0	All pins are GPIO mode.		
	OP0EN=1	OP0P (Vin+)	OP0N (Vin-)	OP0O (Vout)
OP1	OP1EN=0	All pins are GPIO mode.		
	OP1EN=1	OP1P (Vin+)	OP1N (Vin-)	OP1O (Vout)

OP0/OP1 output pins are also connected to ADC internal AIN12 and AIN13 channel =>CHS[4:0].

12.2 OP AMP REGISTER

09BH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPM	PW3CM1	PW3CM0	PW2CM1	PW2CM0	PW1CM1	PW1CM0	OP1EN	OP0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit 1 **OP1EN**: OP Amp enable bit.
0 = Disable. OP Amp disable, OP1O/OP1P/OP1N pins are GPIO mode.
1 = Enable. OP Amp enables, OP1O/OP1P/OP1N pins are OP Amp input and output pins.

Bit 0 **OP0EN**: OP Amp enable bit.
0 = Disable. OP Amp disable, OP0O/OP0P/OP0N pins are GPIO mode.
1 = Enable. OP Amp enables, OP0O/OP0P/OP0N pins are OP Amp input and output pins.

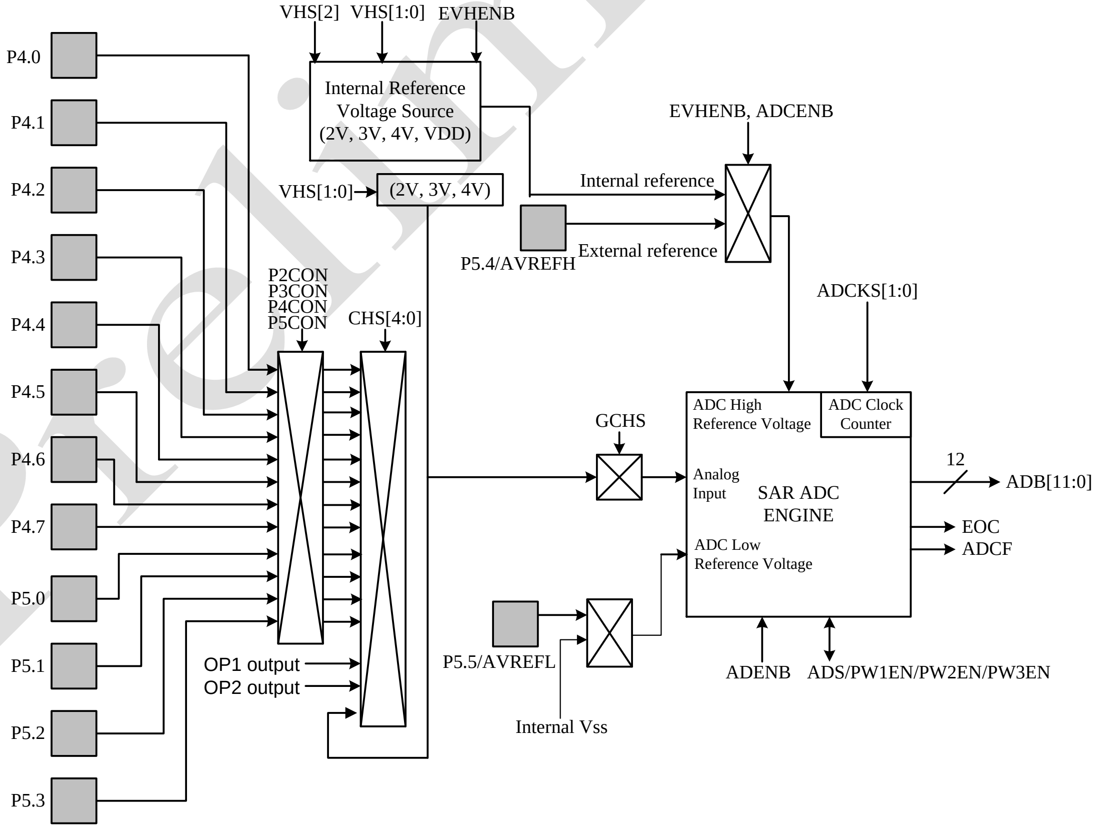
09FH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3CON	P3CON7	P3CON6	P3CON5	P3CON4	P3CON3	P3CON2	P3CON1	P3CON0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **P3CON [7:0]**: P3 configuration control bit.
0 = P3 can be analog input pin (OP/CMP input pin) or digital GPIO pin.
1 = P3 is pure analog input pin and can’t be a digital GPIO pin.

13 12 CHANNEL ANALOG TO DIGITAL CONVERTER (ADC)

13.1 OVERVIEW

The analog to digital converter (ADC) is SAR structure with 12-input sources and up to 4096-step resolution to transfer analog signal into 12-bits digital buffers. The ADC builds in 12-channel input source (AIN0~AIN11) to measure 12 different analog signal sources controlled by CHS[4:0] and GCHS bits. The ADC resolution is 12-bit. The ADC converting rate can be selected by ADCKS[1:0] bits to decide ADC converting time. The ADC reference high voltage includes 5 sources controlled by VREFH register. Four internal power source including Vdd, 4V, 3V and 2V. The other one is external reference voltage input pin from AVREFH pin. The ADC reference low voltage includes 2 sources controlled by VREFH register. One is internal Vss and the other one is external reference voltage input pin from AVREFL. The ADC builds in P2CON/P3CON/P4CON/P5CON registers to set pure analog input pin. It is necessary to set ADC input pin as input mode without pull-up resistor by program. After setup ADENB and ADS bits, the ADC starts to convert analog signal to digital data. When the conversion is complete, the ADC circuit will set EOC and ADCF bits to “1” and the digital data outputs in ADB and ADR registers. Besides ADS bit can start to convert analog signal, PW1EN/ PW2EN/ PW3EN also have convert analog signal ADC function. If the EADC = 1, the ADC interrupt request occurs and executes interrupt service routine when ADCF = 1 after ADC converting. If ADC interrupt function is enabled (EADC=1), the system will execute interrupt procedure. The interrupt procedure is system program counter points to interrupt vector and executes interrupt service routine after ADC converting. Clear ADCF by program is necessary in interrupt procedure. ADC can work in idle mode. After ADC operating, the system would be waked up from green mode to normal mode if interrupt enable.



13.2 ADC MODE REGISTER

ADM is ADC mode control register to configure ADC configurations including ADC start, ADC channel selection, ADC high reference voltage source and ADC processing indicator…These configurations must be setup completely before starting ADC converting.

0D2H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADM	ADENB	ADS	EOC	CHS4	CHS3	CHS2	CHS1	CHS0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit 7 **ADENB:** ADC control bit. **In stop mode, disable ADC to reduce power consumption.**
0 = Disable
1 = Enable

Bit 6 **ADS:** ADC start control bit. **ADS bit is cleared after ADC processing automatically.**
0 = ADC converting stops.
1 = Start to execute ADC converting.

Bit 5 **EOC:** ADC status bit.
0 = ADC progressing.
1 = End of converting and reset ADS bit.

Bit [4:0] **CHS[4:0]:** ADC input channel select bit.
00000 = CT0/AIN0, 00001 = CT1/AIN1, 00010 = CT2/AIN2, 00011 = CT3/AIN3, 00100 = CT4/AIN4,
00101 = CT5/AIN5, 00110 = CT6/AIN6, 00111 = CT7/AIN7, 01000 = CT8/AIN8, 01001 = CT9/AIN9,
01010 = CT10/AIN10, 01011 = CT11/AIN11, 01100 = CT12, 01101 = CT13, 01110 = CT14, 01111 = CT15,
10000 = CT16, 10001 = CT17, 10010 = CT18, 10011 = CT19, 10100 = CT20, 10101 = CT21,
10110 = CT22, 10111 = CT23, 11000 = AIN12, 11001 = AIN13, 11010 = AIN14, others = Reserved.

AIN12, AIN13 channels are OP1 and OP2-Amp output terminal. The AIN14 is internal 2V or 3V or 4V input channel. There is no any input pin from outside. In this time ADC reference voltage must be internal VDD and External voltage, not internal 2V or 3V or 4V. AIN14 can be a good battery detector for battery system. To select appropriate internal AVREFH level and compare value, a high performance and cheaper low battery detector is built in the system.

ADR register includes ADC mode control and ADC low-nibble data buffer. ADC configurations including ADC clock rate. These configurations must be setup completely before starting ADC converting.

0D4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADR	TCHEN	GCHS	ADCKS1	ADCKS0	ADB3	ADB2	ADB1	ADB0
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
After reset	0	0	0	0	-	-	-	-

Bit 7 **TCHEN:** Capacitive Sensing function control bit.
0 = Disable Capacitive Sensing function. CA1 and CA2 are GPIO mode (P54 and P55).
1 = Enable Capacitive Sensing function. CA1 and CA2 are analogy pins.

Bit 6 **GCHS:** ADC global channel select bit.
0 = Disable AIN channel
1 = Enable AIN channel

Bit [5:4] **ADCKS1, ADCKS0:** ADC's clock source select bit.
00 = Fhosc/16, 01 = Fhosc/8, 10 = Fhosc/1, 11 = Fhosc/2

13.3 ADC DATA BUFFER REGISTERS

ADC data buffer is 12-bit length to store ADC converter result. The high byte is ADB register, and the low-nibble is ADR[3:0] bits. The ADB register is only 8-bit register including bit 4~bit11 ADC data. To combine ADB register and the low-nibble of ADR will get full 12-bit ADC data buffer. The ADC data buffer is a read-only register and the initial status is unknown after system reset.

➢ **ADB[11:0]: In 12-bit ADC mode, the ADC data is stored in ADB and ADR registers.**

0D3H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADB	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4
Read/Write	R	R	R	R	R	R	R	R
After reset	-	-	-	-	-	-	-	-

Bit [7:0] **ADB [11:4]** = ADC Result Bit [11:4] in 12-bit ADC resolution mode

0D4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADR	TCHEN	GCHS	ADCKS1	ADCKS0	ADB3	ADB2	ADB1	ADB0
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
After reset	0	0	0	0	-	-	-	-

Bit [3:0] **ADB [3:0]** = ADC Result Bit [3:0] in 12-bit ADC resolution mode

The AIN input voltage v.s. ADB output data

AIN n	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
0/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	0
1/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	1
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
4094/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	0
4095/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	1

✸ ***Note: The initial status of ADC data buffer including ADB register and ADR low-nibble after the system reset is unknown.***

13.4 ADC REFERENCE VOLTAGE REGISTERS

The ADC builds in five high reference voltage source controlled through VREFH register. There are one external voltage source and four internal voltage source (VDD, 4V, 3V, 2V). When EVHENB bit is “1”, ADC reference voltage is external voltage source from AVREFH/P5.4. The ADC reference low voltage includes two sources controlled by EVLENB bit. One is internal Vss (EVLENB =0), and the other one is external reference voltage input pin from AVREFL/P5.5 (EVLENB=1). If EVHENB bit is “0”, ADC reference high voltage is from internal voltage source selected by VHS[1:0] bits. If VHS[1:0] is “11”, ADC reference high voltage is VDD. If VHS[1:0] is “10”, ADC reference high voltage is 4V. If VHS[1:0] is “01”, ADC reference high voltage is 3V. If VHS[1:0] is “00”, ADC reference high voltage is 2V.The limitation of internal high reference voltage application is VDD can’t below each of internal high voltage level, or the level is equal to VDD

0D5H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VREFH	EVHENB	EVLENB	-	ADPWS	-	VHS2	VHS1	VHS0
Read/Write	R/W	R/W	-	R/W	-	R/W	R/W	R/W
After reset	0	0	-	0	-	0	0	0

Bit 7 **EVHENB**: ADC internal reference high voltage control bit.
0 = Enable ADC internal VREFH function. AVREFH pin is GPIO.
1 = Disable ADC internal VREFH function. AVREFH pin is external AVREFH input pin.

Bit 6 **EVLENB**: ADC internal reference low voltage control bit.
0 = Enable ADC internal VREFL function. AVREFL pin is GPIO.
1 = Disable ADC internal VREFL function. AVREFL pin is external AVREFL input pin.

Bit 4 **ADPWS**: PWM trigger ADC start control bit.
0 = Disable PWM trigger ADC start.
1 = Enable PWM trigger ADC start.

Bit 2 **VHS[2]:** ADC internal reference high voltage select bit for AIN14.
0 = ADC internal VREFH function is depend on VHS[1:0].
1 = ADC internal VREFH function is internal VDD.

Bit [1:0] **VHS[1:0]:** ADC internal reference high voltage select bits.

VHS1	VHS0	Internal reference high voltage
0	0	2.0V
0	1	3.0V
1	0	4.0V
1	1	VDD

✸ ***Note:***

1. ***The AVREFH level must be between the VDD and AVREFL + 2.0V.***
2. ***The AVREFL level must be between the VSS and AVREFH - 2.0V.***
3. ***If AIN14 channel is selected as internal 2V or 3V or 4V input channel. There is no any input pin from outside. In this time ADC reference high voltage must be internal VDD or External voltage, not internal 2V/3V/4V.***

13.5 ADC OPERATION DESCRIPTION AND NOTIC

13.5.1 ADC SIGNAL FORMAT

ADC sampling voltage range is limited by high/low reference voltage. The ADC low reference voltage includes internal Vss and external reference voltage source from P5.5/AVREFL controlled by EVLENB bit. If EVLENB=0, ADC reference low voltage is from internal voltage source. If EVLENB=1, ADC reference low voltage is from external voltage source (P5.5/AVREFL). The ADC high reference voltage includes internal Vdd/4V/3V/2V and external reference voltage source from P5.4/AVREFH pin controlled by EVHENB bit. If EVHENB=0, ADC reference high voltage is from internal voltage source. If EVHENB=1, ADC reference high voltage is from external voltage source (P5.4/AVREFH). ADC reference voltage range limitation is **"(ADC high reference voltage – low reference voltage) ≧ 2V"**. So **ADC high reference voltage range is AVREFL+2V ~ Vdd**. The range is ADC external high reference voltage range. **ADC low reference voltage range is VSS ~ AVREFH-2V**. The range is ADC external high reference voltage range.

- **ADC Internal Low Reference Voltage = 0V. (EVLENB=0)**
- **ADC External Low Reference Voltage = VSS ~ AVREFH-2V. (EVLENB=1)**
- **ADC Internal High Reference Voltage = Vdd/4V/3V/2V. (EVHENB=0)**
- **ADC External High Reference Voltage = AVREFL+2V ~ Vdd. (EVHENB=1)**

ADC sampled input signal voltage must be from ADC low reference voltage to ADC high reference. If the ADC input signal voltage is over the range, the ADC converting result is error (full scale or zero).

- **ADC Low Reference Voltage ≦ ADC Sampled Input Voltage ≦ ADC High Reference Voltage**

13.5.2 ADC CONVERTING TIME

The ADC converting time is from ADS=1 (Start to ADC convert) to EOC=1 (End of ADC convert). The converting time duration is depend on ADC clock rate. 12-bit ADC's converting time is 1/(ADC clock /4)*16 sec. ADC clock source is Fhosc and includes Fhosc/1, Fhosc/2, Fhosc/8 and Fhosc/16 controlled by ADCKS[1:0] bits.
The ADC converting time affects ADC performance. If input high rate analog signal, it is necessary to select a high ADC converting rate. If the ADC converting time is slower than analog signal variation rate, the ADC result would be error. So to select a correct ADC clock rate to decide a right ADC converting rate is very important.

12-bit ADC conversion time = 1/(ADC clock rate/4)*16 sec

ADCKS1, ADCKS0	ADC Clock Rate	Fhosc=16MHz		Fhosc=32MHz	
		ADC Converting time	ADC Converting Rate	ADC Converting time	ADC Converting Rate
00	Fhosc/16	1/(16MHz/16/4)*16 = 64 us	15.625KHz	1/(32MHz/16/4)*16 = 32 us	31.25KHz
01	Fhosc/8	1/(16MHz/8/4)*16 = 32 us	31.25KHz	1/(32MHz/8/4)*16 = 16 us	62.5KHz
10	Fhosc	1/(16MHz/4)*16 = 4 us	250KHz	1/(32MHz/4)*16 = 2 us	500KHz
11	Fhosc/2	1/(16MHz/2/4)*16 = 8 us	125KHz	1/(32MHz/2/4)*16 = 4 us	250KHz

13.5.3 ADC PIN CONFIGURATION

ADC input channels are shared with Port2, Port3, Port4 and Port5. ADC channel selection is through CHS[4:0] bit. CHS[4:0] value points to the ADC input channel directly. CHS[4:0]=0000 selects AIN0. CHS[4:0]=0001 selects AIN1…… Only one pin of Port2, Port3, Port4 and Port5 can be configured as ADC input in the same time. The pins of Port2, Port3, Port4 and Port5 configured as ADC input channel must be set input mode, disable internal pull-up and enable P2CON/P3CON/P4CON/P5CON first by program. After selecting ADC input channel through CHS[4:0], set GCHS bit as "1" to enable ADC channel function.

- **The GPIO mode of ADC input channels must be set as input mode.**
- **The internal pull-up resistor of ADC input channels must be disabled.**
- **P2CON, P3CON, P4CON and P5CON bits of ADC input channel must be set.**

The P5.4/CA1 can be ADC external high reference voltage input pin when EVHENB=1. In the condition, P5.4 GPIO mode must be set as input mode and disable internal pull-up resistor. The P5.5/CA2 can be ADC external high reference voltage input pin when EVLENB=1. In the condition, P5.5 GPIO mode must be set as input mode and disable internal pull-up resistor.

- **The GPIO mode of ADC external high/low reference voltage input pin must be set as input mode.**
- **The internal pull-up resistor of ADC external high/low reference voltage input pin must be disabled.**

ADC input pins are shared with digital I/O pins. Connect an analog signal to COMS digital input pin, especially, the analog signal level is about 1/2 VDD will cause extra current leakage. In the power down mode, the above leakage current will be a big problem. Unfortunately, if users connect more than one analog input signal to Port2, Port3, Port4 or Port5 will encounter above current leakage situation. P2CON is Port2 configuration register. Write "1" into P2CON [7:4] will configure related Port2 pin as pure analog input pin to avoid current leakage. P3CON is Port3 configuration register. Write "1" into P3CON [7:0] will configure related Port3 pin as pure analog input pin to avoid current leakage. P4CON is Port4 configuration register. Write "1" into P4CON [7:0] will configure related port4 pin as pure analog input pin to avoid current leakage. P5CON is Port5 configuration register. Write "1" into P5CON [5:0] will configure related Port5 pin as pure analog input pin to avoid current leakage.

0D7H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P5CON	-	-	P5CON5	P5CON4	P5CON3	P5CON2	P5CON1	P5CON0
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	R/W
After reset	-	-	0	0	0	0	0	0

Bit [5:0] **P5CON [5:0]**: P5 configuration control bit.
0 = P5 can be analog input pin (ADC input pin) or digital GPIO pin.
1 = P5 is pure analog input pin and can't be a digital GPIO pin.

0D6H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4CON	P4CON7	P4CON6	P4CON5	P4CON4	P4CON3	P4CON2	P4CON1	P4CON0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **P4CON [7:0]**: P4 configuration control bit.
0 = P4 can be analog input pin (ADC input pin) or digital GPIO pin.
1 = P4 is pure analog input pin and can't be a digital GPIO pin.

09FH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3CON	P3CON7	P3CON6	P3CON5	P3CON4	P3CON3	P3CON2	P3CON1	P3CON0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **P3CON [7:0]**: P3 configuration control bit.
0 = P3 can be analog input pin (OP/CMP input pin) or digital GPIO pin.
1 = P3 is pure analog input pin and can't be a digital GPIO pin.

09EH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2CON	P2CON7	P2CON6	P2CON5	P2CON4	-	-	-	-
Read/Write	R/W	R/W	R/W	R/W	-	-	-	-
After reset	0	0	0	0	-	-	-	-

Bit [7:4] **P2CON [7:4]**: P2 configuration control bit.

0 = P2 can be analog input pin (OP/CMP input pin) or digital GPIO pin.
1 = P2 is pure analog input pin and can’t be a digital GPIO pin.

✸ ***Note: When ADC pin is general I/O mode, the bit of P2CON, P3CON, P4CON and P5CON must be set to “0”, or the digital I/O signal would be isolated.***

14 Universal Asynchronous Receiver/Transmitter (UART)

14.1 OVERVIEW

The UART provides a flexible full-duplex synchronous/asynchronous receiver/transmitter. The serial interface is applied to low speed data transfer and communicate with low speed peripheral devices. It can operate in four modes (one synchronous and three asynchronous). Mode0 is a shift register mode. It operates as synchronous transmitter/ receiver. In Mode1~Mode3 the UART operates as asynchronous transmitter/receiver with 8-bit or 9-bit data. The transfer format has start bit, 8-bit/ 9-bit data and stop bit. Transmission is started by writing to the S0BUF register. After reception, input data are available after completion of the reception in the S0BUF register. TB80/RB80 bit can be used as the 9th bit for transmission and reception in 9-bit UART mode. Programmable baud rate supports different speed peripheral devices.

The UART features include the following:

- **Full-duplex, 2-wire synchronous/asynchronous data transfer.**
- **Programmable baud rate.**
- **8-bit shift register**: operates as synchronous transmitter/receiver
- **8-bit / 9-bit UART:** operates as asynchronous transmitter/receiver with 8 or 9-bit data bits and programmable baud rate.

14.2 UART OPERATION

The UART UTX(P0.5) and URX(P0.6) pins are shared with GPIO. In synchronous mode, the UTX/URX shared pins must set output high by software. In asynchronous mode (8-bit/9-bit UART), the UTX shared pins must set output high and URX set input high by software. Thus, URX/UTX pins will transfers to UART purpose. When UART disables, the UART pins returns to GPIO last status.

The UTX/URX pins also support open-drain structure. The open-drain option is controlled by P05OC and P06OC bit. When P05OC=0/P06OC=0, disable UTX/URX open-drain structure. When P05OC=1/P06OC=1, enable UTX/URX open-drain structure. If enable open-drain structure, UTX/URX pin must set high level (IO mode control will be ignored) and need external pull-up resistor.

The UART supports interrupt function. ES0 is UART transfer interrupt function control bit. ES0=0, disable both transmitter and receiver interrupt function. ES0=1, enable both UART transmitter and receiver interrupt function. When UART interrupt function enable, the program counter points to interrupt vector (ORG 0x23) to do UART interrupt service routine after UART operating. TI0/RI0 is UART interrupt request flag, and also to be the UART operating status indicator when interrupt is disabled. TI0 and RI0 must clear by software.

UART provides four operating mode (one synchronous and three asynchronous) controlled by S0CON register. These modes can be support in different baud rate and communication protocols.

SM1	SM0	Mode	Synchronization	Clock Rate	Start Bit	Data Bits	Stop Bit	UART pins' mode and data
0	0	0	Synchronous	Fcpu/12	X	8	X	UTX pin: P05M=1 and P05=1. URX pin: Transmitter: P06M=1 and P06=1 Receiver: P06M=0 and P06=1
0	1	1	Asynchronous	Baud Rate Generator or T1 Overflow Rate	1	8	1	UTX pin: P05M=1 and P05=1. URX pin: P06M=0.
1	0	2	Asynchronous	Fcpu/64 or Fcpu/32	1	9	1	
1	1	3	Asynchronous	Baud Rate Generator or T1 Overflow Rate	1	9	1	

14.2.1 MODE0: SYNCHRONOUS 8-BIT SHIFT REGISTER

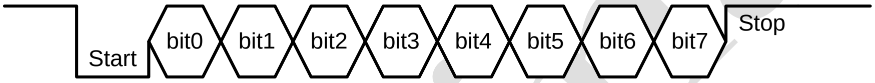
Mode0 is a shift register mode. It operates as synchronous transmitter/receiver. The UTX/P0.5 pin output shift clock for both transmit and receive condition. The URX/P0.6 pin is used to transmit and receive data. 8-bit data will be transmit and receive with LSB first. The baud rate is Fcpu/12. Data transmission is started by writing data to S0BUF register. In the end of the 8th bit transmission, the TI0 flag is set. Data reception is controlled by REN0 bit. When REN0=1, data transmission starts and the RI0 flag is set at the end of the 8th bit reception.

14.2.2 MODE1: 8-Bit UART with Variable Baud Rate

Mode1 supports an asynchronous 8-bit UART with variable baud rate. The transfer format includes 1 start bit, 8 data bits (LSB first) and 1 stop bit. Data is transmitted by UTX/P0.5 pin and received by URX/P0.6 pin. The baud rate clock source can be baud rate generator or T1 overflow controlled by BD bit. When BD=0, the baud rate clock source is from T1 overflow. When BD=0, the baud rate clock source is from baud rate generator controlled by S0RELH and S0RELL. Additionally, the baud rate can be doubled by SMOD bit.

Data transmission is started by loading data to S0BUF register. After transmission configuration, load transmitted data into S0BUF, and then UART starts to transmit the pocket. The TI0 flag is set at the beginning of the stop bit.

Data reception is controlled by REN0 bit. When REN0=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX detects the falling edge of start bit, and then the RI0 flag is set in the middle of a stop bit. Until reception completion, input data is stored in S0BUF register and the stop bit is stored in RB80.



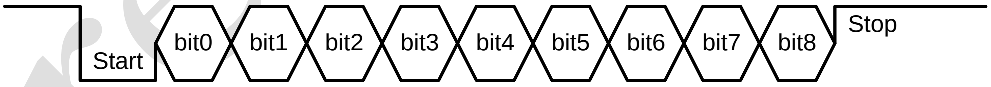
8-Bit UART Transfer Format

14.2.3 MODE2: 9-Bit UART with Fixed Baud Rate

Mode2 supports an asynchronous 9-bit UART with fixed baud rate. The transfer format includes 1 start bit, 9 data bits (LSB first) and 1 stop bit. Data is transmitted by UTX/P0.5 pin and received by URX/P0.6 pin. The baud rate clock source is fixed to Fcpu/64 or Fcpu/32 and is controlled by SMOD bit. When SMOD=0, baud rate is Fcpu/64. When SMOD=1, baud rate is Fcpu/32.

Data transmission is started by loading data to S0BUF register. After transmission configuration, load transmitted data into S0BUF, and then UART starts to transmit the pocket. The 9th data bit is taken from TB80. The TI0 flag is set at the beginning of the stop bit.

Data reception is controlled by REN0 bit. When REN0=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX detects the falling edge of start bit, and then the RI0 flag is set in the middle of a stop bit. Until reception completion, lower 8-bit input data is stored in S0BUF register and the 9th bit is stored in RB80.



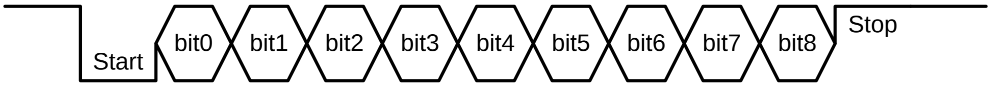
9-Bit UART Transfer Format

14.2.4 MODE3: 9-Bit UART with Variable Baud Rate

Mode3 supports an asynchronous 9-bit UART with variable baud rate. The transfer format includes 1 start bit, 9 data bits (LSB first) and 1 stop bit. Data is transmitted by UTX/P0.5 pin and received by URX/P0.6 pin. The different between Mode2 and Mode3 is baud rate selection. In the Mode3, the baud rate clock source can be baud rate generator or T1 overflow controlled by BD bit. When BD=0, the baud rate clock source is from T1 overflow. When BD=0, the baud rate clock source is from baud rate generator controlled by S0RELH and S0RELL. Additionally, the baud rate can be doubled by SMOD bit.

Data transmission is started by loading data to S0BUF register. After transmission configuration, load transmitted data into S0BUF, and then UART starts to transmit the pocket. The 9th data bit is taken from TB80. The TI0 flag is set at the beginning of the stop bit.

Data reception is controlled by REN0 bit. When REN0=1, data reception function is enabled. Data reception starts by receiving the start bit for master terminal, URX detects the falling edge of start bit, and then the RI0 flag is set in the middle of a stop bit. Until reception completion, lower 8-bit input data is stored in S0BUF register and the 9th bit is stored in RB80.



9-Bit UART Transfer Format

14.3 VARIABLE BAUD RATE BAUD RATE GENERATOR

UART variable baud includes 2 clock sources from Timer 1 overflow time and internal baud rate generator, and it controlled by BD bit in Mode1 and Mode3. When BD=0, baud rate clock is from T1 overflow signal. When BD=1, baud rate clock is from internal baud rate generator with 16-bit buffer. In Mode1 and Mode3, SMOD is baud rate doubler bit. When SMOD=1, the baud rate is doubled.

Timer1 must set in 8-bit auto-reload mode and frequency is controlled by reload register TH1 (The detail is shown in T1 section). The baud ate is generated by T1 overflow time and SMOD bit.

The UART baud rate clock source is T1 overflow rate. The equation is as following.

UART Baud Rate = (2^SMOD)*Fcpu/32*(T1 Overflow Rate)…bps

In internal baud rate generator, the clock source is generated from system clock (Fcpu) and through UART 10-bit buffer (S0REL). SMOD doubler is also used in this case.

The UART baud rate clock source is Fcpu. The equation is as following.

UART Baud Rate = (2^SMOD)*Fcpu/(64*(1024-SOREL))…bps

Fhosc = 32MHz

Baud Rate	Fcpu (MHz)	SMOD	S0REL (Hex)	Accuracy (%)
4800	32	0	398	-0.16%
9600	32	0	3CC	-0.16%
19200	32	0	3E6	-0.16%
38400	32	0	3F3	-0.16%
56000	32	1	3EE	0.79%
57600	32	1	3EF	-2.12%
115200	32	1	3F7	3.55%
128000	32	1	3F8	2.34%
250000	32	1	3FC	0.00%
500000	32	1	3FE	0.00%
1000000	32	1	3FF	0.00%

0BAH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S0RELH	-	-	-	-	-	-	S0RELH1	S0RELH0
Read/Write	-	-	-	-	-	-	R/W	R/W
After reset	-	-	-	-	-	-	1	1

0AAH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S0RELL	S0RELL7	S0RELL6	S0RELL5	S0RELL4	S0RELL3	S0RELL2	S0RELL1	S0RELL0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	0	1	1	0	0	1

Bit [7:0] **S0RELH, S0RELL:** UART Reload Register is used for UART baud rate generation. Only 10 bits are used. 8 bits from the S0RELL as lower bits and 2 bits from the S0RELH as higher bits.

✸ ***Note: When baud rate generator source is T1 overflow rate, the max counter value is 0XFB. (Only supports 0x00~0xFB). UART maximum baud rate is 1Mbps.***

14.4 MULTIPROCESSOR COMMUNICATION

UART supports multiprocessor communication between a master device and one or more slaver device in Mode2 and Mode3 (9-bit UART). The master identifies correct slavers by using the 9th data bit. When the communication starts, the master transmits a specific address byte with the 9th bit is set "1" to selected slavers, and then transmits a data byte with the 9th bit is set "0" in the following transmission.

Multiprocessor communication is controlled by SM20 bit. When SM20=0, disable multiprocessor communication. When SM20=1, enable multiprocessor communication. If SM20 is set, the UART reception interrupt is only generated when the 9th received bit is "1" (RB80). The slavers will compare received data with its own address data by software. If address byte is match, the slavers clear SM20 bit to enable interrupt function in the following data transmission. The slavers with unmatched address, their SM20 keep in "1" and will not generate interrupt in the following data transmission.

14.5 UART CONTROL REGISTER

098H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:6] **SM[1:0]:** UART mode select.
00 = Mode 0, Synchronous shift register.
01 = Mode1, 8-bit UART with variable Baud Rate
10 = Mode2, 9-bit UART with fixed Baud Rate
11 = Mode3, 9-bit UART with variable Baud Rate

Bit 5 **SM20:** Multiprocessor communication enable.
0 = Disable multiprocessor communication
1 = Enable multiprocessor communication.

Bit 4 **REN0:** UART receive control bit.
0 = Disable UART receive function.
1 = Enable UART receive function.

Bit 3 **TB80:** 9th transmission bit.
TB80 is the 9th transmission bit in Mode2 and Mode3. The state of this bit corresponds with the state of the 9th transmitted bit (e.g. parity check or multiprocessor communication). It is controlled by software

Bit 2 **RB80:** 9th receive bit.
RB80 is the 9th bit received in Mode2 and Mode3. It reflects the state of the 9th received bit. In Mode 1, if multiprocessor communication is enabled (sm20 = 0), this bit is the stop bit that was received. In Mode 0 this bit is not used

Bit 1 **TI0:** UART transmit interrupt request flag. It indicates completion of a serial transmission at UART. It is set by hardware at the end of bit 8 in mode 0 or at the beginning of a stop bit in other modes. It must be cleared by software.
0 = None UART transmit interrupt request.
1 = UART transmit interrupt request.

Bit 0 **RI0:** UART receive interrupt request flag. It is set by hardware after completion of a serial reception at UART. It is set by hardware at the end of bit 8 in mode 0 or in the middle of a stop bit in other modes. It must be cleared by software.
0 = None UART receive interrupt request.

1 = UART receive interrupt request.

099H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S0BUF	S0BUF7	S0BUF6	S0BUF5	S0BUF4	S0BUF3	S0BUF2	S0BUF1	S0BUF0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **S0BUF[7:0]:** Writing data to this register sets data in serial output buffer and starts the transmission through UART. Reading from the S0BUF reads data from the serial receive buffer.

087H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	SMOD	-	-	-	P2SEL	GF0	STOP	IDLE
Read/Write	R/W	-	-	-	R/W	R/W	R/W	R/W
After reset	0	-	-	-	1	0	0	0

Bit 7 **SMOD:** UART baud rate select (baud rate doubler).

Bit 3 **P2SEL:** High-order address byte configuration bit.
Chooses the higher byte of address (“memaddr[15:8]”) during MOVX @Ri operations; when 0, the “memaddr[15:8]” = “p2reg” when 1, the “memaddr[15:8]” =0X00. The “p2reg” is the contents of Port2 output register.

Bit 2 **GF0:** General Purpose Flag.

Bit 1 **STOP:** Stop mode control.
Setting this bit activates the Stop Mode. This bit is always read as 0.

Bit 0 **IDLE:** Idle mode control
Setting this bit activates the Idle Mode. This bit is always read as 0.

0D8H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S0CON2	BD	-	-	-	-	-	-	-
Read/Write	R/W	-	-	-	-	-	-	-
After reset	0	-	-	-	-	-	-	-

The MSB of this register is used by UART for baud rate generation.
Bit 7 **BD:** UART baud rate select (in modes 1 and 3) When 1, additional internal baud rate generator is used, otherwise Timer 1 overflow is used.

0E4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1OC	PW3EN	PW2EN	PW1EN	P06OC	P05OC	P13OC	P12OC	P11OC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [4:0] **P05OC~P06OC:** Open drain control bit.
0 = Disable.
1 = Enable open-drain structure.

15 Serial Peripheral Interface (SPI)

15.1 OVERVIEW

The SPI a serial communicate interface for data exchanging from one MCU to one MCU or other hardware peripherals. It is a simple 8-bit interface without a major definition of protocol, packet or control bits. The SPI transceiver includes three pins, clock (SCK), data input and data output ((MISO/ MOSI) to send data between master and slaver terminals. The SPI interface builds in 4-mode which are the clock idle status and the clock phases.

- **Full-duplex, 3-wire synchronous data transfer.**
- **Master (SCK is clock output) or Slave (SCK is clock input) operation.**
- **Seven SPI Master baud rates.**
- **Slave Clock rate up to Fhosc/8.**
- **8-bit data transmitted MSB first, LSB last.**
- **Serial clock with programmable polarity and phase.**
- **Master Mode fault error flag with MCU interrupt capability.**
- **Write collision flag protection.**

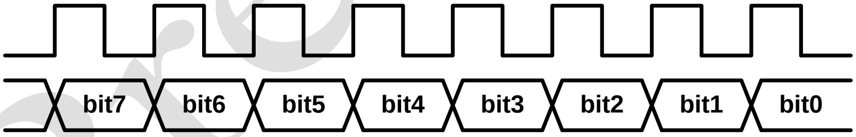
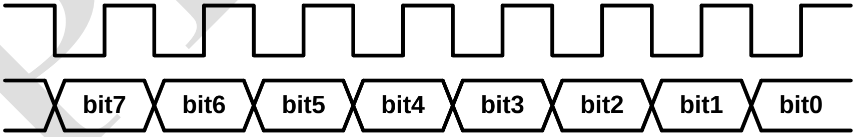
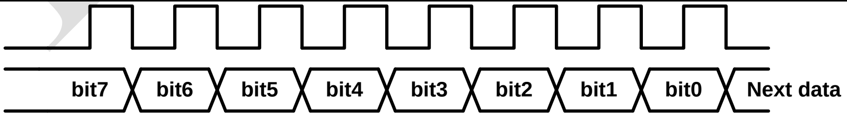
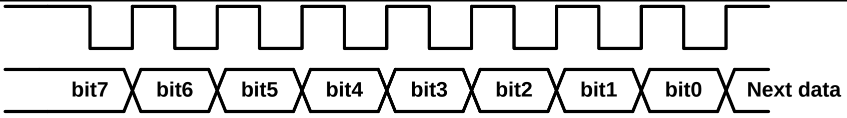
15.2 SPI OPERATION

The SPCON register can control SPI operating function, such as: transmit/receive, clock rate, data transfer direction, SPI clock idle status and clock control phase and enable this circuit. This SPI circuit will transmit or receive 8-bit data automatically by setting SPEN in SPCON register and write or read SPDAT register.

CPOL bit is designed to control SPI clock idle status. CPHA bit is designed to control the clock edge direction of data receive. CPOL and CPHA bits decide the SPI format. The SPI data transfer direction is MSB bit to LSB bit.

The SPI supports 4-mode format controlled by CPOL and CPHA bits. The edge direction is “Data Transfer Edge”. When setting rising edge that means to receive and transmit one bit data at SCK rising edge, and data transition is at SCK falling edge. When setting falling edge, that means to receive and transmit one bit data at SCK falling edge, and data transition is at SCK rising edge.

“CPHA” is the clock phase bit controls the phase of the clock on which data is sampled. When CPHA=1, the SCK first edge is for data transition, and receive and transmit data is at SCK 2nd edge. When CPHA=0, the 1st bit is fixed already, and the SCK first edge is to receive and transmit data. The SPI data transfer timing as following figure:

CPOL	CPHA	Diagrams	Description
0	1		SCK idle status = Low. The transfer first bit = MSB. SCK data transfer edge = Falling edge.
1	1		SCK idle status = High. The transfer first bit = MSB. SCK data transfer edge = Rising edge.
0	0		SCK idle status = Low. The transfer first bit = MSB. SCK data transfer edge = Rising edge.
1	0		SCK idle status = High. The transfer first bit = MSB. SCK data transfer edge = Falling edge.

SPI may be programmed to work as master or as slave device. In master mode (the MSTR bit of SPCON register is set) the SPI waits on write operation to SPDAT register. If writes operation to SPDAT register is done, transmission is started. Data shifts out on “MOSI” pin. Simultaneously, another data byte shifts in from the slave on master's “MISO” pin. SPI has to be configured as a slave by writing MSTR = 0 in SPCON register. Then it has to be enabled by setting the SPEN = 1. In slave mode the SPI waits on low level on “SSN” input. The “SSN” input must remain low until the transmission is completed. The beginning of transmission depends on the state of the CPHA bit of SPCON register. When CPHA is “0”, then the slave must begin driving its data before the first “SCK” edge, and a falling edge on the “SSN” input is used to start the transmission. When the CPHA bit is “1”, then the slave uses the first edge of “SCK” input as a transmission start signal.

The SPI supports interrupt function. ESPI is SPI interrupt function control bit. ESPI=0, disable SPI interrupt function. ESPI=1, enable SPI interrupt function. When SPI interrupt function enable, the program counter points to interrupt vector to do SPI interrupt service routine after SPI operating. SPIF is SPI interrupt request flag, and also to be the SPI operating status indicator when ESPI= 0, but cleared by reading the SPSTA register.
SPI builds in chip selection function to implement SPI multi-device mode. One master communicating with several slave devices in SPI bus, and the chip selection decides the pointed device. The chip selection pin is SSN pin.

The SPI pins also support open-drain structure. The open-drain option is controlled by P11OC~P13OC bits. When P11OC~P13OC=0, disable SPI open-drain structure. When P11OC~P13OC=1, enable SPI open-drain structure. If enable open-drain structure, SPI pins must be set input mode and need external pull-up resistor.

✸ ***Note: The first step of SPI operation is to setup the SPI pins’ mode. Must be set “input mode” in SCK/MOSI/MISO/SSN pins.***

15.3 SPI MODE REGISTER

0E2H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPCON	SPR2	SPEN	SSDIS	MSTR	CPOL	CPHA	SPR1	SPR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	1	0	1	0	0

The SPCON register is used in configuration of the SPI. It controls the master clock output rate (bits SPR0, SPR1 and SPR2), the clock polarity (CPOL) and phase (CPHA), configures the SPI either as master or slave (MSTR bit), enables or disables the “SSN” input (SSDIS bit) and enables or disables the whole SPI component (SPEN bit).

Bit 7,[1:0] **SPR[2:0]:** Serial Peripheral Rate in master mode. **These 3-bits are workless when MSTR=0.**

SPR2	SPR1	SPR0	Serial Peripheral Rate
0	0	0	Fclk / 2
0	0	1	Fclk / 4
0	1	0	Fclk / 8
0	1	1	Fclk / 16
1	0	0	Fclk / 32
1	0	1	Fclk / 64
1	1	0	Fclk / 128
1	1	1	Reserved

Bit 6 **SPEN:** Serial Peripheral Enable.
0 = Disable SPI function. SPI pins are GPIO.
1 = Enable SPI function. GPIO pins are SPI pins.
SPI pin structure can be push-pull structure and open-drain structure controlled by P1OC register.

Bit 5 **SSDIS:** SS Disable.
0 = Enables the “SSN” input in both Master and Slave modes.
1 = Disables the “SSN” input in both Master and Slave modes.
In Slave mode, this bit has no effect if CPHA = 0. When SSDIS is set, no “MODF” interrupt request will be generated.

Bit 4 **MSTR:** Serial Peripheral Master.
0 = SPI is Slave mode.
1 = SPI is Master mode.

Bit 3 **CPOL:** SCK idle status control bit.
0 = SCK idle status is low status.
1 = SCK idle status is high status.

Bit 2 **CPHA:** The Clock Phase bit controls the phase of the clock on which data is sampled.
0 = Data receive at the first clock phase.
1 = Data receive at the second clock phase.

15.4 SPDAT DATA BUFFER

0E3H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPDAT	SPDAT7	SPDAT6	SPDAT5	SPDAT4	SPDAT3	SPDAT2	SPDAT1	SPDAT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

The SPDAT register is used during transmission process. After transmission process is completed the received data can be read from SPDAT register.

Bit [7:0] **SPDAT[7:0]**: The SPDAT is a read/write buffer for the “receive data” register. While writing to the SPDAT data is placed directly into the shift register (there is no transmit buffer). Reading the SPDAT returns the value located in the receive buffer, not the shift register.

15.5 SPI STATUS REGISTER

0E1H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPSTA	SPIF	WCOL	SSERR	MODF	-	-	-	-
Read/Write	R	R	R	R	-	-	-	-
After reset	0	0	0	0	-	-	-	-

The SPSTA register reflects the current status of the SPI. The SPIF flag informs that there is a transfer in progress or a transfer has finished. The WCOL flag indicates that a write collision on the SPDAT has occurred, i.e. the SPDAT register was written through the SFR interface while there was a transfer on the SPI interface. The SERR bit informs that the “SSN” input was removed before the end of receive sequence when the SPI was configured as slave. Finally, the MODF bit notifies when the state of the “SSN” input is in conflict with the actual mode settings (i.e. when “SSN” = 0 and the SPI is configured as a master).

Bit 7 **SPIF:** Serial Peripheral Data Transfer Flag.
Set by hardware upon data transfer completion. Cleared by reading the SPSTA register with the SPIF bit set, and then reading the SPDAT register.

Bit 6 **WCOL:** Write Collision Flag.
Set by hardware upon write collision to SPDAT. Cleared by an access to SPSAT register and an access to SPDAT register.

Bit 5 **SSERR:** Synchronous Serial Slave Error Flag.
Set by hardware when “SSN” input is deasserted before the end of receive sequence. Cleared by disabling the SPI (clearing SPEN bit in SPCON register).

Bit 4 **MODF:** Mode Fault Flag.
Set by hardware when the “SSN” pin level is in conflict with actual mode of the SPI controller (configured as master while externally selected as slave). Cleared by hardware when the “SSN” pin is at appropriate level and the SPCON register be write any value.

0E4H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1OC	PW3EN	PW2EN	PW1EN	P06OC	P05OC	P13OC	P12OC	P11OC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [2:0] **P11OC~P13OC**: Open drain control bit.
0 = Disable.
1 = Enable open-drain structure.

16 Inter-Integrated Circuit (I2C)

16.1 OVERVIEW

The I2C is a serial communication interface for data exchanging from one MCU to one MCU or other hardware peripherals. The device can transmit data as a master or a slave with two bi-directional IO, SDA (Serial data output) and SCL (Serial clock input). For data exchanging, “WRITE” and “READ” operation will be executed. When a master transmit data to a slave, it’s called “WRITE” operation; when a slave transmit data to a master, it’s called “READ” operation. It also supports multi-master communication and keeps data transmission correctly by an arbitration method to decide one master has the control on bus and transmit its data.

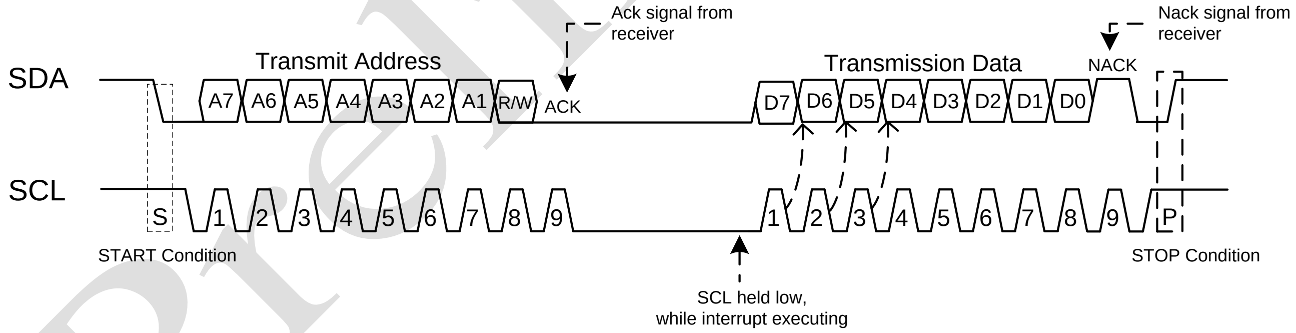
- **Master Tx,Rx Mode**
- **Slave Tx,Rx mode (with general address call) for multiplex slave in single master situation.**

The I2C features include the following:

- **2-wire synchronous data transfer/receiver.**
- **Master (SCL is clock output) or Slave (SCL is clock input) operation.**
- **Support 100K/400K clock rate.**

16.2 I2C TRANSMISSION FORMAT

I2C transmission structure includes a START(S) condition, 8-bit address byte, one or more data byte and a STOP (P) condition. START condition is generated by master to initial any transmission. Data is transmitted with the Most Significant Bit (MSB) first. In address byte, the higher 7-bit is address bit and the lowest bit is data direction (R/W) bit. When R/W=0, it assigns a “WRITR” operation. When R/W=1, it assigns a “READ” operation. After each byte is received, the receiver (a master or a slave) must send an acknowledge (ACK). If transmitter can’t receive an ACK, it will recognize a not acknowledge (NACK). In WRITE operation, the master will transmit data to the slave and then waits for ACK from slave. In READ operation, the slave will transmit data to the master and then waits for ACK from master. In the end, the master will generate a STOP condition to finish transmission.



Transmission Timing Diagram

16.3 SYNCHRONIZATION AND ARBITRATION

In multi-master condition, more than one master may transmit on bus in the same time. It must be decided which master has the control of bus and complete its transmission. Clock synchronization and arbitration are used to configure multi-master transmission.

Clock synchronization is executed by synchronizing the SCL signal with anther devices. When two masters want to transmit in the same, the clock synchronization will start by the High to Low transition on the SCL. If master 1 clock set LOW first, it holds the SCL in LOW status until the clock transit to HIGH status. However, if anther master clock still keep LOW status, the Low to High transition of master 1 may not change SCL status (SCL keep LOW). In the other word, SCL keep LOW by the master with the longest clock time in LOW status. The SCL will transit from LOW to HIGH when the all devices clock transit to HIGH status. In the duration, the master1 will keep in HIGH status and wait for

SCL transition (from LOW to HIGH), then continue its transmission. After clock synchronization, all devices clock and SCL clock are the same.

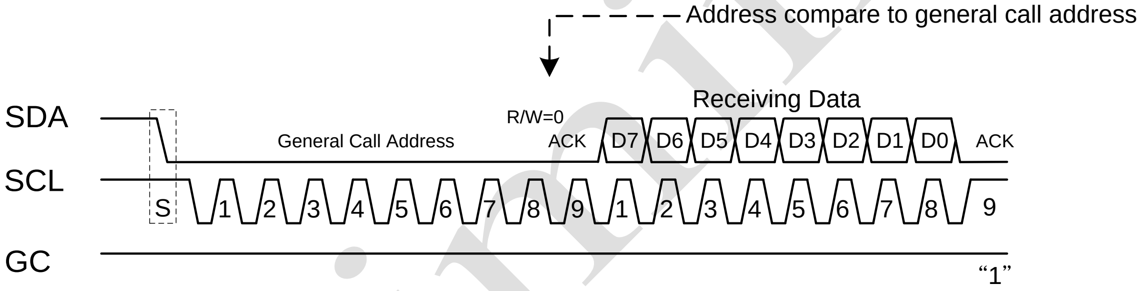
Arbitration is used to decide which master can complete its transmission by SDA signal. Two masters may send out a START condition and transmit data on bus in the same time. They may influence by each other. Arbitration will force one master to lose the control on bus. Data transmission will keep until masters output different data signal. If one master transmits HIGH status and anther master transmits LOW status, the SDA will be pull low. The master output High will detect the different with SDA and lose the control on bus. The mater with LOW status wins the bus control and continues its transmission. There is no data miss during arbitration

> ✸ ***Note: The first step of I2C operation is to setup the I2C pins' mode. Must be set "input mode" in SDA/SCL pins.***

16.4 GENERAL CALL ADDRESS

In I2C bus, the first 7-bit is the Slave address. Only the address match Slave address the Slave will response an ACK. The exception is the general call address which can address all slave devices. When this address occur, all devices should response an acknowledge.

The general call address is a special address which is reserved as all "0" of 7-bit address. The general call address function is control by GC bit. Set this bit will enable general call address and clear it will disable. When GC=1, the general call address will be recognized. When GC=0, the general call address will be ignored.



General Call Address Timing Diagram

16.5 SERIAL CLOCK GENERATOR

In master mode, the SCL clock rate generator's is controlled by CR[2:0] bit of I2CCON register.
When CR[2:0]=000~110, SCL clock rate is from internal clock generator.

SCL Clock Rate = Fcpu/Pre-scaler (Pre-scaler = 256~60)

When CR[2:0]=111, SCL clock rate is from Timer 1 overflow rate .

SCL Clock Rate = (Timer 1 Overflow Rate)/8

The table below shows the clock rate under different setting.

CR2	CR1	CR0	I2C Pre-scaler	Bit Frequency (kHz)			
				6MHz	12MHz	16MHz	24MHz
0	0	0	256	23	47	63	92
0	0	1	224	27	54	71	108
0	1	0	192	31	63	83	124
0	1	1	160	37	75	100	148
1	0	0	960	6.25	12.5	17	25
1	0	1	120	50	100	133	200
1	1	0	60	100	200	266	400
1	1	1	(Timer 1 overflow rate)/8				

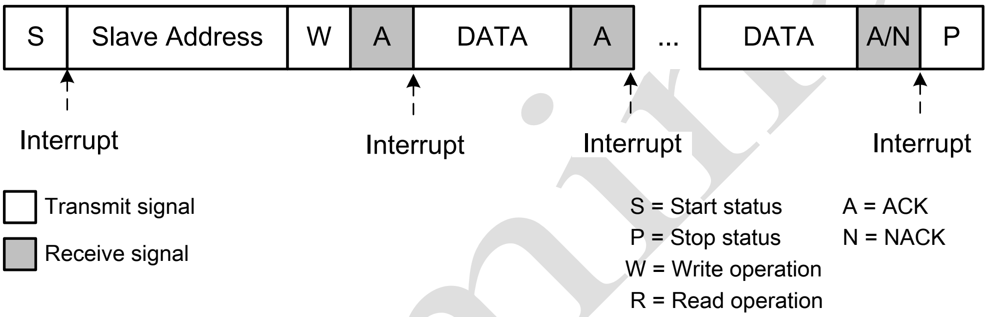
✸ ***Note: When clock generator source is T1 overflow rate, the max counter value is 0XFB (Only supports 0x00~0xFB). And in this time if T1 clock rate is IHRC_32MHz, SCL maximum clock rate is 800 KHz.***

16.6 I2C TRANSMISSION MODE

The I2C module can operate as a master/slave to execute the 8-bit serial data transmission/reception operation. Thus, the module can operate in one of four modes: Master Transmitter, Master Receiver, Slave Transmitter and Slave Receiver.

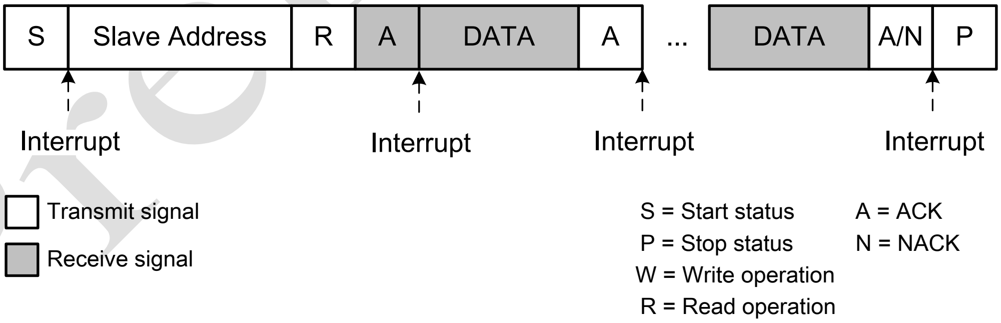
16.6.1 MASTER TRANSMITTER MODE

The master transmits information to the slave. The serial data is output via SDA while the serial clock is output on SCL. Data transmission starts via generate a START(S) signal. After the START signal, the specific address byte of slave device is sent. The address byte includes 7-bit address bit and an 8th data direction (R/W) bit. The R/W is set “0” to enable the master transmission. In the following, the master transmits one or more data byte to the slaver. After each data is transmitted, the master waits for the acknowledge (ACK) from the slave. In the end, the master generates a STOP (P) signal to terminate the data transmission.



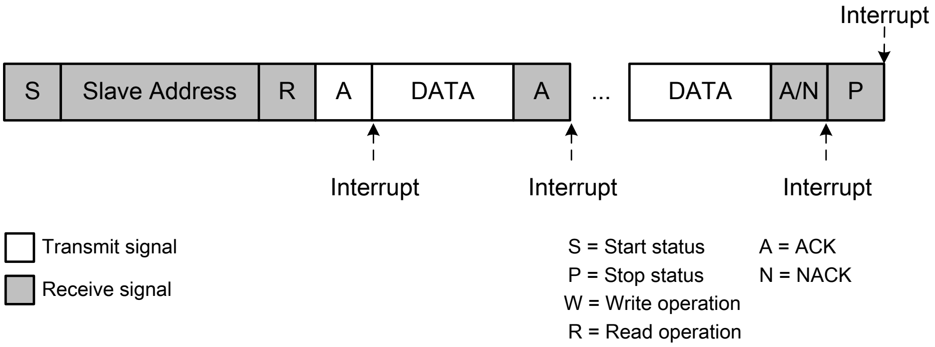
16.6.2 MASTER RECEIVER MODE

The master receives the information from the slave. The serial data input via SDA while the serial clock output on SCL. Data reception starts via generate a START(S) signal. After the START signal, the specific address byte of slave device is sent. The address byte includes 7-bit address bit and an 8th data direction (R/W) bit. The R/W is set “1” to enable the master reception. In the following, the master receives one or more data byte from the slaver. After each data is received, the master generates the acknowledge (ACK) or not acknowledge (NACK) to the slave via the status of AA bit. In the end, the master generates a STOP (P) signal to terminate the data transmission.



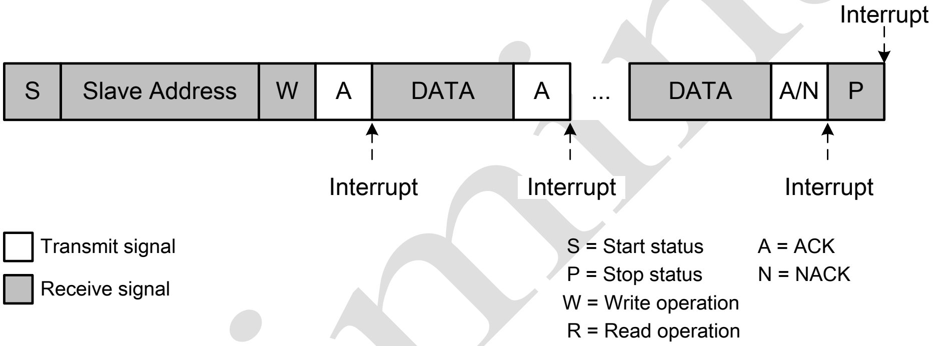
16.6.3 SLAVE TRANSMITTER MODE

The slave transmits information to the master. The serial data output via SDA while the serial clock input on SCL. Data transmission starts via receive a START(S) signal from the master. After the START signal, the specific address byte of slave device is received. The address byte includes 7-bit address bit and an 8th data direction (R/W) bit. The R/W is set “1” to enable the slave transmission. If the received address byte match the address in I2CADR register, the slave generate an acknowledge (ACK). Otherwise, if general call address condition is set (GC=1), the slave also generate an acknowledge (ACK) after general call address (0x00) is received. In the following, the slave transmits one or more data byte to the master. After each data is transmitted, the slave waits for the acknowledge (ACK) from the master. In the end, the slave receives a STOP (P) signal from the master to terminate the data transmission.



16.6.4 SLAVER RECEIVER MODE

The slave receives information from the master. Both the serial data and the serial clock are input on SDA and SCL. Data reception starts via receive a START(S) signal from the master. After the START signal, the specific address byte of slave device is received. The address byte includes 7-bit address bit and an 8th data direction (R/W) bit. The R/W is set “0” to enable the slave reception. If the received address byte match the address in I2CADR register, the slave generate an acknowledge (ACK). Otherwise, if general call address condition is set (GC=1), the slave also generate an acknowledge (ACK) after general call address (0x00) is received. In the following, the slave receives one or more data byte from the master. After each data is receives, the slave generates the acknowledge (ACK) or not acknowledge (NACK) to the master via the status of AA bit. In the end, the slave receives a STOP (P) signal from the master to terminate the data transmission.



16.7 I2C TRANSMISSION REGISTER

Data transmission in I2C interface via through 4 special registers: I2CCON (control register), I2CSTA (status register), I2CDAT (data register) and I2CADR (own slave address register)

16.7.1 CONTROL REGISTER - I2CCON

I2C Function can be enabled/disable by setting ENSI bit. When ENSI=1, I2C function is enabled. Otherwise, I2C function is disabled. I2C supports open-drain structure. When ENSI=1, SDA pin (P1.5) and SCL pin (P1.4) change to open-drain automatically. Before transmission, P1.4/P1.5 must be set “**input mode**” by software with external pull-up resistor.

STA bit is set by the master and generate a START condition if the bus is free. If bus stays busy, the interface will wait for STOP condition to release bus and then generate a START condition. If STA=1 while I2C is master mode and bus is busy for data transfer, a repeated START condition will be generated. STO bit is set by the master and generate a STOP condition. STO is set by software and clear by hardware.

I2C is also supports interrupt function by EI2C control bit. EI2C=0, disable I2C interrupt function. EI2C=1, enable I2C interrupt function. When I2C interrupt function enable, the program counter points to interrupt vector (ORG 0x43) to do I2C interrupt service routine. SI is I2C interrupt request flag, and also to be the operating status indicator when interrupt is disabled. SI is set by hardware and must clear by software. SI is set when I2C stays in one of the 25 out of 26 possible states and an interrupt is requested. The only state that does not generate an interrupt is the F8h state.

The AA bit is used to control transmit ACK or NACK after receiving slave address, receiving general call address or receiving data byte in master or slave mode. When AA=1, an ACK is generated (pull SDA low) and sent during the 9th clock. When AA=0, a NACK is generated (pull SDA high) and sent during the 9th clock.

0DCH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CCON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

The I2CCON register controls the operation of I2C interface. The CPU can read from and write to this 8-bit, directly addressable SFR. Two bits of this register are affected by the I2C hardware: the SI bit is set when serial interrupt is requested, and the STO bit is cleared when STOP condition is present on the I2C bus.

Bit 6 **ENS1:** I2C enable bit.
0 = I2C function is enabled
1 = I2C function is disabled

Bit 5 **STA:** START Flag.
0 = No START condition is transmitted.
1 = A START condition is transmitted if the bus is free.

Bit 4 **STO:** STOP Flag.
0 = No STOP condition is transmitted.
1 = A STOP condition is transmitted to the I2C bus in master mode.

Bit 3 **SI:** Serial Interrupt Flag.
The SI is set by hardware when one of 25 out of 26 possible I2C states is entered. The only state that does not set the SI is state F8h, which indicates that no relevant state information is available. The SI flag must be cleared by software. In order to clear the SI bit, '0' must be written to this bit. Writing a '1' to SI bit does not change value of the SI.

Bit 2 **AA:** Assert Acknowledge Flag.
0 = An "not acknowledge" will be returned when:
- a data byte has been received while I2C was in master receiver mode
- a data byte has been received while I2C was in slave receiver mode

1 = An "acknowledge" will be returned when:
- the "own slave address" has been received
- the general call address has been received while GC bit in I2CADDR register was set
- a data byte has been received while I2C was in master receiver mode
- a data byte has been received while I2C was in slave receiver mode

Bit 7,[1:0] **CR[2:0]:** Clock rate bit [2:0].

CR2	CR1	CR0	I2C Pre-scaler	Bit Frequency (kHz)			
				6MHz	12MHz	16MHz	24MHz
0	0	0	256	23	47	63	92
0	0	1	224	27	54	71	108
0	1	0	192	31	63	83	124
0	1	1	160	37	75	100	148
1	0	0	960	6.25	12.5	17	25
1	0	1	120	50	100	133	200
1	1	0	60	100	200	266	400
1	1	1	(Timer 1 overflow rate)/8				

16.7.2 DATA REGISTER – I2CDAT

The I2CDAT contain 1-byte data to be transmitted to receiver or 1-byte data received from transmitter. MCU should read or write data to I2CDAT during interrupt occurs (SI=1). As I2CDAT register is not shadowed or double buffered, data maybe read or write error when data is shifting in/out the register.

0DAH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CDAT	I2CDAT7	I2CDAT6	I2CDAT5	I2CDAT4	I2CDAT3	I2CDAT2	I2CDAT1	I2CDAT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **I2CDAT[7:0]:** The I2CDAT register contains a byte to be transmitted through I2C bus or a byte which has just been received through I2C bus. The CPU can read from and write to this 8-bit, directly addressable SFR while it is not in the process of byte shifting. The I2CDAT register is not shadowed or double buffered so the user should only read I2CDAT when an I2C interrupt occurs.

16.7.3 SLAVE ADDREDD REGISTER – I2CADR

The I2CADR contain the I2C slave address and general call address control bit. In the master mode, the slave address and general call address control bit are un-use. The slave address is set in the higher 7-bit of I2CADR (Bit7~BIT1). The lowest bit (Bit 0) is general call address control bit (GC). When GC=1, the general call address (0x00) recognition is enabled. When GC=0, the general call address (0x00) recognition is disabled.

0DBH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CADR	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	GC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:1] **ADR[6:0]:** Own I2C slave address (7 bit).

Bit 0 **GC:** General Call Address (0X00) Acknowledge.
0 = The general call address is ignored.
1 = The general call address is recognized.

16.7.4 STATUS REGISTER – I2CSTA

The I2CSTA register contain 8-bit status code to reflect the current status of I2C interface. The lower 3-bit of I2CSTA are always zero. There are 26 possible interface conditions with specific status code as shown in the status code table. Note that, user should not write I2CSTA register by software. I2CSTA only define by hardware during interrupt occurs (SI=1).

0DDH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CSTA	I2CSTA7	I2CSTA6	I2CSTA5	I2CSTA4	I2CSTA3	-	-	-
Read/Write	R/W	R/W	R/W	R/W	R/W	-	-	-
After reset	1	1	1	1	1	-	-	-

Bit [7:3] **I2CSTA[7:3]:** I2C Status Code. The contents of this register reflect the actual state of I2C interface.

I2C status code and status

Mode	Status Code	Status of the I2C	Application software response					Next action taken by I2C hardware
			To/from I2CDAT	TO I2CCON				
				STA	STO	SI	AA	
Master Transmitter/ Receiver	08H	A START condition has been transmitted	Load SLA+R	X	0	0	X	SLA+R/W will be transmitted; ACK will be received
	10H	A repeated START condition has been transmitted.	Load SLA+R	X	0	0	X	SLA+R/W will be transmitted; ACK will be received
			Load SLA+W					SLA+W will be transmitted; I2C will be swithed to MST/TRX mode.
Master Transmitter	18H	SLA+W has been transmitted; ACK has been received	Load data byte	0	0	0	X	Data byte will be transmitted; ACK will be received.
			No action	1	0	0	X	Repeated START will be transmitted.
			No action	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
			No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
	20H	SLA+W has been transmitted; not ACK has been received	Load data byte*	0	0	0	X	Data byte will be transmitted; ACK will be received.
			No action	1	0	0	X	Repeated START will be transmitted.
			No action	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
			No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
	28H	Data byte in I2CDAT has been transmitted; ACK has been received	Load data byte	0	0	0	X	Data byte will be transmitted; ACK bit will be received.
			No action	1	0	0	X	Repeated START will be transmitted.
			No action	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
			No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
	30H	Data byte in I2CDAT has been transmitted; not ACK has been received	Load data byte*	0	0	0	X	Data byte will be transmitted; ACK will be received.
			No action	1	0	0	X	Repeated START will be transmitted.
			No action	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
			No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
	38H	Arbitration lost	No action	0	0	0	X	I2C will be released; A start condition will be transmitted.
			No action	1	0	0	X	When the bus becomes free. (enter to a master mode)
Master Receiver	40H	SLA+R has been transmitted; ACK has been received	No action	0	0	0	0	Data byte will be received; not ACK will be returned
			No action	0	0	0	1	Data byte will be received; ACK will be returned
	48H	SLA+R has been transmitted; not ACK has been received	No action	1	0	0	X	Repeated START condition will be transmitted
			No action	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
			No action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset
	50H	Data byte has been received; ACK has been returned	Read data byte	0	0	0	0	Data byte will be received; not ACK will be returned
			Read data byte	0	0	0	1	Data byte will be received; ACK will be returned
	54H	Data byte has been received; not ACK has been returned	Read data byte	1	0	0	X	Repeated START condition will be transmitted
			Read data byte	0	1	0	X	STOP condition will be transmitted; STO flag will be reset
			Read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset

* “SLA” means slave address, “R” means R/W=1, “W” means R/W=0

*For applications where NACK doesn’t mean the end of communication.

Mode	Status Code	Status of the I2C	Application software response					Next action taken by I2C hardware
			To/from I2CDAT	TO I2CCON				
				STA	STO	SI	AA	
Slave Receiver	60H	Own SLA+W has been received; ACK has been returned	No action	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	68H	Arbitration lost in SLA+R/W as master; own SLA+W has been received, ACK returned	No action	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	70H	General cal address (00H) has been received; ACK has been retirned	No action	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	78H	Arbitration lost in SLA+R/W as master; general call address has been received, ACK returned	No action	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	80H	Previously addressed with own SLV address; DATA has been received; ACK returned	Read data byte	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	88H	Previously addressed with own SLA; DATA byte has been received; not ACK returned	Read data byte	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address
			Read data byte	0	0	0	1	Switched to not addressed SLV mode; own SLA or gernerl call address will be recognized
			Read data byte	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free
			Read data byte	1	0	0	1	Switched to not addressed SLV mode; own SLA or gernerl call address will be recognized; START condition will be transmitted when the bus becomes free
	90H	Previously addressed with general call address; DATA has been received; ACK returned	Read data byte	X	0	0	0/1	Data byte will be received and not ACK/ACK will be returned
	98H	Previously addressed with general call address; DATA has been received; not ACK returned	Read data byte	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address
			Read data byte	0	0	0	1	Switched to not addressed SLV mode; own SLA or gernerl call address will be recognized
			Read data byte	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free
			Read data byte	1	0	0	1	Switched to not addressed SLV mode; own SLA or gernerl call address will be recognized; START condition will be transmitted when the bus becomes free
	A0H	A STOP condition or repeated START condition has been received while still addressed as SLV/REC or SLV/TRX	No action	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address
			No action	0	0	0	1	Switched to not addressed SLV mode; own SLA or gernerl call address will be recognized
			No action	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free
			No action	1	0	0	1	Switched to not addressed SLV mode; own SLA or gernerl call address will be recognized; START condition will be transmitted when the bus becomes free
Slave Transmitter	A8H	Own SLA+R has been received; ACK has been returned	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
			Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
	B0H	Arbitration lost in SLA+R/W as master; own SLA+R has been received, ACK ha been returned .	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
			Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
	B8H	Data byte has been transmitted; ACK will be received.	Load data byte	X	0	0	0	Last data byte will be transmitted and ACK will be received
			Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
	C0H	Data byte has been transmitted; not ACK has been received.	No action	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address.
			No action	0	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized.
			No action	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free.
			No action	1	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free.

	C8H	Last data byte has been transmitted; ACK has been received.	No action	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address.
			No action	0	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized.
			No action	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or general call address; START condition will be transmitted when the bus becomes free.
			No action	1	0	0	1	Switched to not addressed SLV mode; own SLA or general call address will be recognized; START condition will be transmitted when the bus becomes free.
All	F8H	No relevant state information available; SI=0	No action	No action				Wait or proceed current transfer
	00H	Bus error during MST or selected slave modes	No action	0	1	0	X	Only the internal hardware is affected in the MST or addressed SLV modes. In all cases, the bus is released and I2C is switched to the not addressed SLV mode. STO flag is reser.

* “SLA” means slave address, “R” means R/W=1, “W” means R/W=0
*For applications where NACK doesn’t mean the end of communication.

16.8 SYSTEM MANGMENT BUS EXTENSION

The optional System Management Bus (SMBus) protocol hardware supports 3 types timeout detection:

- **Tmext Timeout Detection:** The cumulative stretch clock cycles within one byte.
- **Tsext Timeout Detection:** The cumulative stretch clock cycles between start and stop condition.
- **Timeout Detection:** The clock low measurement.

Timeout detection is controlled by SMBSEL and SMBDST registers. The SMBEXE bit of SMBSEL is SMBus extension function enable bit. When SMBEXE=1, SMBus extension function is enabled. Otherwise, Disable SMBus extension function. Timeout type and period setting is controlled by SMBTOP[2:0] and SMBDST.

The period of SMBus timeout is controlled by three 16-bit biffers of Tmex, Tsext and Tout. The equation is as following.

$$\boxed{\textit{Tmext / Tsext / Tout = Timeout Period (sec) * Fcpu(Hz)/1024}}$$

Tmext is support by two 8-bit register of Tmext_L and Tmext_H . Tmext_L hold the low byte and Tmext_H hold high byte.
Tsext is support by two 8-bit register of Tsext_L and Tsext_H . Tsext_L hold the low byte and Tsext_H hold high byte.
Tout is support by two 8-bit register of Tout_L and Tout_H . Tout_L hold the low byte and Tout_H hold high byte.

Type	Time out period	Fcpu=24MHz		Fcpu=32MHz	
		DEC	HEX	DEC	HEX
Tmext	5ms	118	76	157	9D
Tsext	25ms	586	24A	782	30E
Tout	35ms	821	335	1094	446

By the setting of SMBTOP[2:0] to choose register type (as the table below), and write to register by write data to SMBDST register.

SMBTOP[2:0]	SMBDST	Description
000	Tmext_L	Select the low byte of Tmext register.
001	Tmext_H	Select the high byte of Tmext register.
010	Tsext_L	Select the low byte of Tsext register.
011	Tsext_H	Select the high byte of Tsext register.
100	Tout_L	Select the low byte of Tout register.
101	Tout_H	Select the high byte of Tout register.

When the SMBus extension function is enabled the lower 3-bit of I2CSTA hold the information about time out as the table below.

I2CSTA	Description
XXXX X000	No timeout errors.
XXXX XXX1	Tout timeout error.
XXXX XX1X	Tsext timeout error.
XXXX X1XX	Tmext timeout error.

0DEH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMBSEL	SMBEXE	-	-	-	-	SMBTOP2	SMBTOP1	SMBTOP0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

This register is used to enable the SMBus extension to the primary I2C interface and to provide a read/write access port to the SMBus timeout registers.

Bit 7 **SMBEXE:** SMBus Extension Enable.
0 = Disable SMBus Extension function.
1 = Eable SMBus Extension function.

Bit [2:0] **SMBTOP[2:0]:** SMBus Timeout Register Pointer. Selects one of the Timeout Registers for read/write access through the SMBDST register.

0DFH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMBDST	SMBDST7	SMBDST6	SMBDST5	SMBDST4	SMBDST3	SMBDST2	SMBDST1	SMBDST0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **SMBDST[7:0]:** This register is used to provide a read/write access port to the SMBus timeout registers. Data read or written to that register is actually read or written to the Timeout Register which is pointed by the SMBSEL register.

17 IN SYSTEM PROGRAM FLASH ROM

17.1 OVERVIEW

The SN8F5708 MCU integrated device feature in-system programmable (ISP) FLASH memory for convenient, upgradeable code storage. The FLASH memory may be programmed via the SONiX 8 bit MCU programming interface or by application code. The SN8F5708 provides security options at the disposal of the designer to prevent unauthorized access to information stored in FLASH memory. ISP Flash ROM provided user an easy way to storage data into Flash ROM. The ISP concept is memory mapping idea that is to move RAM buffer to flash ROM. Choice ROM/RAM address and executing ROM programming command – PECMD, after programming words which controlled by PERAM data will be programmed into address PEROML/PEROMH.

RAM (byte)			Flash ROM (byte)	
RAM Address	bit7 ~ bit0		ROM Address	bit7 ~ bit0
X	DATA0		Y	DATA0
X+1	DATA1		Y+1	DATA1
X+2	DATA2	=>	Y+2	
X+3	DATA3		Y+3	
…			…	
…			…	
X+N	DATAN		Y+M	DATAN

During Flash program or erase operation, the MCU is stalled, although peripherals (Timers, WDT, I/O, PWM, etc.) remain active. When PECMD register is set to execute ISP program and erase operations, the program counter stops, op-code can't be dumped from flash ROM, instruction stops operating, and program execution is hold not to active. At this time hardware depends on ISP operation configuration to do flash ROM erasing and flash ROM programming automatically. After ISP operation is finished, hardware releases system clock to make program counter running, system returns to last operating mode, and the next instruction is executed. Recommend to add two "NOP" instructions after ISP operations.

- **ISP flash ROM program time = 8ms……32-byte.**

✸ ***Note:***

1. ***Watch dog timer should be clear before the Flash write (program) operation, or watchdog timer would overflow and reset system during ISP operating.***
2. ***Besides program execution, all functions keep operating during ISP operating, e.g. timer, ADC, SPI, UART… All interrupt events still active and latch interrupt flags automatically. If any interrupt request occurs during ISP operating, the interrupt request will be process by program after ISP finishing.***
3. ***ISP function only support IHRC_32M and IHRC_32M_RTC mode.***

17.2 ISP FLASH ROM PROGRAM OPERATION

ISP flash ROM program operation is to write data into flash ROM by program. Program ROM supports 32-byte length density of one page. The number of ISP flash ROM program operation fixed 32-byte at one time. ISP flash ROM program ROM map is as following:

ISP ROM MAP		ROM address bit0~bit4 (hex) =0
ROM address bit5~bit15 (hex)	0000	**These pages include reset vector and interrupt sector. We strongly recommend to reserve the area not to do ISP erase.**
	0020	
	0040	
	…	
	00C0	
	00E0	
	0100	One ISP Program Page
	0120	One ISP Program Page
	…	One ISP Program Page
	3000	One ISP Program Page
	3020	One ISP Program Page
	…	One ISP Program Page
	3700	One ISP Program Page
	3720	One ISP Program Page
	…	One ISP Program Page
	3FE0	**This page includes ROM reserved area. We strongly recommend to reserve the area not to do ISP erase.**

ISP flash ROM program page density is 32-byte which limits program page boundary. The first 8-page of flash ROM (0x0000~0x00FF) includes reset vector and interrupt vectors related essential program operation, and the last page 32-word of flash ROM (0x3FE0~0x3FFF) includes system reserved ROM area, we strongly recommend do not execute ISP flash ROM program operation in the two pages. Flash ROM area 0x0100~0x3FDF includes 503-page for ISP flash ROM program operation.

ISP flash ROM program operation is a simple memory mapping operation. The first step is to plan a RAM area to store programmed data and keeps the RAM address for ISP RAM addressing. The second step is to plan a ROM area will be programmed from RAM area in ISP flash ROM program operation. The RAM addressing is through PERAML[7:0] 8-bit buffer to configure the start RAM address.

ISP programming length is 32-byte. ISP flash ROM programming length is fixed to 32 bits. PEROML [7:0] and PEROMH [7:0] define the target starting address [15:0] of flash ROM. Write the start address into PEROML and PEROMH registers, set PECMD[11:0] register to “0xA5A”, and the system start to execute ISP flash ROM program operation.

- **Case 1:** 32-byte ISP program. RAM buffer length is 32-byte and RAM address is X ~ X+31. The page address of flash ROM is Y ~ Y+31. The Y is the start address and set to PEROML, PEROMH registers.

RAM (byte)

RAM Address 32-byte	bit7 ~ bit0
X	**DATA0**
X+1	**DATA1**
X+2	**DATA2**
X+3	**DATA3**
…	**…**
X+30	**DATA30**
X+31	**DATA31**

=>

Flash ROM (word)

ROM Address 32-byte	bit7 ~ bit0	
Y	**DATA0**	**The head of the page.** **The start address of ISP.**
Y+1	**DATA2**	
Y+2	**…**	
Y+3	**…**	
…	**…**	
…	**…**	
Y+31	**DATA62**	**The end of the page.** **The end address of ISP.**

- **Example: Use ISP flash ROM program to program 32-byte data to flash ROM as case 1. Set RAM buffer start address is 0x020. Set flash ROM programmed start address is 0x0100.**

```
; Load data into 32-byte RAM buffer.
              …
              …

; Set RAM start address of 32-byte buffer.
              MOV          A, #0x20
              MOV          PERAM, A                     ; Set PERAM[7:0] to 0x20.

; Set programmed start address of flash ROM to 0x0100.
              MOV          A, #0x01
              MOV          PEROMH, A                    ; Move high byte address 0x01 to PEROMH.
              MOV          A, #0x0A
              MOV          PEROML, A                    ; Move low byte address 0x00 to PEROML

; Clear watchdog timer.
              MOV          A,#0X5A
              MOV          WDTR,A

; Start to execute ISP flash ROM program operation.
              ORL          PEROML, #0X0A
              MOV          A,#0X5A                      ; Start to program flash ROM.
              MOV          PECMD, A
              NOP                                       ; NOP Delay
              NOP
                                                        ; The end of ISP flash ROM program operation.
```

The two “NOP” instructions make a short delay to let system stable after ISP flash ROM program operation.

> ✸ ***Note: Don’t execute ISP flash ROM program operation for the first page and the last page, or affect program operation.***

17.3 ISP PROGRAM/ERASE CONTROL REGISTER

094H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PECMD	PECMD7	PECMD6	PECMD5	PECMD4	PECMD3	PECMD2	PECMD1	PECMD0
Read/Write	W	W	W	W	W	W	W	W
After reset	-	-	-	-	-	-	-	-

Bit [7:0] **PECMD [7:0]**

095H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEROML	PEROM7	PEROM6	PEROM5	-	PECMD11	PECMD10	PECMD9	PECMD8
Read/Write	R/W	R/W	R/W	-	W	W	W	W
After reset	0	0	0	-	0	0	0	0

Bit [3:0] **PEROM [3:0]** = ISP CMD [11:8].

PECMD [11:0] ISP operation control register.
0xA5A: Page Program (32 bytes / page).
Others: Reserved.

17.4 ISP ROM ADDRESS REGISTER

ISP ROM address length is 16-bit and separated into PEROML and PEROMH registers. Before ISP execution, set the head address of ISP ROM by program.

095H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEROML	PEROM7	PEROM6	PEROM5	-	PECMD11	PECMD10	PECMD9	PECMD8
Read/Write	R/W	R/W	R/W	-	W	W	W	W
After reset	0	0	0	-	0	0	0	0

Bit [7:5] **PEROM [7:5]** = ISP page write target start ROM address [7:5]. PEROM [4:0] always 0.

096H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEROMH	PEROM15	PEROM14	PEROM13	PEROM12	PEROM11	PEROM10	PEROM9	PEROM8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **PEROM[15:8]** = ISP page write target start ROM address[15:8].

17.5 ISP ROM ADDRESS REGISTER

ISP RAM address length is 8-bit PERAM. Before ISP execution, set the head address of ISP RAM by program.

097H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PERAM	PERAM7	PERAM6	PERAM5	PERAM4	PERAM3	PERAM2	PERAM1	PERAM0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Bit [7:0] **PERAM[7:0]:** ISP fetch data start RAM address [7:0].

18 INSTRUCTION TABLE

18.1 INSTRUCTION SET

Field	Mnemonic	Description	Code	Bytes	Cycles
ARITHMETIC	ADD A,Rn	Add register to accumulator	0x28-0x2F	1	1
	ADD A,direct	Add directly addressed data to accumulator	0x25	2	2
	ADD A,@Ri	Add indirectly addressed data to accumulator	0x26-0x27	1	2
	ADD A,#data	Add immediate data to accumulator	0x24	2	2
	ADDC A,Rn	Add register to accumulator with carry	0x38-0x3F	1	1
	ADDC A,direct	Add directly addressed data to accumulator with carry	0x35	2	2
	ADDC A,@Ri	Add indirectly addressed data to accumulator with carry	0x36-0x37	1	2
	ADDC A,#data	Add immediate data to accumulator with carry	0x34	2	2
	SUBB A,Rn	Subtract register from accumulator with borrow	0x98-0x9F	1	1
	SUBB A,direct	Subtract directly addressed data from accumulator with borrow	0x95	2	2
	SUBB A,@Ri	Subtract indirectly addressed data from accumulator with borrow	0x96-0x97	1	2
	SUBB A,#data	Subtract immediate data from accumulator with borrow	0x94	2	2
	INC A	Increment accumulator	0x04	1	1
	INC Rn	Increment register	0x08-0x0F	1	1
	INC direct	Increment directly addressed location	0x05	2	2
	INC @Ri	Increment indirectly addressed location	0x06-0x07	1	2
	INC DPTR	Increment data pointer	0xA3	1	1
	DEC A	Decrement accumulator	0x14	1	1
	DEC Rn	Decrement register	0x18-0x1F	1	1
	DEC direct	Decrement directly addressed location	0x15	2	2
	DEC @Ri	Decrement indirectly addressed location	0x16-0x17	1	2
	MUL AB	Multiply A and B	0xA4	1	4
	DIV AB	Divide A by B	0x84	1	4
	DA A	Decimally adjust accumulator	0xD4	1	1
LOGIC	ANL A,Rn	AND register to accumulator	0x58-0x5F	1	1
	ANL A,direct	AND directly addressed data to accumulator	0x55	2	2
	ANL A,@Ri	AND indirectly addressed data to accumulator	0x56-0x57	1	2
	ANL A,#data	AND immediate data to accumulator	0x54	2	2
	ANL direct,A	AND accumulator to directly addressed location	0x52	2	2
	ANL direct,#data	AND immediate data to directly addressed location	0x53	3	3
	ORL A,Rn	OR register to accumulator	0x48-0x4F	1	1
	ORL A,direct	OR directly addressed data to accumulator	0x45	2	2
	ORL A,@Ri	OR indirectly addressed data to accumulator	0x46-0x47	1	2
	ORL A,#data	OR immediate data to accumulator	0x44	2	2
	ORL direct,A	OR accumulator to directly addressed location	0x42	2	2
	ORL direct,#data	OR immediate data to directly addressed location	0x43	3	3
	XRL A,Rn	Exclusive OR register to accumulator	0x68-0x6F	1	1
	XRL A,direct	Exclusive OR directly addressed data to accumulator	0x65	2	2
	XRL A,@Ri	Exclusive OR indirectly addressed data to accumulator	0x66-0x67	1	2
	XRL A,#data	Exclusive OR immediate data to accumulator	0x64	2	2
	XRL direct,A	Exclusive OR accumulator to directly addressed location	0x62	2	2
	XRL direct,#data	Exclusive OR immediate data to directly addressed location	0x63	3	3
	CLR A	Clear accumulator	0xE4	1	1
	CPL A	Complement accumulator	0xF4	1	1
	RL A	Rotate accumulator left	0x23	1	1
	RLC A	Rotate accumulator left through carry	0x33	1	1
	RR A	Rotate accumulator right	0x03	1	1
	RRC A	Rotate accumulator right through carry	0x13	1	1
	SWAP A	Swap nibbles within the accumulator	0xC4	1	1
DATA TRANSFER	MOV A,Rn	Move register to accumulator	0xE8-0xEF	1	1
	MOV A,direct	Move directly addressed data to accumulator	0xE5	2	2
	MOV A,@Ri	Move indirectly addressed data to accumulator	0xE6-0xE7	1	2
	MOV A,#data	Move immediate data to accumulator	0x74	2	2
	MOV Rn,A	Move accumulator to register	0xF8-0xFF	1	1
	MOV Rn,direct	Move directly addressed data to register	0xA8-0xAF	2	2
	MOV Rn,#data	Move immediate data to register	0x78-0x7F	2	2
	MOV direct,A	Move accumulator to direct	0xF5	2	2
	MOV direct,Rn	Move register to direct	0x88-0x8F	2	2
	MOV direct1,direct2	Move directly addressed data to directly addressed location	0x85	3	3
	MOV direct,@Ri	Move indirectly addressed data to directly addressed location	0x86-0x87	2	2
	MOV direct,#data	Move immediate data to directly addressed location	0x75	3	3
	MOV @Ri,A	Move accumulator to indirectly addressed location	0xF6-0xF7	1	1
	MOV @Ri,direct	Move directly addressed data to indirectly addressed location	0xA6-0xA7	2	2
	MOV @Ri,#data	Move immediate data to in directly addressed location	0x76-0x77	2	2

	MOV DPTR,#data16	Load data pointer with a 16-bit immediate	0x90	3	3
	MOVC A,@A+DPTR	Load accumulator with a code byte relative to DPTR	0x93	1	3
	MOVC A,@A+PC	Load accumulator with a code byte relative to PC	0x83	1	3
	MOVX A,@Ri	1) Move external RAM (8-bit addr.) to accumulator	0xE2-0xE3	1	3-10
	MOVX A,@DPTR	1) Move external RAM (16-bit addr.) to accumulator	0xE0	1	3-10
	MOVX @Ri,A	1) Move accumulator to external RAM (8-bit addr.)	0xF2-0xF3	1	3-12
	MOVX @DPTR,A	1) Move accumulator to external RAM (16-bit addr.)	0xF0	1	3-12
	PUSH direct	Push directly addressed data onto stack	0xC0	2	2
	POP direct	Pop directly addressed location from stack	0xD0	2	2
	XCH A,Rn	Exchange register with accumulator	0xC8-0xCF	1	1
	XCH A,direct	Exchange directly addressed location with accumulator	0xC5	2	2
	XCH A,@Ri	Exchange indirect RAM with accumulator	0xC6-0xC7	1	2
	XCHD A,@Ri	Exchange low-order nibbles of indirect and accumulator	0xD6-0xD7	1	2
BRANCH	ACALL addr11	Absolute subroutine call	Xxx10001b	2	2
	LCALL addr16	Long subroutine call	0x12	3	3
	RET	Return from subroutine	0x22	1	4
	RETI	Return from interrupt	0x32	1	4
	AJMP addr11	Absolute jump	Xxx00001b	2	2
	LJMP addr16	Long jump	0x02	3	3
	SJMP rel	Short jump (relative address)	0x80	2	**3**
	JMP @A+DPTR	Jump indirect relative to the DPTR	0x73	1	**2**
	JZ rel	Jump if accumulator is zero	0x60	2	3
	JNZ rel	Jump if accumulator is not zero	0x70	2	3
	JC rel	Jump if carry flag is set	0x40	2	3
	JNC rel	Jump if carry flag is not set	0x50	2	3
	JB bit,rel	Jump if directly addressed bit is set	0x20	3	4
	JNB bit,rel	Jump if directly addressed bit is not set	0x30	3	4
	JBC bit,rel	Jump if directly addressed bit is set and clear bit	0x10	3	4
	CJNE A,direct,rel	Compare directly addressed data to accumulator and jump if not equal	0xB5	3	4
	CJNE A,#data,rel	Compare immediate data to accumulator and jump if not equal	0xB4	3	4
	CJNE Rn,#data,rel	Compare immediate data to register and jump if not equal	0xB8-0xBF	3	4
	CJNE @Ri,#data,rel	Compare immediate to indirect and jump if not equal	0xB6-0xB7	3	5
	DJNZ Rn,rel	Decrement register and jump if not zero	0xD8-0xDF	2	3
	DJNZ direct,rel	Decrement directly addressed location and jump if not zero	0xD5	3	4
	NOP	No operation	0x00	1	1
BIT	CLR C	Clear carry flag	0xC3	1	1
	CLR bit	Clear directly addressed bit	0xC2	2	2
	SETB C	Set carry flag	0xD3	1	1
	SETB bit	Set directly addressed bit	0xD2	2	2
	CPL C	Complement carry flag	0xB3	1	**1**
	CPL bit	Complement directly addressed bit	0xB2	2	2
	ANL C,bit	AND directly addressed bit to carry flag	0x82	2	2
	ANL C,/bit	AND complement of directly addressed bit to carry	0xB0	2	2
	ORL C,bit	OR directly addressed bit to carry flag	0x72	2	2
	ORL C,/bit	OR complement of directly addressed bit to carry	0xA0	2	2
	MOV C,bit	Move directly addressed bit to carry flag	0xA2	2	2
	MOV bit,C	Move carry flag to directly addressed bit	0x92	2	2

Note: The MOVX instructions perform one of two actions depending on the state of “PMW” bit (PCON.4).

All R8051XC2 instructions are binary code compatible and perform the same functions as they do within the industry standard 8051. The following tables give a summary of instruction cycles of the R8051XC2 microcontroller core.Table show instruction hexadecimal codes, numbers of bytes and machine cycles that each instruction takes to be executed. Note the number of cycles is given for no program memory wait states.

18.2 NOTES ON DATA & PROGRAM ADDRESSING MODES

Symbol	Description
Rn	Working register R0-R7
Direct	One of 128 internal RAM locations or any Special Function Register
@Ri	Indirect internal or external RAM location addressed by register R0 or R1
#data	8-bit constant included in instruction (immediate operand)
#data16	16-bit constant included as bytes 2 and 3 of instruction (immediate operand)
Bit	One of 128 software flags located in internal RAM, or any flag of bit-addressable Special Function Registers, including I/O pins and status word
A	Accumulator
Addr16	Destination address for LCALL or LJMP, can be anywhere within the 64-Kbyte page of program memory address space
Addr11	Destination address for ACALL or AJMP, within the same 2-Kbyte page of program memory as the first byte of the following instruction
rel	SJMP and all conditional jumps include an 8-bit offset byte. Its range is +127/-128 bytes relative to the first byte of the following instruction

18.3 THE DURATION OF EACH INSTRUCTION CALCULATION

The duration of each instruction can be calculated using the formula below.

```
If (BYTES > 1 or CYCLES = 1) then
    DURATION = CYCLES + (BYTES+R)*P + X*D
else
    DURATION = CYCLES + (2+R)*P + X*D
```

Where:
- BYTES is the number of bytes for the instruction (see tables above)
- CYCLES is the number of cycles for no wait states (see tables above)
- R = 1 for the MOVC instruction, otherwise R = 0
- X = 1 for MOVX instructions, otherwise X = 0
- P = number of program memory wait states (= “CKCON[6:4]”)
- D = number of data memory wait states (= “CKCON[2:0]”).

In the Program Memory Write mode (PMW) the formula for MOVX is as follows:

DURATION = CYCLES + (2+X)*P

18.4 READ-MODIFY-WRITE INSTRUCTIONS

Instructions that read a byte from SFR or internal RAM, modify it and rewrite it back, are called "Read-Modify-Write" instructions. When the destination is an I/O port (P0-P3), or a Port bit, these instructions read the output latch rather than the pin.

Field	Mnemonic	Description	Code	Bytes	Cycles
RMW	ANL direct,A	AND accumulator to direct	0x52	2	2
	ANL direct,#data	AND immediate data to direct	0x53	3	3
	ORL direct,A	OR accumulator to direct	0x42	2	2
	ORL direct,#data	OR immediate data to direct	0x43	3	3
	XRL direct,A	Exclusive OR accumulator to direct	0x62	2	2
	XRL direct,#data	Exclusive OR immediate data to direct	0x63	3	3
	JBC bit, rel	Jump if bit is set and clear bit	0x10	3	4
	CPL bit	Complement bit	0xB2	2	2
	INC direct	Increment direct	0x05	2	2
	INC @Ri	Increment indirect	0x06-0x07	1	2
	DEC direct	Decrement direct	0x15	2	2
	DEC @Ri	Decrement indirect	0x16-0x17	1	2
	DJNZ direct,rel	Decrement and jump if not zero	0xD5	3	4
	MOV bit,C	Move carry flag to direct bit	0x92	2	2
	CLR bit	Clear bit	0xC2	2	2
	SETB bit	Set bit	0xD2	2	2

19 ELECTRICAL CHARACTERISTIC

19.1 ABSOLUTE MAXIMUM RATING

Supply voltage (Vdd)…………………………………………………………………………………………………… - 0.3V ~ 6.0V
Input in voltage (Vin)………………………………………………………………………………………… Vss – 0.2V ~ Vdd + 0.2V
Operating ambient temperature (Topr)
SN8F5708, SN8F57081, SN8F5707, SN8F5705 …………………………………………………………………… –40°C ~ + 85°C
Storage ambient temperature (Tstor) ………………………………………………………………………………… –40°C ~ + 125°C

19.2 ELECTRICAL CHARACTERISTIC

- **SN8F5708 DC CHARACTERISTIC**

(All of voltages refer to Vss, Vdd = 5.0V, Fosc = 16MHz, ambient temperature is 25℃ unless otherwise note.)

PARAMETER	SYM.	DESCRIPTION		MIN.	TYP.	MAX.	UNIT
Operating voltage	Vdd	*Fcpu = 1MHz, ISP is inactive.*		1.8	-	5.5	V
		Fcpu = 1MHz, ISP actives.		1.8		5.5	V
RAM Data Retention voltage	Vdr			1.5	-	-	V
*Vdd rise rate	Vpor	Vdd rise rate to ensure internal power-on reset		0.05	-	-	V/ms
Input Low Voltage	ViL	All input ports		Vss	-	0.3*Vdd	V
Input High Voltage	ViH	All input ports		0.7*Vdd	-	Vdd	V
I/O port input leakage current	Ilekg	Pull-up resistor disable, Vin = Vdd		-	-	2	uA
I/O port pull-up resistor	Rup	Vin = Vss , Vdd = 3V, P0/P1/P2/P3/P4/P5 pins.		100	200	300	
		Vin = Vss , Vdd = 5V, P0/P1/P2/P3/P4/P5 pins.		50	100	150	
I/O output source current	IoH	Vop = Vdd – 0.5V, P0/P1/P4/P5 pins.		8	14	-	mA
I/O output sink current	IoL1	Vop = Vss + 0.5V, P14~P17/P2/P3/P4/P5 pins.		15	20	-	
	IoL2	Vop = Vss + 1.5V, P0/P10~P13 pins.		80	100	-	
*INTn trigger pulse width	Tint0	INT0 interrupt request pulse width		2/fcpu	-	-	cycle
Supply Current (Disable ADC)	Idd1	Run Mode (No loading)	Vdd= 3V, Fcpu = 8MHz	-	3.5	-	mA
			Vdd= 5V, Fcpu = 8MHz	-	3.55	-	mA
			Vdd= 3V, Fcpu = 4MHz	-	2.8	-	mA
			Vdd= 5V, Fcpu = 4MHz	-	2.85	-	mA
			Vdd= 3V, Fcpu = 1MHz	-	2.25	-	mA
			Vdd= 5V, Fcpu = 1MHz	-	2.3	-	mA
	Idd2	Sleep Mode	Vdd= 3V	-	3.5	-	uA
			Vdd= 5V	-	4	-	uA
	Idd3	Idle Mode (No loading, Watchdog Disable)	Vdd= 3V, IHRC=32MHz/32	-	0.63	-	mA
			Vdd= 5V, IHRC=32MHz/32	-	0.65	-	mA
			Vdd= 3V, Ext. 16MHz X'tal/16	-	0.65	-	mA
			Vdd= 5V, Ext. 16MHz X'tal/16	-	1.25	-	mA
			Vdd= 3V, Ext. 4MHz X'tal/1	-	0.6	-	mA
			Vdd= 5V, Ext. 4MHz X'tal/1	-	0.75	-	mA
Internal High Oscillator Freq.	Fihrc	Internal Hihg RC (IHRC)	*25℃*, Vdd=1.8V~ 5.5V	31.36	32	32.64	MHz
			-40℃~85℃, Vdd=1.8V~ 5.5V	30.88	32	33.12	MHz
LVD Voltage	Vdet0	Low voltage reset level. *25℃*		1.7	1.8	1.9	V
		Low voltage reset level. *-40℃~85℃*		1.6	1.8	2.0	V
	Vdet1	Low voltage reset/indicator level. *25℃*		2.3	2.4	2.5	V
		Low voltage reset/indicator level. *-40℃~85℃*		2.2	2.4	2.6	V
	Vdet2	Low voltage reset/indicator level. *25℃*		3.2	3.3	3.4	V
		Low voltage reset/indicator level. *-40℃~85℃*		3.0	3.3	3.6	V

" *" These parameters are for design reference, not tested.

ADC CHARACTERISTIC

(All of voltages refer to Vss, Vdd = 5.0V, Fosc = 16MHz, Fcpu=4MHz, ambient temperature is 25℃ unless otherwise note.)

PARAMETER	SYM.	DESCRIPTION	MIN.	TYP.	MAX.	UNIT
AIN0 ~ AIN11 input voltage	Vani	Vdd = 5.0V	0	-	Avrefh	V
AVREFH input voltage	Varfh	Vdd = 5.0V	Varfl+2V		Vdd	V
AVREFL input voltage	Varfl	Vdd = 5.0V	Vss		Varfh–2V	V
ADC reference Voltage	Verf	External reference voltage, Vdd = 5V.	2.0	-	Vdd	V
	Virf1	Internal VDD reference voltage, Vdd = 5V.		Vdd		V
	Virf2	Internal 4V reference voltage, Vdd = 5V.	3.92	4	4.08	V
	Virf3	Internal 3V reference voltage, Vdd = 5V.	2.94	3	3.06	V
	Virf4	Internal 2V reference voltage, Vdd = 5V.	1.96	2	2.04	V
*ADC enable time	Tast	Ready to start convert after set ADENB = “1”	100	-	-	us
*ADC current consumption	I_{ADC}	Vdd =5.0V	-	0.7	-	mA
		Vdd =3.0V	-	0.65	-	mA
ADC Clock Frequency	F_{ADCLK}	Vdd =5.0V	-	-	32M	Hz
		Vdd =3.0V	-	-	16M	Hz
ADC Conversion Cycle Time	F_{ADCYL}	Vdd =2.0V~5.5V	64	-	-	$1/F_{ADCLK}$
ADC Sampling Rate (Set FADS=1 Frequency)	F_{ADSMP}	Vdd =5.0V	-	-	500	K/sec
		Vdd =3.0V	-	-	250	K/sec
Differential Nonlinearity	DNL	Vdd =5.0V , AVREFH=2.4V, F_{ADSMP} =62.5K	-	±2	-	LSB
		Vdd =5.0V , AVREFH=2.4V, F_{ADSMP} =250K	-	±2	-	LSB
		Vdd =5.0V , AVREFH=2.4V, F_{ADSMP} =500K	-	±5	-	LSB
Integral Nonlinearity	INL	Vdd =5.0V , AVREFH=2.4V, F_{ADSMP} =62.5K	-	±3	-	LSB
		Vdd =5.0V , AVREFH=2.4V, F_{ADSMP} =250K	-	±3	-	LSB
		Vdd =5.0V , AVREFH=2.4V, F_{ADSMP} =500K	-	±5	-	LSB
No Missing Code	NMC	Vdd =5.0V , AVREFH=2.4V, F_{ADSMP} =62.5K	10	11	12	Bits
		Vdd =5.0V , AVREFH=2.4V, F_{ADSMP} =250K	-	11	-	Bits
		Vdd =5.0V , AVREFH=2.4V, F_{ADSMP} =500K	-	9	-	Bits
ADC offset Voltage	$V_{ADCoffset}$	Non-trimmed	-10	0	+10	mV
		Trimmed	-2	0	+2	mV

“ *” These parameters are for design reference, not tested.

OP AMP CHARACTERISTIC

(All of voltages refer to Vss, Vdd = 5.0V, Fosc = 16MHz, Fcpu = 4MHz, ambient temperature is 25℃ unless otherwise note.)

PARAMETER	SYM.	DESCRIPTION	MIN.	TYP.	MAX.	UNIT
Operating voltage	Vop		2.0	-	5.5	V
*Supply current	Iop	Vdd = 5V, Unit-gain-buffer (V+ tie to Vss).	-	100	-	uA
		Vdd = 3V, Unit-gain-buffer (V+ tie to Vss).	-	90	-	
*Common Mode Input Voltage Range	Vcmr	Vdd = 5.0V	Vss	-	Vdd	mV
*Input Offset Voltage	Vos	Unit-gain-buffer (V+ tie to 1/2*Vdd).	-	+5	-	mV
*Power Supply Rejection Ratio	PSRR	Vcm = Vss	50	-	70	dB
*Common Mode Rejection Ratio	CMRR	Vcm=-0.3V~5.0V. Vdd=5V.	50		-	dB
*Open-Loop Gain (Large Signal)	Aol	Vout = 0.2V~Vdd-0.2V. Vcm=Vss.	90			dB
Output Voltage Swing	Vos	Vopp = 2.5V.	Vss+15		Vdd-15	mV
Output Short Current	Isc	Unit Gain Buffer, Vi=Vdd~Vss, Vo=Vss~Vdd, Vdd=5V. (Vdd-1/2*Vdd and Vss+1/2*Vdd)	-	25	-	mA
*Output Slew Rate	Tosr	Unit-gain buffer, Vo=rising Vss~Vdd. Vdd=5V.	-	5	-	us
		Unit-gain buffer, Vo=falling Vdd~Vss. Vdd=5V.	-	5	-	

“ *” These parameters are for design reference, not tested.

COMPARATOR CHARACTERISTIC

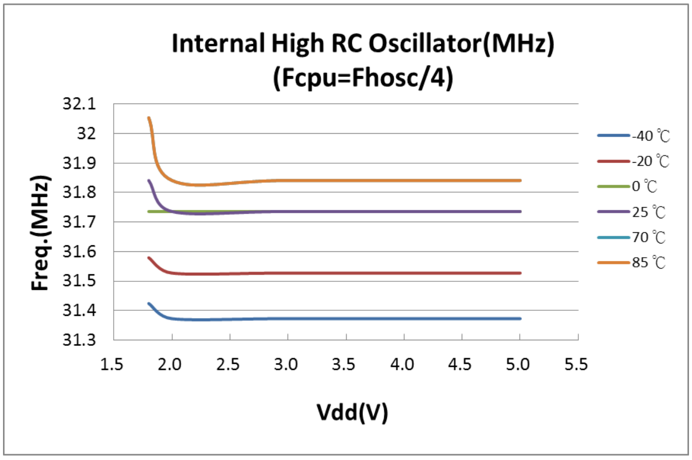
(All of voltages refer to Vss, Vdd = 5.0V, Fosc = 16MHz, Fcpu = 4MHz, ambient temperature is 25℃ unless otherwise note.)

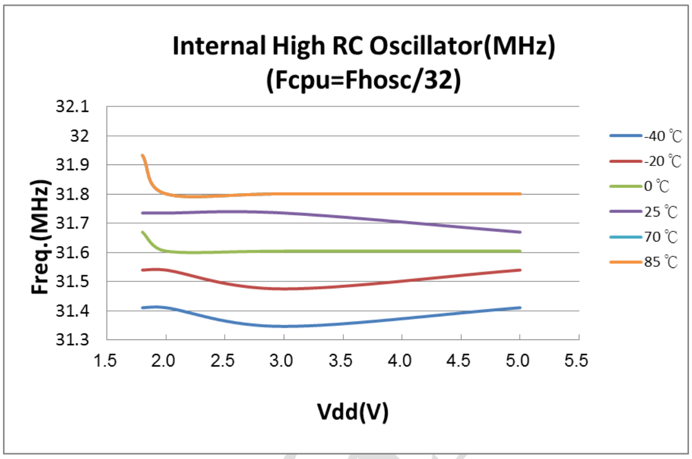
PARAMETER	SYM.	DESCRIPTION	MIN.	TYP.	MAX.	UNIT
Operating voltage	Vcm		2.0	-	5.5	V
*Supply current	Icm	CM0P=Vdd, CM0N=Vss, VDD = 5V	-	100	-	uA
*Input Offset Voltage	Vos	Vcm=Vdd/2, VDD=5V	-5	-	+5	mV
Response Time	Trs1	Vp=Vo=rising Vss~Vdd. Vn=2.5V. Vdd=5V.	-	100	-	ns
	Trs2	Vp=Vo=falling Vdd~Vss. Vn=2.5V. Vdd=5V.	-	100	-	
Output Slew Rate Time	Tsr	Vo=rising Vss~Vdd or falling Vdd~Vss. Vdd=5V.	-	20	-	ns
Common Mode Input Voltage Range	Vcmr	Vdd=5.0V	Vss+0.5		Vdd-0.5	V

“ *” These parameters are for design reference, not tested.

19.3 CHARACTERISTIC GRAPHS

The Graphs in this section are for design guidance, not tested or guaranteed. In some graphs, the data presented are outside specified operating range. This is for information only and devices are guaranteed to operate properly only within the specified range (-40℃~+85℃ curves are for design reference).

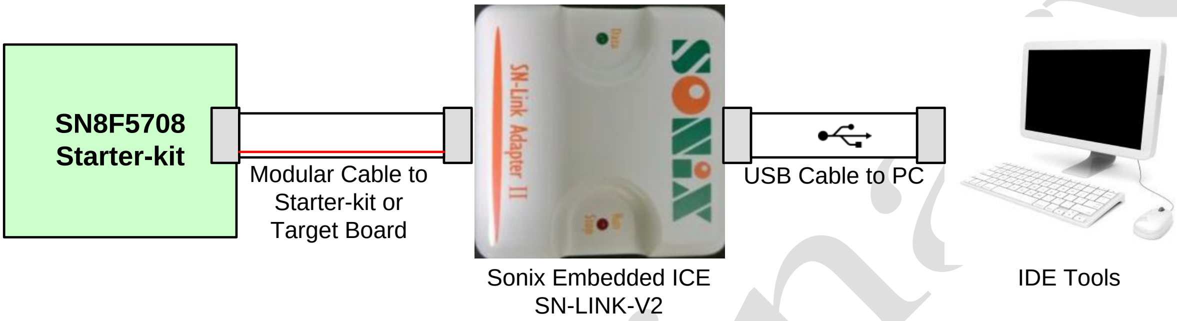




20 DEVELOPMENT TOOL

SONIX provides an Embedded ICE emulator system to offer SN8F5708 firmware development. The platform is a in-circuit debugger and controlled by Keil uVision IDE software on Microsoft Windows platform. The platform includes SN-LINK-V2, SN8F5708 Starter-kit and Keil uVision IDE software to build a high-speed, low cost, powerful and multi-task development environment including emulator, debugger and programmer. To execute emulation is like run real chip because the emulator circuit integrated in SN8F5708 to offer a real development environment.

SN8F5708 Embedded ICE Emulator System:



SN8F5708 Embedded ICE Emulator includes:

- SN-LINK-V2
- USB cable to provide communications between the SN-LINK-V2 and a PC.
- SN8F5708 Starter-Kit.
- Modular cable to connect the SN-LINK-V2 and SN8F5708 Starter-Kit or target board.
- Keil uVision IDE C51 V9.50 or greater.

SN8F5708 Embedded ICE Emulator Feature:

- Target's Operating Voltage: 3.0V~5.5V.
- Up to 5 hardware break points.
- System clock rate up to 32MHz (Fcpu=32mips, Finstruction=8mips).
- Oscillator supports internal high speed RC, internal low speed RC, external crystal/resonator and external clock.

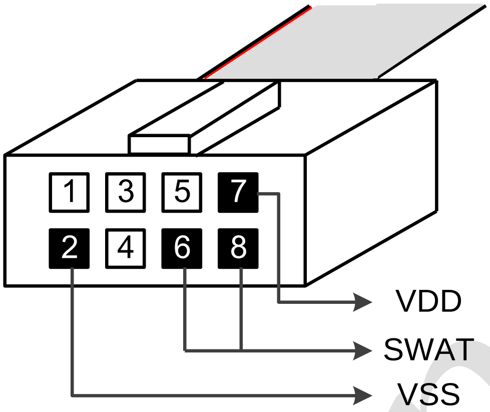
SN8F5708 Embedded ICE Emulator Limitation:

- SWAP pin is shared with GPIO pin. In embedded ICE mode, the shared GPIO function can't work. We strongly recommend planning this pins as simple function which can be verified without debugger platform.

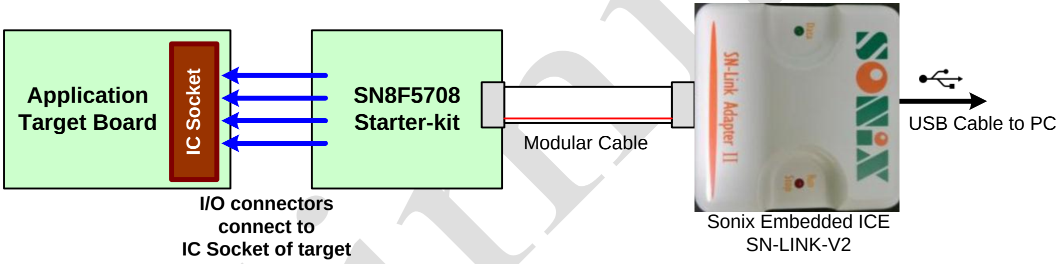
20.1 SN-LINK-V2

SN-LINK-V2 is a high speed emulator for Sonix Embedded ICE type flash MCU. It debugs and programs Sonix flash MCU and transfers MCU's system status, RAM data and system register between Keil uVision IDE and Sonix flash MCU through USB interface. The other terminal connected to SN8F5708 Starter-kit or Target board is a 4-wire serial interface. In addition to debugger functions, the Smart Starter-Kit system also may be used as a programmer to load firmware from PC to MCU for engineering production, even mass production.

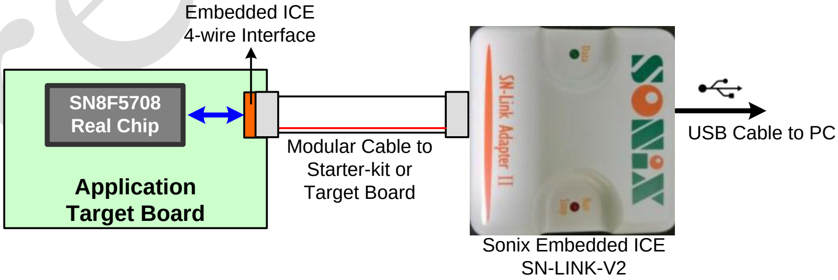
SN-LINK-V2 communication with SN8F5708 flash MCU is through a 4-wire bus. The pin definition of the Modular cable is as following:



The modular cable can be inserted into SN8F5708 Starter-Kit plugged into the target board or inserted into a matching socket at the target device on the target board.



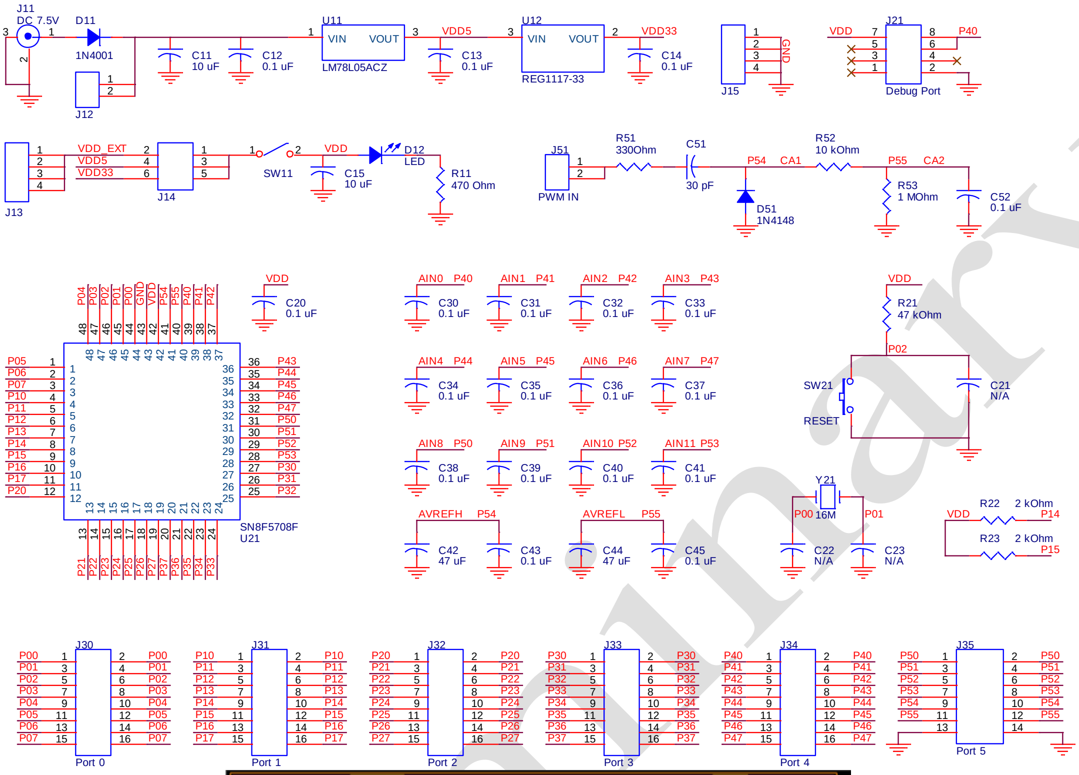
If the target board of application is designed and ready, the modular cable can be inserted into the target directly to replace SN8F5708 Starter-Kit. Design the 4-wire interface connected with SN8F5708 IC to build a real application environment. In the mode, set SN8F5708 IC on the target is necessary, or the emulation would be error without MCU.

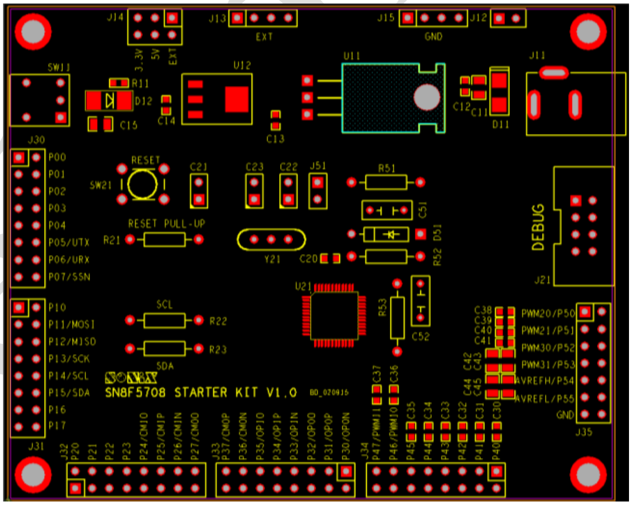

Embedded ICE 4-wire Interface
SN8F5708 Real Chip
Application Target Board
Modular Cable to Starter-kit or Target Board
USB Cable to PC
Sonix Embedded ICE SN-LINK-V2

SWAT share with P4.0 GPIO. In emulation mode, SWAT is Embedded ICE interface and not execute GPIO functions. The P4.0 GPIO status still display on Keil uVision IDE window to simulate P4.0 program execution.

20.2 SN8F5708 STARTER-KIT

SN8F5708 Starter-kit is an easy-development platform. It includes SN8F5708 real chip and I/O connectors to input signal or drive extra device of user's application. It is a simple platform to develop application as target board not ready. The starter-kit can be replaced by target board, because SN8F5708 integrates embedded ICE in-circuit debugger circuitry. The schematic and outline of SN8F5708 Starter-Kit is as following:





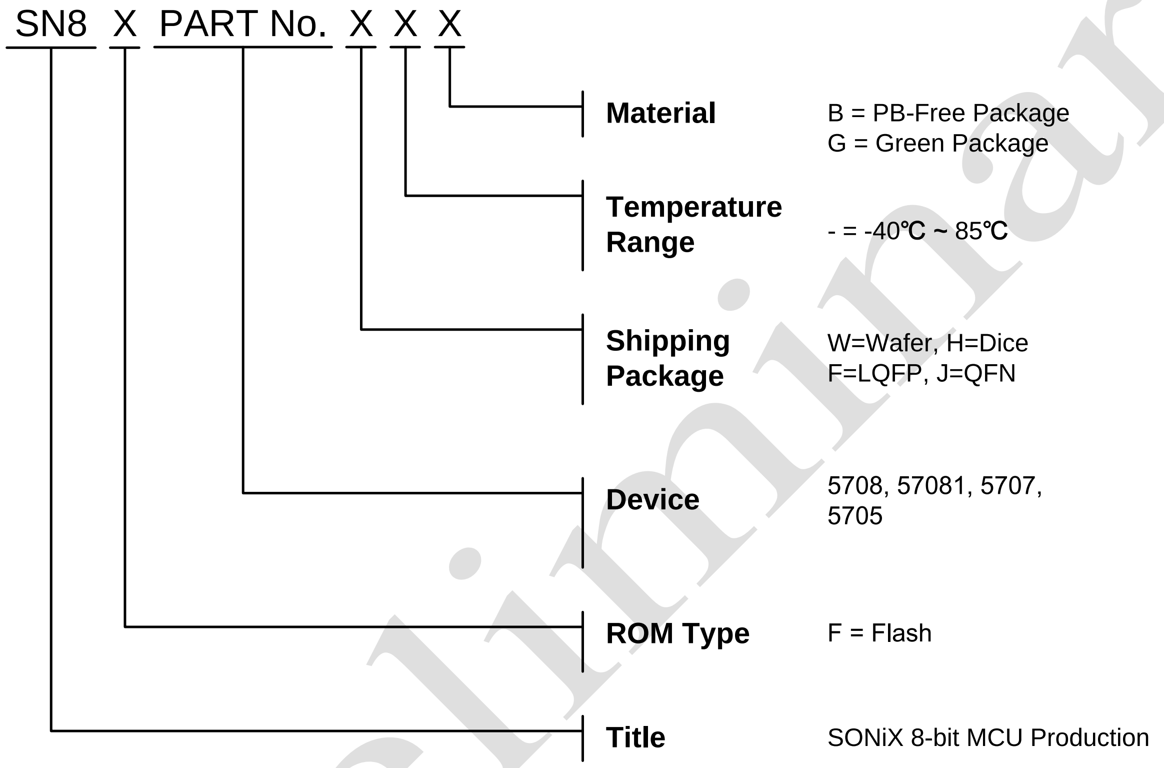
- J11: DC 7.5V power adapter.
- J14: VDD power source is 5.0V or 3.3V or external power.
- J13/J15: External power source.
- SW11: Target power switch.
- U21: SN8F5708F real chip (Sonix standard option).
- D12: MCU LED.
- C30~C41: 12-ch ADC capacitors.
- C42~C45: AVREFH and AVREFL capacitors.
- SW21: External reset trigger source.
- J30~J35: I/O connector.
- Y21, C22, C23: External crystal/resonator oscillator components.
- R22, R23: SCL and SDA pull-up resistors.
- J21: Debug connector.

21 MARKING DEFINITION

21.1 INTRODUCTION

There are many different types in Sonix 8-bit MCU production line. This note listed the production definition of all 8-bit MCU for order or obtain information. This definition is only for Blank Flash ROM MCU.

21.2 MARKING INDETIFICATION SYSTEM



SN8 X PART No. X X X

Material	B = PB-Free Package
	G = Green Package

Temperature Range	- = -40℃ ~ 85℃

Shipping Package	W=Wafer, H=Dice
	F=LQFP, J=QFN

Device	5708, 57081, 5707, 5705
ROM Type	F = Flash
Title	SONiX 8-bit MCU Production

21.3 MARKING EXAMPLE

- **Wafer, Dice:**

Name	ROM Type	Device	Package	Temperature	Material
S8F5708W	FLASH	5708	Wafer	-40℃~85℃	-
SN8F5708H	FLASH	5708	Dice	-40℃~85℃	-

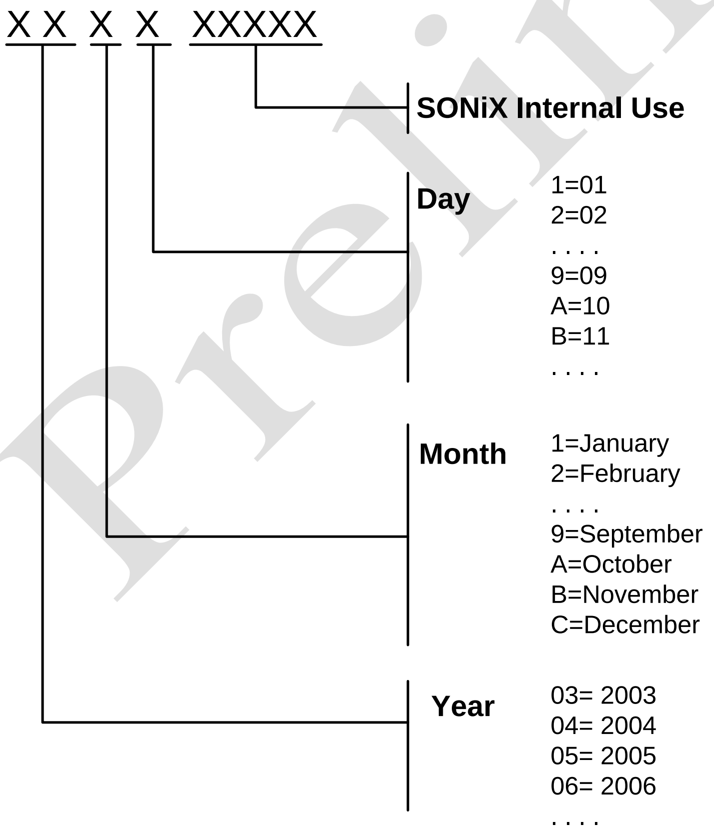
- **Green Package:**

Name	ROM Type	Device	Package	Temperature	Material
SN8F5708FG	FLASH	5708	LQFP	-40℃~85℃	Green Package
SN8F5708JG	FLASH	5708	QFN	-40℃~85℃	Green Package
SN8F57081JG	FLASH	5708	QFN	-40℃~85℃	Green Package
SN8F5707FG	FLASH	5707	LQFP	-40℃~85℃	Green Package
SN8F5705FG	FLASH	5705	LQFP	-40℃~85℃	Green Package
SN8F5705JG	FLASH	5705	QFN	-40℃~85℃	Green Package

- **PB-Free Package:**

Name	ROM Type	Device	Package	Temperature	Material
SN8F5708FB	FLASH	5708	LQFP	-40℃~85℃	PB-Free Package
SN8F5708JB	FLASH	5708	QFN	-40℃~85℃	PB-Free Package
SN8F57081JB	FLASH	5708	QFN	-40℃~85℃	PB-Free Package
SN8F5707FB	FLASH	5707	LQFP	-40℃~85℃	PB-Free Package
SN8F5705FB	FLASH	5705	LQFP	-40℃~85℃	PB-Free Package
SN8F5705JB	FLASH	5705	QFN	-40℃~85℃	PB-Free Package

21.4 DATECODE SYSTEM



X X X X XXXXX

SONiX Internal Use

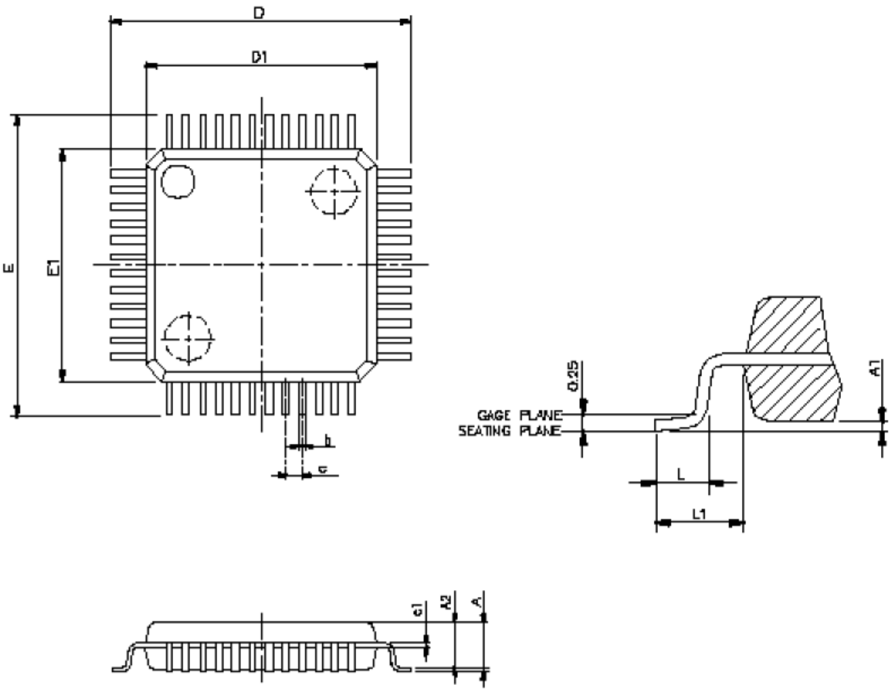
Day
1=01
2=02
. . . .
9=09
A=10
B=11
. . . .

Month
1=January
2=February
. . . .
9=September
A=October
B=November
C=December

Year
03= 2003
04= 2004
05= 2005
06= 2006
. . . .

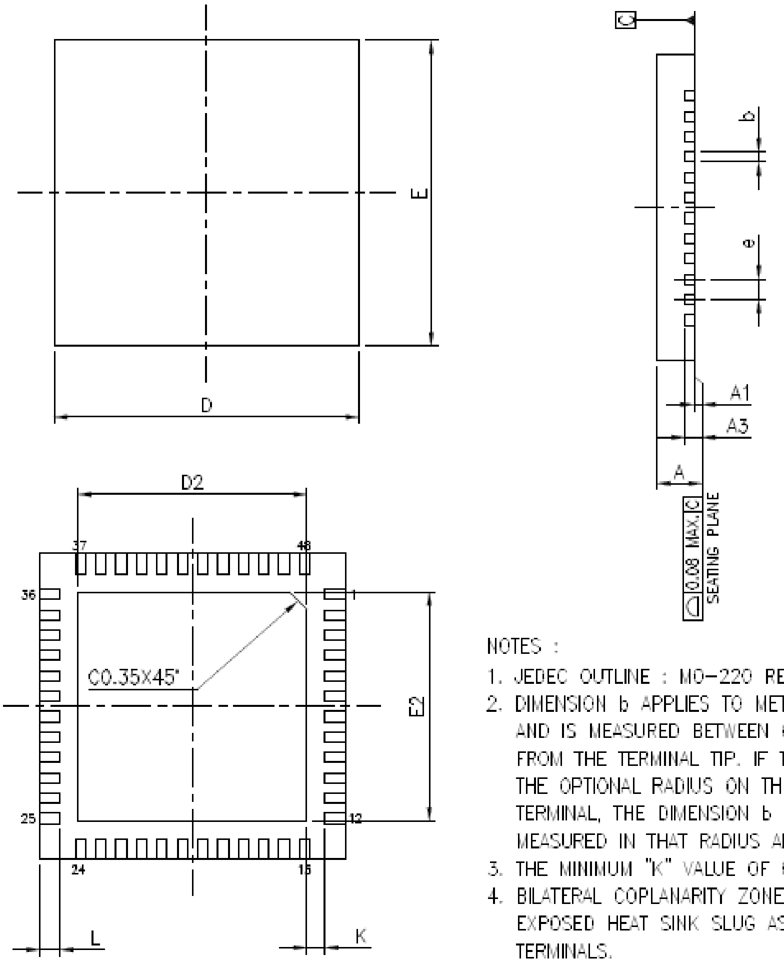
22 PACKAGE INFORMATION

22.1 LQFP 48 PIN



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	-	-	0.063	-	-	1.6
A1	0.002	-	0.006	0.05	-	0.15
A2	0.053	-	0.057	1.35	-	1.45
c1	0.004	-	0.006	0.09	-	0.16
D	0.354 BSC			9 BSC		
D1	0.276 BSC			7 BSC		
E	0.354 BSC			9 BSC		
E1	0.276 BSC			7 BSC		
e	0.020 BSC			0.5 BSC		
B	0.007	-	0.011	0.17	-	0.27
L	0.018	-	0.030	0.45	-	0.75
L1	0.039 REF			1 REF		

22.2 QFN 48 PIN



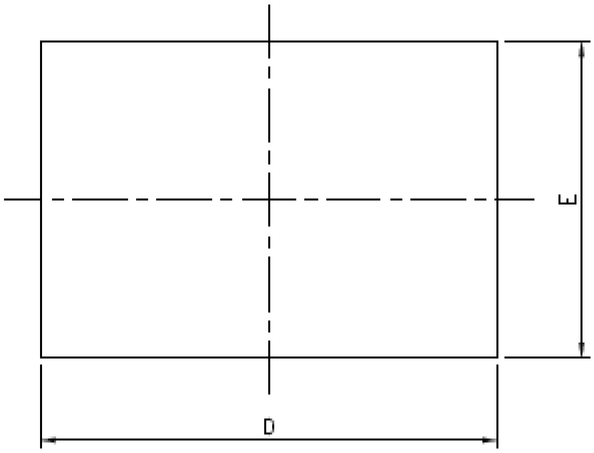
NOTES :

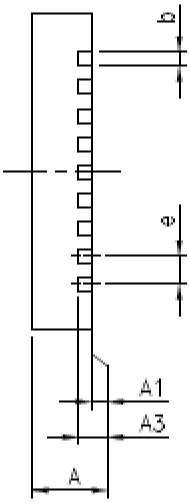
1. JEDEC OUTLINE : MO-220 REV.J(VJJE-1)
2. DIMENSION b APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15mm AND 0.30mm FROM THE TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION b SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
3. THE MINIMUM "K" VALUE OF 0.20mm APPLIES.
4. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.

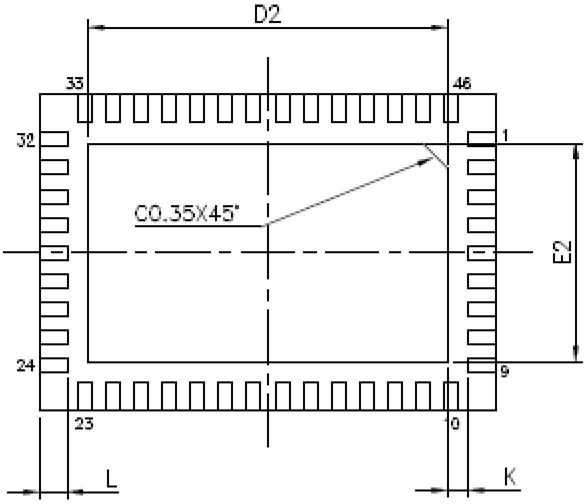
SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
		(inch)			(mm)	
A	0.031	0.033	0.035	0.80	0.85	0.90
A1	0	0.001	0.002	0.00	0.02	0.05
A3		0.008 REF.			0.20 REF.	
b	0.006	0.008	0.010	0.15	0.20	0.25
D		0.236 BSC			6.00 BSC	
E		0.236 BSC			6.00 BSC	
e		0.016 BSC			0.40 BSC	
L	0.014	0.016	0.018	0.35	0.40	0.45
K	0.008	-	-	0.20	-	-

PAD SIZE	D2 (mm)			E2 (mm)		
	MIN	NOR	MAX	MIN	NOR	MAX
185x185 MIL	4.40	4.50	4.55	4.40	4.50	4.55

22.3 QFN 46 PIN





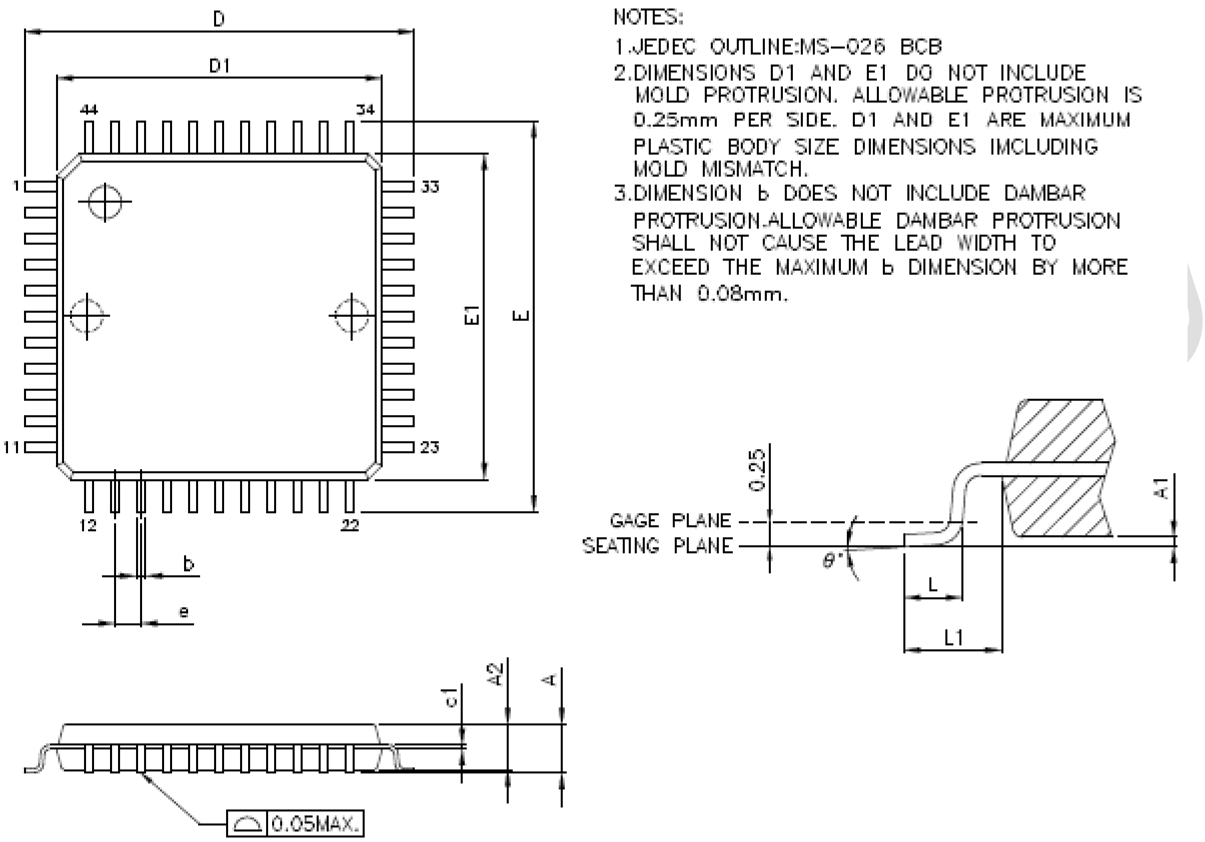


NOTES :

1. JEDEC OUTLINE : N/A.
2. DIMENSION b APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15mm AND 0.30mm FROM THE TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION b SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
3. THE MINIMUM "K" VALUE OF 0.20mm APPLIES.
4. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.

SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
		(inch)			(mm)	
A	0.031	0.033	0.035	0.80	0.85	0.90
A1	0	0.001	0.002	0.00	0.035	0.05
A3		0.008 REF.			0.203 REF.	
b	0.007	0.010	0.012	0.18	0.25	0.30
D	0.252	0.256	0.260	6.40	6.50	6.60
E	0.173	0.177	0.181	4.40	4.50	4.60
e		0.016 BSC			0.40 BSC	
D2	0.197	0.201	0.205	5.00	5.10	5.20
E2	0.118	0.122	0.126	3.00	3.10	3.20
L	0.014	0.016	0.018	0.35	0.40	0.45
K	0.008	-	-	0.20	-	-

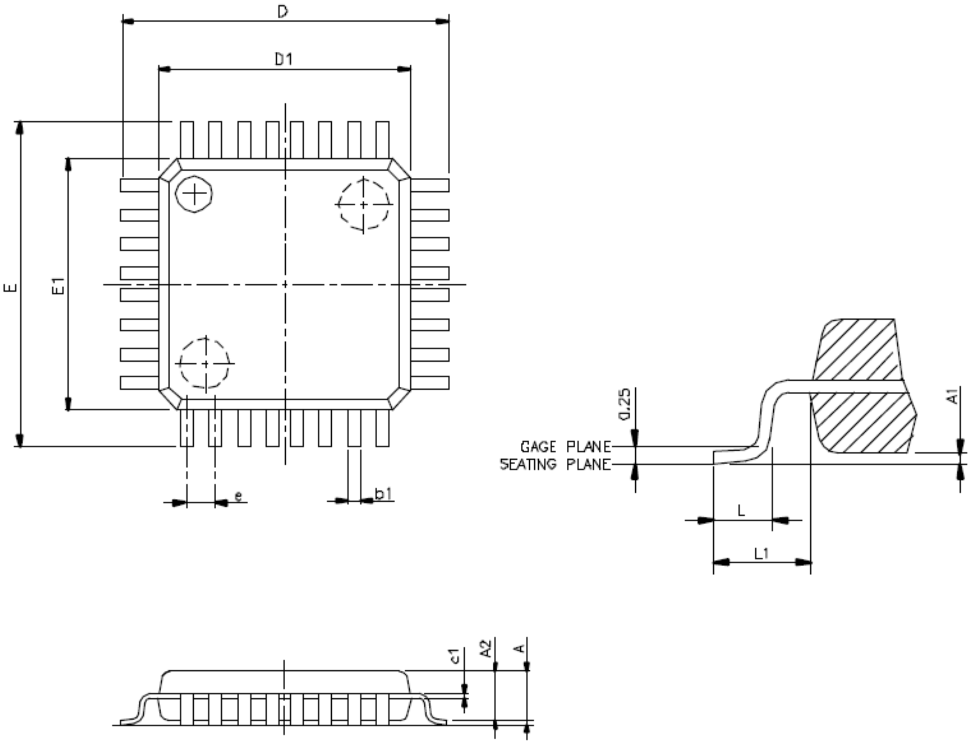
22.4 LQFP 44 PIN



NOTES:
1.JEDEC OUTLINE:MS-026 BCB
2.DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25mm PER SIDE. D1 AND E1 ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS IMCLUDING MOLD MISMATCH.
3.DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE MAXIMUM b DIMENSION BY MORE THAN 0.08mm.

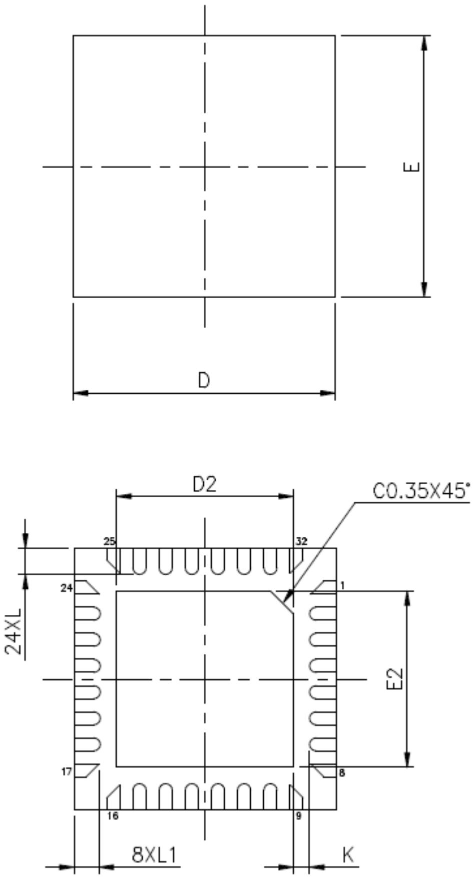
SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	-	-	0.063	-	-	1.6
A1	0.002	-	0.006	0.05	-	0.15
A2	0.053	0.055	0.057	1.35	1.4	1.45
c1	0.004	-	0.006	0.09	-	0.16
D	0.472 BSC			12 BSC		
D1	0.394 BSC			10 BSC		
E	0.472 BSC			12 BSC		
E1	0.394 BSC			10 BSC		
e	0.031 BSC			0.8 BSC		
b	0.010	0.012	0.014	0.25	0.30	0.35
L	0.018	0.024	0.030	0.45	0.60	0.75
L1	0.039 REF			1 REF		
θ°	0°	0.138°	0.276°	0°	3.5°	7°

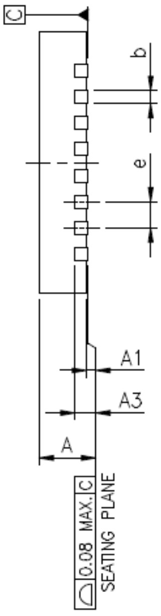
22.5 LQFP 32 PIN



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
		(inch)			(mm)	
A	-	-	0.063	-	-	1.6
A1	0.002	0.004	0.006	0.05	0.1	0.15
A2	0.053	0.055	0.057	1.35	1.4	1.45
c1	0.004	0.005	0.006	0.09	0.125	0.16
D	0.354 BSC			9 BSC		
D1	0.276 BSC			7 BSC		
E	0.354 BSC			9 BSC		
E1	0.276 BSC			7 BSC		
e	0.031 BSC			0.8 BSC		
b	0.012	0.015	0.018	0.3	0.375	0.45
L	0.018	0.024	0.030	0.45	0.6	0.75
L1	0.039 REF			1 REF		

22.6 QFN 4X4 32 PIN





NOTES :

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSION b APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15mm AND 0.30mm FROM THE TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION b SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
3. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.

SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
		(inch)			(mm)	
A	0.028	0.030	0.031	0.70	0.75	0.80
A1	0.000	0.001	0.002	0.00	0.02	0.05
A3		0.008 REF.			0.20 REF.	
b	0.006	0.008	0.010	0.15	0.20	0.25
D		0.157 BSC			4.00 BSC	
E		0.157 BSC			4.00 BSC	
e		0.016 BSC			0.40 BSC	
L	0.014	0.016	0.018	0.35	0.40	0.45
L1	0.013	0.015	0.017	0.332	0.382	0.432
K	0.008	-	-	0.20	-	-

PAD SIZE	D2 (mm)			E2 (mm)		
	MIN	NOR	MAX	MIN	NOR	MAX
114x114 MIL	2.60	2.70	2.75	2.60	2.70	2.75

SONIX reserves the right to make change without further notice to any products herein to improve reliability, function or design. SONIX does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. SONIX products are not designed, intended, or authorized for us as components in systems intended, for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SONIX product could create a situation where personal injury or death may occur. Should Buyer purchase or use SONIX products for any such unintended or unauthorized application. Buyer shall indemnify and hold SONIX and its officers , employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that SONIX was negligent regarding the design or manufacture of the part.

Corporate Headquarters:
10F-1, No.36, Taiyuan Street, Chupei City, Hsinchu, Taiwan
TEL :(886)3-5600-888 FAX :(886)3-5600-889
Taipei Sales Office:
15F-2, No.171 Song Ted Road, Taipei, Taiwan
TEL:(886)2-2759-1980 FAX:(886)2-2759-8180
mkt@sonix.com.tw | sales@sonix.com.tw
Hong Kong Sales Office:
Unit 2603, 26/F CCT Telecom Building, No. 11 Wo Shing Street,
Fo Tan, New Territories, Hong Kong
TEL:(852)2723-8086 FAX:(852)2723-9179
hk@sonix.com.tw
Shenzhen Contact Office:
High Tech Industrial Park, Shenzhen, China
TEL:(86)755-2671-9666 FAX:(86)755-2671-9786
mkt@sonix.com.tw | sales@sonix.com.tw
Sonix Japan Office:
Kobayashi bldg. 2F, 4-8-27,Kudanminami, Chiyodaku,Tokyo,
102-0074, Japan
TEL:(81)3-6272-6070 FAX:(81)3-6272-6165
jpsales@sonix.com.tw
FAE Support via Email:
8-bit Microcontroller Products: sa1fae@sonix.com.tw
All Products: fae@sonix.com.tw