



十速

**TM56E6422*****DATA SHEET******Rev 1.00***

**tenx** reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **tenx** 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. **tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses **tenx** products for any such unintended or unauthorized application, Buyer shall indemnify and hold **tenx** 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 **tenx** was negligent regarding the design or manufacture of the part.

## PRECAUTION

1. The user must switch RDCTL to “4ns” to enhance the performance of minimal operating voltage
2. A sleep instruction is necessary before the event of pin-change otherwise pin-change event may be missed.
3. To read the chapter of "FAMILY OVERVIEW" before the emulation to understand the limitation.

## CONTENTS

|                                                   |           |
|---------------------------------------------------|-----------|
| <b>PRECAUTION</b> .....                           | <b>2</b>  |
| <b>CONTENTS</b> .....                             | <b>3</b>  |
| <b>FAMILY OVERVIEW</b> .....                      | <b>5</b>  |
| <b>FEATURES</b> .....                             | <b>6</b>  |
| <b>SYSTEM BLOCK DIAGRAM</b> .....                 | <b>9</b>  |
| <b>PIN ASSIGNMENT</b> .....                       | <b>10</b> |
| <b>PIN DESCRIPTIONS</b> .....                     | <b>11</b> |
| <b>PIN SUMMARY</b> .....                          | <b>12</b> |
| <b>FUNCTION DESCRIPTION</b> .....                 | <b>13</b> |
| 1 CPU Core.....                                   | 13        |
| 1.1 Program ROM (PROM) .....                      | 13        |
| 1.1.1 Reset Vector (000h) .....                   | 13        |
| 1.1.2 Interrupt Vector (004h) .....               | 13        |
| 1.2 System Configuration Register (SYSCFG).....   | 14        |
| 1.3 EEPROM.....                                   | 14        |
| 1.3.1 The Precaution of EEPROM Programming .....  | 16        |
| 1.4 RAM Addressing Mode .....                     | 18        |
| 1.5 Programming Counter (PC) and Stack.....       | 21        |
| 1.5.1 ALU and Working (W) Register .....          | 24        |
| 1.5.2 STATUS Register (003h/083h/103h/183h) ..... | 24        |
| 2 Reset .....                                     | 26        |
| 2.1 Power on Reset (POR) .....                    | 26        |
| 2.2 Low Voltage Reset (LVR) .....                 | 26        |
| 2.3 External Pin Reset (XRST) .....               | 26        |
| 2.4 Watchdog Timer Reset (WDTR) .....             | 27        |
| 3 Clock Circuitry and Operation Mode .....        | 27        |
| 3.1 System Clock.....                             | 27        |
| 3.2 Dual System Clock Modes Transition .....      | 30        |
| 3.3 System Clock Oscillator.....                  | 33        |
| 4 Interrupt .....                                 | 34        |
| 5 I/O Port .....                                  | 39        |
| 5.1 PA0-PA7, PB0-PB7, PD0-PD1.....                | 39        |
| 5.2 Pin-change Wake-up & Interrupt .....          | 45        |
| 6 Peripheral Functional Block .....               | 46        |
| 6.1 Watchdog (WDT) /Wake-up (WKT) Timer .....     | 46        |
| 6.2 Timer0 .....                                  | 49        |
| 6.3 Timer1 .....                                  | 53        |
| 6.4 T2:15-bit Timer .....                         | 56        |
| 6.5 PWM: 16 bits PWM.....                         | 58        |
| 6.5.1 PWM0 .....                                  | 58        |

|                                              |                                     |            |
|----------------------------------------------|-------------------------------------|------------|
| 6.5.2                                        | PWM1~5 .....                        | 61         |
| 6.6                                          | Analog-to-Digital Converter .....   | 66         |
| 6.6.1                                        | Normal Trigger ADC .....            | 66         |
| 6.6.2                                        | PWM Trigger ADC .....               | 67         |
| 6.7                                          | Comparator .....                    | 70         |
| 6.8                                          | Cyclic Redundancy Check (CRC).....  | 73         |
| 6.9                                          | S/W Control LCD Driver .....        | 74         |
| <b>MEMORY MAP.....</b>                       |                                     | <b>76</b>  |
| <b>INSTRUCTION SET.....</b>                  |                                     | <b>86</b>  |
| <b>ELECTRICAL CHARACTERISTICS .....</b>      |                                     | <b>100</b> |
| 1.                                           | Absolute Maximum Ratings .....      | 100        |
| 2.                                           | DC Characteristics .....            | 100        |
| 3.                                           | Clock Timing .....                  | 101        |
| 4.                                           | Reset Timing Characteristics .....  | 101        |
| 5.                                           | LVR Circuit Characteristics .....   | 102        |
| 6.                                           | LVD Circuit Characteristics .....   | 102        |
| 7.                                           | ADC Electrical Characteristics..... | 103        |
| 8.                                           | EEPROM Characteristics .....        | 103        |
| 9.                                           | Comparator Characteristics .....    | 104        |
| 10.                                          | Characteristics Graphs .....        | 104        |
| <b>PACKAGING INFORMATION.....</b>            |                                     | <b>108</b> |
| <b>APPENDIX: PIN ASSIGNMENT DIAGRAM.....</b> |                                     | <b>115</b> |
| <b>AMENDMENT HISTORY .....</b>               |                                     | <b>117</b> |

## FAMILY OVERVIEW

|                                  |                                                                 |                                                                                                                   |
|----------------------------------|-----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
|                                  | TM56F1552 (TK)<br>TM56F1522 (IO)                                | TM56E6422                                                                                                         |
| EV board                         | On chip debug                                                   | TM56F1552 (TK)<br>TM56F1522 (IO)                                                                                  |
| ROM                              | 4k x 16                                                         | Please refer to the data in the chapter of “ <a href="#">FEATURES</a> ”. (Left click the link to go to that page) |
| RAM                              | 336                                                             |                                                                                                                   |
| EEPROM                           | 128                                                             |                                                                                                                   |
| EEPROM write                     | Halt CPU until EEP write complete                               | CPU is still running. Use EEPIF to confirm whether the write is complete                                          |
| EEPROM read                      | No limit                                                        | Before EEPIF=1, EEPROM cannot be read                                                                             |
| EEPIF                            | X                                                               | V                                                                                                                 |
| CTK                              | V                                                               | X                                                                                                                 |
| SIRC                             | 84 KHz@5V/25°C                                                  | 65.5 KHz±2%@25°C/5V                                                                                               |
| WDT                              | 96ms, 192ms, 768ms, 1536ms @5V                                  | 125ms, 250ms, 1001ms, 2001 ms @5V                                                                                 |
| WKT                              | 12ms, 24ms, 48ms, 96ms @5V                                      | 15.6ms, 31.3ms, 62.5ms, 125ms @5V                                                                                 |
| SFR.RDCTL                        | X                                                               | V (suggest RDCTL=4ns)                                                                                             |
| OPA                              | V                                                               | X                                                                                                                 |
| SFR.OPOF (CMPP to OPO)           | OPOF=0 (POR, CMPP ≤ OPO)<br>OPOF=1 (CMPP ≤ CIPx )               | No OPA<br>must set OPOF =1 in emulation<br>(CMPP connect to CIPx)                                                 |
| SFR.ADVREFS                      | VCC / 2.48V                                                     | VCC / 1.2V / 2V / 2.56V<br>ADVREFS=1.2V/2V, could not be emulated                                                 |
| SFR.BG2TRIM                      | X                                                               | Read BG2TRIM and Write into BGTRIM, obtain ADVREFS=2.0V                                                           |
| SFR.SVRF (DAC VREF)              | VCC / 1.2 / 2.48V                                               | VCC / 1.2 / 2.56V                                                                                                 |
| High Sink                        | 75mA@5V                                                         | 62mA@5V for all pins except PA7; <b>PA7 has no high sink, 1/2 bias and resistor pull-down</b>                     |
| POR                              | 1.95V<br>No PORSEL                                              | 1.63V<br>Has PORSEL                                                                                               |
| Minimal Operating Voltage        | 1.9V @16MHz                                                     | 2.3V @16MHz(PWMCKS=FIRC*1)                                                                                        |
| LVR <sub>th</sub>                | 2.05V~4.15V                                                     | 1.73V~3.5V                                                                                                        |
| LVD <sub>th</sub>                | 2.2V~4.15V                                                      | 1.73V~3.5V                                                                                                        |
| PWM                              | PWM0CLK: CPUCLK or FIRC (16MHz) or FIRC*2 (32MHz)<br>Normal PWM | PWM0CLK: CPUCLK or FIRC/256 or FIRC (16MHz) or FIRC*2 (32MHz)<br>Normal and Half-bridge PWM                       |
| ATD                              | X                                                               | V                                                                                                                 |
| All Pin-change Wake-up Interrupt | has pin-change wake-up, but no relative interrupt and flag      | V                                                                                                                 |
| PWM Trigger ADC                  | X                                                               | V                                                                                                                 |

## FEATURES

### 4. ROM: 4K x 16 bits MTP

### 5. EEPROM: 256 x 8 bits

### 6. RAM: 256 x 8 bits

### 7. STACK: 8 Levels

### 8. System Clock type selections:

- Fast clock from 1~20 MHz Crystal (FXT)
- Fast clock from Internal RC (FIRC, 16 MHz)
- Slow clock from 32768 Hz Crystal (SXT)
- Slow clock from Internal RC (SIRC, 65.5 KHz+/-2%@25°C/V<sub>CC</sub>=5V)

### 9. System Clock Prescaler:

- System Clock can be divided by 1/2/4/8 option

### 10. Power Saving Operation Mode

- FAST Mode: Slow-clock is enabled, Fast-clock keeps CPU running
- SLOW Mode: Fast-clock can be disabled or enabled, Slow-clock keeps CPU running
- IDLE Mode: Fast-clock and CPU stop. Slow-clock, T2, or Wake-up Timer keep running
- STOP Mode: All clocks stop, T2 and Wake-up Timer stop

### 11. 3 Independent Timers

- Timer0
  - 8-bit timer divided by 1~256 pre-scale option / auto-reload / counter / interrupt / stop function
- Timer1
  - 8-bit timer divided by 1~256 pre-scale option / auto-reload / interrupt / stop function
  - Overflow and Toggle out
- T2
  - 15-bit timer with 4 interrupt interval time options
  - IDLE mode wake-up timer or used as one simple 15-bit time base
  - Clock source: Slow-clock, F<sub>sys</sub>/128, or FIRC/512 (16 MHz/512)

### 12. Interrupt

- Three External Interrupt pins
  - 1 pin is falling edge wake-up triggered & Interrupts
  - 2 pins are rising or falling edge wake-up triggered & Interrupt
- Timer0 / Timer1 / T2 / Wake-up Timer Interrupt
- ADC Interrupt
- Comparator Interrupt
- PWM Interrupt

- LVD Interrupt
- All Port Pin-change Wake-up Interrupt
- EEPROM Interrupt

### 13. Wake-up Timer (WKT)

- Clocked by built-in RC oscillator with 4 adjustable interrupt times  
- 15.6 ms / 31.3 ms / 62.5 ms / 125 ms @ $V_{CC}=5V$

### 14. Watchdog Timer (WDT)

- Clocked by built-in RC oscillator with 4 adjustable reset times  
- 125 ms / 250 ms / 1001 ms / 2001 ms @ $V_{CC}=5V$
- Watchdog timer can be disabled / enabled in STOP mode

### 15. Six 16 bits PWMs

- Six individual duty-adjustable, shared period-adjustable
- PWM clock source: System clock ( $F_{sys}$ ), FIRC/256, FIRC (16 MHz), FIRC\*2 (32 MHz)
- PWM0 supports complementary output (PWM0P, PWM0N)
- PWM0 output with dead-zone(non-overlap) time durations adjustable:  $(0\sim15)*(PWMCLK)$
- PWM0N/0P/1/2/3/4/5 has two outputs
- Half-bridge phase control output

### 16. 12-bit ADC with 14 channels for External Pin Input and 2 channels for Internal Voltage

- Two internal voltage channels: VBG,  $1/4V_{CC}$
- ADC reference voltage:  $V_{CC}$ ,  $V_{BG}$  (1.2V),  $V_{BG}$  (2.56V) and  $V_{BG}$  (2V)
- PWM trigger ADC

### 17. Comparator

- Comparator x 1  
- With 7-bit DAC input  
- DAC reference voltage:  $V_{CC}$  or  $V_{BG}$  (1.20V or 2.56V)

### 18. Reset Sources

- Power On Reset
- Watchdog Timer Reset
- Low Voltage Reset
- External Pin Reset

### 19. Low Voltage Reset (LVR) and Low Voltage Detection (LVD)

- 15-Level Low Voltage Reset: 1.73V ~ 3.5V, can be disabled
- 15-Level Low Voltage Detection: 1.73V ~ 3.5V, can be disabled

### 20. Operating Voltage

- $F_{sys}=16\text{ MHz}$ ,  $PWMCKS=FIRC*1$ ,  $LVR\sim5.5V$ . Suggest  $LVR\geq 2.30V$

- $F_{sys} = 8 \text{ MHz}$ ,  $PWMCKS = FIRC * 1$ ,  $LVR \sim 5.5V$ . Suggest  $LVR \geq 1.6V$

Note: Refer to the “Electrical Characteristics Graphs”.

**21. Operating Temperature Range :  $-40^{\circ}\text{C}$  to  $+105^{\circ}\text{C}$**

**22. Table Read Instruction: 16-bit ROM data lookup table**

**23. Integrated 16-bit Cyclic Redundancy Check (CRC) function**

**24. Instruction set: 39 Instructions**

**25. I/O ports:**

- Maximum 18 programmable I/O pins
  - Open-Drain Output
  - CMOS Push-Pull Output
  - Schmitt Trigger Input with pull-up / pull-down resistor option (PA7 has no pull-down resistor)
  - All I/O with High-Sink except PA7
  - $1/2 V_{CC}$  (LCD 1/2 bias) Output (except PA7)
- All pin-change wake-up (falling edge and rising edge trigger) and interrupt

**26. LCD Driver**

- Maximum 17 software controlled COM
- LCD 1/2 bias

**27. Programming connectivity support 5-wire (ICP) or 6-wire program**

**28. RDCTL: Read signal delay control for Program ROM**

- The user must switch this register to “4ns” to enhance the performance of minimal operating voltage

**29. Trimmed VBG1.2V/2V**

- The users could move BG2TRIM to BGTRIM for exact 2V VBG.

**30. ATD: Automatic transient detection(Read signal length control for Program ROM) to enhance the performance of power consumption at slow mode**

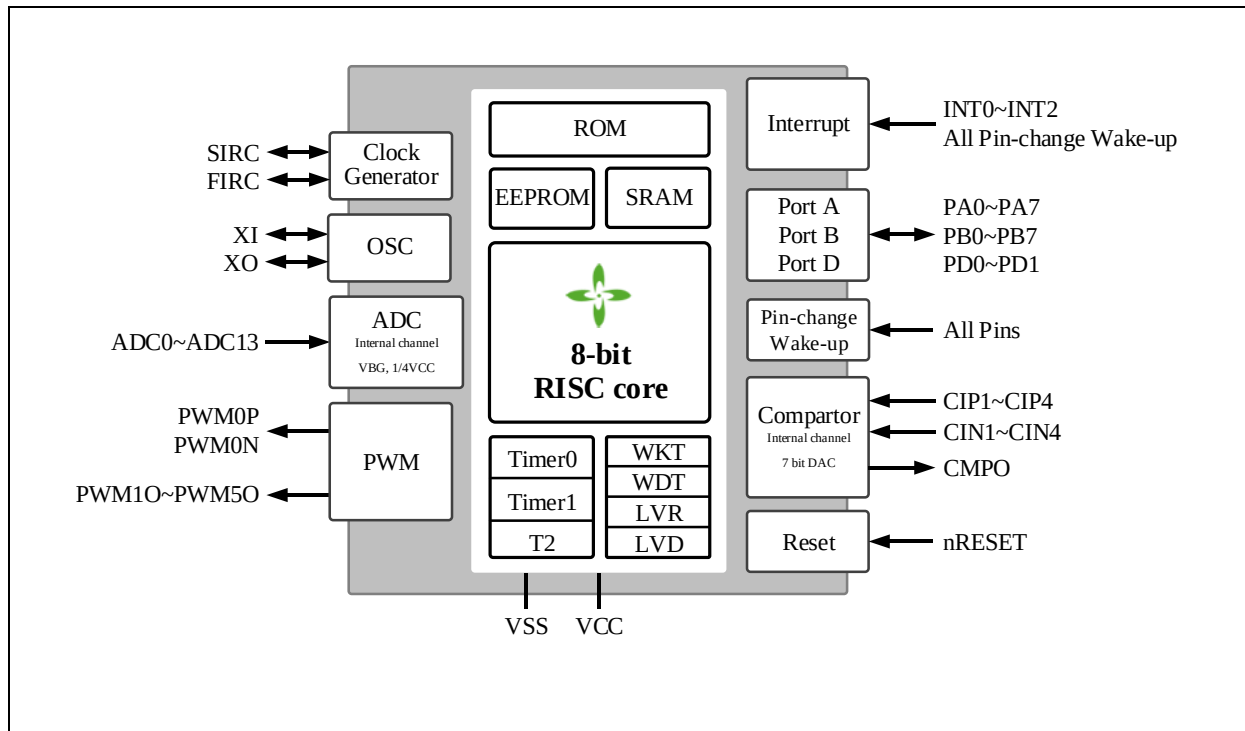
**31. Package Types: Please refer to the chapter of “PACKAGING INFORMATION” (Left click the link to go to that page)**

**32. Supported EV board**

- TM56F1552/22



## SYSTEM BLOCK DIAGRAM



**Block Diagram**

## PIN ASSIGNMENT

Please refer to “[APPENDIX: PIN ASSIGNMENT DIAGRAM](#)” (Left click the link to go to that page)

## PIN DESCRIPTIONS

| Name                          | In/Out | Pin Description                                                                                                                                                                                                                                                             |
|-------------------------------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PA0~PA7<br>PB0~PB7<br>PD0~PD1 | I/O    | Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output, open-drain output or 1/2V <sub>CC</sub> (LCD 1/2 bias) output. Pull-up/Pull-down resistors are assignable by software. <b>PA7 has no high-sink, 1/2 bias and resistor pull-down capability.</b> |
| nRESET                        | I      | External active low reset                                                                                                                                                                                                                                                   |
| VCC, VSS                      | P      | Power Voltage input pin and ground                                                                                                                                                                                                                                          |
| VPP                           | I      | Programming high voltage(9.5V) input(not applicable for EEPROM)                                                                                                                                                                                                             |
| XIN, XOUT                     | –      | Crystal/Resonator oscillator connection for System clock (FXT or SXT)                                                                                                                                                                                                       |
| INT0~INT2                     | I      | External interrupt input                                                                                                                                                                                                                                                    |
| TM0CKI                        | I      | Timer0's input in counter mode                                                                                                                                                                                                                                              |
| PWM0P                         | O      | 16 bits PWM0 positive output                                                                                                                                                                                                                                                |
| PWM0N                         | O      | 16 bits PWM0 negative output                                                                                                                                                                                                                                                |
| PWM1O~PWM5O                   | O      | 16 bits PWM1~PWM5 output                                                                                                                                                                                                                                                    |
| CMPO                          | O      | Comparator status output                                                                                                                                                                                                                                                    |
| TCOUT                         | O      | Fsys/2 clock output                                                                                                                                                                                                                                                         |
| TM1OUT                        | O      | Timer1 overflow toggle output                                                                                                                                                                                                                                               |
| ADC0~ADC13                    | I      | ADC channel input                                                                                                                                                                                                                                                           |
| CIN1~CIN4                     | I      | Comparator negative port input                                                                                                                                                                                                                                              |
| CIP1~CIP4                     | I      | Comparator positive port input                                                                                                                                                                                                                                              |
| COM0~COM16                    | O      | LCD 1/2 bias output                                                                                                                                                                                                                                                         |
| PSCL                          | I      | Clock for programmer                                                                                                                                                                                                                                                        |
| PSDA                          | I/O    | Data for programmer                                                                                                                                                                                                                                                         |

Programming pins:

Normal mode (6-wire): VCC / VSS / PA0(PSDA) / PA1(PSCL) / PA5 / PA7(VPP)

ICP mode (5-wire): VCC / VSS / PA0(PSDA) / PA1(PSCL) / PA7(VPP) - When using ICP (In-Circuit Program) mode, the PCB needs to remove all components of PA0, PA1.

**PIN SUMMARY**

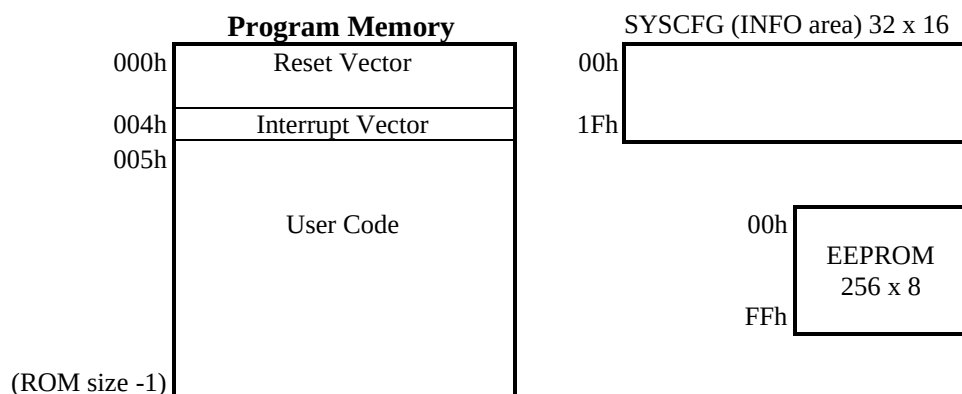
| Pin Number | Pin Name                                                      | Type | GPIO            |                   |                |                   |            |                |                                    | Alternate Function |     |            |             |
|------------|---------------------------------------------------------------|------|-----------------|-------------------|----------------|-------------------|------------|----------------|------------------------------------|--------------------|-----|------------|-------------|
|            |                                                               |      | Input           |                   |                |                   | Output     |                |                                    | PWM                | ADC | Comparator | MISC        |
|            |                                                               |      | Pull-up Control | Pull-down Control | Ext. Interrupt | Wake-up Interrupt | Open Drain | CMOS Push-Pull | 1/2 V <sub>CC</sub> (LCD 1/2 Bias) |                    |     |            |             |
| 6          | VCC                                                           | P    |                 |                   |                |                   |            |                |                                    |                    |     |            |             |
| 2          | PA4/ <b>ADC4</b> / <b>PWM0P</b> /COM4/XIN                     | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  | ●                  | ●   |            | XIN         |
| 3          | PA5/ <b>ADC5</b> / <b>PWM3O</b> /COM5/XOUT/ <b>CMPO</b>       | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  | ●                  | ●   | ●          | XOUT        |
| 5          | PA7/nRESET/INT2/VPP                                           | I/O  | ●               |                   | ●              | ●                 | ●          |                |                                    |                    |     |            | nRESET/VPP  |
| 7          | PA6/ <b>ADC6</b> / <b>PWM0N</b> /COM6                         | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  | ●                  | ●   |            |             |
| 8          | PB0/ <b>ADC7</b> / <b>PWM1O</b> /COM7                         | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  | ●                  | ●   |            |             |
| 9          | PB1/ <b>ADC8</b> / <b>PWM2O</b> /COM8/INT1                    | I/O  | ●               | ●                 | ●              | ●                 | ●          | ●              | ●                                  | ●                  | ●   |            |             |
| 10         | PB2/ <b>ADC9</b> / <b>PWM3O</b> /COM9/INT0                    | I/O  | ●               | ●                 | ●              | ●                 | ●          | ●              | ●                                  | ●                  | ●   |            |             |
| 11         | PB3 / <b>PWM4O</b> /COM10                                     | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  | ●                  |     |            |             |
| 12         | PD0/COM15                                                     | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  |                    |     |            |             |
| 13         | PD1/COM16/TCOUT/ <b>CIP4</b>                                  | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  |                    |     | ●          | TCOUT       |
| 17         | PB4/ <b>ADC10</b> / <b>PWM0P</b> /COM11/ <b>CIN4</b>          | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  | ●                  | ●   | ●          |             |
| 16         | PB5/ <b>ADC11</b> / <b>PWM5O</b> /COM12/INT2                  | I/O  | ●               | ●                 | ●              | ●                 | ●          | ●              | ●                                  | ●                  | ●   |            |             |
| 14         | PB6/ <b>ADC12</b> / <b>PWM0N</b> /COM13/ <b>CIP3</b>          | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  | ●                  | ●   | ●          |             |
| 15         | PB7/ <b>ADC13</b> /COM14/TM1OUT/ <b>CIN3</b>                  | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  |                    | ●   | ●          | TM1OUT      |
| 18         | PA2/ <b>ADC2</b> / <b>PWM4O</b> /COM2/TM0CKI/ <b>CIP2</b>     | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  | ●                  | ●   | ●          | TM0CKI      |
| 19         | PA0/ <b>ADC0</b> / <b>PWM5O</b> /COM0/ <b>CIN2</b> /PSDA      | I/O  | ●               | ●                 |                | ●                 | ●          | ●              | ●                                  | ●                  | ●   | ●          | Programming |
| 20         | PA1/ <b>ADC1</b> / <b>PWM1O</b> /COM1/INT1/ <b>CIP1</b> /PSCL | I/O  | ●               | ●                 | ●              | ●                 | ●          | ●              | ●                                  | ●                  | ●   | ●          | Programming |
| 1          | PA3/ <b>ADC3</b> / <b>PWM2O</b> /COM3/INT0/ <b>CIN1</b>       | I/O  | ●               | ●                 | ●              | ●                 | ●          | ●              | ●                                  | ●                  | ●   | ●          |             |
| 4          | VSS                                                           | P    |                 |                   |                |                   |            |                |                                    |                    |     |            |             |

## FUNCTION DESCRIPTION

### 1 CPU Core

#### 1.1 Program ROM (PROM)

The ROM of this device has an extra 32-Word INFO area to store the SYSCFG and an extra 256-Byte EEPROM. The ROM can be written multi-times and can be read as long as the PROTECT (CFGWH.15) bit of SYSCFG is not set. The SYSCFG can be read no matter PROTECT is set or cleared, but PROTECT bit can be cleared only when User ROM Code area is erased. On the other hand, if PROTECT bit is set, the user ROM code area will not be read by writer, and the user ROM code can't be updated until the PROTECT bit is cleared. The endurance of ROM is 1000 times @Vcc=5V/25°C。



| 113h  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| RDCTL | —     | —     | —     | —     | —     | —     | RDCTL |       |
| R/W   | —     | —     | —     | —     | —     | —     | R/W   |       |
| Reset | —     | —     | —     | —     | —     | —     | 1     | 0     |

113h.1~0 **RDCTL:** Read signal delay control for Program ROM

00: 16ns delay for read signal of Program ROM

01: 12ns delay for read signal of Program ROM

10: 8ns delay for read signal of Program ROM

11: 4ns delay for read signal of Program ROM

Change this register at slow clock for safety.

**The user must switch this register to “4ns” to enhance the performance of minimal operating voltage.**

This feature can't be emulated.

##### 1.1.1 Reset Vector (000h)

After reset, system will restart the program counter (PC) at the address 000h, all registers will revert to the default value.

##### 1.1.2 Interrupt Vector (004h)

When an interrupt occurs, the program counter (PC) will be pushed onto the stack and jumps to address 004h.

## 1.2 System Configuration Register (SYSCFG)

The System Configuration Register (SYSCFG) is located at INFO area; it contains a 16 bits register (CFGWH). The SYSCFG determines the option for initial condition of CPU. It is written by PROM Write only. User can select LVR operation mode and chip operation mode by SYSCFG register. The 15<sup>th</sup> bit of CFGWH is code-protected selection bit. If this bit is 1, the data in PROM will be protected when user reads PROM.

| Bit           |       | 15~0                                                                                                                                                                                                        |                                                                                                    |
|---------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| Default Value |       | 0000_0110_0000_0000                                                                                                                                                                                         |                                                                                                    |
| Bit           |       | Description                                                                                                                                                                                                 |                                                                                                    |
| CFGWH         | 15    | <b>PROTECT</b> : Code protection selection                                                                                                                                                                  |                                                                                                    |
|               |       | 0                                                                                                                                                                                                           | Disable                                                                                            |
|               |       | 1                                                                                                                                                                                                           | Enable                                                                                             |
|               | 13-12 | <b>WDTE</b> : WDT Reset Enable                                                                                                                                                                              |                                                                                                    |
|               |       | 0X                                                                                                                                                                                                          | Disable                                                                                            |
|               |       | 10                                                                                                                                                                                                          | Enable in FAST/SLOW mode, Disable in IDLE/STOP mode                                                |
|               |       | 11                                                                                                                                                                                                          | Always Enable                                                                                      |
|               | 11-8  | <b>LVR</b> : Low Voltage Reset Voltage Selection<br>Please refer to the table of LVR voltage in the section of “ <a href="#">LVR Circuit Characteristics</a> ”.<br>(Left click the link to go to that page) |                                                                                                    |
|               |       |                                                                                                                                                                                                             |                                                                                                    |
|               | 7     | <b>XRSTE</b> : External Pin (PA7) Reset Enable                                                                                                                                                              |                                                                                                    |
|               |       | 0                                                                                                                                                                                                           | Disable (PA7 as I/O pin)                                                                           |
|               |       | 1                                                                                                                                                                                                           | Enable                                                                                             |
|               | 5     | <b>FIRCPSC</b> : FIRC Prescaler                                                                                                                                                                             |                                                                                                    |
|               |       | 0                                                                                                                                                                                                           | Divided by 1 (16 MHz)                                                                              |
|               |       | 1                                                                                                                                                                                                           | Divided by 2 (8 MHz)                                                                               |
|               | 4     | <b>PORSEL</b> : POR duty cycle selection                                                                                                                                                                    |                                                                                                    |
|               |       | 0                                                                                                                                                                                                           | POR enables at 100% duty cycle(POR is always on)                                                   |
|               |       | 1                                                                                                                                                                                                           | POR enables at 1/16 duty cycle(This feature can't be emulated)(POR is only on at part of the time) |
|               | 3     | <b>ATDOFF</b> : Automatic transient detection(Read signal length control for Program ROM)                                                                                                                   |                                                                                                    |
|               |       | 0                                                                                                                                                                                                           | ATD on(for power saving in SLOW mode)                                                              |
|               |       | 1                                                                                                                                                                                                           | ATD off(recommend for EFT issue)                                                                   |
|               | 2-0   | tenx Reserved                                                                                                                                                                                               |                                                                                                    |

## 1.3 EEPROM

The Chip contains 256 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. According the physical characteristic the EEPROM need more long access time than Program ROM. Before the writing of EEPROM is complete, the main program can still continue to execute. EEPIF will become 1 after EEPROM write is completed and then the users could clear it. Before EEPIF is 1, no more writing or reading can be done to the EEPROM.

**The EEPROM Read** usage is same as use Table Read instruction except EEPROM enable bit must be set to high. By writing 0xE2 to register EEPEN (191h) can set the EEPROM enable bit, writing other value to EEPEN (191h) will clear the EEPROM enable bit. To access EEPROM, the DPTR[11:8] must be set to 0.

◇ Example: read EEPROM in assembly method

```

MOVLW    E2h
MOVW    EEPEN          ; set EEPROM enable bit
CLR     DPH            ; set DPH = 00h
MOVLW    23h

```

```
MOVW DPL ; set DPL = 23h, DPTR = 0023h
TABRL ; W = TABR = data of EEPROM[23h]
; read EEPROM data into W by using TABRL
```

◇ Example: read EEPROM in C language method

```
EEPEN=0xE2; // set EEPROM enable bit
DPH=0; // set DPH = 00h
DPL=0x23; // set DPL = 23h, DPTR = 0023h
TABR=1; // write 01h to TABR = opcode TABRL
rData=TABR; // rData = data of EEPROM[23h]
```

*Note: When using C language to look up the table, the lookup table instruction can only be used outside the interrupt or within the interrupt, if the lookup table instruction is used inside and outside the interrupt, it may cause an error.*

**The EEPROM Write** usage is similar to read EEPROM. When F/W writes data to the register EEPDT (192h), the data will also be written to EEPROM. EEPROM writing requires typ 4ms@V<sub>CC</sub>=2.5V/25°C, 1.5ms@V<sub>CC</sub>=5V/25°C. This Chip has a build-in EEPROM Time-out function for escaping write fail state. EEPROM writing needs V<sub>CC</sub>>2.5V. Before the writing of EEPROM is complete, the main program can still continue to execute. EEPIF will become 1 after EEPROM write is completed and then the users could clear it. Before EEPIF is 1, no more writing or reading can be done to the EEPROM. **To clear watchdog first before writing EEPROM to avoid watchdog timer reset during writing procedure for safety.**

◇ Example: write EEPROM data A5 to address 23h

```
ORG 0h
LGOTO WREEP

ORG 4h
Interrupt:
;Save WREG/STATUS
BTXSS EEPIE ;check EEPIE
LGOTO $+3 ;check EEPROM write time-out flag
BTXSC EEPIF ;check EEPIF
LGOTO EEPINT
:
LGOTO IntEnd

EEPINT:
BTXSC EEPTO ;check EEPROM write time-out flag
BSX RAM21,1 ;set RAM21,1 eep Write Timeout (fail)

RDEEP:
TABRL ;W = TABR = data of EEPROM[23h]
MOVW RAM20 ;RAM20 = data of EEPROM[23h]
CLRX EEPEN ;clear EEPROM enable bit
MOVLW DFh
MOVW INTIF1 ;clear EEPIF
BSX RAM21,0 ;set RAM21,0 (EEPIF)
:

IntEnd:
;Restore STATUS/WREG
RETI

ORG 100h

WREEP:
CLRX INTIE ; disable INTIE
CLRX INTIE1 ; disable INTIE1
CLRX RAM20
CLRX RAM21
```

```

MOVLW      DFh
MOVWX      INTIF1           ;clear EEPIF
BSX        EEPIE           ;enable EEPIE Interrupt
MOVLW      00000011b
MOVWX      EEPCTL           ;set EEPROM write with 62.5ms time out
CLRXL      DPH              ;set DPH = 00h
MOVLW      23h
MOVWX      DPL              ;set DPL = 23h, DPTR = 0023h
CLRWDW      WDT              ;To clear WDT first before writing EEPROM
MOVLW      E2h
MOVWX      EEPEN           ;set EEPROM enable bit
MOVLW      A5h
MOVWX      EEPDT           ;write A5h to EEPDT
                        ;the data also save to EEPROM @Address 23h

BTXSS      RAM21,0          ;check RAM21,0 (EEPIF)
LGOTO      $-1
BTFSC      RAM21,1          ;check RAM21,1 eep Write Timeout (fail)
LGOTO      WREEP

MOVXL      RAM20
XORLW      A5h
BTXSS      STATUS, 2        ;Check RAM20 = A5h
LGOTO      WREEP
:

```

### 1.3.1 The Precaution of EEPROM Programming

#### 1.3.1.1 The Programming Characteristics of EEPROM

- (1) The EEPROM programming time is not fixed, and programming different data requires different times.
- (2) The programming time is affected by voltage, temperature, and whether the data is inverted. The programming time is longer for higher temperatures, lower VCC and larger number of data inversion.
- (3) This chip has a built-in EEPROM write time-out function to protect the write timeout and ensure that the system can execute the program normally.
- (4) VPP could be VCC, VSS or floating when EEPROM is programming.

#### 1.3.1.2 EEPROM Write Endurance

The number of EEPROM programming cycle is related to voltage and temperature. The write endurance is at least 30,000(2.5V < VCC < 5.5V, -20 °C ~85 °C ). Please refer to the table of “EEPROM Characteristics” in the chapter of “ELECTRICAL CHARACTERISTICS”.

#### 1.3.1.3 EEPROM Write Verification

Depending on the specific application, it is generally required to read back the data written into EEPROM for verification.

#### 1.3.1.4 Protection against Miswriting

When a write operation is initiated, the following operations can prevent miswriting operations:

- (1) Low voltage detection: when writing EEPROM, VCC must be >2.5V, you can use the LVD function to monitor the voltage  
(LVD monitoring voltage is recommended to be 3.5V to prevent power outage and allow sufficient time for writing EEPROM)
- (2) Clear the watchdog (WDT) every time a byte is written to prevent the watchdog from being reset when multiple bytes are written sequentially.
- (3) When writing data, temporarily turn off all of the interrupt and resume them after the completion of writing.
- (4) Software failure: add EEPROM read-back mechanism to the program to ensure that data is written correctly.
- (5) Timeout protection: enable the write timeout function (EEPTE) in the program to protect the system from getting stuck when write time-out occurred.



(6) To reduce power supply glitches: connect capacitors between VCC and GND to stabilize the system power supply.

(7) Before writing another byte, it is necessary to confirm that EEPIF=0.

| 0Dh    | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| INTIE1 | –     | PCIE  | EEPIE | CMPIE | –     | –     | PWMIE | LVDIE |
| R/W    | –     | R/W   | R/W   | R/W   | –     | –     | R/W   | R/W   |
| Reset  | –     | 0     | 0     | 0     | –     | –     | 0     | 0     |

0Dh.5 **EEPIE**: EEPROM interrupt enable

0: disable

1: enable

| 0Eh    | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| INTIF1 | –     | PCIF  | EEPIF | CMPIF | –     | –     | PWMIF | LVDIF |
| R/W    | –     | R/W   | R/W   | R/W   | –     | –     | R/W   | R/W   |
| Reset  | –     | 0     | 0     | 0     | –     | –     | 0     | 0     |

0Eh.5 **EEPIF**: EEPROM interrupt event pending flag

This bit is set by H/W when EEPROM writing is finishing or time-out event occurred. Writing DFh to INTIF1 will clear this flag.

| 190h   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| EEPCTL | EEPTO | –     | –     | –     | –     | –     | EEPTE |       |
| R/W    | R     | –     | –     | –     | –     | –     | R/W   |       |
| Reset  | 0     | –     | –     | –     | –     | –     | 0     | 0     |

190h.7 **EEPTO**: EEPROM write time-out flag

0: EEPROM write no time-out

1: EEPROM write is time-out. EEPTO will be cleared when EEPEN is disabled.

190h.1~0 **EEPTE**: EEPROM write time-out enable (access wait time)

00: Disable

01: 2 ms@5.0V, 2.5 ms@3.0V

10: 7.8 ms@5.0V, 10 ms@3.0V

11: 62.5 ms@5.0V, 79.6 ms@3.0V

| 191h  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| EEPEN | EEPEN |       |       |       |       |       |       |       |
| R/W   | W     |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

191h.7~0 **EEPEN**: EEPROM access enable

Write 0xE2 to this register will enable EEPROM access

Write others value to this register will disable EEPROM access

| 192h  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| EEPDT | EEPDT |       |       |       |       |       |       |       |
| R/W   | W     |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

192h.7~0 **EEPDT**: EEPROM data to write

Write data to this register will let H/W write the data to EEPROM when EEPROM access is enable.

**note: To clear watchdog first before writing EEPROM to avoid watchdog timer reset during writing procedure for safety.**

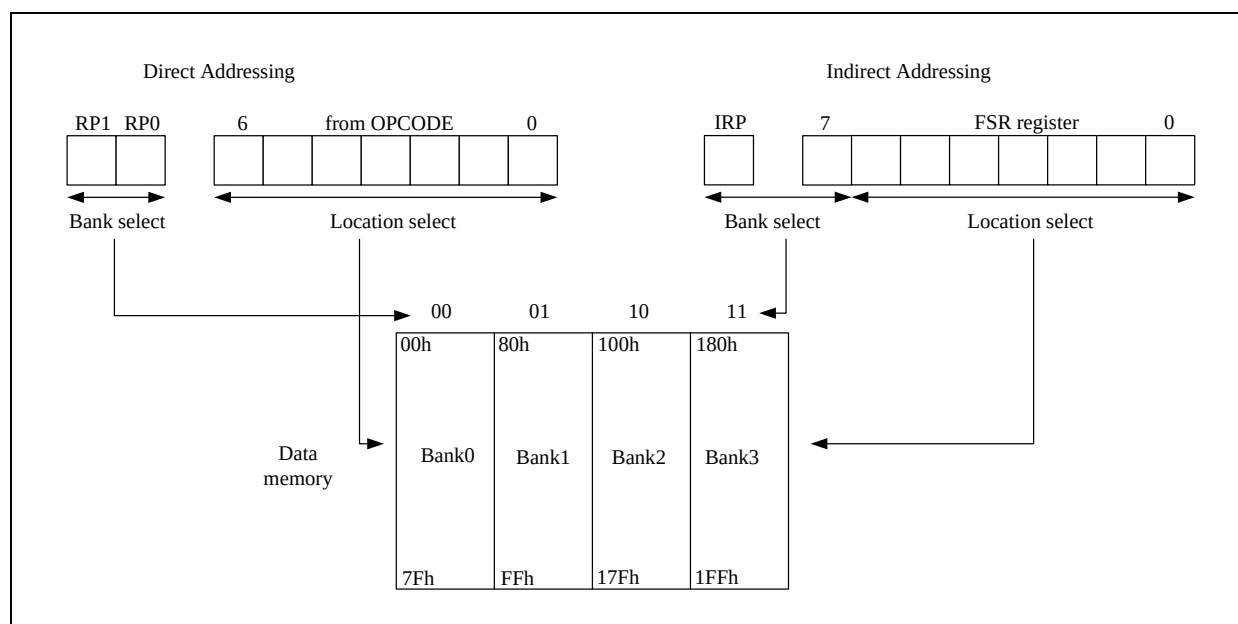
## 1.4 RAM Addressing Mode

There is one Data Memory Plane in CPU. The Plane is partitioned into four banks. Each bank extends up to 7Fh (128 bytes). The lower locations of each bank are reserved for Special Function Register (SFR). Above the SFR are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

Bit RP1 and RP0 (STATUS[6:5]) are the bank select bits.

| [RP1, RP0] | BANK |
|------------|------|
| 00         | 0    |
| 01         | 1    |
| 10         | 2    |
| 11         | 3    |

The plane can be addressed directly or indirectly. The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing. Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by File Select Register, FSR. Reading the INDF register itself, indirectly (FSR = '0') will read 00h. Writing to the INDF register indirectly (FSR = '0') results in a no operation (although status bit may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS[7]). Refer to the figure below.



Direct / Indirect Addressing

### Keeping RP0=RP1=0 in the beginning of the F/W code and using the new instruction set.

The advantage of using new instruction is user can ignore the bank location of registers and the code size can be saved. The new instruction is almost the same as the old instruction. By replacing the “F” to “X” in the instruction set can easily use the new instruction without switching the bank.

For example:

|                 |          |   |                 |          |
|-----------------|----------|---|-----------------|----------|
| BC <b>F</b>     | TM0IE    | → | BC <b>X</b>     | TM0IE    |
| DEC <b>F</b>    | CNT, 1   | → | DEC <b>X</b>    | CNT, 1   |
| INC <b>F</b> SZ | RAM25, 0 | → | INC <b>X</b> SZ | RAM25, 0 |
| MOVW <b>F</b>   | PAMOD10  | → | MOVW <b>X</b>   | PAMOD10  |
| RL <b>F</b>     | RAMA0, 0 | → | RL <b>X</b>     | RAMA0, 0 |
| SWAP <b>F</b>   | ADCTL, 0 | → | SWAP <b>X</b>   | ADCTL, 0 |

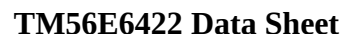
| 【BANK0】<br>000~07Fh |                              | 【BANK1】<br>080h~0FFh  |                              | 【BANK2】<br>100h~17Fh  |                              | 【BANK3】<br>180h~1FFh  |           |
|---------------------|------------------------------|-----------------------|------------------------------|-----------------------|------------------------------|-----------------------|-----------|
| 000h                | INDF                         | 080h                  | INDF                         | 100h                  | INDF                         | 180h                  | INDF      |
| 001h                | TM0                          | 081h                  | OPTION                       | 101h                  | TM0                          | 181h                  | OPTION    |
| 002h                | PCL                          | 082h                  | PCL                          | 102h                  | PCL                          | 182h                  | PCL       |
| 003h                | STATUS                       | 083h                  | STATUS                       | 103h                  | STATUS                       | 183h                  | STATUS    |
| 004h                | FSR                          | 084h                  | FSR                          | 104h                  | FSR                          | 184h                  | FSR       |
| 005h                | PAD                          | 085h                  | PAMOD10                      | 105h                  | PINMOD                       | 185h                  | DPL       |
| 006h                | PBD                          | 086h                  | PAMOD32                      | 106h                  |                              | 186h                  | DPH       |
| 007h                | PDD                          | 087h                  | PAMOD54                      | 107h                  | PWMCTL2                      | 187h                  | CRCDL     |
| 008h                |                              | 088h                  | PAMOD76                      | 108h                  |                              | 188h                  | CRCDH     |
| 009h                |                              | 089h                  | PWMCTL                       | 109h                  | LVRPD                        | 189h                  | CRCIN     |
| 00Ah                | PCLATH                       | 08Ah                  | PCLATH                       | 10Ah                  | PCLATH                       | 18Ah                  | PCLATH    |
| 00Bh                | INTIE                        | 08Bh                  | INTIE                        | 10Bh                  | INTIE                        | 18Bh                  | INTIE     |
| 00Ch                | INTIF                        | 08Ch                  | PBMOD10                      | 10Ch                  | PCH                          | 18Ch                  | TABR      |
| 00Dh                | INTIE1                       | 08Dh                  | PBMOD32                      | 10Dh                  |                              | 18Dh                  | CMPCTL    |
| 00Eh                | INTIF1                       | 08Eh                  | PBMOD54                      | 10Eh                  | BGTRIM                       | 18Eh                  | CMPPNS    |
| 00Fh                | CLKCTL                       | 08Fh                  | PBMOD76                      | 10Fh                  | IRCF                         | 18Fh                  | DACTL     |
| 010h                | TM0RLD                       | 090h                  | PDMOD10                      | 110h                  |                              | 190h                  | EEPCTL    |
| 011h                | TM0CTL                       | 091h                  | OPTION2                      | 111h                  | BG2TRIM                      | 191h                  | EEPEN     |
| 012h                | TM1                          | 092h                  | PWMPRDH                      | 112h                  |                              | 192h                  | EEPDT     |
| 013h                | TM1RLD                       | 093h                  | PWMPRDL                      | 113h                  | RDCTL                        | 193h                  | Don't Use |
| 014h                | TM1CTL                       | 094h                  | PWM0DH                       | 114h                  |                              | 194h                  |           |
| 015h                | T2CTL                        | 095h                  | PWM0DL                       | 115h                  | SIRCF                        | 195h                  |           |
| 016h                | LVCTL                        | 096h                  | PWM1DH                       | 116h                  |                              | 196h                  |           |
| 017h                | ADCDH                        | 097h                  | PWM1DL                       | 117h                  |                              | 197h                  |           |
| 018h                | ADCTL                        | 098h                  | PWM2DH                       | 118h                  |                              | 198h                  |           |
| 019h                | ADCTL2                       | 099h                  | PWM2DL                       | 119h                  |                              | 199h                  |           |
| 01Ah                |                              | 09Ah                  | PWM3DH                       | 11Ah                  |                              | 19Ah                  |           |
| 01Bh                |                              | 09Bh                  | PWM3DL                       | 11Bh                  |                              | 19Bh                  |           |
| 01Ch                |                              | 09Ch                  | PWM4DH                       | 11Ch                  |                              | 19Ch                  |           |
| 01Dh                |                              | 09Dh                  | PWM4DL                       | 11Dh                  |                              | 19Dh                  |           |
| 01Eh                |                              | 09Eh                  | PWM5DH                       | 11Eh                  |                              | 19Eh                  |           |
| 01Fh                |                              | 09Fh                  | PWM5DL                       | 11Fh                  |                              | 19Fh                  |           |
| 020h                | RAM Bank0 area<br>(80 Bytes) | 0A0h                  | RAM Bank1 area<br>(80 Bytes) | 120h                  | RAM Bank2 area<br>(80 Bytes) | 1A0h                  | Don't Use |
| 06Fh                |                              | 0EFh                  |                              | 16Fh                  |                              | 1EFh                  |           |
| 070h                |                              | 0F0h                  |                              | 170h                  |                              | 1F0h                  |           |
| 07Fh                |                              | 0FFh                  |                              | 17Fh                  |                              | 1FFh                  |           |
|                     |                              |                       |                              |                       |                              |                       |           |
|                     |                              |                       |                              |                       |                              |                       |           |
|                     |                              |                       |                              |                       |                              |                       |           |
|                     | common area<br>(16 Bytes)    | accesses<br>070h~07Fh |                              | accesses<br>070h~07Fh |                              | accesses<br>070h~07Fh |           |

◇ Example: read / write register by using direct addressing (**force RP0=RP1=0**)

|         |         |      |                                          |
|---------|---------|------|------------------------------------------|
| CLKCTL  | equ     | 00Fh | ; SFR in Bank0                           |
| TM1     | equ     | 012h | ; SFR in Bank0                           |
| OPTION2 | equ     | 091h | ; SFR in Bank1                           |
| LVRPD   | equ     | 109h | ; SFR in Bank2                           |
| IRCF    | equ     | 10Fh | ; SFR in Bank2                           |
| DPL     | equ     | 185h | ; SFR in Bank3                           |
| RAM020  | equ     | 020h | ; RAM in Bank0                           |
| RAM0A0  | equ     | 0A0h | ; RAM in Bank1                           |
|         |         |      |                                          |
| MOVXW   | TM1     |      | ; read TM1 (Bank0) to W                  |
| MOVXW   | OPTION2 |      | ; read OPTION2 (Bank1) to W              |
| MOVXW   | IRCF    |      | ; read IRCF (Bank2) to W                 |
| MOVXW   | DPL     |      | ; read DPL (Bank3) to W                  |
|         |         |      |                                          |
| MOVLW   | 16h     |      | ; W = 16h                                |
| MOVWX   | RAM020  |      | ; RAM[0x020] = W = 16h                   |
| MOVWX   | RAM0A0  |      | ; RAM[0x0A0] = W = 16h                   |
|         |         |      |                                          |
| MOVLW   | 37h     |      | ; W = 37h                                |
| MOVWX   | LVRPD   |      | ; LVRPD = W = 37h, force LVR/POR disable |
|         |         |      |                                          |
| MOVXW   | CLKCTL  |      | ; read SFR CLKCTL (00Fh) to W            |
| MOVXW   | IRCF    |      | ; read SFR IRCF (10Fh) to W              |
|         |         |      |                                          |
| MOVLW   | 0Bh     |      | ; W = 0Bh                                |
| MOVWX   | CLKCTL  |      | ; CLKCTL (00Fh) = W = 0Bh                |
| MOVWX   | IRCF    |      | ; IRCF (10Fh) = W = 0Bh                  |

◇ Example: read / write register by using indirect addressing (**force RP0=RP1=0**)

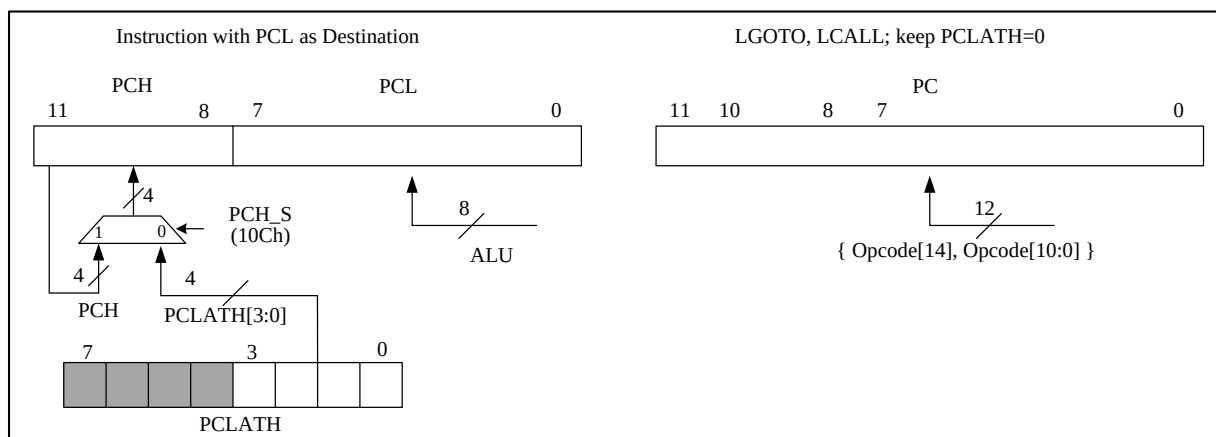
|       |      |  |                               |
|-------|------|--|-------------------------------|
| BSX   | IRP  |  | ; IRP = 1 => Bank2/3          |
| MOVLW | 0Fh  |  | ; W = 0Fh                     |
| MOVWX | FSR  |  | ; FSR = W = 0Fh               |
| MOVXW | INDF |  | ; read SFR IRCF (10Fh) to W   |
|       |      |  |                               |
| BSX   | IRP  |  | ; IRP = 1 => Bank2/3          |
| MOVLW | 0Fh  |  | ; W = 0Fh                     |
| MOVWX | FSR  |  | ; FSR = W = 0Fh               |
| MOVLW | 0Bh  |  | ; W = 0Bh                     |
| MOVWX | INDF |  | ; IRCF (10Fh) = W = 0Bh       |
|       |      |  |                               |
| BCX   | IRP  |  | ; IRP = 0 => Bank0/1          |
| MOVLW | 0Fh  |  | ; W = 0Fh                     |
| MOVWX | FSR  |  | ; FSR = W = 0Fh               |
| MOVXW | INDF |  | ; read SFR CLKCTL (00Fh) to W |
|       |      |  |                               |
| BCX   | IRP  |  | ; IRP = 0 => Bank0/1          |
| MOVLW | 0Fh  |  | ; W = 0Fh                     |
| MOVWX | FSR  |  | ; FSR = W = 0Fh               |
| MOVLW | 0Bh  |  | ; W = 0Bh                     |
| MOVWX | INDF |  | ; CLKCTL (00Fh) = W = 0Bh     |



The Programming Counter is 12-bit wide. As a program instruction is executed, the PC will contain the address of the next program instruction to be executed. The PC value is normally increased by one except for the following cases. The Reset Vector (000h) and the Interrupt Vector (004h) are provided for PC initialization and Interrupt. For CALL/GOTO instruction, PC loads lower 11 bits address from instruction word and upper 1 bit from PCLATH[3]. For RET/RETI/RETLW instruction, PC retrieves its content from the top level STACK.

The low byte data of the Programming Counter (PC[7:0]) can be read and written by PCL register (002h/082h/102h/182h). The high byte data of Programming Counter (PC[11:8]) can only be read by PCH register (10Ch). The internal flag PCH\_S is used to select the source of PCH, when executing any instruction with the PCL register as the destination. Write 0x1C to PCH register can set PCH\_S, write others value to PCH register will clear PCH\_S. After reset, the PCH\_S is cleared.

When PCH\_S is set to '1', executing any instruction with the PCL register as the destination the low byte is written to the PCL register and will not change the PCH. It is recommended to setting PCH\_S to '1' when using any instruction with the PCL register as the destination, but C language doesn't support this function.



|       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 002h  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| PCL   | PCL   |       |       |       |       |       |       |       |
| R/W   | R/W   |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

DS-TM56E6422 E

| 00Ah   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|--------|-------|-------|-------|
| PCLATH | GPR   |       |       |       | PCLATH |       |       |       |
| R/W    | R/W   |       |       |       | R/W    |       |       |       |
| Reset  | 0     | 0     | 0     | 0     | 0      | 0     | 0     | 0     |

00Ah.3~0 **PCLATH**: Programming Counter high byte data when instruction with PCL as destination is executed, and PCH\_S is cleared

00Ah.3 **PCLATH**: Programming Counter upper 1 bit when CALL/GOTO instruction is executed

Note: When using LCALL/LGOTO instruction must keep cleared

| 10Ch  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| PCH   | PCH   |       |       |       |       |       |       |       |
| R/W   | W     |       |       |       | R/W   |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

10Ch.7~0 **PCH (W)**: Programming Counter high byte source selection when instruction with PCL as destination is executed

write 0x1C to set PCH\_S = 1: PCH keep the original value

write others to clear PCH\_S = 0: PCH is from PCLATH

10Ch.3~0 **PCH (R)**: Programming Counter data bit 11~8

The STACK is 12-bit wide and 8-level in depth. The LCALL instruction and hardware interrupt will push STACK level in order, while the RET/RETI/RETLW instruction pops STACK level in order. For table lookup, the device offer the powerful table read instructions TABRL, TABRH to return the 16-bit ROM data into W register by setting DPTR={DPH, DPL} registers. It also offers another way to read the 16-bit ROM data into W register by setting TABR (18Ch) for C language.

◇ Example: To look up the PROM data located RETLW.

```

ORG      000h                                ; Reset Vector
LGOTO    START

START:
    MOVLW 00h
    MOVWX RAM020                            ; Set lookup table's address
    MOVLW 1Ch                               ; Write 1Ch to PCH to set PCH_S flag
    MOVWX PCH

LOOP:
    MOVXW  RAM020                            ; Move index value to W register
    LCALL  TABLE1                          ; To lookup data
    ...
    INCX   RAM020, 1                        ; Increment the index address for next address
    ...
    LGOTO  LOOP                             ; Go to LOOP label
    ...

ORG      X00h
TABLE1:
    ADDWX  PCL, 1                           ; Add the W with PCL, the result back in PCL.
    RETLW  55h                               ; W=55h when return
    RETLW  56h                               ; W=56h when return
    RETLW  58h                               ; W=58h when return

```

**Note:** The chip define 256 ROM address as one page, so that ROM has 16 pages, 000h~0FFh, 100h~1FFh, ..., F00h~FFFh. On the other words, PC[11:8] can be define as page. A lookup table must be

located at the same page to avoid getting wrong data. Thus, the lookup table has maximum 255 data for above example with starting a lookup table at X00h (X = 1, 2, 3, ..., E, F). If a lookup table has fewer data, it needs not setting the starting address at X00h, but only confirms all lookup table data are located at the same page.

◇ Example: To look up the PROM data located using TABRL / TABRH in assembly method.

```

        MOVLW    (TABLE2 >>8) & 0xff
        MOVWX    DPH
        MOVLW    (TABLE2) & 0xff
        MOVWX    DPL                ; DPTR = {DPH, DPL} = TABLE2
; Table Read by instructions TABRL / TABRH
        TABRL                    ; Read PROM low byte data to W (W = 86h)
        TABRH                    ; Read PROM high byte data to W (W = 19h)
        ...

TABLE2:
        .DT      0x1986                ; 16-bit ROM data
        ...

```

◇ Example: To look up the PROM data located using TABR register in C language method.

```

; Table Read by SFR TABR
const unsigned int TABLE[]={0x1234};
DPL = &TABLE;
DPH = &TABLE >> 8;
TABR=1;                                // write 01h to TABR = opcode TABRL
                                        // TABR= low byte data of TABLE[0] =34h
rData1=TABR;                          // rData1 = TABR= low byte data of TABLE[0] =34h
TABR=2;                                // write 02h to TABR = opcode TABRH
                                        // TABR= high byte data of TABLE[0] =12h
rData2=TABR;                          // rData2 = TABR= high byte data of TABLE[0] =12h

```

**Note:** When using TABR register in C language to look up the table, the lookup table instruction can only be used outside the interrupt or within the interrupt, if the lookup table instruction is used inside and outside the interrupt, it may cause an error.

| 18Ch  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| TABR  | TABR  |       |       |       |       |       |       |       |
| R/W   | R/W   |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

18Ch.7~0 The TABR register is used for lookup tables when using the C language as following step.

1. TABR write 01h to get PROM low byte data to W and TABR register (as instruction TABRL)
  2. TABR write 02h to get PROM high byte data to W and TABR register (as instruction TABRH)
  3. After step.1 or step.2, read TABR to get main ROM table read value
  4. After step.1 or step.2, read TABR to get main ROM table read value for C language
- Table Read for ASM: Support instruction TABRL / TABRH or register TABR. Suggest not using the method of register TABR. SFR HWAUTO=1 is also suggested.*
- Table Read for C: using register TABR. Only be used outside or inside the interrupt service routine. Don't utilize it inside and outside interrupt service routine simultaneously. Otherwise, something will be wrong.*

### 1.5.1 ALU and Working (W) Register

The ALU is 8-bit wide and capable of addition, subtraction, shift and logical operations. In two-operand instructions, typically one operand is the W register, which is an 8-bit non-addressable register used for ALU operations. The other operand is either a file register or an immediate constant. In single operand instructions, the operand is either W register or a file register. Depending on the instruction executed, the ALU may affect the values of Carry (C), Digit Carry (DC), and Zero (Z) Flags in the STATUS register. The C and DC flags operate as a /Borrow and /Digit Borrow, respectively, in subtraction.

Note: /Borrow represents inverted of Borrow register.

/Digit Borrow represents inverted of Digit Borrow register.

### 1.5.2 STATUS Register (003h/083h/103h/183h)

This register contains the arithmetic status of ALU and the Reset status. The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. It is recommended, therefore, that only BCX, BSX and MOVWX instructions are used to alter the STATUS Register because these instructions do not affect those bits.

| STATUS      | Bit 7                                                                                                                                                                                                                | Bit 6 | Bit 5 | Bit 4 | Bit 3                                                                     | Bit 2 | Bit 1 | Bit 0 |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|-------|---------------------------------------------------------------------------|-------|-------|-------|
| Reset Value | 0                                                                                                                                                                                                                    | 0     | 0     | 0     | 0                                                                         | 0     | 0     | 0     |
| R/W         | R/W                                                                                                                                                                                                                  | R/W   | R/W   | R     | R                                                                         | R/W   | R/W   | R/W   |
| Bit         | Description                                                                                                                                                                                                          |       |       |       |                                                                           |       |       |       |
| 7           | <b>IRP:</b> Register Bank Select bit (used for indirect addressing)<br>0 = Bank 0,1 (000h - 0FFh)<br>1 = Bank 2,3 (100h - 1FFh)                                                                                      |       |       |       |                                                                           |       |       |       |
| 6:5         | <b>RP1:RP0:</b> Register Bank Select bits (used for direct addressing)<br>00 = Bank 0 (000h - 07Fh)<br>01 = Bank 1 (080h - 0FFh)<br>10 = Bank 2 (100h - 17Fh)<br>11 = Bank 3 (180h - 1FFh)<br>Each bank is 128 bytes |       |       |       |                                                                           |       |       |       |
| 4           | <b>TO:</b> Time Out Flag<br>0: after Power On Reset or CLRWDWT/SLEEP instruction<br>1: WDT time out occurs                                                                                                           |       |       |       |                                                                           |       |       |       |
| 3           | <b>PD:</b> Power Down Flag<br>0: after Power On Reset or CLRWDWT instruction<br>1: after SLEEP instruction                                                                                                           |       |       |       |                                                                           |       |       |       |
| 2           | <b>Z:</b> Zero Flag<br>0: the result of a logic operation is not zero<br>1: the result of a logic operation is zero                                                                                                  |       |       |       |                                                                           |       |       |       |
| 1           | <b>DC:</b> Decimal Carry Flag or Decimal / Borrow Flag                                                                                                                                                               |       |       |       |                                                                           |       |       |       |
|             | ADD instruction                                                                                                                                                                                                      |       |       |       | SUB instruction                                                           |       |       |       |
|             | 0: no carry<br>1: a carry from the low nibble bits of the result occurs                                                                                                                                              |       |       |       | 0: a borrow from the low nibble bits of the result occurs<br>1: no borrow |       |       |       |
| 0           | <b>C:</b> Carry Flag or /Borrow Flag                                                                                                                                                                                 |       |       |       |                                                                           |       |       |       |
|             | ADD instruction                                                                                                                                                                                                      |       |       |       | SUB instruction                                                           |       |       |       |
|             | 0: no carry<br>1: a carry occurs from the MSB                                                                                                                                                                        |       |       |       | 0: a borrow occurs from the MSB<br>1: no borrow                           |       |       |       |



◇ Example: Write immediate data into STATUS register.

```
MOVLW    00h
MOVWX    STATUS           ; Clear STATUS register
```

◇ Example: Bit addressing set and clear STATUS register.

```
BSX      STATUS, 0        ; Set C=1
BCX      STATUS, 0        ; Clear C=0
```

◇ Example: Determine the C flag by BTXSS instruction.

```
BTXSS    STATUS, 0        ; Check the carry flag
LGOTO    LABEL_1         ; If C=0, goto LABEL_1
LGOTO    LABEL_2         ; If C=1, goto LABEL_2
```

## 2 Reset

This device can be RESET in four ways.

- Power-On-Reset (POR)
- Low Voltage Reset (LVR)
- External Pin Reset (XRST)
- Watchdog Timer Reset (WDTR)

Resets can be caused by Power on Reset (POR), External Pin Reset (XRST), Watchdog Timer Reset (WDTR), or Low Voltage Reset (LVR). The CFGWH controls the Reset functionality. After Reset, the SFRs are returned to their default value, the program counter (PC) is cleared, and the system starts running from the reset vector 000h place. The TO and PD flags at status register (STATUS) are indicate system reset status.

### 2.1 Power on Reset (POR)

After Power-On-Reset, all system and peripheral control registers are then set to their default hardware Reset values.

### 2.2 Low Voltage Reset (LVR)

The Low Voltage Reset features static reset when supply voltage is below a threshold level. There are 15 threshold levels can be selected. The LVR's operation mode is defined by the CFGWH register. Please refer to the table of LVR voltage in the chapter of Electrical Characteristics. Vcc must be below maximum operating voltage and above LVR voltage. The user must consider the minimal operating voltage of corresponding operating frequency such that LVR voltage must be above minimum operating voltage.

Different  $F_{sys}$  have different system minimum operating voltage, reference to Operating Voltage of DC characteristics, if current system voltage is low than minimum operating voltage and lower LVR is selected, then the system maybe enters dead-band and error occurs.

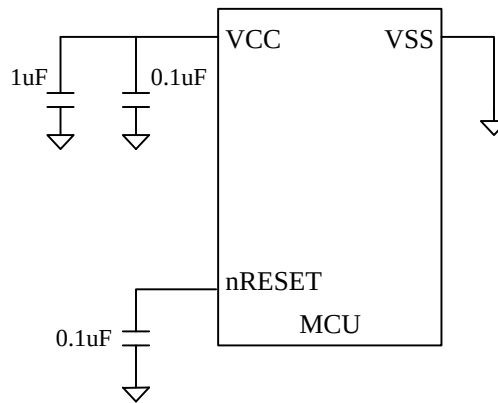
| 16h   | Bit 7 | Bit 6  | Bit 5  | Bit 4  | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|--------|--------|--------|-------|-------|-------|-------|
| LVCTL | LVDF  | LVDHYS | LVRSAV | LVDSAV | LVDS  |       |       |       |
| R/W   | R     | R/W    | R/W    | R/W    | R/W   |       |       |       |
| Reset | 0     | 0      | 1      | 1      | 0     | 0     | 0     | 0     |

- 16h.5     **LVRSAV**: POR/LVR auto power off in STOP/IDLE mode  
0: disable POR/LVR auto power off in STOP/IDLE mode  
1: enable POR/LVR auto power off in STOP/IDLE mode

### 2.3 External Pin Reset (XRST)

The External Pin Reset (XRST) can be disabled or enabled by XRSTE at CFGWH register. External pin reset should be kept low for at least 2 SIRC clock cycles to ensure reset can active. The External Pin Reset also sets all the control registers to their default value but the TO/PD flags will not affected by these resets.

External reset pin (nRESET) is 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 can reset the system during power on duration, and good external reset circuit can protect the system to avoid operating at inappropriate power condition.



## 2.4 Watchdog Timer Reset (WDTR)

The WDT reset can be disabled or enabled through the CFGWH register. Set WDTPSC to define the period during which WDT reset occurs. WDT reset counter can be cleared by device Reset or CLRWDT bit. WDT reset also set all the control registers to their default value. The TO/PD flags are not affected by WDT resets.

◇ Example: Defining Reset Vector

```

ORG      000h                ; Reset Vector
LGOTO    START                ; Jump to user program address.

ORG      010h
START:
...      ; 010h, The head of user program
...
LGOTO    START

```

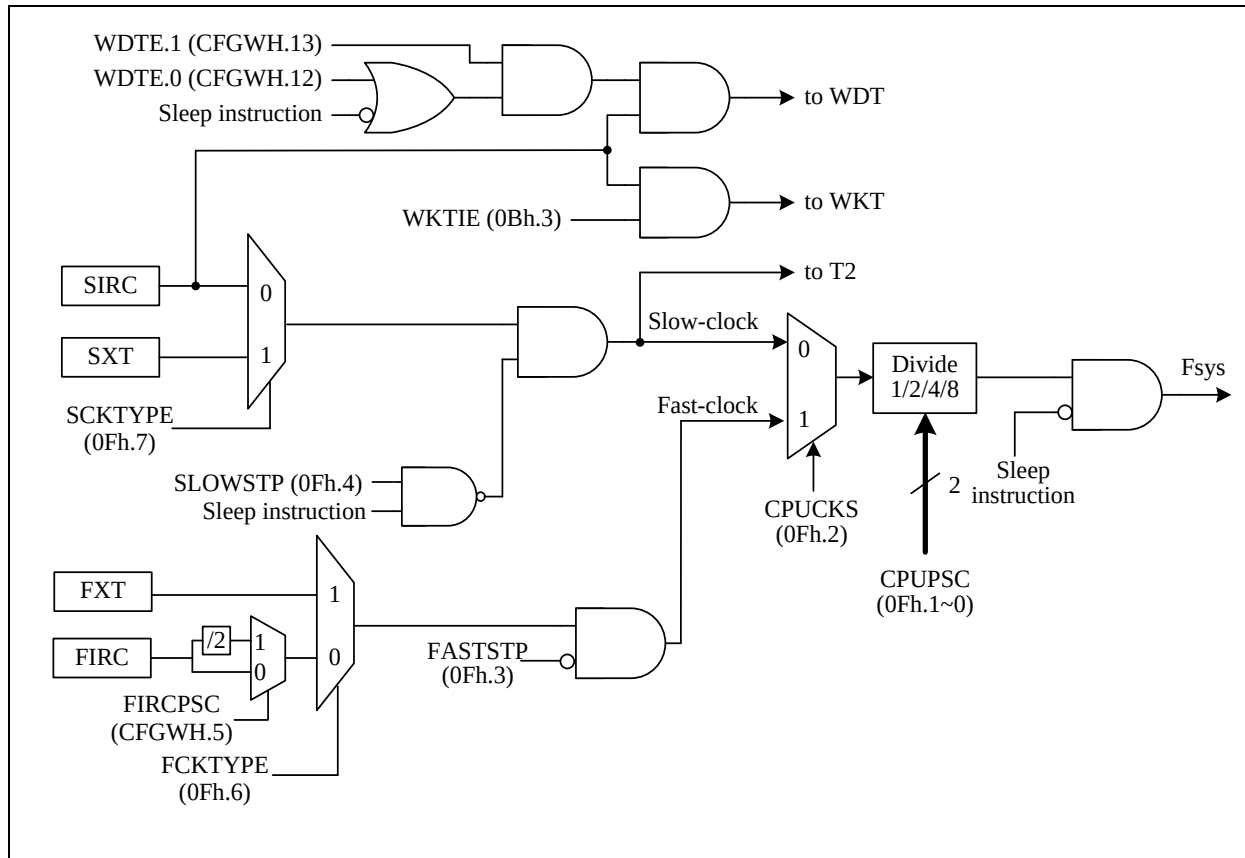
## 3 Clock Circuitry and Operation Mode

### 3.1 System Clock

The device is designed with dual-clock system. There are four kinds of clock source, FXT (Fast Crystal) Clock, SXT (Slow Crystal) Clock, SIRC (Slow Internal RC) Clock and FIRC (Fast Internal RC) Clock. Each clock source can be applied to CPU kernel as system clock. When in IDLE mode, the Slow-clock (SIRC or SXT) can be configured to keep oscillating to provide clock source to T2 block, or the SIRC provides clock source to WKT/WDT block. Refer to the Figure as below.

After Reset, the device is running at SLOW mode with SIRC. S/W should select the proper clock rate for chip operation safety. The higher  $V_{CC}$  allows the chip to run at a higher System clock frequency. In a typical condition, a 16 MHz System clock rate requires  $V_{CC} > 2V @ (25^{\circ}C)$ .

The CLKCTL (0Fh) SFR controls the System clock operating. H/W automatically blocks the S/W abnormally setting for this register. Never to write both FASTSTP=1 and CPUCKS=1. It is recommended to write this SFR bit by bit.



**Clock Scheme Block Diagram**

The frequency of FIRC can be adjusted by IRCF (10Fh). When IRCF=00h, frequency is the lowest. When IRCF=7Fh, frequency is the highest. With this function, we can adjust the frequency of FIRC after power on. Each IC may have different default value of IRCF, to make sure the frequency of FIRC=16 MHz after Power on Reset.

**FAST Mode:**

In this mode, the program is executed using FIRC or FXT as CPU clock (Fsys). The Timer0, Timer1 blocks are also driven by Fast-clock. The PWM0 block can be driven by Fsys, FIRC/256, FIRC (16 MHz), or FIRC\*2 (32 MHz) by setting PWMCKS (91h.5~4). T2 can be driven by Slow-clock, Fsys/128, or FIRC/512 (16 MHz/512) by setting T2CKS (15h.3~2).

**SLOW Mode:**

After power-on or reset, device enters SLOW mode, the default Slow-clock is SIRC. In this mode, the Fast-clock can stopped (by FASTSTP=1, for power saving) or running (by FASTSTP=0), and Slow-clock is enabled. All peripheral blocks (Timer0, Timer1, etc...) clock source are Slow-clock in the SLOW mode, except PWM and T2 blocks, which can select other clock source. There are two kinds of SLOW clock can be selected, SIRC and SXT.

**IDLE Mode:**

After executing the SLEEP instruction, if SIRC or SXT is still oscillating, it means entering IDLE mode. IDLE mode is terminated by Reset or enabled Interrupts wake-up. There are two ways to keep SIRC or SXT oscillating in IDLE mode.

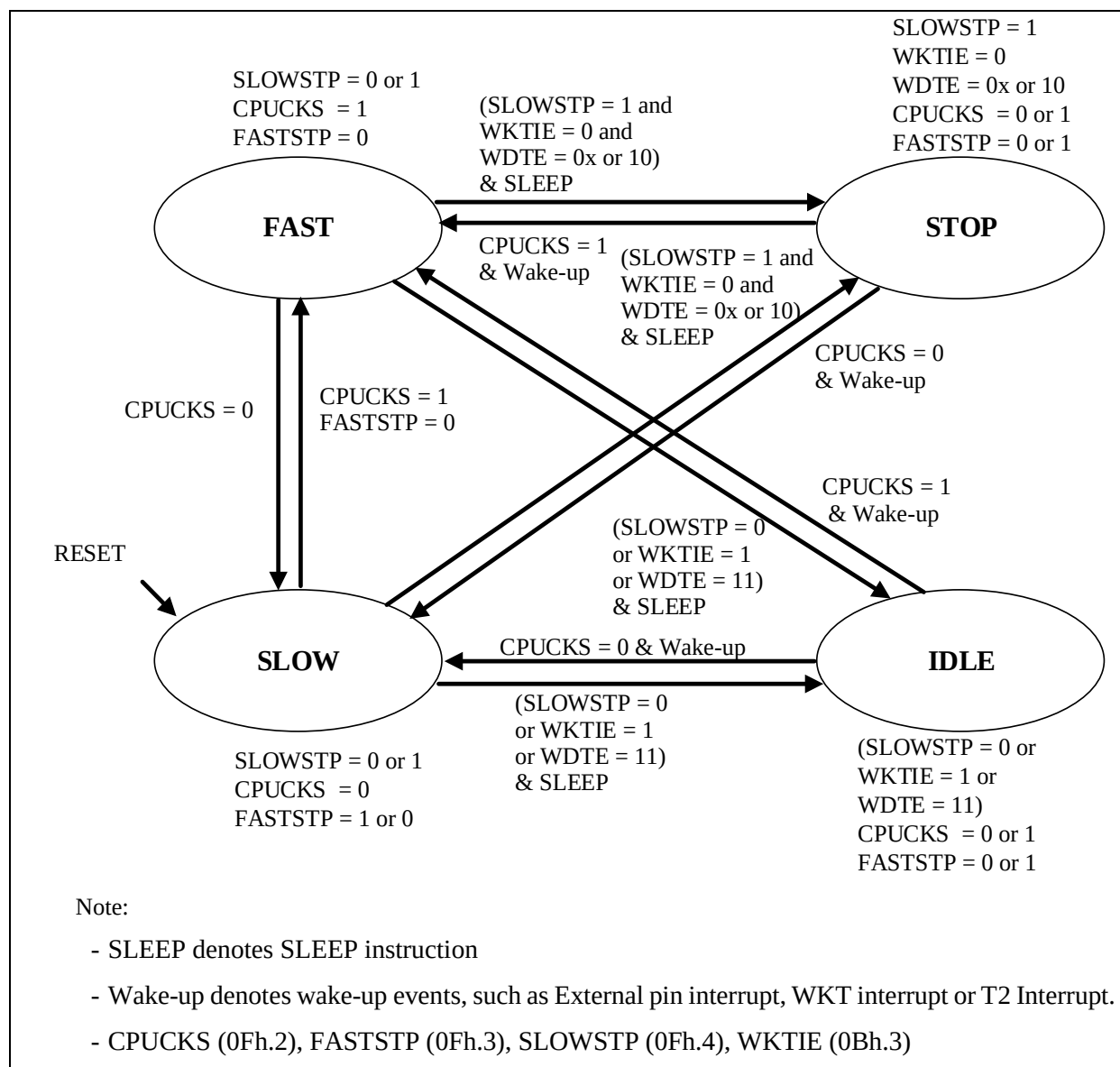
- (1) Set SLOWSTP=0, before executing the SLEEP instruction, the SIRC or SXT can still oscillate. In this situation, Slow-clock can continue to oscillate to provide T2 block running in IDLE mode.
- (2) Set WKTIE=1 or WDTE=11, before executing the SLEEP instruction, the SIRC can still oscillate to keep WKT/WDT operating in IDLE mode.

**STOP Mode:**

When SLOWSTP (0Fh.4) is set, WKTIE (0Bh.3) is cleared and WDTE=0x or 10, all blocks will be turned off and the Chip will enter the “STOP Mode” after executing the SLEEP instruction. STOP mode is similar to IDLE mode. The difference is all clock oscillators either Fast-clock or Slow-clock are stopped and no clocks are generated.

### 3.2 Dual System Clock Modes Transition

The device is operated in one of four modes: FAST mode, SLOW mode, IDLE mode, and STOP mode.



**CPU Operation Block Diagram**

**CPU Mode & Clock Functions Table:**

| Mode | Fsys       | Fast-clock     | Slow-clock | TM0/TM1 | T2           | WKT          | WDT         | Wake-up event |
|------|------------|----------------|------------|---------|--------------|--------------|-------------|---------------|
| FAST | Fast-clock | Run            | Run        | Run     | Run          | Run          | Run         | X             |
| SLOW | Slow-clock | Set by FASTSTP | Run        | Run     | Run          | Run          | Run         | X             |
| IDLE | Stop       | Stop           | Run        | Stop    | Set by T2CKS | Set by WKTIE | Set by WDTE | WKT/IO/T2     |
| STOP | Stop       | Stop           | Stop       | Stop    | Stop         | Stop         | Stop        | IO            |

### ● FAST mode switches to SLOW mode

The following steps are suggested to be executed by order when FAST mode switches to SLOW mode:

- (1) Switch to Slow-clock (CPUCKS=0)
- (2) Stop Fast-clock (FASTSTP=1)

◇ Example: Switch FAST mode to SLOW mode.

|     |         |                      |
|-----|---------|----------------------|
| BCX | CPUCKS  | ; Fsys=Slow-clock    |
| BSX | FASTSTP | ; Disable Fast-clock |

### ● SLOW mode switches to FAST mode

SLOW mode can be enabled by CPUCKS=0 in CLKCTL register. The following steps are suggested to be executed by order when SLOW mode switches to FAST mode:

- (1) Enable Fast-clock (FASTSTP=0)
- (2) Switch to Fast-clock (CPUCKS=1)

◇ Example: Switch SLOW mode to FAST mode (The Fast-clock stop).

|     |         |                     |
|-----|---------|---------------------|
| BCX | FASTSTP | ; Enable Fast-clock |
| NOP |         |                     |
| BSX | CPUCKS  | ; Fsys=Fast-clock   |

### ● IDLE mode Setting

The IDLE mode can be configured by following setting in order:

- (1) Enable Slow-clock (SLOWSTP=0) or WKT (WKTIE=1) or WDT (WDTE=11b)
- (2) Switch T2 clock source to Slow-clock (T2CKS=0)
- (3) Execute SLEEP instruction

IDLE mode can be waked up by External interrupt, WKT interrupt and T2 interrupt.

◇ Example: Switch FAST/SLOW mode to IDLE mode.

|       |                    |                                                     |
|-------|--------------------|-----------------------------------------------------|
| BCX   | SLOWSTP            | ; Enable Slow-clock after execute SLEEP instruction |
| MOVLW | 0000 <u>00</u> 00b |                                                     |
| MOVWX | T2CTL              |                                                     |
| SLEEP |                    | ; Enter IDLE mode                                   |

### ● STOP Mode Setting

The STOP mode can be configured by following setting in order:

- (1) Stop Slow-clock (SLOWSTP=1)
- (2) Stop WKT (WKTIE=0)
- (3) Execute SLEEP instruction

STOP mode can be woken up only by external pin interrupt and pin-change.

Note: CPU will not enter STOP mode if WDTE=11b

◇ Example: Switch FAST/SLOW mode to STOP mode.

```

BSX          SLOWSTP          ; Disable Slow-clock after execute SLEEP instruction
MOVLW        00000000b        ; Disable WKT counting
MOVWX        INTIE
SLEEP        ; Enter STOP mode.
  
```

| 0Bh   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIE | ADCIE | T2IE  | TM1IE | TM0IE | WKTIE | INT2IE | INT1IE | INT0IE |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

0Bh.3 **WKTIE**: Wake-up Timer interrupt enable and Wake-up Timer enable  
 0: disable  
 1: enable

| 0Fh    | Bit 7   | Bit 6   | Bit 5 | Bit 4   | Bit 3   | Bit 2  | Bit 1  | Bit 0 |
|--------|---------|---------|-------|---------|---------|--------|--------|-------|
| CLKCTL | SCKTYPE | FCKTYPE | —     | SLOWSTP | FASTSTP | CPUCKS | CPUPSC |       |
| R/W    | R/W     | R/W     | —     | R/W     | R/W     | R/W    | R/W    |       |
| Reset  | 0       | 0       | —     | 0       | 1       | 0      | 1      | 1     |

0Fh.7 **SCKTYPE**: Slow-clock select. This bit could only be changed in Fast mode(CPUCKS=1)  
 0: Slow-clock is SIRC  
 1: Slow-clock is SXT. PA4 and PA5 are crystal pins.

0Fh.6 **FCKTYPE**: Fast-clock select. This bit could only be changed in Slow mode(CPUCKS=0)  
 0: Fast-clock is FIRC  
 1: Fast-clock is FXT. PA4 and PA5 are crystal pins. FXT oscillator gain is higher than that of SXT.

0Fh.4 **SLOWSTP**: Stop Slow-clock after execute SLEEP instruction  
 0: Slow-clock keeps running after execute SLEEP instruction  
 1: Slow-clock stops running after execute SLEEP instruction

0Fh.3 **FASTSTP**: Fast-clock stop  
 0: Fast-clock is running  
 1: Fast-clock stops running

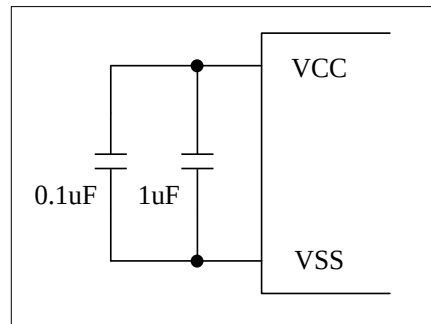
0Fh.2 **CPUCKS**: System clock source select  
 0: Slow-clock  
 1: Fast-clock

0Fh.1~0 **CPUPSC**: System clock source prescaler. System clock source  
 00: divided by 8  
 01: divided by 4  
 10: divided by 2  
 11: divided by 1

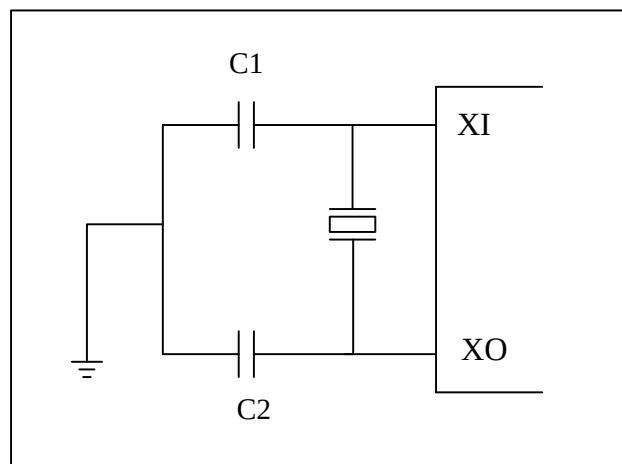


### 3.3 System Clock Oscillator

In the Fast Internal RC (FIRC) mode, the on-chip oscillator generates 16 MHz system clock. In Slow/Fast Crystal (SXT/FXT) mode, a crystal or ceramic resonator is connected to XI and XO pins to establish oscillation. Since power noise degrades the performance of Internal Clock Oscillator, placing power supply bypass capacitors 1  $\mu$ F and 0.1  $\mu$ F very close to VCC/VSS pins improves the stability of clock and the overall system.



Internal RC Mode



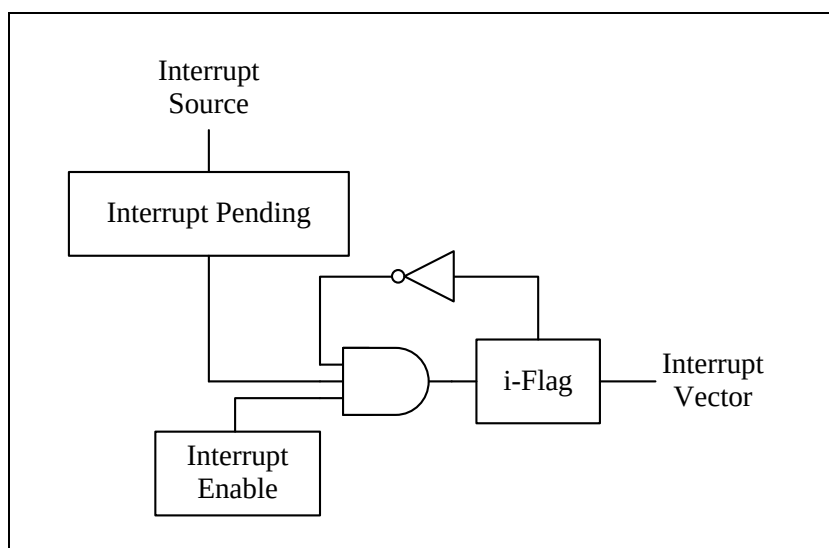
External Crystal/Ceramic  
Oscillator

## 4 Interrupt

The Chip has 1 level, 1 vector and 11 interrupt sources. Each interrupt source has its own enable control bit. An interrupt event will set its individual pending flag, no matter its enable control bit is 0 or 1.

If the corresponding interrupt enable bit (INTIE[7:0], INTIE1[6:4], INTIE1[1:0]) has been set, it would trigger CPU to service the interrupt. CPU accepts interrupt at the end of current executed instruction cycle. In the meanwhile, a “LCALL 004” instruction is inserted to CPU, and i-flag is set to prevent recursive interrupt nesting.

The i-flag is cleared in the instruction after the “RETI” instruction. That is, at least one instruction in main program is executed before service the pending interrupt. The interrupt event is level triggered. F/W must clear the interrupt event register while serving the interrupt routine.



◇ Example: Setup INT1 (PA1) interrupt request with rising edge trigger

```

ORG      000h                ; Reset Vector
LGOTO    START              ; Goto user program address

ORG      004h                ; All interrupt vector
LGOTO    INT                ; If INT1 (PA1) input occurred rising edge

ORG      005h
START:
MOVLW    0000xxxxb
MOVWX    PAMOD10            ; Select INT1 Pin Mode as mode 0000b
                                ; Open drain output low or input with Pull-up

MOVLW    xxxxxx1xb
MOVWX    PAD                ; Release INT1, it becomes Schmitt-trigger
                                ; input with input pull-up resistor

MOVLW    xx1xxxxxb
MOVWX    OPTION            ; Set INT1 interrupt trigger as rising edge
MOVLW    11111101b
MOVWX    INTIF             ; Clear INT1 interrupt request flag
MOVLW    00000010b
MOVWX    INTIE             ; Enable INT1 interrupt

MAIN:
...
LGOTO    MAIN

INT:
MOVWX    20h                ; Store W data to SRAM 20h
MOVXW    STATUS            ; Get STATUS data
MOVWX    21h                ; Store STATUS data to SRAM 21h

BTXSC    INT1IF            ; Check INT1IF bit
LCALL    INT1_SUB          ; INT1IF = 1, jump to INT1 interrupt service routine
...

EXIT_INT:
MOVXW    21h                ; Get SRAM 21h data
MOVWX    STATUS            ; Restore STATUS data
MOVXW    20h                ; Restore W data
RETI                     ; Return from interrupt

INT1_SUB:                    ; INT1 interrupt service routine
...
MOVLW    11111101b
MOVWX    INTIF             ; Clear INT1 interrupt request flag
RET

```

| 0Bh   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIE | ADCIE | T2IE  | TM1IE | TM0IE | WKTIE | INT2IE | INT1IE | INT0IE |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

- 0Bh.7 **ADCIE:** ADC interrupt enable  
0: disable  
1: enable
- 0Bh.6 **T2IE:** T2 interrupt enable  
0: disable  
1: enable
- 0Bh.5 **TM1IE:** Timer1 interrupt enable  
0: disable  
1: enable
- 0Bh.4 **TM0IE:** Timer0 interrupt enable  
0: disable  
1: enable
- 0Bh.3 **WKTIE:** Wake-up Timer interrupt enable and Wake-up Timer enable  
0: disable  
1: enable
- 0Bh.2 **INT2IE:** INT2 interrupt enable  
0: disable  
1: enable
- 0Bh.1 **INT1IE:** INT1 interrupt enable  
0: disable  
1: enable
- 0Bh.0 **INT0IE:** INT0 interrupt enable  
0: disable  
1: enable

| 0Ch   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIF | ADCIF | T2IF  | TM1IF | TM0IF | WKTIF | INT2IF | INT1IF | INT0IF |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

- 0Ch.7 **ADCIF:** ADC interrupt event pending flag  
This bit is set by H/W after ADC end of conversion, write 7Fh to INTIF will clear this flag
- 0Ch.6 **T2IF:** T2 interrupt event pending flag  
This bit is set by H/W while T2 overflows, write BFh to INTIF will clear this flag
- 0Ch.5 **TM1IF:** Timer1 interrupt event pending flag  
This bit is set by H/W while Timer1 overflows, write DFh to INTIF will clear this flag
- 0Ch.4 **TM0IF:** Timer0 interrupt event pending flag  
This bit is set by H/W while Timer0 overflows, write EFh to INTIF will clear this flag
- 0Ch.3 **WKTIF:** Wake-up Timer interrupt event pending flag  
This bit is set by H/W while Wake-up Timer is timeout, write F7h to INTIF will clear this flag
- 0Ch.2 **INT2IF:** INT2 pin falling interrupt pending flag  
This bit is set by H/W at INT2 pin's falling edge, write FBh to INTIF will clear this flag
- 0Ch.1 **INT1IF:** INT1 pin falling/rising interrupt pending flag  
This bit is set by H/W at INT1 pin's falling/rising edge, write FDh to INTIF will clear this flag
- 0Ch.0 **INT0IF:** INT0 pin falling/rising interrupt pending flag  
This bit is set by H/W at INT0 pin's falling/rising edge, write FEh to INTIF will clear this flag

| 0Dh    | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| INTIE1 | —     | PCIE  | EEPIE | CMPIE | —     | —     | PWMIE | LVDIE |
| R/W    | —     | R/W   | R/W   | R/W   | —     | —     | R/W   | R/W   |
| Reset  | —     | 0     | 0     | 0     | —     | —     | 0     | 0     |

0Dh.6 **PCIE:** All port pin-change wake-up interrupt enable

0: disable

1: enable

0Dh.5 **EEPIE:** EEPROM interrupt enable

0: disable

1: enable

0Dh.4 **CMPIE:** Comparator interrupt enable

0: disable

1: enable

0Dh.1 **PWMIE:** PWM interrupt enable

0: disable

1: enable

0Dh.0 **LVDIE:** LVD interrupt enable

0: disable

1: enable

| 0Eh    | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| INTIF1 | —     | PCIF  | EEPIF | CMPIF | —     | —     | PWMIF | LVDIF |
| R/W    | —     | R/W   | R/W   | R/W   | —     | —     | R/W   | R/W   |
| Reset  | —     | 0     | 0     | 0     | —     | —     | 0     | 0     |

0Eh.6 **PCIF:** All port pin-change wake-up interrupt event pending flag

This bit is set by H/W at all pin's falling/rising edge, write BFh to INTIF1 will clear this flag. A sleep instruction is necessary before the event of pin-change otherwise pin-change event may be missed.

0Eh.5 **EEPIF:** EEPROM interrupt event pending flag

This bit is set by H/W when EEPROM writing is finishing or time-out event occurred. Writing DFh to INTIF1 will clear this flag.

0Eh.4 **CMPIF:** Comparator interrupt event pending flag

This bit is set by H/W while CMPO match trigger condition, write EFh to INTIF1 will clear this flag

0Eh.1 **PWMIF:** PWM interrupt event pending flag

This bit is set by H/W after PWM period counter roll over, write FDh to INTIF1 will clear this flag

0Eh.0 **LVDIF:** LVD interrupt event pending flag

This bit is set by H/W after  $V_{CC} < V_{LVD}$ , write FEh to INTIF1 will clear this flag

| 81h    | Bit 7  | Bit 6   | Bit 5   | Bit 4 | Bit 3  | Bit 2 | Bit 1  | Bit 0 |
|--------|--------|---------|---------|-------|--------|-------|--------|-------|
| OPTION | HWAUTO | INT0EDG | INT1EDG | —     | WDTPSC |       | WKTPSC |       |
| R/W    | R/W    | R/W     | R/W     | —     | R/W    |       | R/W    |       |
| Reset  | 0      | 0       | 0       | —     | 1      | 1     | 1      | 1     |

81h.6 **INT0EDG:** INT0 pin interrupt edge selection

0: falling edge to trigger

1: rising edge to trigger

81h.5 **INT1EDG:** INT0 pin interrupt edge selection

0: falling edge to trigger

1: rising edge to trigger

| 91h     | Bit 7 | Bit 6 | Bit 5  | Bit 4 | Bit 3 | Bit 2   | Bit 1   | Bit 0   |
|---------|-------|-------|--------|-------|-------|---------|---------|---------|
| OPTION2 | —     | —     | PWMCKS |       | —     | INT2SEL | INT1SEL | INT0SEL |
| R/W     | —     | —     | R/W    |       | —     | R/W     | R/W     | R/W     |
| Reset   | —     | —     | 0      | 0     | —     | 0       | 0       | 0       |

91h.2 **INT2SEL:** INT2 pin select

0: PA7

1: PB5

91h.1 **INT1SEL:** INT1 pin select

0: PA1

1: PB1

91h.0 **INT0SEL:** INT0 pin select

0: PA3

1: PB2

| 16h   | Bit 7 | Bit 6  | Bit 5  | Bit 4  | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|--------|--------|--------|-------|-------|-------|-------|
| LVCTL | LVDF  | LVDHYS | LVRSAV | LVDSAV | LVDS  |       |       |       |
| R/W   | R     | R/W    | R/W    | R/W    | R/W   |       |       |       |
| Reset | 0     | 0      | 1      | 1      | 0     | 0     | 0     | 0     |

16h.7 **LVDF:** Low voltage detection flag

0:  $V_{CC} > V_{LVD}$

1:  $V_{CC} < V_{LVD}$

16h.6 **LVDHYS:** LVD Hysteresis

0: disable

1: enable

16h.4 **LVDSAV:** LVD auto power off in STOP/IDLE mode

0: disable LVD auto power off in STOP/IDLE mode

1: enable LVD auto power off in STOP/IDLE mode

16h.3~0 **LVDS:** LVD voltage ( $V_{LVD}$ ) select

Please refer to the table of LVD voltage in the section of “[LVD Circuit Characteristics](#)”. (Left click the link to go to that page)

## 5 I/O Port

### 5.1 PA0-PA7, PB0-PB7, PD0-PD1

Each IO has 4 bits as the mode setting. The mode setting can include the following functions: open drain output, CMOS output, pull-up resistor, pull-down resistor, pin-change wake-up, PWMO, TCOUT, TM1OUT and so on. All IO support two sink current options, which are defined by the HSINK (105h.2).

**PA7 has no high-sink, 1/2 bias and resistor pull-down capability.**

These pins can be operated in different modes as below table.

| PAxMOD<br>PBxMOD<br>PDxMOD | PADx<br>PBDx<br>PDDx | PA0~PA7, PB0~PB7, PD0~PD1<br>pin function           | Pin State  | Resistor<br>Pull-up | Digital<br>Input | Pin-<br>changed<br>Wake-up |
|----------------------------|----------------------|-----------------------------------------------------|------------|---------------------|------------------|----------------------------|
| <b>0000b</b>               | 0                    | Open Drain                                          | Drive Low  | -                   | -                | -                          |
|                            | 1                    | Input                                               | Pull-up    | Y                   | Y                | -                          |
| <b>0001b</b>               | 0                    | Open Drain                                          | Drive Low  | -                   | -                | -                          |
|                            | 1                    | Input                                               | Hi-Z       | -                   | Y                | -                          |
| <b>0010b</b>               | 0                    | CMOS Output (except PWMx)                           | Drive Low  | -                   | -                | -                          |
|                            | 1                    |                                                     | Drive High | -                   | -                | -                          |
| <b>0011b</b>               | X                    | Analog input/output for<br>ADCx / CINx / CIPx / XT* | Hi-Z       | -                   | -                | -                          |

\*: XT mean crystal oscillator

I/O Pin Function Table 1

| PAxMOD<br>PBxMOD<br>PDxMOD | PADx<br>PBDx<br>PDDx | PA0~PA7*, PB0~PB7,<br>PD0~PD1<br>pin function     | Pin State  | Resistor<br>Pull-down | Digital<br>Input | Pin-<br>changed<br>Wake-up |
|----------------------------|----------------------|---------------------------------------------------|------------|-----------------------|------------------|----------------------------|
| <b>0100b</b>               | 0                    | Open Drain                                        | Drive Low  | -                     | -                | -                          |
|                            | 1                    | Input                                             | Pull-down* | Y*                    | Y                | -                          |
| <b>0101b</b>               | 0                    | Open Drain                                        | Drive Low  | -                     | -                | -                          |
|                            | 1                    | Input                                             | Hi-Z       | -                     | Y                | -                          |
| <b>0110b</b>               | 0                    | CMOS Output (except PWMx)                         | Drive Low  | -                     | -                | -                          |
|                            | 1                    |                                                   | Drive High | -                     | -                | -                          |
| <b>0111b</b>               | X                    | Function CMOS output for<br>PWMx / TCOUT / TM1OUT | -          | -                     | -                | -                          |

\*: PA7 has no high-sink, 1/2 bias and resistor pull-down capability.

I/O Pin Function Table 2

| PAxMOD<br>PBxMOD<br>PDxMOD | PADx<br>PBDx<br>PDDx | PA0~PA7, PB0~PB7, PD0~PD1<br>pin function | Pin State  | Resistor<br>Pull-up | Digital<br>Input | Pin-<br>changed<br>Wake-up |
|----------------------------|----------------------|-------------------------------------------|------------|---------------------|------------------|----------------------------|
| <b>1000b</b>               | 0                    | Open Drain                                | Drive Low  | -                   | -                | -                          |
|                            | 1                    | Input                                     | Pull-up    | Y                   | Y                | Y                          |
| <b>1001b</b>               | 0                    | Open Drain                                | Drive Low  | -                   | -                | -                          |
|                            | 1                    | Input                                     | Hi-Z       | -                   | Y                | Y                          |
| <b>1010b</b>               | 0                    | CMOS Output (except PWMx)                 | Drive Low  | -                   | -                | -                          |
|                            | 1                    |                                           | Drive High | -                   | -                | -                          |
| <b>1011b</b>               |                      | Reserved                                  |            |                     |                  |                            |

I/O Pin Function Table 3

| PAxMOD<br>PBxMOD<br>PDxMOD | PADx<br>PBDx<br>PDDx | PA0~PA7*, PB0~PB7,<br>PD0~PD1<br>pin function                                   | Pin State           | Resistor<br>Pull-down | Digital<br>Input | Pin-<br>changed<br>Wake-up |
|----------------------------|----------------------|---------------------------------------------------------------------------------|---------------------|-----------------------|------------------|----------------------------|
| <b>1100b</b>               | 0                    | Open Drain                                                                      | Drive Low           | -                     | -                | -                          |
|                            | 1                    | Input                                                                           | Pull-down*          | Y*                    | Y                | Y                          |
| <b>1101b</b>               | 0                    | Open Drain                                                                      | Drive Low           | -                     | -                | -                          |
|                            | 1                    | Input                                                                           | Hi-Z                | -                     | Y                | Y                          |
| <b>1110b</b>               | 0                    | CMOS Output (except PWMx)                                                       | Drive Low           | -                     | -                | -                          |
|                            | 1                    |                                                                                 | Drive High          | -                     | -                | -                          |
| <b>1111b</b>               | X                    | Analog output for<br>1/2 V <sub>CC</sub> (LCD 1/2 bias)<br>Does not contain PA7 | 1/2 V <sub>CC</sub> | -                     | -                | -                          |
|                            |                      | - (PA7)                                                                         | Pull-up             |                       |                  |                            |

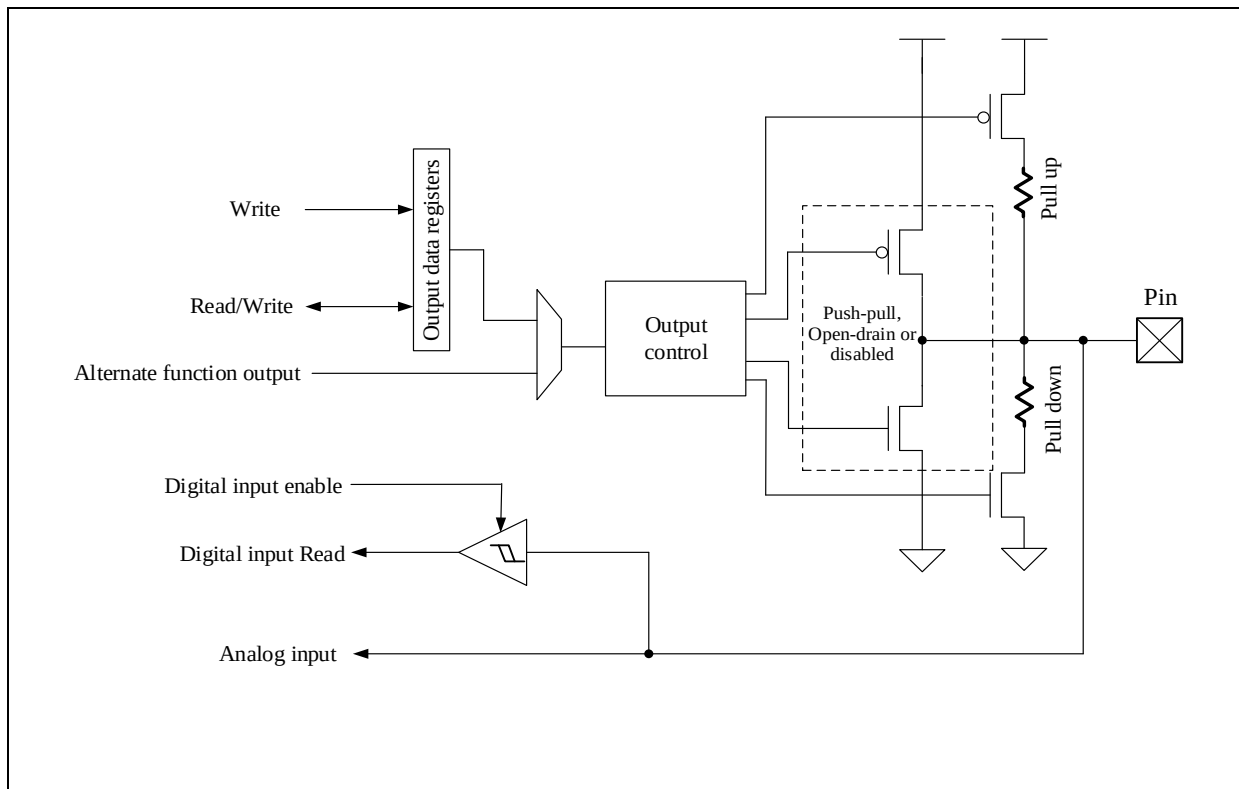
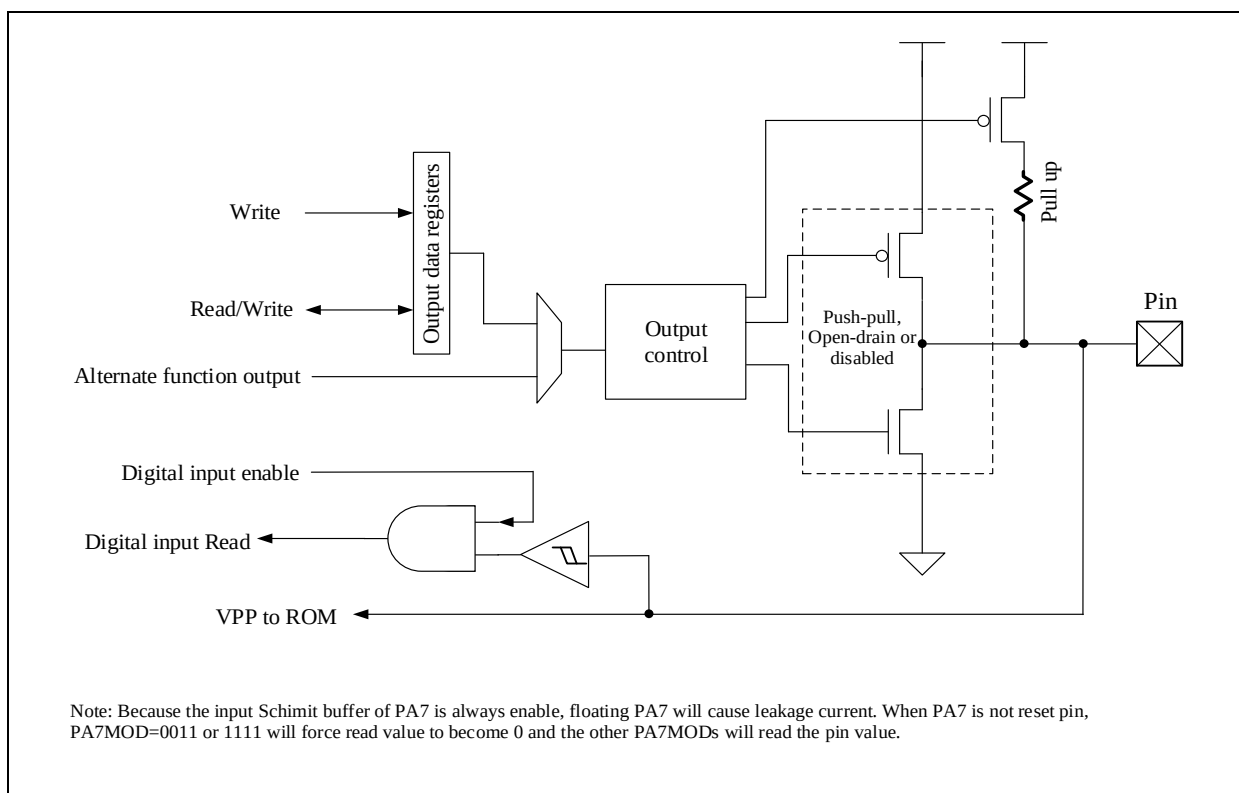
\*: PA7 has no high-sink, 1/2 bias and resistor pull-down capability.

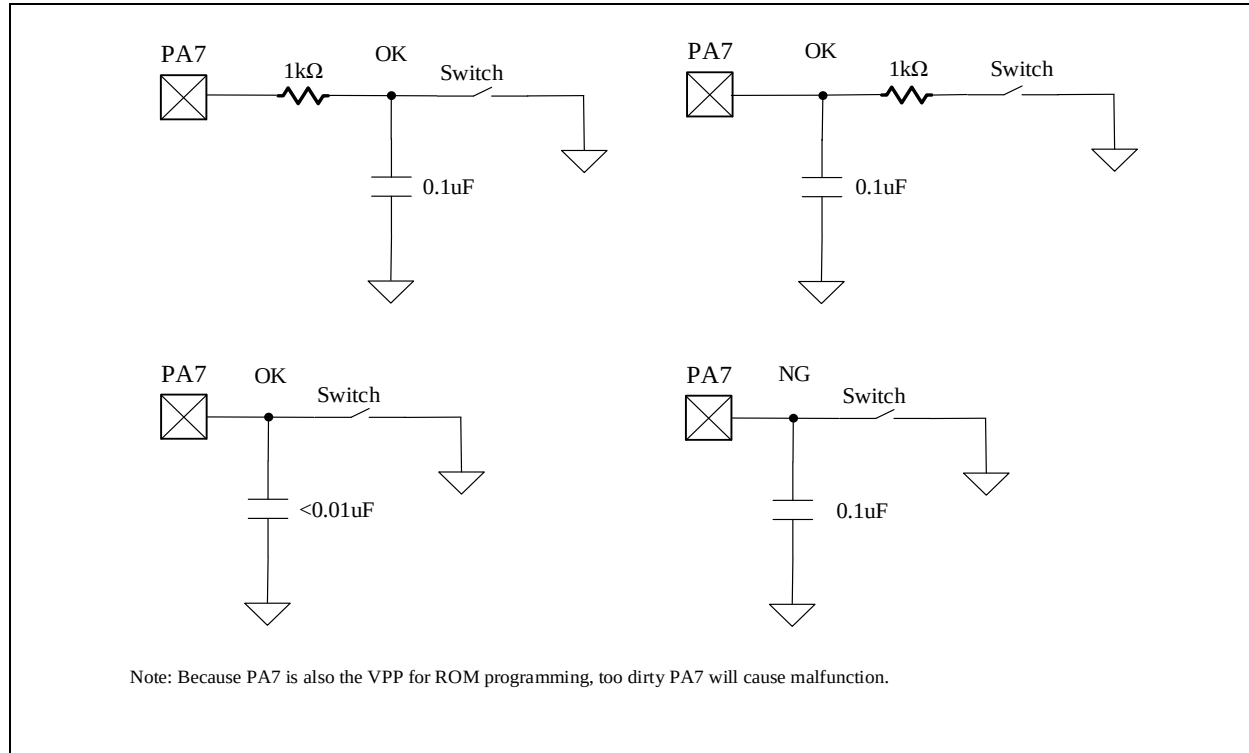
**I/O Pin Function Table 4**

| Pin Name | PAxMOD / PBxMOD / PDxMOD Setting |                           |                          |
|----------|----------------------------------|---------------------------|--------------------------|
|          | 0011b<br>(Analog in/out)         | 0111b<br>(Digital output) | 1111b<br>(Analog output) |
| PA0      | ADC0<br>CIN2                     | PWM5O                     | COM0 (LCD1/2 bias)       |
| PA1      | ADC1<br>CIP1                     | PWM1O                     | COM1 (LCD1/2 bias)       |
| PA2      | ADC2<br>CIP2                     | PWM4O                     | COM2 (LCD1/2 bias)       |
| PA3      | ADC3<br>CIN1                     | PWM2O                     | COM3 (LCD1/2 bias)       |
| PA4      | ADC4<br>XIN                      | PWM0P                     | COM4 (LCD1/2 bias)       |
| PA5      | ADC5<br>XOUT                     | PWM3O                     | COM5 (LCD1/2 bias)       |
| PA6      | ADC6                             | PWM0N                     | COM6 (LCD1/2 bias)       |
| PA7      |                                  |                           | Pull-up                  |
| PB0      | ADC7                             | PWM1O                     | COM7 (LCD1/2 bias)       |
| PB1      | ADC8                             | PWM2O                     | COM8 (LCD1/2 bias)       |
| PB2      | ADC9                             | PWM3O                     | COM9 (LCD1/2 bias)       |
| PB3      |                                  | PWM4O                     | COM10 (LCD1/2 bias)      |
| PB4      | ADC10<br>CIN4                    | PWM0P                     | COM11 (LCD1/2 bias)      |
| PB5      | ADC11                            | PWM5O                     | COM12 (LCD1/2 bias)      |
| PB6      | ADC12<br>CIP3                    | PWM0N                     | COM13 (LCD1/2 bias)      |
| PB7      | ADC13<br>CIN3                    | TM1OUT                    | COM14 (LCD1/2 bias)      |
| PD0      |                                  |                           | COM15 (LCD1/2 bias)      |
| PD1      | CIP4                             | TCOUT                     | COM16 (LCD1/2 bias)      |

**Special function for PAxMOD/PBxMOD/PDxMOD Table**




**General Pin Structure**

**PA7 Structure**



### Constraint on PA7

| 85h     | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|---------|--------|-------|-------|-------|--------|-------|-------|-------|
| PAMOD10 | PA1MOD |       |       |       | PA0MOD |       |       |       |
| R/W     | R/W    |       |       |       | R/W    |       |       |       |
| Reset   | 0      | 0     | 0     | 1     | 0      | 0     | 0     | 1     |

| 86h     | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|---------|--------|-------|-------|-------|--------|-------|-------|-------|
| PAMOD32 | PA3MOD |       |       |       | PA2MOD |       |       |       |
| R/W     | R/W    |       |       |       | R/W    |       |       |       |
| Reset   | 0      | 0     | 0     | 1     | 0      | 0     | 0     | 1     |

| 87h     | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|---------|--------|-------|-------|-------|--------|-------|-------|-------|
| PAMOD54 | PA5MOD |       |       |       | PA4MOD |       |       |       |
| R/W     | R/W    |       |       |       | R/W    |       |       |       |
| Reset   | 0      | 0     | 0     | 1     | 0      | 0     | 0     | 1     |

| 88h     | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|---------|--------|-------|-------|-------|--------|-------|-------|-------|
| PAMOD76 | PA7MOD |       |       |       | PA6MOD |       |       |       |
| R/W     | R/W    |       |       |       | R/W    |       |       |       |
| Reset   | 0      | 0     | 0     | 0     | 0      | 0     | 0     | 1     |

88h.7~4 **PA7MOD ~ PA0MOD:** PA7~PA0 Pin Mode Control

88h.3~0 0000: Open drain or digital input with pull-up

87h.7~4 0001: Open drain or digital input

87h.3~0 0010: CMOS Push-pull

86h.7~4 0011: Analog input/output

86h.3~0 0100: Open drain or digital input with pull-down(PA7 has no pull-down)

85h.7~4 0101: Open drain or digital input

85h.3~0 0110: CMOS Push-pull

0111: Alternate function output

1000: Open drain or digital input with pull-up and pin-changed wake-up

1001: Open drain or digital input and pin-changed wake-up  
 1010: CMOS Push-pull  
 1011: Reserved  
 1100: Open drain or digital input with pull-down and pin-changed wake-up(PA7 has no pull-down)  
 1101: Open drain or digital input and pin-changed wake-up  
 1110: CMOS Push-pull  
 1111: 1/2 V<sub>CC</sub> (LCD 1/2 bias) (except PA7) or pull-up(PA7)

| 8Ch     | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|---------|--------|-------|-------|-------|--------|-------|-------|-------|
| PBMOD10 | PB1MOD |       |       |       | PB0MOD |       |       |       |
| R/W     | R/W    |       |       |       | R/W    |       |       |       |
| Reset   | 0      | 0     | 0     | 1     | 0      | 0     | 0     | 1     |

| 8Dh     | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|---------|--------|-------|-------|-------|--------|-------|-------|-------|
| PBMOD32 | PB3MOD |       |       |       | PB2MOD |       |       |       |
| R/W     | R/W    |       |       |       | R/W    |       |       |       |
| Reset   | 0      | 0     | 0     | 1     | 0      | 0     | 0     | 1     |

| 8Eh     | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|---------|--------|-------|-------|-------|--------|-------|-------|-------|
| PBMOD54 | PB5MOD |       |       |       | PB4MOD |       |       |       |
| R/W     | R/W    |       |       |       | R/W    |       |       |       |
| Reset   | 0      | 0     | 0     | 1     | 0      | 0     | 0     | 1     |

| 8Fh     | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|---------|--------|-------|-------|-------|--------|-------|-------|-------|
| PBMOD76 | PB7MOD |       |       |       | PB6MOD |       |       |       |
| R/W     | R/W    |       |       |       | R/W    |       |       |       |
| Reset   | 0      | 0     | 0     | 1     | 0      | 0     | 0     | 1     |

8Fh.7~4 **PB7MOD ~ PB0MOD:** PB7~PB0 Pin Mode Control  
 8Fh.3~0 0000: Open drain or digital input with pull-up  
 8Eh.7~4 0001: Open drain or digital input  
 8Eh.3~0 0010: CMOS Push-pull  
 8Dh.7~4 0011: Analog input  
 8Dh.3~0 0100: Open drain or digital input with pull-down  
 8Ch.7~4 0101: Open drain or digital input  
 8Ch.3~0 0110: CMOS Push-pull  
 0111: Alternate function output  
 1000: Open drain or digital input with pull-up and pin-changed wake-up  
 1001: Open drain or digital input and pin-changed wake-up  
 1010: CMOS Push-pull  
 1011: Reserved  
 1100: Open drain or digital input with pull-down and pin-changed wake-up  
 1101: Open drain or digital input and pin-changed wake-up  
 1110: CMOS Push-pull  
 1111: 1/2 V<sub>CC</sub> (LCD 1/2 bias)

| 90h     | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|---------|--------|-------|-------|-------|--------|-------|-------|-------|
| PDMOD10 | PD1MOD |       |       |       | PD0MOD |       |       |       |
| R/W     | R/W    |       |       |       | R/W    |       |       |       |
| Reset   | 0      | 0     | 0     | 1     | 0      | 0     | 0     | 1     |

90h.7~4 **PD1MOD ~ PD0MOD:** PD1~PD0 Pin Mode Control  
 90h.3~0 0000: Open drain or digital input with pull-up  
 0001: Open drain or digital input  
 0010: CMOS Push-pull  
 0011: Analog input

0100: Open drain or digital input with pull-down  
 0101: Open drain or digital input  
 0110: CMOS Push-pull  
 0111: Alternate function output  
 1000: Open drain or digital input with pull-up and pin-changed wake-up  
 1001: Open drain or digital input and pin-changed wake-up  
 1010: CMOS Push-pull  
 1011: Reserved  
 1100: Open drain or digital input with pull-down and pin-changed wake-up  
 1101: Open drain or digital input and pin-changed wake-up  
 1110: CMOS Push-pull  
 1111: 1/2 V<sub>CC</sub> (LCD 1/2 bias)

| 05h   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| PAD   | PAD   |       |       |       |       |       |       |       |
| R/W   | R/W   |       |       |       |       |       |       |       |
| Reset | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |

05h.7~0 **PAD**: PA7~PA0 data

| 06h   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| PBD   | PBD   |       |       |       |       |       |       |       |
| R/W   | R/W   |       |       |       |       |       |       |       |
| Reset | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |

06h.7~0 **PBD**: PB7~PB0 data

| 07h   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| PDD   | —     | —     | —     | —     | —     | —     | PDD   |       |
| R/W   | —     | —     | —     | —     | —     | —     | R/W   |       |
| Reset | —     | —     | —     | —     | —     | —     | 1     | 1     |

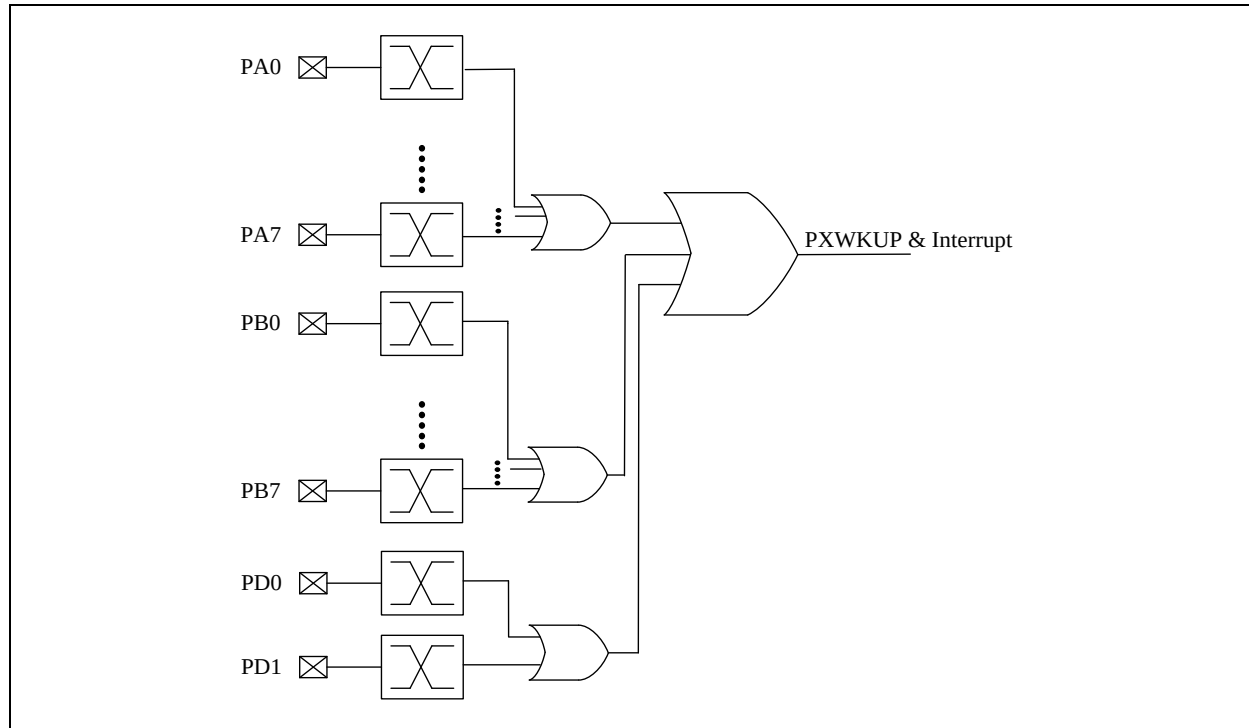
07h.1~0 **PDD**: PD1~PD0 data

| 105h   | Bit 7 | Bit 6 | Bit 5    | Bit 4    | Bit 3 | Bit 2 | Bit 1    | Bit 0    |
|--------|-------|-------|----------|----------|-------|-------|----------|----------|
| PINMOD | —     | —     | Reserved | Reserved | —     | HSINK | Reserved | Reserved |
| R/W    | —     | —     | R        | R/W      | —     | R/W   | R/W      | R/W      |
| Reset  | —     | —     | x        | 0        | —     | 1     | 0        | 0        |

105h.5 **Reserved**: read as unknown after reset  
 105h.4 **Reserved**: must be kept at 0  
 105h.2 **HSINK**: All IO ports high sink current enable  
     0: low sink current  
     1: high sink current  
 105h.1 **Reserved**: must be kept at 0  
 105h.0 **Reserved**: must be kept at 0

## 5.2 Pin-change Wake-up & Interrupt

All of the IO pins also have the pin-change wake-up and interrupt capability. A sleep instruction is necessary before the event of pin-change otherwise pin-change event may be missed.

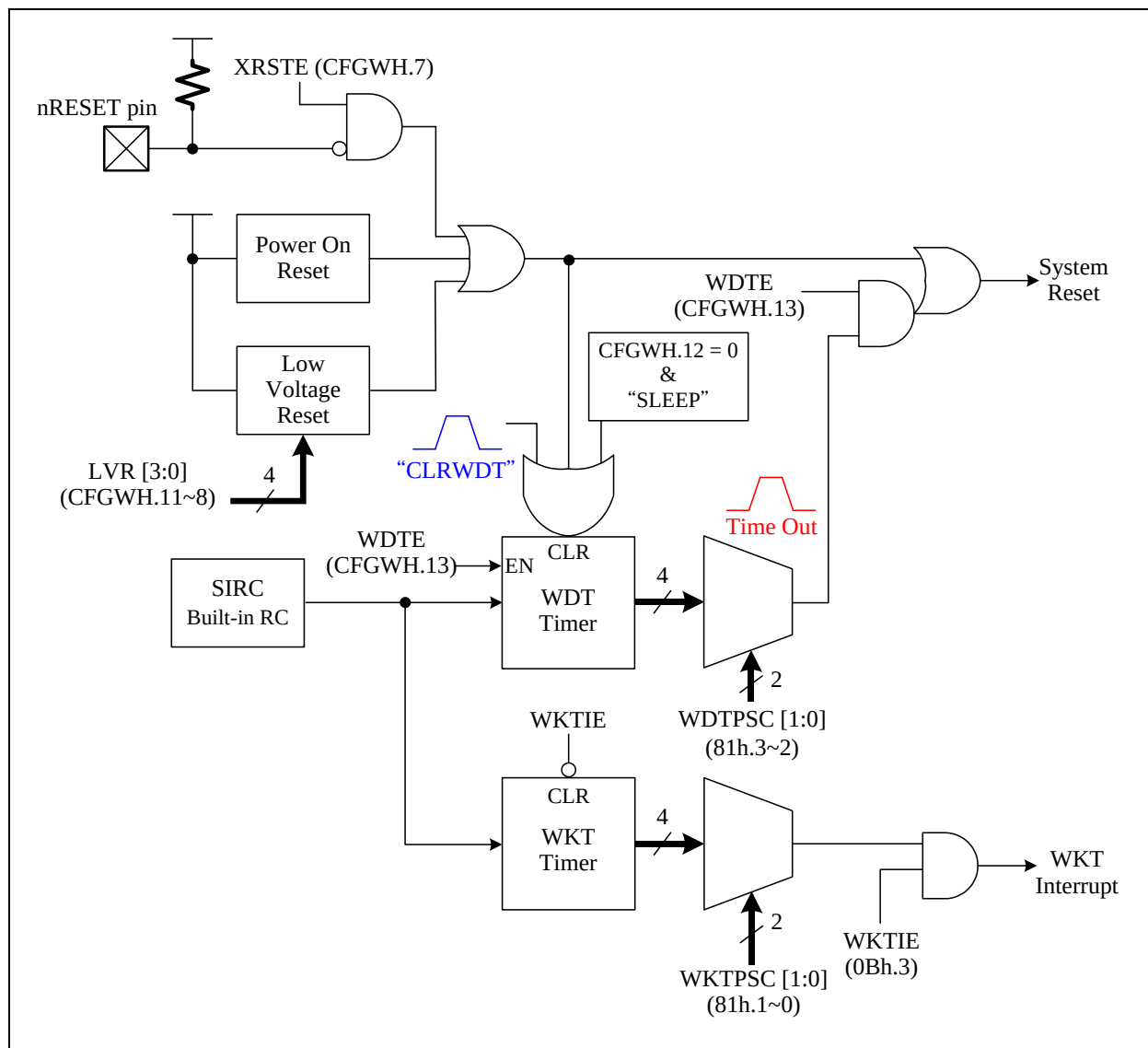


## 6 Peripheral Functional Block

### 6.1 Watchdog (WDT) /Wake-up (WKT) Timer

The WDT and WKT share the same built-in internal RC Oscillator and have individual counters. The overflow period of WDT, WKT can be selected by individual prescaler (WDTPSC[1:0], WKTTPSC[1:0]). The WDT timer is cleared by the CLRWDT instruction. If the Watchdog is enabled, the WDT generates the chip reset signal.

The WKT timer is an interval timer, WKT time out will generate WKT Interrupt Flag (WKTIF). The WKT timer is cleared/stopped by WKTIE=0. Set WKTIE=1, the WKT timer will always count regardless at any CPU operating mode.



WDT/WKT Block Diagram

The WDT's behavior in different Mode is shown as below table.

| Mode                    | CFGWH[13:12] |         | WDT  |
|-------------------------|--------------|---------|------|
|                         | WDTE[1]      | WDTE[0] |      |
| Normal Mode             | 0            | 0       | Stop |
|                         | 0            | 1       | Stop |
|                         | 1            | 0       | Run  |
|                         | 1            | 1       | Run  |
| Power-down Mode (SLEEP) | 0            | 0       | Stop |
|                         | 0            | 1       | Stop |
|                         | 1            | 0       | Stop |
|                         | 1            | 1       | Run  |

Watchdog clear is controlled by CLRWDT instruction.

◇ Example: Clear watchdog timer by CLRWDT instruction.

```

MAIN:    ...                ; Execute program.
          CLRWDT             ; Execute CLRWDT instruction.
          ...
          LGOTO    MAIN

```

◇ Example: Setup WDT time.

```

          MOVLW    00000111b
          MOVWX    OPTION           ; Select WDT Time out=250 ms @5V
          ...

```

◇ Example: Set WKT period and interrupt function.

```

          MOVLW    00000110b
          MOVWX    OPTION           ; Select WKT period=62.5 ms @5V
          MOVLW    11110111b       ; Clear WKT interrupt flag by using byte operation
          MOVWX    INTIF            ; Don't use bit operation "BCX WKTIF" to clear

          BSX      WKTIE            ; Enable WKT interrupt function

```

| 03h    | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| STATUS | IRP   | RP1   | RP0   | TO    | PD    | Z     | DC    | C     |
| R/W    | R/W   | R/W   | R/W   | R     | R     | R/W   | R/W   | R/W   |
| Reset  | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

03h.4 **TO:** WDT time out flag, read-only  
0: after Power On Reset or CLRWDT / SLEEP instructions  
1: WDT time out occurs

| 0Ch   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIF | ADCIF | T2IF  | TM1IF | TM0IF | WKTIF | INT2IF | INT1IF | INT0IF |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

0Ch.3 **WKTIF:** Wake-up Timer interrupt event pending flag  
This bit is set by H/W while Wake-up Timer is timeout, write F7h to INTIF will clear this flag

| 0Bh   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIE | ADCIE | T2IE  | TM1IE | TM0IE | WKTIE | INT2IE | INT1IE | INT0IE |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

0Bh.3 **WKTIE:** Wake-up Timer interrupt enable and Wake-up Timer enable  
 0: disable  
 1: enable

| 81h    | Bit 7  | Bit 6   | Bit 5   | Bit 4 | Bit 3  | Bit 2 | Bit 1   | Bit 0 |
|--------|--------|---------|---------|-------|--------|-------|---------|-------|
| OPTION | HWAUTO | INT0EDG | INT1EDG | –     | WDTPSC |       | WKTTPSC |       |
| R/W    | R/W    | R/W     | R/W     | –     | R/W    |       | R/W     |       |
| Reset  | 0      | 0       | 0       | –     | 1      | 1     | 1       | 1     |

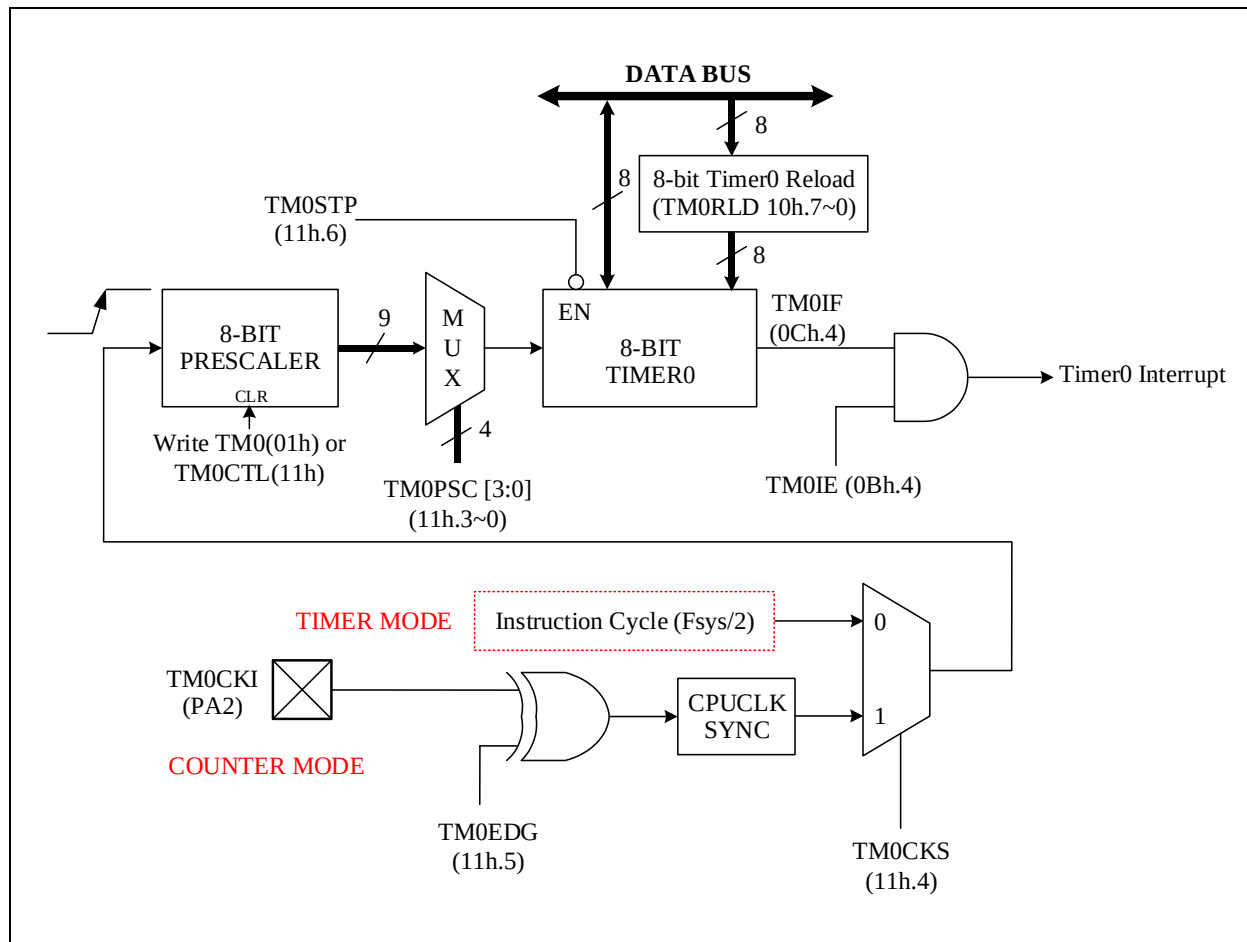
81h.3~2 **WDTPSC:** WDT period (@V<sub>CC</sub>=5V)  
 00: 125 ms  
 01: 250 ms  
 10: 1001 ms  
 11: 2001 ms

81h.1~0 **WKTTPSC:** WKT period (@V<sub>CC</sub>=5V)  
 00: 15.6 ms  
 01: 31.3 ms  
 10: 62.5 ms  
 11: 125 ms



## 6.2 Timer0

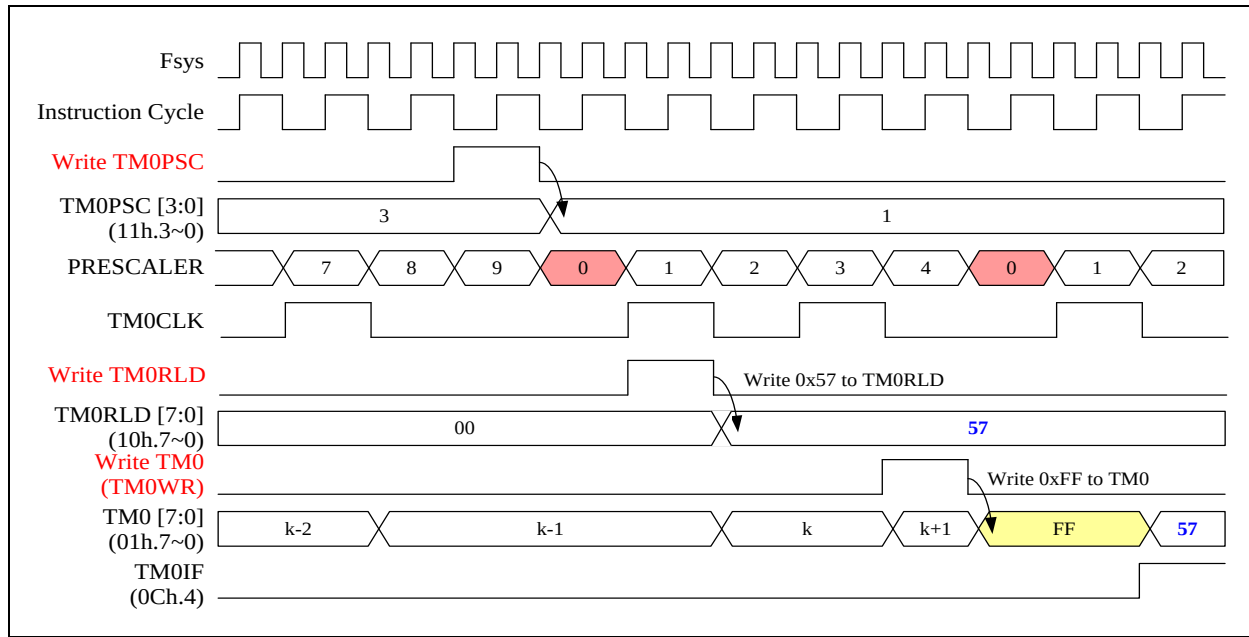
Timer0(TM0) (01h.7~0) is an 8-bit wide register. It can be read or written as any other register. Besides, Timer0 increases itself periodically and automatically rolls over a new "offset value" (TM0RLD) while it rolls over based on the pre-scaled clock source, which can be  $F_{sys}/2$  or TM0CKI (PA2) rising/falling input. The Timer0 increase rate is determined by "Timer0 Pre-Scale" (TM0PSC) register. The Timer0 always generates TM0IF (0Ch.4) when its count rolls over. It generates Timer0 Interrupt if TM0IE (0Bh.4) is set. Timer0 can be stopped counting if the TM0STP (11h.6) bit is set.



Timer0 Block Diagram

The following timing diagram describes the Timer0 works in pure Timer mode.

When the Timer0 prescaler (TM0PSC) is written, the internal 8-bit prescaler will be cleared to 0 to make the counting period correct at the first Timer0 count. TM0CLK is the internal signal that causes the Timer0 to increase by 1 at the end of TM0CLK. TM0WR is also the internal signal that indicates the Timer0 is directly written by instruction; meanwhile, the internal 8-bit prescaler will be cleared. When Timer0 counts from FFh to TM0RLD, TM0IF (Timer0 Interrupt Flag) will be set to 1, and generate interrupt if TM0IE (Timer0 Interrupt Enable) is set.



**Timer0 works in Timer mode (TM0CKS=0)**

The equation of Timer0 interrupt time value is as following:

$$\text{Timer0 interrupt interval cycle time} = F_{\text{sys}} / 2 / \text{TM0PSC} / (256 - \text{TM0RLD})$$

◇ Example: Setup Timer0 work in Timer mode, if  $F_{\text{sys}} = 8 \text{ MHz}$

; Setup Timer0 clock source and divider

```
MOVLW    00x00101b    ; TM0CKS = 0, Timer0 clock is instruction cycle
MOVWX    TM0CTL        ; TM0PSC = 0101b, divided by 32
```

; Setup Timer0 reload data

```
MOVLW    80h
MOVWX    TM0RLD        ; Set Timer0 reload data = 128
```

; Setup Timer0

```
BSX      TM0STP        ; Timer0 stops counting
CLR      TM0           ; Clear Timer0 content
```

; Enable Timer0 and interrupt function

```
MOVLW    11101111b
MOVWX    INTIF        ; Clear Timer0 request interrupt flag
BSX      TM0IE        ; Enable Timer0 interrupt function
BCX      TM0STP        ; Enable Timer0 counting
```

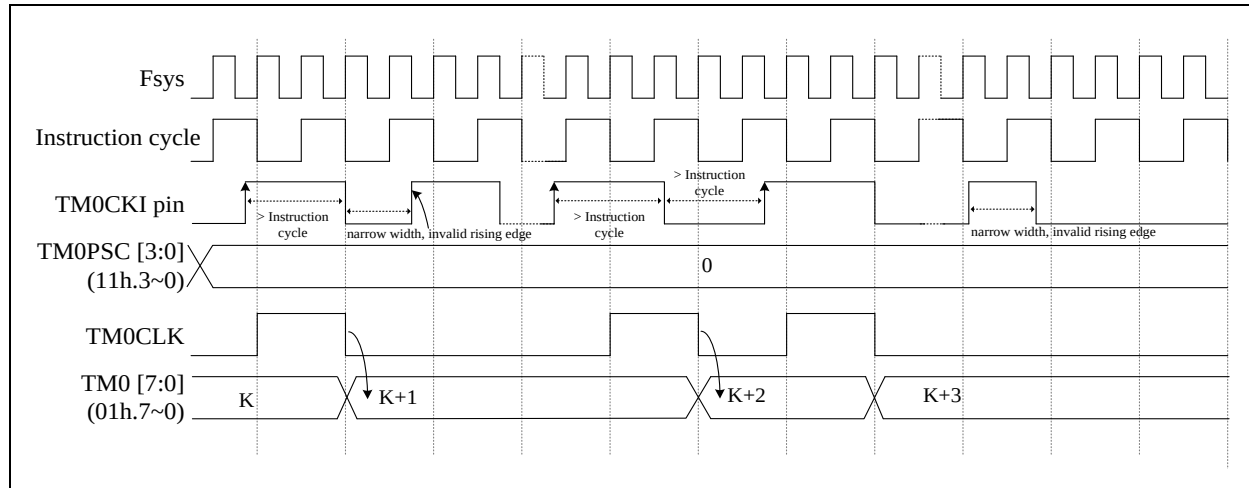
Timer0 interrupt frequency =  $F_{\text{sys}} / 2 / \text{TM0PSC} / (256 - \text{TM0RLD})$ ,

$F_{\text{sys}} = 8 \text{ MHz}$ ,  $\text{TM0PSC} = \text{div } 32$ ,  $\text{TM0RLD} = 128$

Timer0 interrupt frequency =  $8 \text{ MHz} / 2 / 32 / (256 - 128) = 0.976 \text{ KHz}$

The following timing diagram describes the Timer0 works in Counter mode.

If TM0CKS=1 then Timer0 counter source clock is from TM0CKI pin. TM0CKI signal is synchronized by instruction cycle ( $F_{sys}/2$ ) that means the high/low time durations of TM0CKI must be longer than one instruction cycle time ( $F_{sys}/2$ ) to guarantee each TM0CKI's change will be detected correctly by the synchronizer.



Timer0 works in Counter mode for TM0CKI (TM0EDG=0), TM0CKS=1

◇ Example: Setup TM0 work in Counter mode and clock source from TM0CKI pin (PA2)

```

; Setup Timer0 clock source and divider
    MOVLW    00110000B    ; TM0EDG = 1, counting edge is falling edge
    MOVWX    TM0CTL        ; TM0CKS = 1, Timer0 clock is TM0CKI
                                ; TM0PSC = 0000b, divided by 1

; Setup Timer0
    BSX      TM0STP        ; Timer0 stops counting
    CLRX     TM0           ; Clear Timer0 content

; Enable Timer0 and read Timer0 counter
    BCX      TM0STP        ; Enable Timer0 counting
    ...
    BSX      TM0STP        ; Timer0 stops counting
    MOVXW    TM0           ; Read Timer0 content
    
```

| 01h   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| TM0   | TM0   |       |       |       |       |       |       |       |
| R/W   | R/W   |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

01h.7~0 **TM0**: Timer0 content

| 0Bh   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIE | ADCIE | T2IE  | TM1IE | TM0IE | WKTIE | INT2IE | INT1IE | INT0IE |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

0Bh.4 **TM0IE:** Timer0 interrupt enable  
0: disable  
1: enable

| 0Ch   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIF | ADCIF | T2IF  | TM1IF | TM0IF | WKTIF | INT2IF | INT1IF | INT0IF |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

0Ch.4 **TM0IF:** Timer0 interrupt event pending flag  
This bit is set by H/W while Timer0 overflows, write EFh to INTIF will clear this flag

| 10h    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| TM0RLD | TM0RLD |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

10h.7~0 **TM0RLD:** Timer0 reload data

| 11h    | Bit 7 | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|--------|--------|--------|--------|-------|-------|-------|
| TM0CTL | —     | TM0STP | TM0EDG | TM0CKS | TM0PSC |       |       |       |
| R/W    | —     | R/W    | R/W    | R/W    | R/W    |       |       |       |
| Reset  | —     | 0      | 0      | 0      | 0      | 0     | 0     | 0     |

11h.6 **TM0STP:** Stop Timer0  
0: Timer0 runs  
1: Timer0 stops

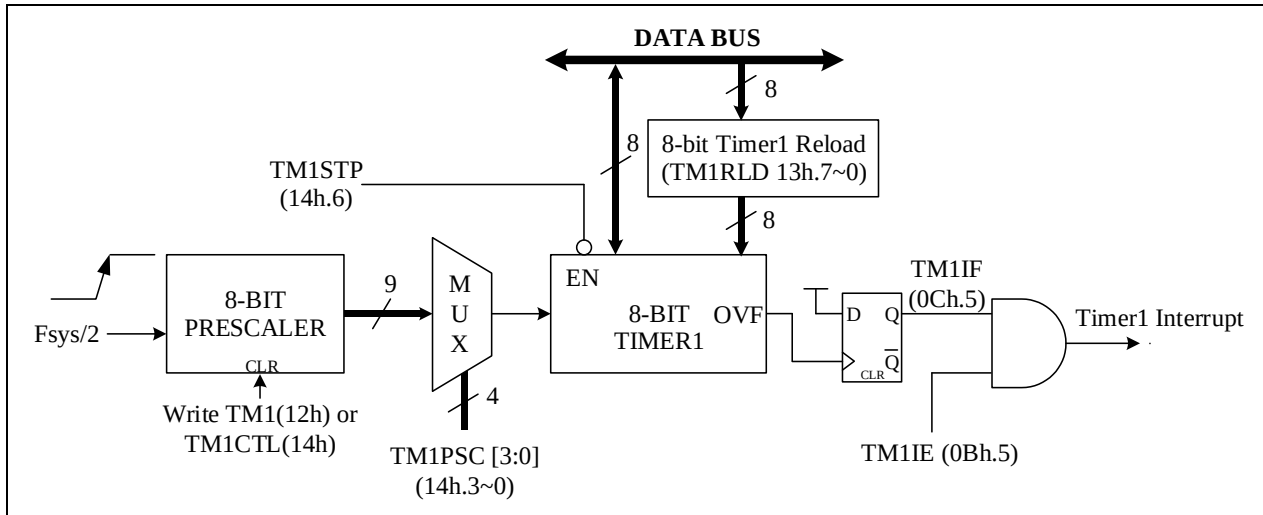
11h.5 **TM0EDG:** Timer0 prescaler counting edge for TM0CKI pin  
0: rising edge  
1: falling edge

11h.4 **TM0CKS:** Timer0 prescaler clock source  
0: Fsys/2  
1: TM0CKI pin (PA2 pin)

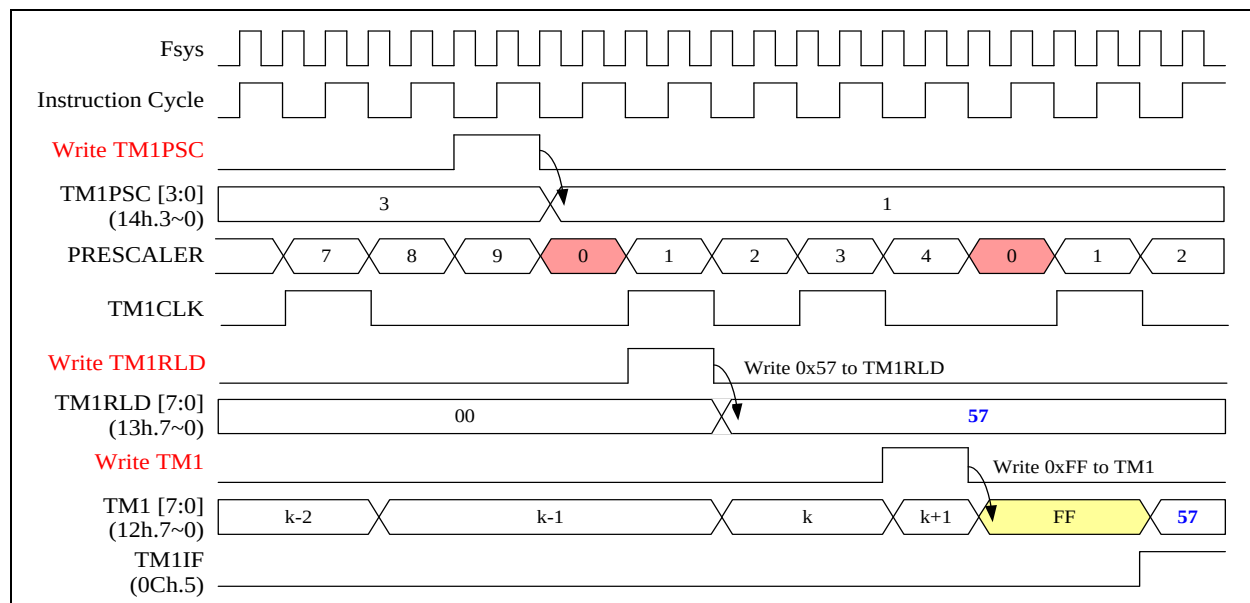
11h.3~0 **TM0PSC:** Timer0 prescaler. Timer0 prescaler clock source divided by  
0000: 1                      0001: 2                      0010: 4                      0011: 8  
0100: 16                      0101: 32                      0110: 64                      0111: 128  
1xxx: 256

### 6.3 Timer1

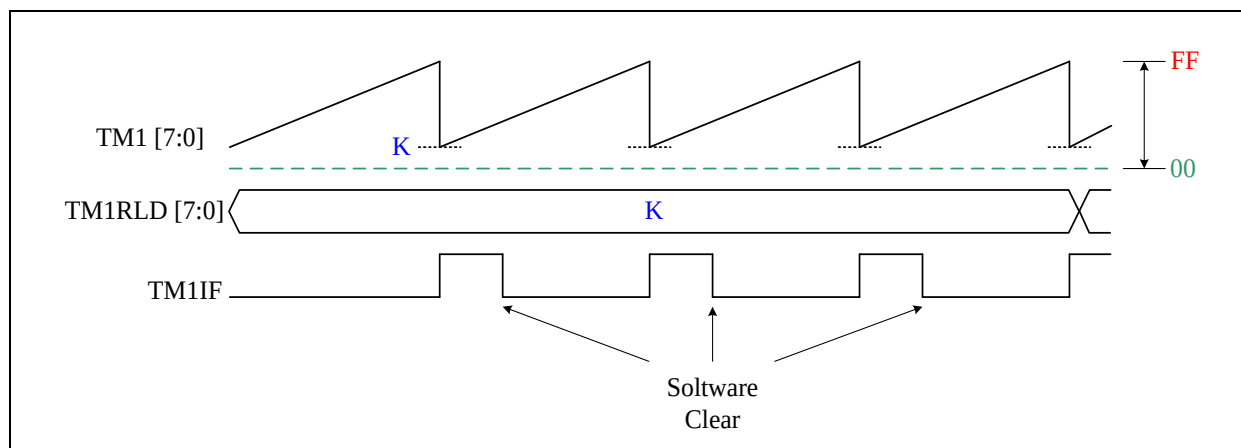
Timer1(TM1) (12h.7~0) is an 8-bit wide register. It can be read or written as any other register. Besides, Timer1 increases itself periodically and automatically reloads a new "offset value" (TM1RLD) while it rolls over based on the pre-scaled instruction clock ( $F_{sys}/2$ ). The Timer1 increase rate is determined by TM1PSC register. It generates Timer1 interrupt if the TM1IE bit is set. Timer1 can be stopped counting if the TM1STP bit is set.



Timer1 Block Diagram



Timer1 Timing Diagram



Timer1 Reload Diagram

◇ Example: CPU is running in SLOW mode,  $F_{sys} = \text{Slow-clock} / \text{CPUPSC} = 32.75 \text{ KHz}$

; Setup Timer1 clock source and divider

```
MOVLW    00000011b
MOVWX    TM1CTL           ; TM1PSC = 0011b, divided by 8
```

; Setup Timer1 reload data

```
MOVLW    FFh
MOVWX    TM1RLD           ; Set Timer1 reload data = 255
```

; Setup Timer1

```
BSX      TM1STP           ; Timer1 stops counting
CLRXL    TM1              ; Clear Timer1 content
```

; Enable Timer1 and interrupt function

```
MOVLW    11011111b
MOVWX    INTIF            ; Clear Timer1 request interrupt flag
BSX      TM1IE            ; Enable Timer1 interrupt function
BCX      TM1STP           ; Enable Timer1 counting
```

Timer1 interrupt frequency =  $F_{sys} / 2 / \text{TM1PSC} / (256 - \text{TM1RLD})$ ,

$F_{sys} = 32.75 \text{ KHz}$ ,  $\text{TM1PSC} = \text{div } 8$ ,  $\text{TM1RLD} = 255$

Timer1 interrupt frequency =  $32.75 \text{ KHz} / 2 / 8 / (256 - 255) = 2.047 \text{ KHz}$

| 0Bh   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIE | ADCIE | T2IE  | TM1IE | TM0IE | WKTIE | INT2IE | INT1IE | INT0IE |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

0Bh.5 **TM1IE**: Timer1 interrupt enable

0: disable

1: enable

| 0Ch   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIF | ADCIF | T2IF  | TM1IF | TM0IF | WKTIF | INT2IF | INT1IF | INT0IF |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

0Ch.5 **TM1IF**: Timer1 interrupt event pending flag  
This bit is set by H/W while Timer1 overflows, write DFh to INTIF will clear this flag

| 12h   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| TM1   | TM1   |       |       |       |       |       |       |       |
| R/W   | R/W   |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

12h.7~0 **TM1**: Timer1 content

| 13h    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| TM1RLD | TM1RLD |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

13h.7~0 **TM1RLD**: Timer1 reload data

| 14h    | Bit 7 | Bit 6  | Bit 5 | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|--------|-------|-------|--------|-------|-------|-------|
| TM1CTL | –     | TM1STP | –     | –     | TM1PSC |       |       |       |
| R/W    | –     | R/W    | –     | –     | R/W    |       |       |       |
| Reset  | –     | 0      | –     | –     | 0      | 0     | 0     | 0     |

14h.6 **TM1STP**: Stop Timer1

0: Timer1 runs

1: Timer1 stops

14h.3~0 **TM1PSC**: Timer1 prescaler. Timer1 prescaler clock source divided by

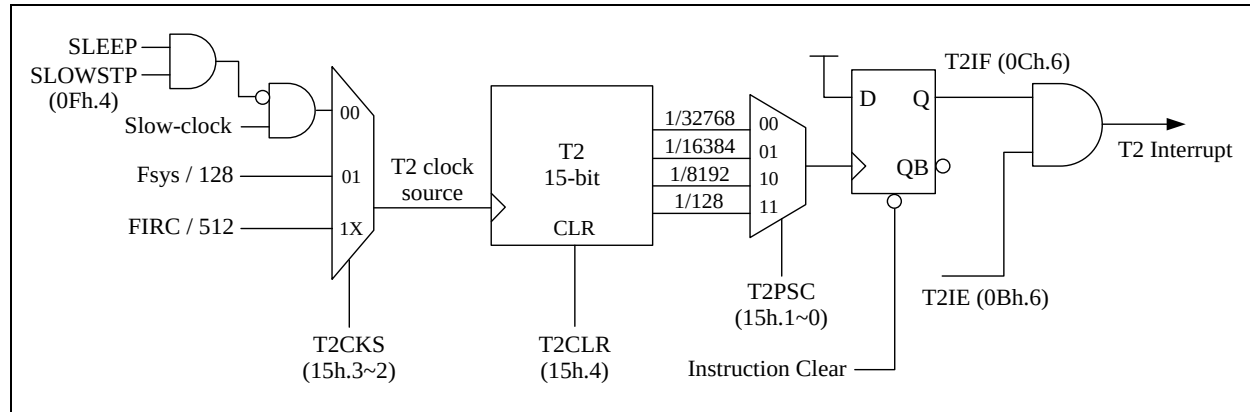
0000: 1      0001: 2      0010: 4      0011: 8

0100: 16      0101: 32      0110: 64      0111: 128

1xxx: 256

## 6.4 T2:15-bit Timer

The T2 is a 15-bit counter and the clock sources are from Slow-clock,  $F_{sys}/128$ , or FIRC/512 (16 MHz/512). It is used to generate time base interrupt and T2 counter block clock. The T2 content cannot be read by instructions. It generates interrupt flag T2IF (0Ch.6) with the clock divided by 32768/16384/8192/128 depends on T2PSC[1:0] (15h.1~0) register bits. The following figure shows the block diagram of T2.



**T2 Block Diagram**

◇ Example: CPU is running at FAST mode,  $F_{sys} = \text{Fast-clock} / \text{CPUPSC} = \text{FIRC} / 2 = 8 \text{ MHz}$

; Setup T2 clock source and divider

```

MOV LW    00000101b      ; T2CKS(15h.3~2) = 1, T2 clock source is Fsys/128
MOV WX    T2CTL           ; T2PSC(15h.1~0) = 1, divided by 16384
BSX       T2CLR           ; T2CLR = 1, clear T2 counter
    
```

; Enable T2 interrupt function

```

MOV LW    10111111b      ; Clear T2 request interrupt flag
MOV WX    INTIF           ; Enable T2 interrupt function
BSX       T2IE            ; T2CLR = 0, Enable T2 counting
BCX       T2CLR
    
```

T2 clock source is  $F_{sys} / 128 = 8 \text{ MHz} / 128 = 62500 \text{ Hz}$ , T2PSC = 16384

T2 frequency =  $62500 \text{ Hz} / 16384 = 3.815 \text{ Hz}$

| 0Bh   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIE | ADCIE | T2IE  | TM1IE | TM0IE | WKTIE | INT2IE | INT1IE | INT0IE |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

0Bh.6 **T2IE:** T2 interrupt enable  
 0: disable  
 1: enable



| 0Ch   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIF | ADCIF | T2IF  | TM1IF | TM0IF | WKTIF | INT2IF | INT1IF | INT0IF |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

0Ch.6 **T2IF:** T2 interrupt event pending flag

This bit is set by H/W while T2 overflows, write BFh to INTIF will clear this flag

| 0Fh    | Bit 7  | Bit 6   | Bit 5 | Bit 4   | Bit 3   | Bit 2  | Bit 1  | Bit 0 |
|--------|--------|---------|-------|---------|---------|--------|--------|-------|
| CLKCTL | SCKTYP | FCKTYPE | —     | SLOWSTP | FASTSTP | CPUCKS | CPUPSC |       |
| R/W    | R/W    | R/W     | —     | R/W     | R/W     | R/W    | R/W    | R/W   |
| Reset  | 0      | 0       | —     | 0       | 1       | 0      | 1      | 1     |

0Fh.4 **SLOWSTP:** Stop Slow-clock after execute SLEEP instruction

0: Slow-clock keeps running after execute SLEEP instruction

1: Slow-clock stops running after execute SLEEP instruction

| 15h   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| T2CTL | —     | —     | —     | T2CLR | T2CKS |       | T2PSC |       |
| R/W   | —     | —     | —     | R/W   | R/W   |       | R/W   |       |
| Reset | —     | —     | —     | 0     | 0     | 0     | 0     | 0     |

15h.4 **T2CLR:** Clear and stop T2

0: T2 runs

1: T2 clear and stops

15h.3~2 **T2CKS:** T2 clock source selection

00: Slow-clock

01: Fsys/128

1x: FIRC/512 (16 MHz/512)

15h.1~0 **T2PSC:** T2 prescaler. T2 clock source divided by

00: 32768

01: 16384

10: 8192

11: 128

## 6.5 PWM: 16 bits PWM

There are six PWMs in this chip. PWM0~PWM5 have independent 16-bit duty control register, and share a set of 16-bit period register. The PWM can generate various frequency waveform with 65536 duty resolution on the basis of the PWM clock. The PWM clock can select  $F_{sys}$ , FIRC/256, FIRC (16 MHz), or  $FIRC*2$  (32 MHz) as its clock source. The following takes PWM0 and PWM1 as an example for description.

The 16-bit PWMPRD, PWMxD( $x=0\sim5$ ) registers both have a low byte and high byte structure. The high bytes can be directly accessed, but the low bytes can only be accessed via an internal 8-bit buffer, reading or writing to these register pairs must be carried out in a specific way. The important point to notes is that data transfer to and from the 8-bit buffer and its related low byte only takes place when write or read operation to its corresponding high bytes is executed. **Briefly speaking, write low byte first and then high byte; read high byte first and then low byte.**

If PWMEN is cleared, the PWM0~5 will be cleared and stopped, otherwise the PWM0~5 remain running. PWMx( $x=0\sim5$ ) share the same period register which can be set by writing the period value to the PWMPRDH and PWMPRDL registers. After writing the PWMxDH( $x=0\sim5$ ) or PWMPRDH register, H/W will update PWM period and duty immediately. PWM0~5 share an interrupt flag, and an interrupt flag is generated at the end of the period.

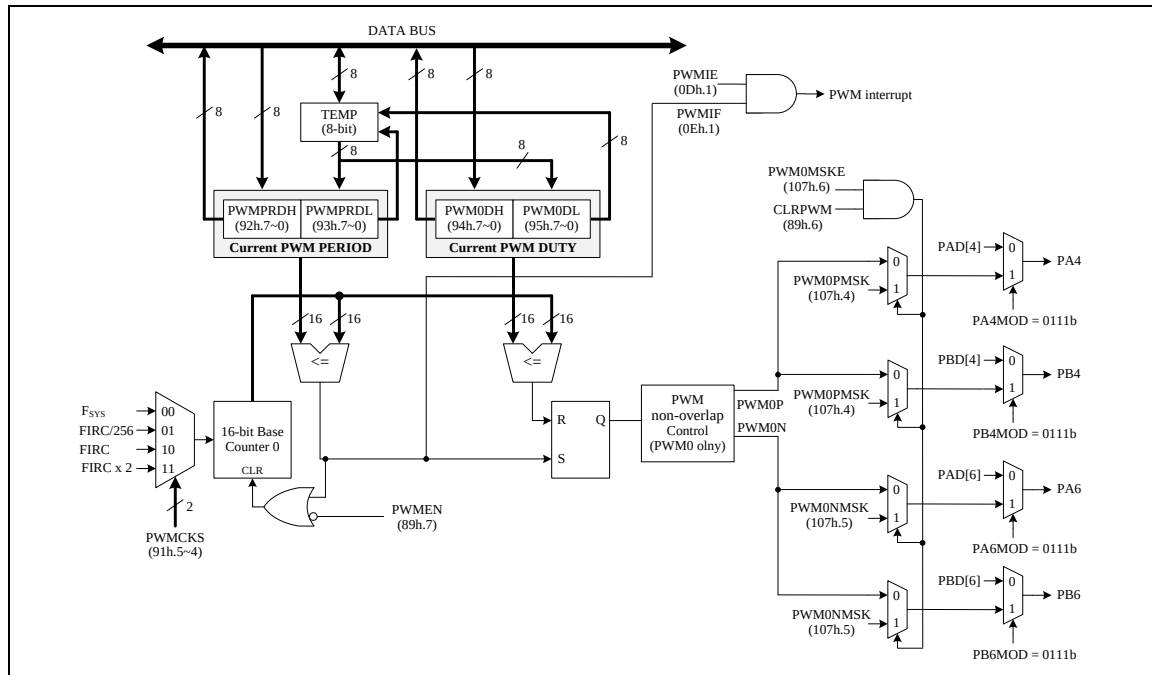
Only PWM0 has dead-zone(non-overlap) control, and is divided into PWM0P and PWM0N outputs, and the remaining PWM1~PWM5 have no dead-zone(non-overlap) control. The PWM1~5 outputs are PWM1O~PWM5O. User can use pin mode setting to output PWMxO to the corresponding IO pin, refer to Chapter 5 for more information on pin settings.

For reading and writing of 16-bit PWM period and duty, it is recommended to update the data only in the main program, or only update the data in the interrupt, to avoid possible errors.

### 6.5.1 PWM0

If PWMEN is cleared, the PWM0~5 will be cleared and stopped, otherwise the PWM0~5 remain running. The PWM0 structure is shown as follow. The PWM0 duty cycle can be changed by writing to PWM0DH and PWM0DL. The PWM0 output signal resets to a low level whenever the 16-bit base counter matches the 16-bit PWM0 duty register {PWM0DH, PWM0DL}. The PWM0 period can be set by writing the period value to the PWMPRDH and PWMPRDL registers. After writing the PWM0DH or PWMPRDH register, H/W will update PWM period and duty immediately. PWM0~5 share an interrupt flag, and an interrupt flag is generated at the end of the period.

The PWM0 has two operation modes, normal mode and half-bridge mode.

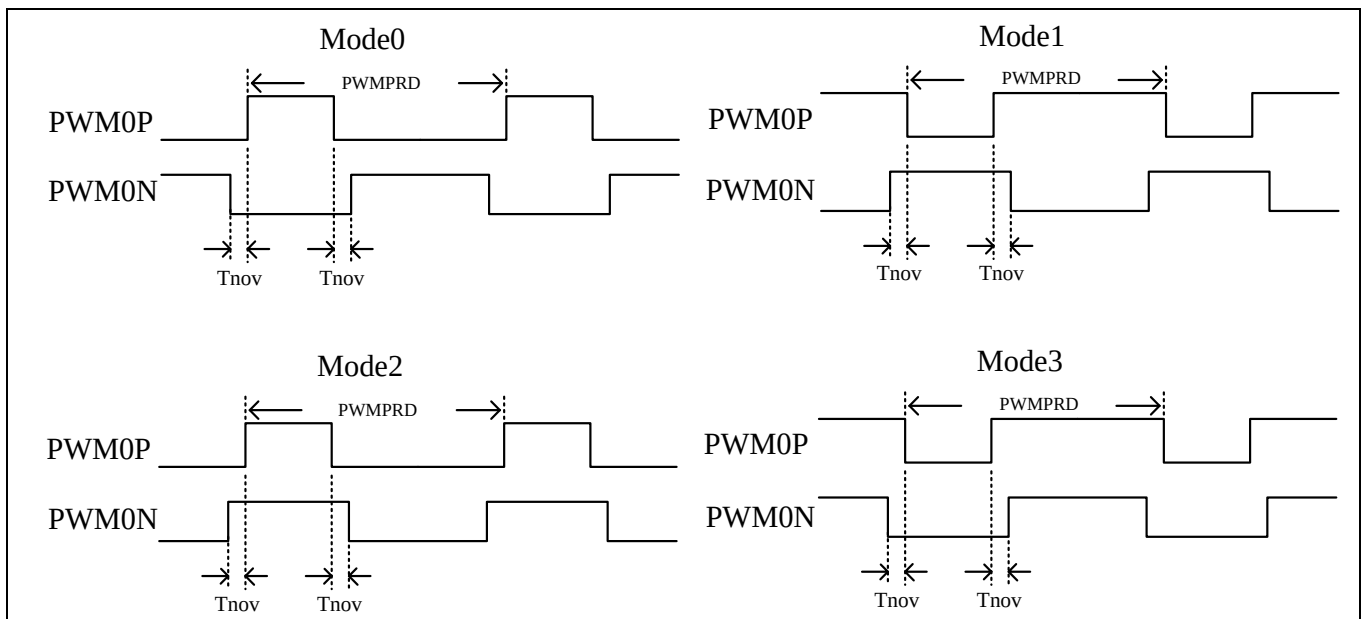


**PWM0 Block Diagram**

### 6.5.1.1 Normal Mode

The normal mode PWM is a simple structure, which switches its output high and low at uniform repeatable intervals. The PWM0D is the output duty cycle, and the output period is PWMPRD+1.

Only PWM0 can be output via PWM0P and PWM0N with four different modes. The edges of the PWM pulse can be separated with 16 different dead-zone(non-overlap) clocks intervals (Tnov). The width of Tnov can be selected by PWM0DZ (89h.3~0) within 0~15 PWM clock. The default output form is Mode0. The waveforms of the four output modes are shown below.



**PWM0 Normal Mode Output Modes**

◇ Example:

; Setup Pin mode

```
MOVLW    xxxx0111b    ;
MOVWX    PAMOD54        ; PA4 Pin as PWM0P
```

```
MOVLW    xxxx0111b    ;
MOVWX    PAMOD76        ; PA6 Pin as PWM0N
```

; Setup PWM0 clock source select

```
MOVLW    xx10xxxxb    ;
MOVWX    OPTION2        ; FIRC 16 MHz as PWM clock source
```

; Setup PWM0 period and duty setting

```
MOVLW    FFh
MOVWX    PWMPRDL        ; write sequence: PWMPRDL then PWMPRDH
MOVLW    7Fh
MOVWX    PWMPRDH        ; Set PWM period = 7FFFh
```

```
MOVLW    00h
MOVWX    PWM0DL        ; write sequence: PWM0DL then PWM0DH
MOVLW    40h
MOVWX    PWM0DH        ; Set PWM0 duty = 4000h
```

; Setup PWM0 enable and dead-zone(non-overlap) control

```
MOVLW    10000000b    ; 89h.7 = 1, PWM0 enable
MOVWX    PWMCTL        ; 89h.5~4 = 0, PWM0 Mode0 output
                        ; 89h.3~0 = 0, PWM0 dead-zone(non-overlap) output
                        ; disable
```

Example:

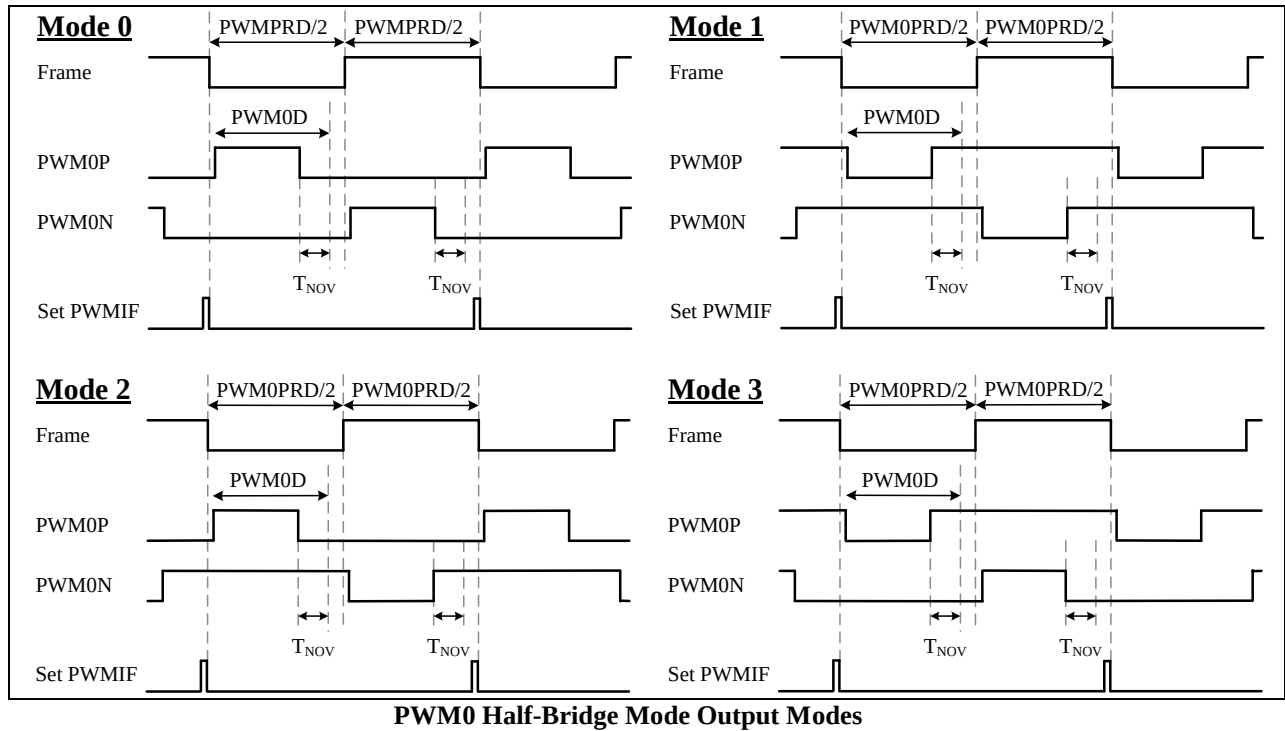
PWM0 clock source = FIRC 16 MHz, PWM period = 7FFFh, PWM duty = 4000h

PWM0 output frequency = 16 MHz / (period+1) = 16 MHz / 32768 = 488 Hz.

PWM0 output duty = duty / (period+1) = 50 %.

### 6.5.1.2 Half-Bridge Mode

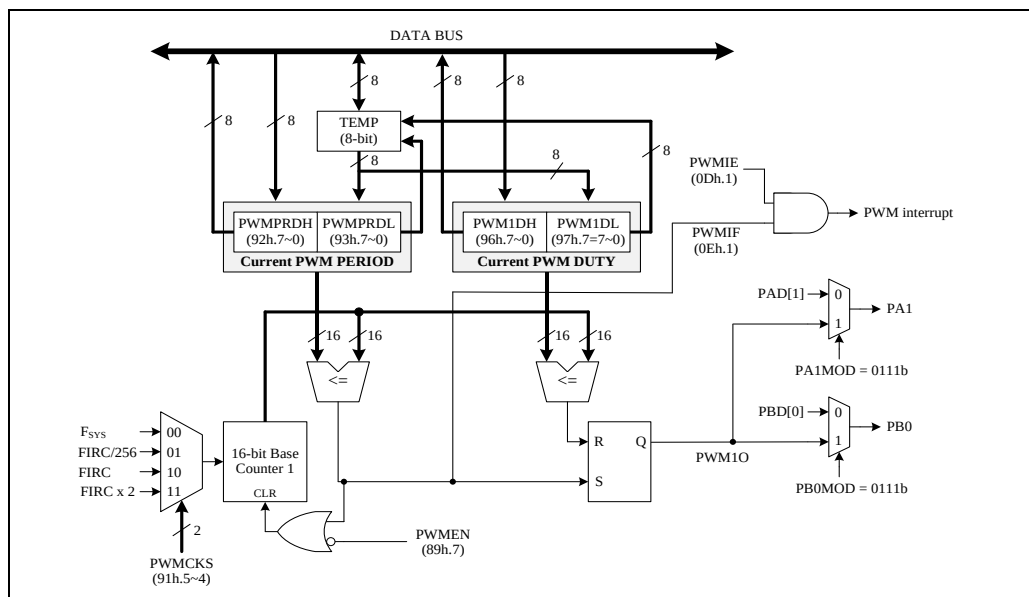
The half-bridge mode PWM is similar to the normal mode but **Dead-zone(non-overlap) is prohibited in half-bridge mode** (SFR PWM0DZ must be 0). It has two frames in a period, PWM0P only output in the first frame, PWM0N only output in the second frame. The width of these two frames must be same, so their width is the integer part of PWM0PRD/2. Because each output channel only output in one frame, the maximum duty cycle is same as the width of a frame. If the PWM0D is larger than PWM0PRD/2, H/W will force set the duty cycle to PWM0PRD/2. Following figure shows the output waveform and the output modes.



PWM0 Half-Bridge Mode Output Modes

### 6.5.2 PWM1~5

If PWMEN is cleared, the PWM0~5 will be cleared and stopped, otherwise the PWM0~5 remain running. The structure of PWM1 is shown bellow and PWM2~5 are like. The PWM1~5 duty cycle can be changed by writing to PWMxDH( $x=1\sim5$ ) and PWMxDL( $x=1\sim5$ ). The PWM1~5 output signal resets to a low level whenever the 16-bit base counter matches the 16-bit PWM1~5 duty register {PWMxDH, PWMxDL}( $x=1\sim5$ ). The PWM1~5 periods, which is shared with PWM0 period, can be set by writing the period value to the PWMPRDH and PWMPRDL registers. After writing the PWMxDH( $x=1\sim5$ ) or PWMPRDH register, H/W will update PWM period and duty immediately. PWM0~5 share an interrupt flag, and an interrupt flag is generated at the end of the period.



PWM1 Block Diagram

| 0Dh    | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| INTIE1 | —     | PCIE  | EEPIE | CMPIE | —     | —     | PWMIE | LVDIE |
| R/W    | —     | R/W   | R/W   | R/W   | —     | —     | R/W   | R/W   |
| Reset  | —     | 0     | 0     | 0     | —     | —     | 0     | 0     |

0Dh.1 **PWMIE:** PWM interrupt enable  
 0: disable  
 1: enable

| 0Eh    | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| INTIF1 | —     | PCIF  | EEPIF | CMPIF | —     | —     | PWMIF | LVDIF |
| R/W    | —     | R/W   | R/W   | R/W   | —     | —     | R/W   | R/W   |
| Reset  | —     | 0     | 0     | 0     | —     | —     | 0     | 0     |

0Eh.1 **PWMIF:** PWM interrupt event pending flag  
 This bit is set by H/W after PWM period counter roll over, write FDh to INTIF1 will clear this flag

| 89h    | Bit 7 | Bit 6  | Bit 5  | Bit 4 | Bit 3  | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|--------|--------|-------|--------|-------|-------|-------|
| PWMCTL | PWMEN | CLRPWM | PWM0OM |       | PWM0DZ |       |       |       |
| R/W    | R/W   | R/W    | R/W    |       | R/W    |       |       |       |
| Reset  | 0     | 0      | 0      | 0     | 0      | 0     | 0     | 0     |

89h.7 **PWMEN:** PWM0~5 enable  
 0: disable  
 1: enable

89h.6 **CLRPWM:** Clear PWM  
 0: not clear  
 1: clear

89h.5~4 **PWM0OM:** PWM0 output mode select  
 00: Mode0  
 01: Mode1  
 10: Mode2  
 11: Mode3

89h.3~0 **PWM0DZ:** PWM0 dead-zone(non-overlap) control  
 0000: no dead-zone(non-overlap)  
 0001: dead-zone(non-overlap) width are 1 PWM clock cycle  
 0010: dead-zone(non-overlap) width are 2 PWM clock cycles  
 ...  
 1111: dead-zone(non-overlap) width are 15 PWM clock cycles

| 91h     | Bit 7 | Bit 6 | Bit 5  | Bit 4 | Bit 3 | Bit 2   | Bit 1   | Bit 0   |
|---------|-------|-------|--------|-------|-------|---------|---------|---------|
| OPTION2 | —     | —     | PWMCKS |       | —     | INT2SEL | INT1SEL | INT0SEL |
| R/W     | —     | —     | R/W    |       | —     | R/W     | R/W     | R/W     |
| Reset   | —     | —     | 0      | 0     | —     | 0       | 0       | 0       |

91h.5~4 **PWMCKS:** PWM clock source select  
 00: Fsys  
 01: FIRC/256  
 10: FIRC (16 MHz)  
 11: FIRC x 2 (32 MHz). Refer to the graph of minimal operating voltage for PWMCKS=FIRC x 2.

| 92h     | Bit 7   | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|---------|-------|-------|-------|-------|-------|-------|-------|
| PWMPRDH | PWMPRDH |       |       |       |       |       |       |       |
| R/W     | R/W     |       |       |       |       |       |       |       |
| Reset   | 1       | 1     | 1     | 1     | 1     | 1     | 1     | 1     |

92h.7~0 **PWMPRDH**: PWM0~5 period high byte  
 write sequence: PWMPRDL then PWMPRDH  
 read sequence: PWMPRDH then PWMPRDL

| 93h     | Bit 7   | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|---------|-------|-------|-------|-------|-------|-------|-------|
| PWMPRDL | PWMPRDL |       |       |       |       |       |       |       |
| R/W     | R/W     |       |       |       |       |       |       |       |
| Reset   | 1       | 1     | 1     | 1     | 1     | 1     | 1     | 1     |

93h.7~0 **PWMPRDL**: PWM0~5 period low byte  
 write sequence: PWMPRDL then PWMPRDH  
 read sequence: PWMPRDH then PWMPRDL

| 94h    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM0DH | PWM0DH |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 1      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

94h.7~0 **PWM0DH**: PWM0 duty high byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL

| 95h    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM0DL | PWM0DL |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

95h.7~0 **PWM0DL**: PWM0 duty low byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL

| 96h    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM1DH | PWM1DH |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 1      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

96h.7~0 **PWM1DH**: PWM1 duty high byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL

| 97h    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM1DL | PWM1DL |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

97h.7~0 **PWM1DL**: PWM1 duty low byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL

| 98h    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM2DH | PWM2DH |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 1      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

98h.7~0 **PWM2DH:** PWM2 duty high byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL

| 99h    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM2DL | PWM2DL |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

99h.7~0 **PWM2DL:** PWM2 duty low byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL

| 9Ah    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM3DH | PWM3DH |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 1      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

9Ah.7~0 **PWM3DH:** PWM3 duty high byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL

| 9Bh    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM3DL | PWM3DL |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

9Bh.7~0 **PWM3DL:** PWM3 duty low byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL

| 9Ch    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM4DH | PWM4DH |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 1      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

9Ch.7~0 **PWM4DH:** PWM4 duty high byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL

| 9Dh    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM4DL | PWM4DL |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

9Dh.7~0 **PWM4DL:** PWM4 duty low byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL



| 9Eh    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM5DH | PWM5DH |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 1      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

9Eh.7~0 **PWM5DH:** PWM5 duty high byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL

| 9Fh    | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|
| PWM5DL | PWM5DL |       |       |       |       |       |       |       |
| R/W    | R/W    |       |       |       |       |       |       |       |
| Reset  | 0      | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

9Fh.7~0 **PWM5DL:** PWM5 duty low byte  
 write sequence: PWMxDL then PWMxDH  
 read sequence: PWMxDH then PWMxDL

| 107h    | Bit 7   | Bit 6    | Bit 5    | Bit 4    | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|---------|----------|----------|----------|-------|-------|-------|-------|
| PWMCTL2 | PWM0MOD | PWM0MSKE | PWM0NMSK | PWM0PMSK | —     | —     | —     | —     |
| R/W     | R/W     | R/W      | R/W      | R/W      | —     | —     | —     | —     |
| Reset   | 0       | 0        | 0        | 0        | —     | —     | —     | —     |

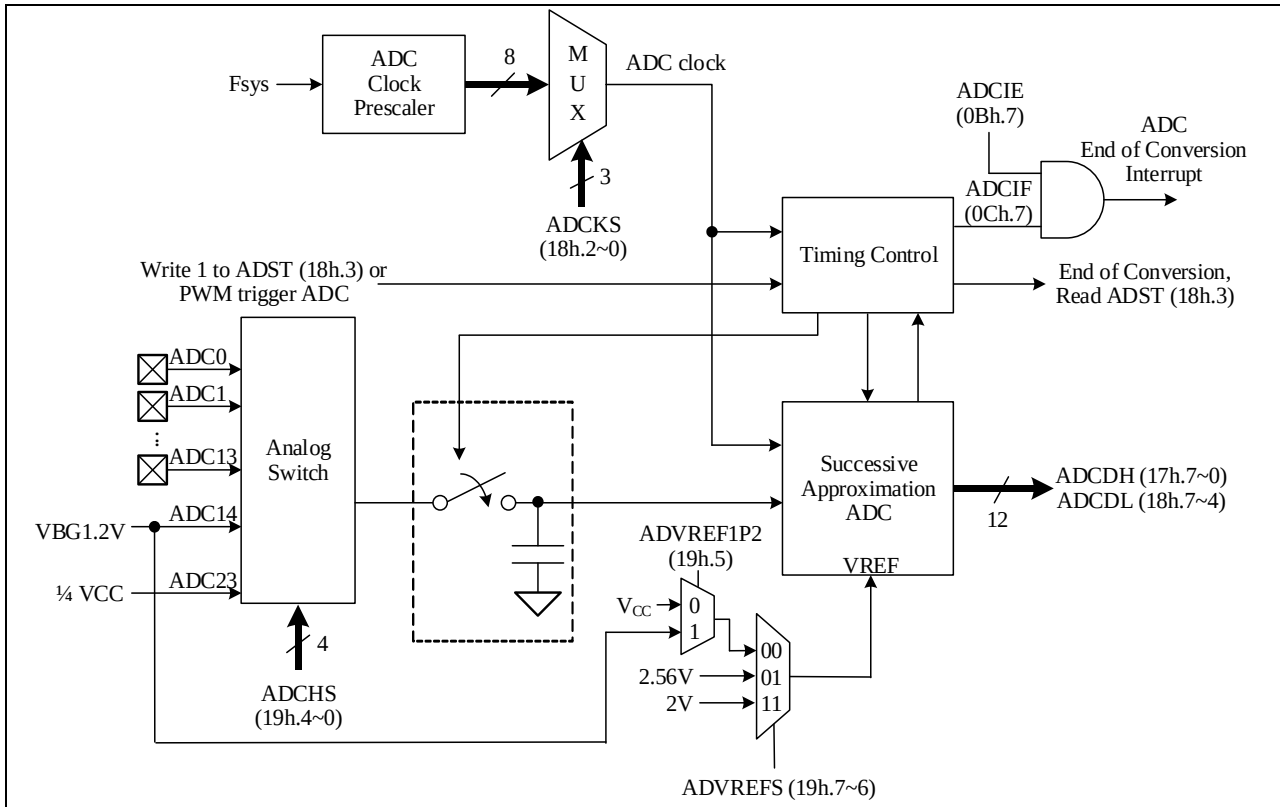
107h.7 **PWM0MOD:** PWM0 mode select  
 0: Normal mode  
 1: Half-bridge mode

107h.6 **PWM0MSKE:** PWM0 mask output enable  
 0: Disable  
 1: Enable, PWM0P/PWM0N output data by PWM0PMSK/PWM0NMSK while CLRPWM=1

107h.5 **PWM0NMSK:** PWM0N mask data  
 If CLRPWM=1 and PWM0MSKE=1, PWM0N will output this mask data.

107h.4 **PWM0PMSK:** PWM0P mask data  
 If CLRPWM=1 and PWM0MSKE=1, PWM0P will output this mask data.

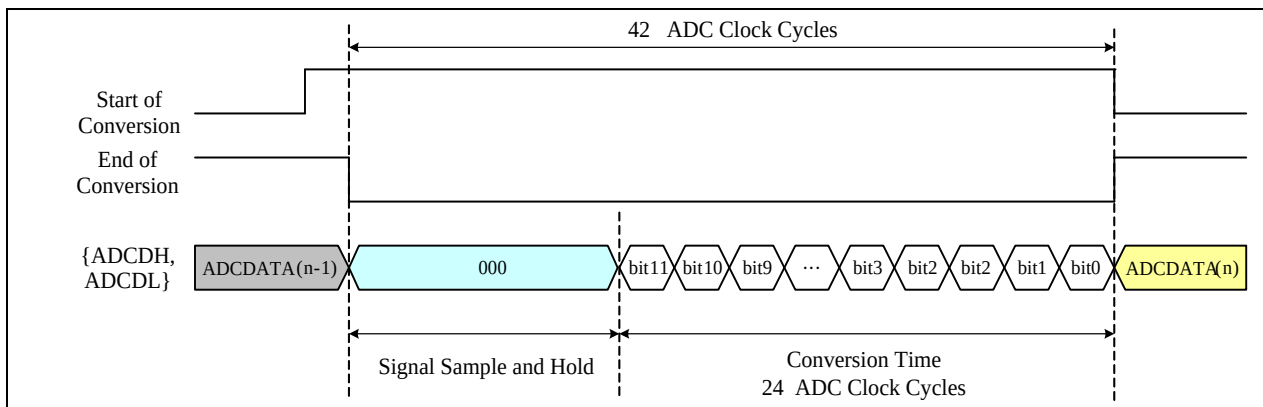
## 6.6 Analog-to-Digital Converter



**ADC Block Diagram**

### 6.6.1 Normal Trigger ADC

The 12-bit ADC (Analog to Digital Converter) consists of a 16-channel analog input multiplexer, control register, clock generator, 12-bit successive approximation register, and output data register. To use the ADC, user needs to set **ADCKS** (18h.2~0) to choose a proper ADC clock frequency, which must be less than 1 MHz. User then launches the ADC conversion by setting the **ADSTEOC** (18h.3) control bit. After end of conversion, H/W automatic clears the **ADSTEOC** (18h.3) bit. User can poll this bit to know the conversion status. When the IO pin is used as the ADC input pin, the corresponding pin mode should be set to 0011b. User needs to set **ADCHS** (19h.4~0) to choose the input channel of ADC. Besides, there are some reference input channel can be selected, **ADC14** is  $V_{BG}$  and **ADC23** is  $1/4V_{CC}$  for ADC. ADC reference voltage can be configured as  $V_{CC}$  or  $V_{BG}$  by **ADVREFS** (19h.7~6). Furthermore, if **ADVREFS** is changed to 2.56V or 2V, it will need 200uS warm-up stable time. When **ADCHS** is selected to  $V_{BG}$ , **ADVREFS** must be set to  $V_{CC}$ , otherwise ADC conversion will be invalid.



Example:

[CPU running at FAST mode , Fsys = FIRC 16 MHz ]  
ADC clock frequency = 1 MHz, ADC channel = ADC2 (PA2).

◇ Example:

```

MOV LW    xxxx0011b          ; ADC2 (PA2) as ADC input
MOV WX    PAMOD32

MOV LW    00000100b          ; ADCKS = Fsys/16, ADC clock = 1 MHz
MOV WX    ADCTL

MOV LW    00000010b          ; ADC reference voltage select VCC
MOV WX    ADCTL2              ; ADC input channel select ADC2

BSX       ADSTEOC              ; 18h.3 (ADSTEOC), ADC start conversion.
  
```

WAIT\_ADC:

```

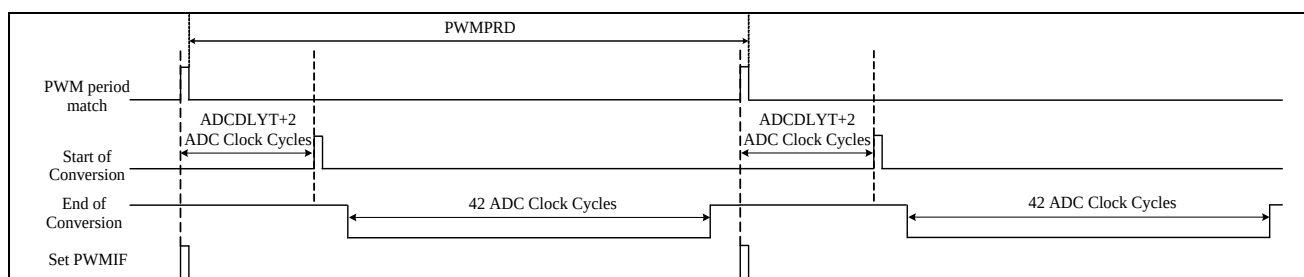
BTX SC    ADSTEOC              ; Wait ADC conversion finish.
LGOTO     WAIT_ADC

MOVXW     ADCDH                ; Read ADC output data bit 11~4
MOVXW     ADCTL                ; Read ADC output data bit 3~0
...
  
```

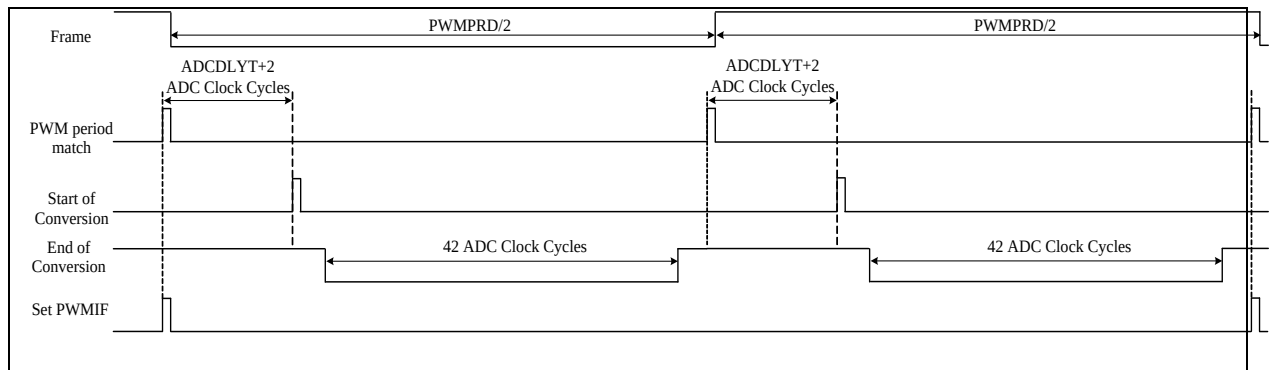
## 6.6.2 PWM Trigger ADC

The mechanism of PWM trigger ADC could be enabled by mean of non-zero ADCDLYT register. When PWM counter matches period register, hardware will start ADC conversion automatically after some delay time if the mechanism of PWM trigger ADC is enabled.

### 6.6.2.1 Normal PWM Trigger ADC

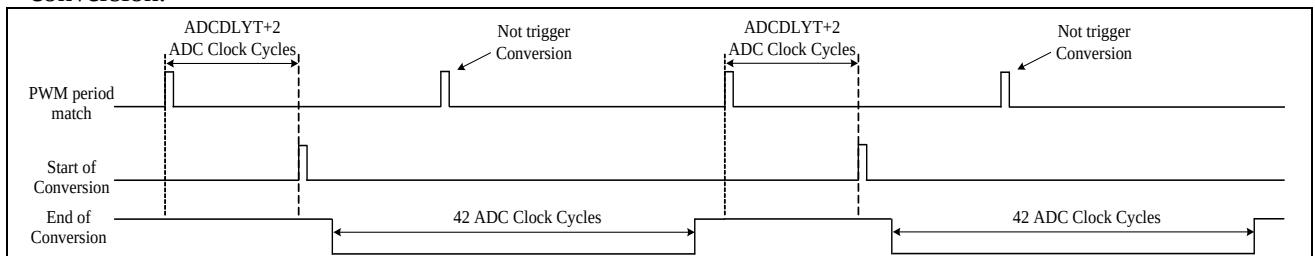


### 6.6.2.2 Half-Bridge PWM Trigger ADC



### 6.6.2.3 Unconverted Situation

When ADC conversion is under progress, PWM period matching event will not trigger new ADC conversion.



| 0Bh   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIE | ADCIE | T2IE  | TM1IE | TM0IE | WKTIE | INT2IE | INT1IE | INT0IE |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

0Bh.7 **ADCIE**: ADC interrupt enable  
 0: disable  
 1: enable

| 0Ch   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2  | Bit 1  | Bit 0  |
|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| INTIF | ADCIF | T2IF  | TM1IF | TM0IF | WKTIF | INT2IF | INT1IF | INT0IF |
| R/W   | R/W   | R/W   | R/W   | R/W   | R/W   | R/W    | R/W    | R/W    |
| Reset | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      |

0Ch.7 **ADCIF**: ADC interrupt event pending flag  
 This bit is set by H/W after ADC end of conversion, write 7Fh to INTIF will clear this flag

| 17h   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| ADCDH | ADCDH |       |       |       |       |       |       |       |
| R/W   | R     |       |       |       |       |       |       |       |
| Reset | —     | —     | —     | —     | —     | —     | —     | —     |

17h.7~0 **ADCDH**: ADC output data bit 11~4

| 18h   | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3   | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|---------|-------|-------|-------|
| ADCTL | ADCDL |       |       |       | ADSTEOC | ADCKS |       |       |
| R/W   | R     |       |       |       | R/W     | R/W   |       |       |
| Reset | —     | —     | —     | —     | 0       | 0     | 0     | 0     |

18h.7~4 **ADCDL**: ADC output data bit 3~0

18h.3 **ADSTEOC(W)**: ADC start bit.

0: H/W clear after end of conversion

1: ADC start conversion

Don't write ADSTEOC when PWM trigger ADC mode (i.e. ADCDLYT != 0) is utilized.

**ADSTEOC(R)**: ADC start or end of conversion bit.

Normal mode(i.e. ADCDLYT = 0): ADC start

0: H/W clear after end of conversion

1: ADC start conversion

PWM trigger ADC mode(i.e. ADCDLYT != 0): ADC end of conversion

0: ADC is in conversion

1: ADC is end of conversion

18h.2~0 **ADCKS**: ADC clock frequency selection:

000: Fsys/256    100: Fsys/16

001: Fsys/128    101: Fsys/8

010: Fsys/64    110: Fsys/4

011: Fsys/32    111: Fsys/2

| 19h    | Bit 7   | Bit 6 | Bit 5     | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|---------|-------|-----------|-------|-------|-------|-------|-------|
| ADCTL2 | ADVREFS |       | ADVREF1P2 | ADCHS |       |       |       |       |
| R/W    | R/W     |       | R/W       | R/W   |       |       |       |       |
| Reset  | 0       | 0     | —         | 1     | 1     | 1     | 1     | 1     |

19h.7~6 **ADVREFS**: ADC reference voltage and  $V_{BG}$  output voltage select

00: ADC reference voltage is  $V_{CC}$  or 1.2V  $V_{BG}$  according to the value of ADVREF1P2.  $V_{BG}$  is 1.20V

01: ADC reference voltage is  $V_{BG}$ ,  $V_{BG}$  is 2.56V

10: Reserved

11: ADC reference voltage is  $V_{BG}$ ,  $V_{BG}$  is 2.00V(This feature can't not be emulated)(Don't use for the selection of DAC's VREF)

19h.5 **ADVREF1P2**: ADC 1.2V reference voltage select

0: ADC reference voltage is  $V_{CC}$  when ADVREFS=00.  $V_{BG}$  is 1.2V

1: ADC reference voltage is 1.2V  $V_{BG}$  when ADVREFS=00.  $V_{BG}$  is 1.2V(This feature can't not be emulated)

19h.4~0 **ADCHS**: ADC channel select

00000: ADC0 (PA0)    01000: ADC8 (PB1)    others: Reserved

00001: ADC1 (PA1)    01001: ADC9 (PB2)

00010: ADC2 (PA2)    01010: ADC10 (PB4)

00011: ADC3 (PA3)    01011: ADC11 (PB5)

00100: ADC4 (PA4)    01100: ADC12 (PB6)

00101: ADC5 (PA5)    01101: ADC13 (PB7)

00110: ADC6 (PA6)    01110: VBG

00111: ADC7 (PB0)    10111: 1/4 VCC

| 193h    | Bit 7   | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|---------|-------|-------|-------|-------|-------|-------|-------|
| ADCDLYT | ADCDLYT |       |       |       |       |       |       |       |
| R/W     | R/W     |       |       |       |       |       |       |       |
| Reset   | 0       | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

193h.7~0 **ADCDLYT**: The delay time between PWM period match and ADC start

8'h00: disable PWM trigger ADC start

8'h01~8'hFF: enable and delay time((ADCDLT[7:0]+2)/Fadc) to ADC start

## 6.7 Comparator

There is a Comparator (CMP) in this device.

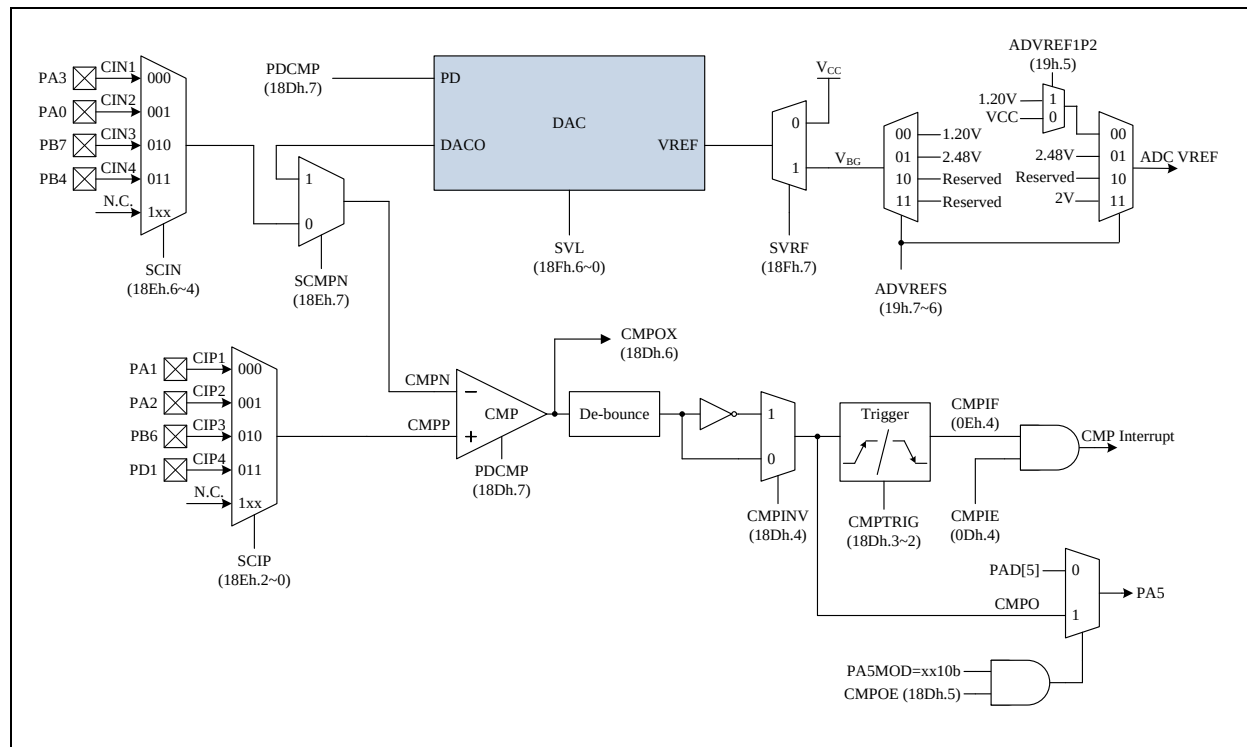
The CMP built in a 7-bit DAC module, which output can be accessed to negative input port of the CMP.

Reference Voltage of DAC can be selected as  $V_{CC}$  or  $V_{BG}$  by setting SVRF (18Fh.7).  $V_{BG}$  will be configured as 1.20V or 2.56V by setting ADVREFS (19h.7~6). A suitable level of voltage can be selected for proper operation of user application by setting SVL (18Fh.6~0), which will change the resistance to transform the value of voltage. Setting the PDCMP=1 (18Dh.7) will let DAC and CMP enter power down mode. By configuring SCMPN (18Eh.7), negative port input source will be external pin input or DAC output. And positive port input source is external pin input. The SCIN (18Eh.6~4) and SCIP (18Eh.2~0) register determine negative and positive port external input source respectively.

Because the input module of the CMP is composed of PMOS, the input voltage range will be affected by  $V_{th}$  of the PMOS. Thus, the maximum input voltage of the CMP will be  $(V_{CC}-0.5)$  V. Meanwhile, the Comparator's hysteresis voltage is about 30mV.

The Comparator original output (CMPOX) can be read by CMPOX (18Dh.6) bit. The Chip provides a de-bounce module to de-bounce the CMPOX signal, user can select de-bounce time by CMPDBS (18Dh.1~0). The de-bounce output signal can select invert or not by CMPINV (18Dh.4) to generate CMPO signal. The CMPO can be output to pin (PA5) by set CMPOE (18Dh.5) and the PA5MOD should be set to xx10b.

The CMPO is also a trigger source for the interrupt trigger module to generate interrupt flag CMPIF (0Eh.4). The trigger mode is selected by CMPTRIG (18Dh.3~2). When Comparator power down, the interrupt flag will still be produced. Therefore, it is necessary to clear the interrupt flag first after turning on the CMP module each time to prevent using the dummy flag.



| 0Dh    | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| INTIE1 | —     | PCIE  | EEPIE | CMPIE | —     | —     | PWMIE | LVDIE |
| R/W    | —     | R/W   | R/W   | R/W   | —     | —     | R/W   | R/W   |
| Reset  | —     | 0     | 0     | 0     | —     | —     | 0     | 0     |

0Dh.4 **CMPIE:** Comparator interrupt enable

0: disable  
1: enable

| 0Eh    | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| INTIF1 | –     | PCIF  | EEPIF | CMPIF | –     | –     | PWMIF | LVDIF |
| R/W    | –     | R/W   | R/W   | R/W   | –     | –     | R/W   | R/W   |
| Reset  | –     | 0     | 0     | 0     | –     | –     | 0     | 0     |

0Eh.4 **CMPIF**: Comparator interrupt event pending flag

This bit is set by H/W while CMPO match trigger condition, write EFh to INTIF1 will clear this flag

| 19h    | Bit 7   | Bit 6 | Bit 5     | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|--------|---------|-------|-----------|-------|-------|-------|-------|-------|
| ADCTL2 | ADVREFS |       | ADVREF1P2 | ADCBS |       |       |       |       |
| R/W    | R/W     |       | R/W       | R/W   |       |       |       |       |
| Reset  | 0       | 0     | –         | 1     | 1     | 1     | 1     | 1     |

19h.7~6 **ADVREFS**: ADC reference voltage and  $V_{BG}$  output voltage select

00: ADC reference voltage is  $V_{CC}$  or 1.2V  $V_{BG}$  according to the value of ADVREF1P2.  $V_{BG}$  is 1.20V

01: ADC reference voltage is  $V_{BG}$ ,  $V_{BG}$  is 2.56V

10: Reserved

11: ADC reference voltage is  $V_{BG}$ ,  $V_{BG}$  is 2V(This feature can't not be emulated) (Don't use for the selection of DAC's VREF)

| 18Dh   | Bit 7 | Bit 6 | Bit 5 | Bit 4  | Bit 3   | Bit 2 | Bit 1  | Bit 0 |
|--------|-------|-------|-------|--------|---------|-------|--------|-------|
| CMPCTL | PDCMP | CMPOX | CMPOE | CMPINV | CMPTRIG |       | CMPDBS |       |
| R/W    | R/W   | R     | R/W   | R/W    | R/W     |       | R/W    |       |
| Reset  | 1     | 1     | 0     | 0      | 0       | 0     | 0      | 0     |

18Dh.7 **PDCMP**: Comparator & DAC power down enable control

0: disable Comparator & DAC power down

1: enable Comparator & DAC power down

18Dh.6 **CMPOX**: Comparator original output (CMPOX) status

0:  $V_{CMPN} < V_{CMPN}$

1:  $V_{CMPN} > V_{CMPN}$  or PDCMP =1

18Dh.5 **CMPOE**: Comparator output (CMPO) signal output to PA5

0: disable

1: enable, PA5MOD should be set to xx10b

18Dh.4 **CMPINV**: Comparator de-bounce output invert select

0: no invert

1: invert

18Dh.3~2 **CMPTRIG**: Comparator interrupt trigger mode

00: Rising edge

01: Falling edge

10: Both edge

11: High level

18Dh.1~0 **CMPDBS**: Comparator original output (CMPOX) de-bounce time

00: none

01: 4 Fsys

10: 8 Fsys

11: 16 Fsys

| 18Eh    | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---------|-------|-------|-------|-------|-------|-------|-------|-------|
| CMP PNS | SCMPN | SCIN  |       |       | –     | SCIP  |       |       |
| R/W     | R/W   | R/W   |       |       | –     | R/W   |       |       |
| Reset   | 1     | 1     | 1     | 1     | –     | 1     | 1     | 1     |

- 18Eh.7 **SCMPN:** Comparator CMPN source select  
 0: Comparator CMPN source is external input (CINx)  
 1: Comparator CMPN source is DAC output

- 18Eh.6~4 **SCIN:** Comparator CMPN external input select  
 000: Comparator CMPN external input is CIN1 (PA3)  
 001: Comparator CMPN external input is CIN2 (PA0)  
 010: Comparator CMPN external input is CIN3 (PB7)  
 011: Comparator CMPN external input is CIN4 (PB4)  
 1xx: No connect

- 18Eh.2~0 **SCIP:** Comparator CMPP external input select  
 000: Comparator CMPP external input is CIP1 (PA1)  
 001: Comparator CMPP external input is CIP2 (PA2)  
 010: Comparator CMPP external input is CIP3 (PB6)  
 011: Comparator CMPP external input is CIP4 (PD1)  
 1xx: No connect

| 18Fh  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| DACTL | SVRF  | SVL   |       |       |       |       |       |       |
| R/W   | R/W   | R/W   |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

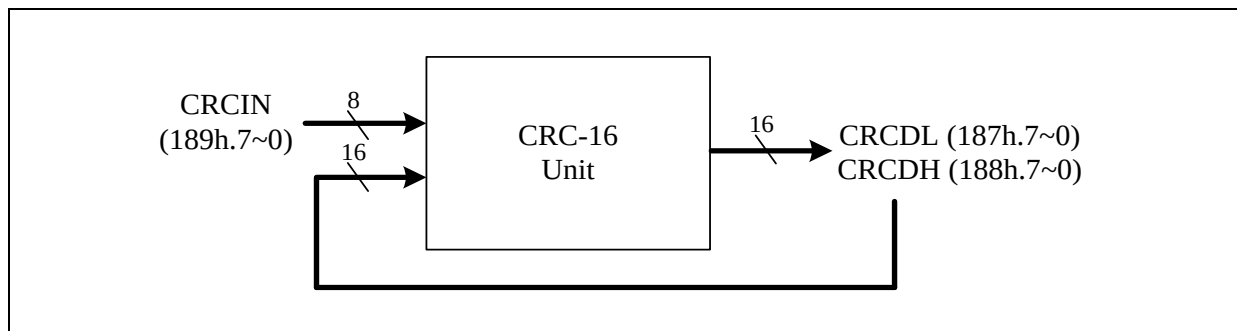
- 18Fh.7 **SVRF:** DAC reference voltage select  
 0:  $V_{CC}$   
 1:  $V_{BG}$  (voltage level is selected by ADVREFS)

- 18Fh.6~0 **SVL:** DAC output voltage select (reference source can be selected as  $V_{CC}$  or  $V_{BG}$ )  
 000\_0000: 0/128 \* reference source  
 000\_0001: 1/128 \* reference source  
 ...  
 111\_1101: 125/128 \* reference source  
 111\_1110: 126/128 \* reference source  
 111\_1111: 127/128 \* reference source



## 6.8 Cyclic Redundancy Check (CRC)

The chip supports an integrated 16-bit Cyclic Redundancy Check function. The Cyclic Redundancy Check (CRC) calculation unit is an error detection technique test algorithm and uses to verify data transmission or storage data correctness. The CRC calculation takes an 8-bit data stream or a block of data as input and generates a 16-bit output remainder. The data stream is calculated by the same generator polynomial.



**CRC16 Block Diagram**

The CRC generator provides the 16-bit CRC result calculation based on the CRC-16-IBM polynomial. In this CRC generator, there is only one polynomial available for the numeric values calculation. It can't support the 16-bit CRC calculations based on any other polynomials. Each write operation to the CRCIN register creates a combination of the previous CRC value stored in the CRCDL and CRCDH registers. It will take one MCU instruction cycle to calculate.

**CRC-16-IBM (Modbus) Polynomial representation:**  $X^{16} + X^{15} + X^2 + 1$

| 187h  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| CRCDL | CRCDL |       |       |       |       |       |       |       |
| R/W   | R/W   |       |       |       |       |       |       |       |
| Reset | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |

187h.7~0 **CRCDL:** 16-bit CRC checksum data bit 7~0

| 188h  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| CRCDH | CRCDH |       |       |       |       |       |       |       |
| R/W   | R/W   |       |       |       |       |       |       |       |
| Reset | 1     | 1     | 1     | 1     | 1     | 1     | 1     | 1     |

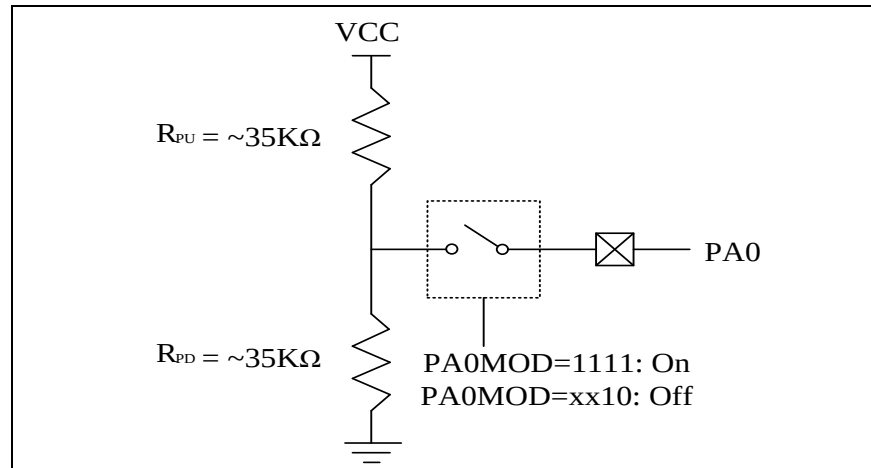
188h.7~0 **CRCDH:** 16-bit CRC checksum data bit 15~8

| 189h  | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| CRCIN | CRCIN |       |       |       |       |       |       |       |
| W     | W     |       |       |       |       |       |       |       |
| Reset | –     | –     | –     | –     | –     | –     | –     | –     |

189h.7~0 **CRCIN:** CRC data input, write this register to start CRC calculation

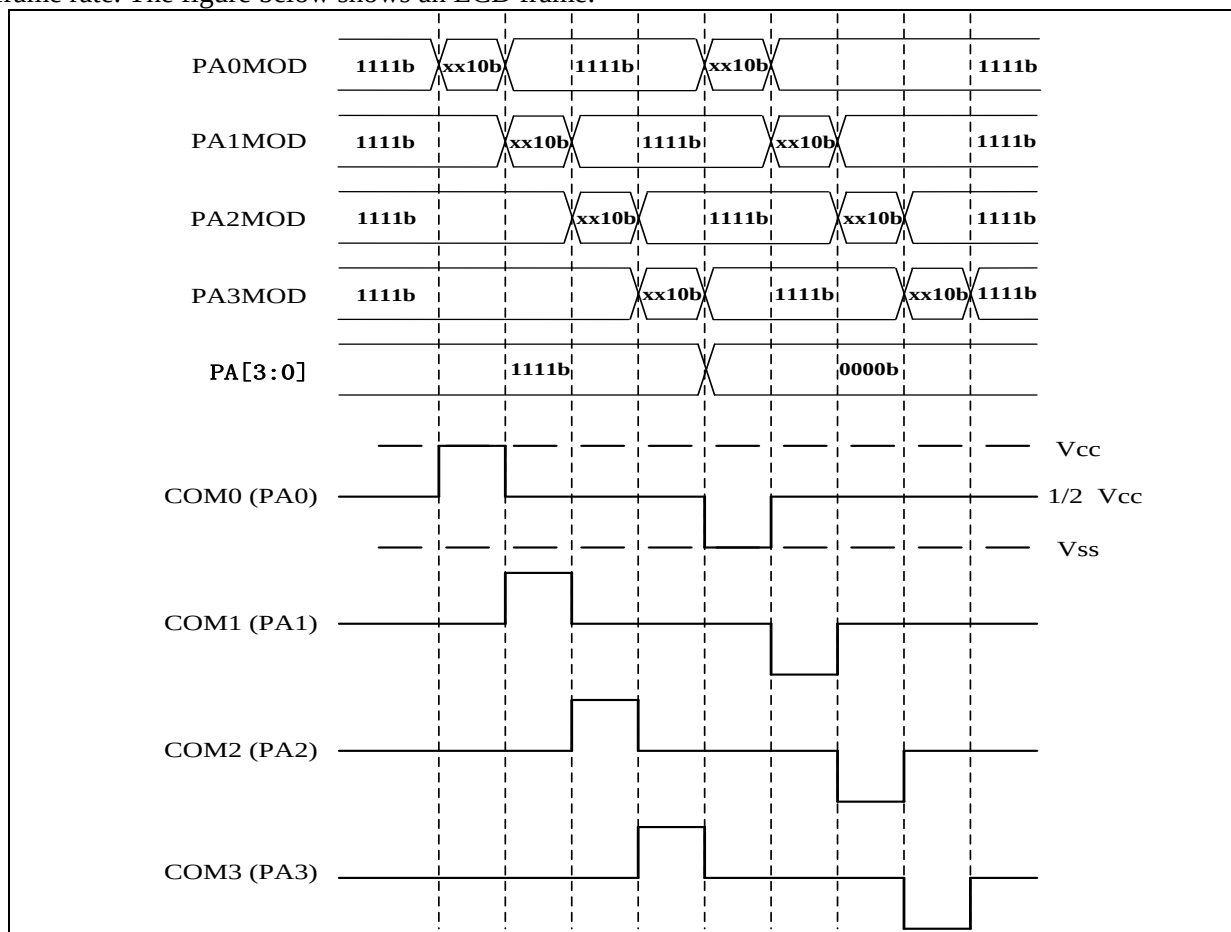
## 6.9 S/W Control LCD Driver

The chip support an S/W controlled method to driving LCD. All IO pins except PA7 could be the Common Pins. Common pins are capable of driving 1/2 bias by setting the register PAxMOD/PBxMOD/PDxMOD. The COM0 circuit is shown below.

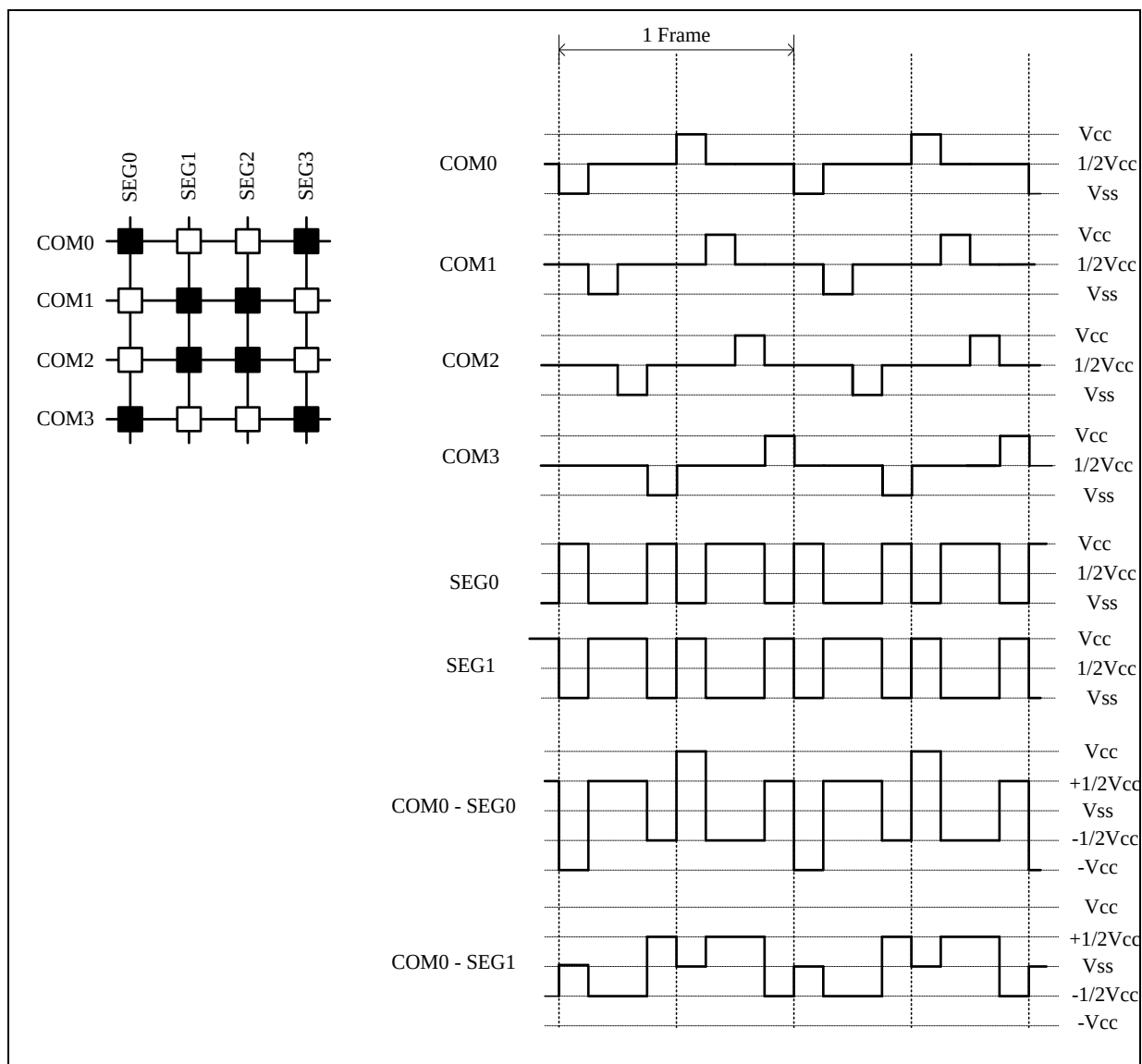


LCD COM0 Circuit

The frequency of any repeating waveform output on the COM pin can be used to represent the LCD frame rate. The figure below shows an LCD frame.



S/W Controlled LCD COM0 ~ COM3 Scanning



**1/4 Duty, 1/2 Bias Output Waveform**

## MEMORY MAP

| Name                       | Address | R/W | Rst | Description                                                                                                            |
|----------------------------|---------|-----|-----|------------------------------------------------------------------------------------------------------------------------|
| INDF (00h/80h/100h/180h)   |         |     |     | Function related to: RAM W/R                                                                                           |
| INDF                       | 00.7~0  | R/W | -   | Not a physical register, addressing INDF actually point to the register whose address is contained in the FSR register |
| TM0 (01h/101h)             |         |     |     | Function related to: Timer0                                                                                            |
| TM0                        | 01.7~0  | R/W | 00  | Timer0 content                                                                                                         |
| PCL (02h/82h/102h/182h)    |         |     |     | Function related to: PROGRAM COUNT                                                                                     |
| PCL                        | 02.7~0  | R/W | 00  | Programming Counter data bit 7~0                                                                                       |
| STATUS (03h/83h/103h/183h) |         |     |     | Function related to: STATUS                                                                                            |
| IRP                        | 03.7    | R/W | 0   | Register Bank Select bit (used for indirect addressing)                                                                |
| RP1                        | 03.6    | R/W | 0   | Register Bank Select bit 1 for direct addressing                                                                       |
| RP0                        | 03.5    | R/W | 0   | Register Bank Select bit 0 for direct addressing                                                                       |
| TO                         | 03.4    | R   | 0   | WDT timeout flag, cleared by PWRST, 'SLEEP' or 'CLRWDT' instruction                                                    |
| PD                         | 03.3    | R   | 0   | Power down flag, set by 'SLEEP', cleared by 'CLRWDT' instruction                                                       |
| Z                          | 03.2    | R/W | 0   | Zero flag                                                                                                              |
| DC                         | 03.1    | R/W | 0   | Decimal Carry flag                                                                                                     |
| C                          | 03.0    | R/W | 0   | Carry flag                                                                                                             |
| FSR (04h/84h/104h/184h)    |         |     |     | Function related to: RAM W/R                                                                                           |
| FSR                        | 04.7~0  | R/W | -   | File Select Register, indirect address mode pointer                                                                    |
| PAD (05h)                  |         |     |     | Function related to: Port A                                                                                            |
| PAD                        | 05.7~0  | R   | -   | Port A pin or "data register" state                                                                                    |
|                            |         | W   | FF  | Port A output data register                                                                                            |
| PBD (06h)                  |         |     |     | Function related to: Port B                                                                                            |
| PBD                        | 06.7~0  | R   | -   | Port B pin or "data register" state                                                                                    |
|                            |         | W   | FF  | Port B output data register                                                                                            |
| PDD (07h)                  |         |     |     | Function related to: Port D                                                                                            |
| PDD                        | 07.1~0  | R   | -   | Port D pin or "data register" state                                                                                    |
|                            |         | W   | 3   | Port D output data register                                                                                            |
| PCLATH (0Ah/8Ah/10Ah/18Ah) |         |     |     | Function related to: PROGRAM COUNT                                                                                     |
| GPR                        | 0A.7~4  | R/W | 0   | General Purpose Register                                                                                               |
| PCLATH                     | 0A.3~0  | R/W | 0   | Write Buffer for the high byte of the Program Counter                                                                  |
| INTIE (0Bh/8Bh/10Bh/18Bh)  |         |     |     | Function related to: Interrupt Enable                                                                                  |
| ADCIE                      | 0B.7    | R/W | 0   | ADC interrupt enable<br>0: disable<br>1: enable                                                                        |
| T2IE                       | 0B.6    | R/W | 0   | T2 interrupt enable<br>0: disable<br>1: enable                                                                         |
| TM1IE                      | 0B.5    | R/W | 0   | Timer1 interrupt enable<br>0: disable<br>1: enable                                                                     |
| TM0IE                      | 0B.4    | R/W | 0   | Timer0 interrupt enable<br>0: disable<br>1: enable                                                                     |
| WKTIE                      | 0B.3    | R/W | 0   | Wake-up Timer interrupt enable and Wake-up Timer enable<br>0: disable<br>1: enable                                     |
| INT2IE                     | 0B.2    | R/W | 0   | INT2 pin (PA7 or PB5) interrupt enable<br>0: disable<br>1: enable                                                      |
| INT1IE                     | 0B.1    | R/W | 0   | INT1 pin (PA1 or PB1) interrupt enable<br>0: disable<br>1: enable                                                      |

| Name         | Address | R/W | Rst | Description                                                                                                                                                                                                      |
|--------------|---------|-----|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INT0IE       | 0B.0    | R/W | 0   | INT0 pin (PA3 or PB2) interrupt enable<br>0: disable<br>1: enable                                                                                                                                                |
| INTIF (0Ch)  |         |     |     | Function related to: Interrupt Flag                                                                                                                                                                              |
| ADCIF        | 0C.7    | R   | -   | ADC interrupt flag, set by H/W after ADC end of conversion                                                                                                                                                       |
|              |         | W   | 0   | write 7Fh to INTIF will clear this flag                                                                                                                                                                          |
| T2IF         | 0C.6    | R   | -   | T2 interrupt event pending flag, set by H/W while T2 overflows                                                                                                                                                   |
|              |         | W   | 0   | write BFh to INTIF will clear this flag                                                                                                                                                                          |
| TM1IF        | 0C.5    | R   | -   | Timer1 interrupt event pending flag, set by H/W while Timer1 overflows                                                                                                                                           |
|              |         | W   | 0   | write DFh to INTIF will clear this flag                                                                                                                                                                          |
| TM0IF        | 0C.4    | R   | -   | Timer0 interrupt event pending flag, set by H/W while Timer0 overflows                                                                                                                                           |
|              |         | W   | 0   | write EFh to INTIF will clear this flag                                                                                                                                                                          |
| WKTIF        | 0C.3    | R   | -   | WKT interrupt event pending flag, set by H/W while WKT time out                                                                                                                                                  |
|              |         | W   | 0   | write F7h to INTIF will clear this flag                                                                                                                                                                          |
| INT2IF       | 0C.2    | R   | -   | INT2 (PA7 or PB5) interrupt event pending flag, set by H/W at INT2 pin's falling edge                                                                                                                            |
|              |         | W   | 0   | write FBh to INTIF will clear this flag                                                                                                                                                                          |
| INT1IF       | 0C.1    | R   | -   | INT1 (PA1 or PB1) interrupt event pending flag, set by H/W at INT1 pin's falling/rising edge                                                                                                                     |
|              |         | W   | 0   | write FDh to INTIF will clear this flag                                                                                                                                                                          |
| INT0IF       | 0C.0    | R   | -   | INT0 (PA3 or PB2) interrupt event pending flag, set by H/W at INT0 pin's falling/rising edge                                                                                                                     |
|              |         | W   | 0   | write FEh to INTIF will clear this flag                                                                                                                                                                          |
| INTIE1 (0Dh) |         |     |     | Function related to: Interrupt Enable                                                                                                                                                                            |
| PCIE         | 0D.6    | R/W | 0   | All port pin-change wake-up interrupt enable<br>0: disable<br>1: enable                                                                                                                                          |
| EEPIE        | 0D.5    | R/W | 0   | EEPROM interrupt enable<br>0: disable<br>1: enable                                                                                                                                                               |
| CMPIE        | 0D.4    | R/W | 0   | Comparator interrupt enable<br>0: disable<br>1: enable                                                                                                                                                           |
| PWMIE        | 0D.1    | R/W | 0   | PWM interrupt enable<br>0: disable<br>1: enable                                                                                                                                                                  |
| LVDIE        | 0D.0    | R/W | 0   | LVD interrupt enable<br>0: disable<br>1: enable                                                                                                                                                                  |
| INTIF1 (0Eh) |         |     |     | Function related to: Interrupt Flag                                                                                                                                                                              |
| PCIF         | 0E.6    | R   | -   | All port pin-change wake-up interrupt event pending flag, set by H/W at all pin's falling/rising edge. A sleep instruction is necessary before the event of pin-change otherwise pin-change event may be missed. |
|              |         | W   | 0   | write BFh to INTIF1 will clear this flag                                                                                                                                                                         |
| EEPIF        | 0E.5    | R   | -   | EEPROM interrupt event pending flag, set by H/W when EEPROM writing is finishing or time-out event occurred.                                                                                                     |
|              |         | W   | 0   | write DFh to INTIF1 will clear this flag                                                                                                                                                                         |
| CMPIF        | 0E.4    | R   | -   | Comparator interrupt event pending flag, set by H/W while CMPO match trigger condition                                                                                                                           |
|              |         | W   | 0   | write EFh to INTIF1 will clear this flag                                                                                                                                                                         |
| PWMIF        | 0E.1    | R   | -   | PWM interrupt event pending flag, set by H/W after PWM period counter roll over                                                                                                                                  |
|              |         | W   | 0   | write FDh to INTIF1 will clear this flag                                                                                                                                                                         |
| LVDIF        | 0E.0    | R   | -   | LVD interrupt event pending flag, set by H/W while $V_{CC} < V_{LVD}$                                                                                                                                            |

| Name                | Address | R/W | Rst | Description                                                                                                                                                                  |
|---------------------|---------|-----|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     |         | W   | 0   | write FEh to INTIF1 will clear this flag                                                                                                                                     |
| <b>CLKCTL (0Fh)</b> |         |     |     | <b>Function related to: Fsys</b>                                                                                                                                             |
| SCKTYPE             | 0F.7    | R/W | 0   | Slow Clock Type. This bit could only be changed in Fast mode (CPUCKS=1)<br>0: SIRC<br>1: SXT. PA4 and PA5 are crystal pins.                                                  |
| FCKTYPE             | 0F.6    | R/W | 0   | Fast Clock Type. This bit could only be changed in Slow mode (CPUCKS=0)<br>0: FIRC<br>1: FXT. PA4 and PA5 are crystal pins. FXT oscillator gain is higher than that of SXT.  |
| SLOWSTP             | 0F.4    | R/W | 0   | Stop Slow-clock after execute SLEEP instruction<br>0: Slow-clock keeps running after execute SLEEP instruction<br>1: Slow-clock stop running after execute SLEEP instruction |
| FASTSTP             | 0F.3    | R/W | 1   | Stop Fast-clock<br>0: Fast-clock is running<br>1: Fast-clock stops running                                                                                                   |
| CPUCKS              | 0F.2    | R/W | 0   | System clock source select<br>0: Slow-clock<br>1: Fast-clock                                                                                                                 |
| CPUPSC              | 0F.1~0  | R/W | 11  | System clock source prescaler. System clock source<br>00: div 8 01: div 4 10: div 2 11: div 1                                                                                |
| <b>TM0RLD (10h)</b> |         |     |     | <b>Function related to: Timer0</b>                                                                                                                                           |
| TM0RLD              | 10.7~0  | R/W | 00  | Timer0 reload data                                                                                                                                                           |
| <b>TM0CTL (11h)</b> |         |     |     | <b>Function related to: Timer0</b>                                                                                                                                           |
| TM0STP              | 11.6    | R/W | 0   | Stop Timer0<br>0: Timer0 runs<br>1: Timer0 stops                                                                                                                             |
| TM0EDG              | 11.5    | R/W | 0   | TM0CKI (PA2) edge<br>0: rising edge<br>1: falling edge                                                                                                                       |
| TM0CKS              | 11.4    | R/W | 0   | Timer0 prescaler clock source<br>0: Fsys/2<br>1: TM0CKI (PA2)                                                                                                                |
| TM0PSC              | 11.3~0  | R/W | 0   | Timer0 prescaler. Timer0 prescaler clock source divided by<br>0000: 1 0011: 8 0110: 64<br>0001: 2 0100: 16 0111: 128<br>0010: 4 0101: 32 1xxx: 256                           |
| <b>TM1 (12h)</b>    |         |     |     | <b>Function related to: Timer1</b>                                                                                                                                           |
| TM1                 | 12.7~0  | R/W | 00  | Timer1 content                                                                                                                                                               |
| <b>TM1RLD (13h)</b> |         |     |     | <b>Function related to: Timer1</b>                                                                                                                                           |
| TM1RLD              | 13.7~0  | R/W | 00  | Timer1 reload data                                                                                                                                                           |
| <b>TM1CTL (14h)</b> |         |     |     | <b>Function related to: Timer1</b>                                                                                                                                           |
| TM1STP              | 14.6    | R/W | 0   | Stop Timer1<br>0: Timer1 runs<br>1: Timer1 stops                                                                                                                             |
| TM1PSC              | 14.3~0  | R/W | 0   | Timer1 prescaler. Timer1 clock source (Fsys/2) divided by<br>0000: 1 0011: 8 0110: 64<br>0001: 2 0100: 16 0111: 128<br>0010: 4 0101: 32 1xxx: 256                            |
| <b>T2CTL (15h)</b>  |         |     |     | <b>Function related to: T2</b>                                                                                                                                               |
| T2CLR               | 15.4    | R/W | 0   | Clear and stop T2<br>0: T2 runs<br>1: T2 clear and stops                                                                                                                     |

| Name                | Address | R/W | Rst | Description                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------|---------|-----|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T2CKS               | 15.3~2  | R/W | 0   | T2 clock source selection<br>00: Slow-clock 11: Fsys/128 1x: FIRC/512 (16 MHz/512)                                                                                                                                                                                                                                                                                                                  |
| T2PSC               | 15.1~0  | R/W | 0   | T2 prescaler. T2 clock source divided by<br>00: 32768 01: 16384 10: 8192 11: 128                                                                                                                                                                                                                                                                                                                    |
| <b>LVCTL (16h)</b>  |         |     |     | <b>Function related to: LVD/LVR</b>                                                                                                                                                                                                                                                                                                                                                                 |
| LVDF                | 16.7    | R   | 0   | Low voltage detection flag<br>0: $V_{CC} > V_{LVD}$<br>1: $V_{CC} < V_{LVD}$                                                                                                                                                                                                                                                                                                                        |
| LVDHYS              | 16.6    | R/W | 0   | LVD Hysteresis<br>0: disable<br>1: enable                                                                                                                                                                                                                                                                                                                                                           |
| LVRSAV              | 16.5    | R/W | 1   | POR/LVR auto power off in STOP/IDLE mode                                                                                                                                                                                                                                                                                                                                                            |
| LVDSAV              | 16.4    | R/W | 1   | LVD auto power off in STOP/IDLE mode                                                                                                                                                                                                                                                                                                                                                                |
| LVDS                | 16.3~0  | R/W | 0   | LVD voltage ( $V_{LVD}$ ) select<br>Please refer to the table of LVD voltage in the section of “ <a href="#">LVD Circuit Characteristics</a> ”. (Left click the link to go to that page)                                                                                                                                                                                                            |
| <b>ADCDH (17h)</b>  |         |     |     | <b>Function related to: ADC</b>                                                                                                                                                                                                                                                                                                                                                                     |
| ADCDH               | 17.7~0  | R   | -   | ADC output data bit 11~4                                                                                                                                                                                                                                                                                                                                                                            |
| <b>ADCTL (18h)</b>  |         |     |     | <b>Function related to: ADC</b>                                                                                                                                                                                                                                                                                                                                                                     |
| ADCDL               | 18.7~4  | R   | -   | ADC output data bit 3~0                                                                                                                                                                                                                                                                                                                                                                             |
| ADSTEOC             | 18.3    | W   | 0   | ADC start bit.<br>0: H/W clear after end of conversion<br>1: ADC start conversion<br>Don't write ADSTEOC when PWM trigger ADC mode (i.e. ADCDLYT != 0) is utilized.                                                                                                                                                                                                                                 |
|                     |         | R   | 0   | ADC start or end of conversion bit.<br>Normal mode(i.e. ADCDLYT = 0): ADC start<br>0: H/W clear after end of conversion<br>1: ADC start conversion<br>PWM trigger ADC mode(i.e. ADCDLYT != 0): ADC end of conversion<br>0: ADC is in conversion<br>1: ADC is end of conversion                                                                                                                      |
| ADCKS               | 18.2~0  | R/W | 0   | ADC clock frequency selection. 1MHz(Typ.)<br>000: Fsys/256 010: Fsys/64 100: Fsys/16 110: Fsys/4<br>001: Fsys/128 011: Fsys/32 101: Fsys/8 111: Fsys/2                                                                                                                                                                                                                                              |
| <b>ADCTL2 (19h)</b> |         |     |     | <b>Function related to: ADC</b>                                                                                                                                                                                                                                                                                                                                                                     |
| ADVREFS             | 19.7~6  | R/W | 00  | ADC reference voltage and $V_{BG}$ output voltage select<br>00: ADC reference voltage is $V_{CC}$ or 1.2V $V_{BG}$ according to the value of ADVREF1P2. $V_{BG}$ is 1.20V<br>01: ADC reference voltage is $V_{BG}$ , $V_{BG}$ is 2.56V<br>10: Reserved<br>11: ADC reference voltage is $V_{BG}$ , $V_{BG}$ is 2.00V(This feature can't not be emulated) (Don't use for the selection of DAC's VREF) |
| ADVREF1P2           | 19.5    | R/W | 0   | ADC 1.2V reference voltage select<br>0: ADC reference voltage is $V_{CC}$ when ADVREFS=00. $V_{BG}$ is 1.2V<br>1: ADC reference voltage is 1.2V $V_{BG}$ when ADVREFS=00. $V_{BG}$ is 1.2V(This feature can't not be emulated)                                                                                                                                                                      |

| Name                     | Address | R/W | Rst | Description                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------------|---------|-----|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADCHS                    | 19.4~0  | R/W | 1F  | ADC channel select<br>00000: ADC0 (PA0)    01000: ADC8 (PB1)    others: Reserved<br>00001: ADC1 (PA1)    01001: ADC9 (PB2)<br>00010: ADC2 (PA2)    01010: ADC10 (PB4)<br>00011: ADC3 (PA3)    01011: ADC11 (PB5)<br>00100: ADC4 (PA4)    01100: ADC12 (PB6)<br>00101: ADC5 (PA5)    01101: ADC13 (PB7)<br>00110: ADC6 (PA6)    01110: VBG<br>00111: ADC7 (PB0)    10111: 1/4 VCC |
| <b>User Data Memory</b>  |         |     |     |                                                                                                                                                                                                                                                                                                                                                                                  |
| RAM                      | 20~6F   | R/W | -   | RAM Bank0 area (80 Bytes)                                                                                                                                                                                                                                                                                                                                                        |
| RAM                      | 70~7F   | R/W | -   | RAM common area (16 Bytes)                                                                                                                                                                                                                                                                                                                                                       |
| <b>OPTION (81h/181h)</b> |         |     |     | <b>Function related to: STATUS/INT0/INT1/WDT/WKT</b>                                                                                                                                                                                                                                                                                                                             |
| HWAUTO                   | 81.7    | R/W | 0   | Enter/Exit interrupt subroutine, HW auto Save/Restore WREG, FSR, TABR, PCLATH, DPL, DPH, and STATUS w/o TO, PD<br>0:disable<br>1:enable                                                                                                                                                                                                                                          |
| INT0EDG                  | 81.6    | R/W | 0   | INT0 pin interrupt edge selection<br>0: falling edge to trigger<br>1: rising edge to trigger                                                                                                                                                                                                                                                                                     |
| INT1EDG                  | 81.5    | R/W | 0   | INT1 pin interrupt edge selection<br>0: falling edge to trigger<br>1: rising edge to trigger                                                                                                                                                                                                                                                                                     |
| WDTOSC                   | 81.3~2  | R/W | 3   | WDT period selections:<br>00: 125ms   01: 250ms   10: 1001ms   11: 2001ms @5V                                                                                                                                                                                                                                                                                                    |
| WKTOSC                   | 81.1~0  | R/W | 3   | WKT period selections:<br>00: 15.6ms   01: 31.3ms   10: 62.5ms   11: 125ms @5V                                                                                                                                                                                                                                                                                                   |
| <b>PAMOD10 (85h)</b>     |         |     |     | <b>Function related to: Port A</b>                                                                                                                                                                                                                                                                                                                                               |
| PA1MOD                   | 85.7~4  | R/W | 1   | PA1 I/O mode control                                                                                                                                                                                                                                                                                                                                                             |
| PA0MOD                   | 85.3~0  | R/W | 1   | PA0 I/O mode control                                                                                                                                                                                                                                                                                                                                                             |
| <b>PAMOD32 (86h)</b>     |         |     |     | <b>Function related to: Port A</b>                                                                                                                                                                                                                                                                                                                                               |
| PA3MOD                   | 86.7~4  | R/W | 1   | PA3 I/O mode control                                                                                                                                                                                                                                                                                                                                                             |
| PA2MOD                   | 86.3~0  | R/W | 1   | PA2 I/O mode control                                                                                                                                                                                                                                                                                                                                                             |
| <b>PAMOD54 (87h)</b>     |         |     |     | <b>Function related to: Port A</b>                                                                                                                                                                                                                                                                                                                                               |
| PA5MOD                   | 87.7~4  | R/W | 1   | PA5 I/O mode control                                                                                                                                                                                                                                                                                                                                                             |
| PA4MOD                   | 87.3~0  | R/W | 1   | PA4 I/O mode control                                                                                                                                                                                                                                                                                                                                                             |
| <b>PAMOD76 (88h)</b>     |         |     |     | <b>Function related to: Port A</b>                                                                                                                                                                                                                                                                                                                                               |
| PA7MOD                   | 88.7~4  | R/W | 0   | PA7 I/O mode control<br>PA7 has no high-sink, 1/2 bias and resistor pull-down capability.                                                                                                                                                                                                                                                                                        |
| PA6MOD                   | 88.3~0  | R/W | 1   | PA6 I/O mode control                                                                                                                                                                                                                                                                                                                                                             |
| <b>PWMCTL (89h)</b>      |         |     |     | <b>Function related to: PWM0</b>                                                                                                                                                                                                                                                                                                                                                 |
| PWMEN                    | 89.7    | R/W | 0   | PWM Clock Enable<br>0: Disable<br>1: Enable                                                                                                                                                                                                                                                                                                                                      |
| CLRPWM                   | 89.6    | R/W | 0   | Clear PWM<br>0: not clear<br>1: clear                                                                                                                                                                                                                                                                                                                                            |
| PWM0OM                   | 89.5~4  | R/W | 0   | PWM0 output mode<br>00: Mode0<br>01: Mode1<br>10: Mode2<br>11: Mode3                                                                                                                                                                                                                                                                                                             |



| Name                 | Address | R/W | Rst | Description                                                                                                                                                                                                                                                          |
|----------------------|---------|-----|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PWM0DZ               | 89.3~0  | R/W | 0   | PWM0 dead-zone(non-overlap) control<br>0000: no dead-zone(non-overlap)<br>0001: dead-zone(non-overlap) width are 1 PWM clock cycle<br>0010: dead-zone(non-overlap) width are 2 PWM clock cycles<br>...<br>1111: dead-zone(non-overlap) width are 15 PWM clock cycles |
| <b>PBMOD10 (8Ch)</b> |         |     |     | <b>Function related to: Port B</b>                                                                                                                                                                                                                                   |
| PB1MOD               | 8C.7~4  | R/W | 1   | PB1 I/O mode control                                                                                                                                                                                                                                                 |
| PB0MOD               | 8C.3~0  | R/W | 1   | PB0 I/O mode control                                                                                                                                                                                                                                                 |
| <b>PBMOD32 (8Dh)</b> |         |     |     | <b>Function related to: Port B</b>                                                                                                                                                                                                                                   |
| PB3MOD               | 8D.7~4  | R/W | 1   | PB3 I/O mode control                                                                                                                                                                                                                                                 |
| PB2MOD               | 8D.3~0  | R/W | 1   | PB2 I/O mode control                                                                                                                                                                                                                                                 |
| <b>PBMOD54 (8Eh)</b> |         |     |     | <b>Function related to: Port B</b>                                                                                                                                                                                                                                   |
| PB5MOD               | 8E.7~4  | R/W | 1   | PB5 I/O mode control                                                                                                                                                                                                                                                 |
| PB4MOD               | 8E.3~0  | R/W | 1   | PB4 I/O mode control                                                                                                                                                                                                                                                 |
| <b>PBMOD76 (8Fh)</b> |         |     |     | <b>Function related to: Port B</b>                                                                                                                                                                                                                                   |
| PB7MOD               | 8F.7~4  | R/W | 1   | PB7 I/O mode control                                                                                                                                                                                                                                                 |
| PB6MOD               | 8F.3~0  | R/W | 1   | PB6 I/O mode control                                                                                                                                                                                                                                                 |
| <b>PDMOD10 (90h)</b> |         |     |     | <b>Function related to: Port D</b>                                                                                                                                                                                                                                   |
| PD1MOD               | 90.7~4  | R/W | 1   | PD1 I/O mode control                                                                                                                                                                                                                                                 |
| PD0MOD               | 90.3~0  | R/W | 1   | PD0 I/O mode control                                                                                                                                                                                                                                                 |
| <b>OPTION2 (91h)</b> |         |     |     | <b>Function related to: PWM0/INT2/INT1/INT0</b>                                                                                                                                                                                                                      |
| PWMCKS               | 91.5~4  | R/W | 00  | PWM Clock Source<br>00: Fsys<br>01: FIRC/256<br>10: FIRC (16 MHz)<br>11: FIRC*2 (32 MHz). Refer to the graph of minimal operating voltage for PWMCKS=FIRC x 2.                                                                                                       |
| INT2SEL              | 91.2    | R/W | 0   | INT2 pin select<br>0: PA7<br>1: PB5                                                                                                                                                                                                                                  |
| INT1SEL              | 91.1    | R/W | 0   | INT1 pin select<br>0: PA1<br>1: PB1                                                                                                                                                                                                                                  |
| INT0SEL              | 91.0    | R/W | 0   | INT0 pin select<br>0: PA3<br>1: PB2                                                                                                                                                                                                                                  |
| <b>PWMPRDH (92h)</b> |         |     |     | <b>Function related to: PWM</b>                                                                                                                                                                                                                                      |
| PWMPRDH              | 92.7~0  | R/W | FF  | PWM Period bit 15~8                                                                                                                                                                                                                                                  |
| <b>PWMPRDL (93h)</b> |         |     |     | <b>Function related to: PWM</b>                                                                                                                                                                                                                                      |
| PWMPRDL              | 93.7~0  | R/W | FF  | PWM Period bit 7~0                                                                                                                                                                                                                                                   |
| <b>PWM0DH (94h)</b>  |         |     |     | <b>Function related to: PWM0</b>                                                                                                                                                                                                                                     |
| PWM0DH               | 94.7~0  | R/W | 80  | PWM0 Duty bit 15~8                                                                                                                                                                                                                                                   |
| <b>PWM0DL (95h)</b>  |         |     |     | <b>Function related to: PWM0</b>                                                                                                                                                                                                                                     |
| PWM0DL               | 95.7~0  | R/W | 00  | PWM0 Duty bit 7~0                                                                                                                                                                                                                                                    |
| <b>PWM1DH (96h)</b>  |         |     |     | <b>Function related to: PWM1</b>                                                                                                                                                                                                                                     |
| PWM1DH               | 96.7~0  | R/W | 80  | PWM1 Duty bit 15~8                                                                                                                                                                                                                                                   |
| <b>PWM1DL (97h)</b>  |         |     |     | <b>Function related to: PWM1</b>                                                                                                                                                                                                                                     |
| PWM1DL               | 97.7~0  | R/W | 00  | PWM1 Duty bit 7~0                                                                                                                                                                                                                                                    |
| <b>PWM2DH (98h)</b>  |         |     |     | <b>Function related to: PWM2</b>                                                                                                                                                                                                                                     |
| PWM2DH               | 98.7~0  | R/W | 80  | PWM2 Duty bit 15~8                                                                                                                                                                                                                                                   |

| Name                    | Address | R/W | Rst | Description                                                                                                                                                                   |
|-------------------------|---------|-----|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PWM2DL (99h)</b>     |         |     |     | <b>Function related to: PWM2</b>                                                                                                                                              |
| PWM2DL                  | 99.7~0  | R/W | 00  | PWM2 Duty bit 7~0                                                                                                                                                             |
| <b>PWM3DH (9Ah)</b>     |         |     |     | <b>Function related to: PWM3</b>                                                                                                                                              |
| PWM3DH                  | 9A.7~0  | R/W | 80  | PWM3 Duty bit 15~8                                                                                                                                                            |
| <b>PWM3DL (9Bh)</b>     |         |     |     | <b>Function related to: PWM3</b>                                                                                                                                              |
| PWM3DL                  | 9B.7~0  | R/W | 00  | PWM3 Duty bit 7~0                                                                                                                                                             |
| <b>PWM4DH (9Ch)</b>     |         |     |     | <b>Function related to: PWM4</b>                                                                                                                                              |
| PWM4DH                  | 9C.7~0  | R/W | 80  | PWM4 Duty bit 15~8                                                                                                                                                            |
| <b>PWM4DL (9Dh)</b>     |         |     |     | <b>Function related to: PWM4</b>                                                                                                                                              |
| PWM4DL                  | 9D.7~0  | R/W | 00  | PWM4 Duty bit 7~0                                                                                                                                                             |
| <b>PWM5DH (9Eh)</b>     |         |     |     | <b>Function related to: PWM5</b>                                                                                                                                              |
| PWM5DH                  | 9E.7~0  | R/W | 80  | PWM5 Duty bit 15~8                                                                                                                                                            |
| <b>PWM5DL (9Fh)</b>     |         |     |     | <b>Function related to: PWM5</b>                                                                                                                                              |
| PWM5DL                  | 9F.7~0  | R/W | 00  | PWM5 Duty bit 7~0                                                                                                                                                             |
| <b>User Data Memory</b> |         |     |     |                                                                                                                                                                               |
| RAM                     | A0~EF   | R/W | -   | RAM Bank1 area (80 Bytes)                                                                                                                                                     |
| <b>PINMOD (105h)</b>    |         |     |     | <b>Function related to: IO Port</b>                                                                                                                                           |
| Reserved                | 105.5   | R   | x   | read as unknown after reset                                                                                                                                                   |
| Reserved                | 105.4   | R/W | 0   | must be kept at 0                                                                                                                                                             |
| HSINK                   | 105.2   | R/W | 1   | All IO port high sink current enable<br>0: low sink current<br>1: high sink current                                                                                           |
| Reserved                | 105.1   | R/W | 0   | must be kept at 0                                                                                                                                                             |
| Reserved                | 105.0   | R/W | 0   | must be kept at 0                                                                                                                                                             |
| <b>PWMCTL2 (107h)</b>   |         |     |     | <b>Function related to: PWM0</b>                                                                                                                                              |
| PWM0MOD                 | 107.7   | R/W | 0   | PWM0 mode select<br>0: Normal mode<br>1: Half-bridge mode                                                                                                                     |
| PWM0MSKE                | 107.6   | R/W | 0   | PWM0 mask output enable<br>0: Disable<br>1: Enable, PWM0P/PWM0N output data by PWM0PSK/PWM0NSK while CLRPWM=1                                                                 |
| PWM0NSK                 | 107.5   | R/W | 0   | PWM0N mask data. If CLRPWM=1 and PWM0MSKE=1, PWM0N will output this mask data.                                                                                                |
| PWM0PSK                 | 107.4   | R/W | 0   | PWM0P mask data. If CLRPWM=1 and PWM0MSKE=1, PWM0P will output this mask data.                                                                                                |
| <b>LVRPD (109h)</b>     |         |     |     | <b>Function related to: LVR/POR</b>                                                                                                                                           |
| LVRPD                   | 109.7~0 | W   | 0   | Write 37h to force LVR+POR Disable<br>Write 38h to force LVR Disable, POR still enable<br>Write 39h to force POR Disable, LVR still enable<br>Write others LVR and POR enable |
| PORPDF                  | 109.1   | R   | 0   | POR force power down flag<br>0: POR enable<br>1: POR is forced power down                                                                                                     |
| LVRPDF                  | 109.0   | R   | 0   | LVR force power down flag<br>0: LVR enable<br>1: LVR is forced power down                                                                                                     |
| <b>PCH (10Ch)</b>       |         |     |     | <b>Function related to: PCH</b>                                                                                                                                               |

| Name                    | Address | R/W | Rst | Description                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------|---------|-----|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PCH                     | 10C.7~0 | W   | 00  | Programming Counter high byte source selection when instruction with PCL as destination is executed<br>write 0x1C to set PCH_S = 1: PCH keep the original value<br>write others to clear PCH_S = 0: PCH is from PCLATH<br>After reset, the PCH_S is cleared                                                                                                                                                                                     |
| PCH                     | 10C.3~0 | R   | 0   | Program Counter data bit 11~8                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>BGTRIM (10Eh)</b>    |         |     |     | <b>Function related to: Bandgap</b>                                                                                                                                                                                                                                                                                                                                                                                                             |
| BGTRIM                  | 10E.4~0 | R/W | CFG | VBG trim value                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>IRCF (10Fh)</b>      |         |     |     | <b>Function related to: Internal RC</b>                                                                                                                                                                                                                                                                                                                                                                                                         |
| IRCF                    | 10F.6~0 | R/W | CFG | FIRC trim value                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>BG2TRIM (111h)</b>   |         |     |     | <b>Function related to: Bandgap</b>                                                                                                                                                                                                                                                                                                                                                                                                             |
| BG2TRIM                 | 111.7~0 | R   | CFG | VBG 2V trim value. The users could move this register to BGTRIM for exact 2V VBG. This feature can't be emulated.                                                                                                                                                                                                                                                                                                                               |
| <b>RDCTL (113h)</b>     |         |     |     | <b>Function related to: Program ROM</b>                                                                                                                                                                                                                                                                                                                                                                                                         |
| RDCTL                   | 113.1~0 | R/W | 02  | Read signal delay control for Program ROM<br>00: 16ns delay for read signal of Program ROM<br>01: 12ns delay for read signal of Program ROM<br>10: 8ns delay for read signal of Program ROM<br>11: 4ns delay for read signal of Program ROM<br>Change this register at slow clock for safety.<br><b>The user must switch this register to “4ns” to enhance the performance of minimal operating voltage.</b><br>This feature can't be emulated. |
| <b>SIRCF (115h)</b>     |         |     |     | <b>Function related to: Slow Internal RC</b>                                                                                                                                                                                                                                                                                                                                                                                                    |
| SIRCF                   | 115.4~0 | R/W | CFG | SIRC trim value                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>User Data Memory</b> |         |     |     |                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| RAM                     | 120~16F | R/W | -   | RAM Bank2 area (80 Bytes)                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>DPL (185h)</b>       |         |     |     | <b>Function related to: Table Read</b>                                                                                                                                                                                                                                                                                                                                                                                                          |
| DPL                     | 185.7~0 | R/W | 00  | TBL Data Pointer bit 7~0                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>DPH (186h)</b>       |         |     |     | <b>Function related to: Table Read</b>                                                                                                                                                                                                                                                                                                                                                                                                          |
| DPH                     | 186.3~0 | R/W | 00  | TBL Data Pointer bit 11~8                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>CRCDL (187h)</b>     |         |     |     | <b>Function related to: CRC16</b>                                                                                                                                                                                                                                                                                                                                                                                                               |
| CRCDL                   | 187.7~0 | R/W | FF  | 16-bit CRC checksum data bit 7~0                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>CRCDH (188h)</b>     |         |     |     | <b>Function related to: CRC16</b>                                                                                                                                                                                                                                                                                                                                                                                                               |
| CRCDH                   | 188.7~0 | R/W | FF  | 16-bit CRC checksum data bit 15~8                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>CRCIN (189h)</b>     |         |     |     | <b>Function related to: CRC16</b>                                                                                                                                                                                                                                                                                                                                                                                                               |
| CRCIN                   | 189.7~0 | W   | 0   | CRC data input, write this register to start CRC calculation                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>TABR (18Ch)</b>      |         |     |     | <b>Function related to: Table Read</b>                                                                                                                                                                                                                                                                                                                                                                                                          |

| Name                  | Address | R/W | Rst                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------|---------|-----|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TABR                  | 18C.7~0 | R/W | 0                                          | 1. TABR write 01h = instruction TABRL (Read PROM low byte data to W and TABR)<br>2. TABR write 02h = instruction TABRH (Read PROM high byte data to W and TABR)<br>3. Don't write the value other than 01h or 02h into register TABR<br>4. After step.1 or step.2, read TABR to get main ROM table read value for C language<br><i>Table Read for ASM: Support instruction TABRL / TABRH or register TABR. Suggest not using the method of register TABR. SFR HWAUTO=1 is also suggested.</i><br><i>Table Read for C: using register TABR. Only be used outside or inside the interrupt service routine. Don't utilize it inside and outside interrupt service routine simultaneously. Otherwise, something will be wrong.</i> |
| <b>CMPCCTL (18Dh)</b> |         |     | <b>Function related to: Comparator</b>     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| PDCMP                 | 18D.7   | R/W | 1                                          | Comparator & DAC power down enable control<br>0: disable Comparator & DAC power down<br>1: enable Comparator & DAC power down                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| CMPOX                 | 18D.6   | R   | 1                                          | Comparator original output (CMPOX) status<br>0: $V_{CMPN} < V_{CMPN}$<br>1: $V_{CMPN} > V_{CMPN}$ or PDCMP =1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| CMPOE                 | 18D.5   | R/W | 0                                          | Comparator output (CMPO) signal output to PA5<br>0: disable<br>1: enable, PA5MOD should be set to xx10b                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| CMPIINV               | 18D.4   | R/W | 0                                          | Comparator de-bounce output invert select<br>0: no invert<br>1: invert                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| CMPTRIG               | 18D.3~2 | R/W | 0                                          | Comparator interrupt trigger mode<br>00: Rising edge<br>01: Falling edge<br>10: Both edge<br>11: High level                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| CMPPDBS               | 18D.1~0 | R/W | 0                                          | Comparator original output (CMPOX) de-bounce time<br>00: none<br>01: 4 Fsys<br>10: 8 Fsys<br>11: 16 Fsys                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>CMPPNS (18Eh)</b>  |         |     | <b>Function related to: Comparator/DAC</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| SCMPN                 | 18E.7   | R/W | 1                                          | Comparator CMPN source select<br>0: Comparator CMPN source is external input (CINx)<br>1: Comparator CMPN source is DAC output                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| SCIN                  | 18E.6~4 | R/W | 7                                          | Comparator CMPN external input select<br>000: Comparator CMPN external input is CIN1 (PA3)<br>001: Comparator CMPN external input is CIN2 (PA0)<br>010: Comparator CMPN external input is CIN3 (PB7)<br>011: Comparator CMPN external input is CIN4 (PB4)<br>1xx: No connect                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| -                     | 18E.3   | -   | -                                          | This bit must be set as 1 in emulation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| SCIP                  | 18E.2~0 | R/W | 7                                          | Comparator CMPP external input select<br>000: Comparator CMPP external input is CIP1 (PA1)<br>001: Comparator CMPP external input is CIP2 (PA2)<br>010: Comparator CMPP external input is CIP3 (PB6)<br>011: Comparator CMPP external input is CIP4 (PD1)<br>1xx: No connect                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

| Name                  | Address | R/W | Rst | Description                                                                                                                                                                                                                                                                                         |
|-----------------------|---------|-----|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DACTL (18Fh)</b>   |         |     |     | <b>Function related to: DAC/Comparator</b>                                                                                                                                                                                                                                                          |
| SVRF                  | 18F.7   | R/W | 0   | DAC reference voltage select<br>0: $V_{CC}$<br>1: $V_{BG}$ (voltage level is selected by ADVREFS)                                                                                                                                                                                                   |
| SVL                   | 18F.6~0 | R/W | 0   | DAC output voltage select<br>(reference source can be selected as $V_{CC}$ or $V_{BG}$ )<br>000_0000: 0/128 * reference source<br>000_0001: 1/128 * reference source<br>...<br>111_1101: 125/128 * reference source<br>111_1110: 126/128 * reference source<br>111_1111: 127/128 * reference source |
| <b>EEPCTL (190h)</b>  |         |     |     | <b>Function related to: EEPROM</b>                                                                                                                                                                                                                                                                  |
| EEPTO                 | 190.7   | R   | 0   | EEPROM Write Time-Out flag<br>0: EEPROM Write no Time-Out<br>1: EEPROM Write is Time-Out. EEPTO will be cleared when EEPEN is disabled.                                                                                                                                                             |
| EEPTE                 | 190.1~0 | R/W | 0   | EEPROM Write Time-Out enable (Access wait time)<br>00: Disable<br>01: 2 ms@5.0V, 2.5 ms@3.0V<br>10: 7.8 ms@5.0V, 10 ms@3.0V<br>11: 62.5 ms@5.0V, 79.6 ms@3.0V                                                                                                                                       |
| <b>EEPEN (191h)</b>   |         |     |     | <b>Function related to: EEPROM</b>                                                                                                                                                                                                                                                                  |
| EEPEN                 | 191.7~0 | W   | 0   | EEPROM Access Enable<br>Write 0xE2 to this register will enable EEPROM access<br>Write others value to this register will disable EEPROM access                                                                                                                                                     |
| <b>EEPDT (192h)</b>   |         |     |     | <b>Function related to: EEPROM</b>                                                                                                                                                                                                                                                                  |
| EEPDT                 | 192.7~0 | W   | 0   | EEPROM Data to write<br>Write data to this register will let H/W write the data to EEPROM when EEPROM access is enable<br><b>note: To clear watchdog first before writing EEPROM to avoid watchdog timer reset during writing procedure for safety.</b>                                             |
| <b>ADCDLYT (193h)</b> |         |     |     | <b>Function related to: ADC</b>                                                                                                                                                                                                                                                                     |
| ADCDLYT               | 193.7~0 | R/W | 0   | The delay time between PWM period match and ADC start<br>8'h00: disable PWM trigger ADC start<br>8'h01~8'hFF: enable and delay time(ADCDLT[7:0]/Fadc) to ADC start                                                                                                                                  |

## INSTRUCTION SET

Each instruction is a 16-bit word divided into an Op Code, which specifies the instruction type, and one or more operands, which further specify the operation of the instruction. The instructions can be categorized as byte-oriented, bit-oriented and literal operations list in the following table.

For byte-oriented instructions, “f” represents the address designator and “d” represents the destination designator. The address designator is used to specify which address in Program memory is to be used by the instruction. The destination designator specifies where the result of the operation is to be placed. If “d” is “0”, the result is placed in the W register. If “d” is “1”, the result is placed in the address specified in the instruction.

For bit-oriented instructions, “b” represents a bit field designator, which selects the number of the bit affected by the operation, while “f” represents the address designator. For literal operations, “k” represents the literal or constant value.

| Field/Legend | Description                                                        |
|--------------|--------------------------------------------------------------------|
| f            | Register File Address                                              |
| b            | Bit address                                                        |
| k            | Literal. Constant data or label                                    |
| d            | Destination selection field, 0: Working register, 1: Register file |
| W            | Working Register                                                   |
| Z            | Zero Flag                                                          |
| C            | Carry Flag or /Borrow Flag                                         |
| DC           | Decimal Carry Flag or Decimal /Borrow Flag                         |
| PC           | Program Counter                                                    |
| TOS          | Top Of Stack                                                       |
| GIE          | Global Interrupt Enable Flag (i-Flag)                              |
| []           | Option Field                                                       |
| ()           | Contents                                                           |
| .            | Bit Field                                                          |
| B            | Before                                                             |
| A            | After                                                              |
| ←            | Assign direction                                                   |

| Mnemonic                                       |      | Op Code              | Cycle  | Flag Affect | Description                                      |
|------------------------------------------------|------|----------------------|--------|-------------|--------------------------------------------------|
| <b>Byte-Oriented File Register Instruction</b> |      |                      |        |             |                                                  |
| ADDW <del>X</del>                              | f, d | ff00 0111 dfff ffff  | 1      | C, DC, Z    | Add W and "f"                                    |
| ANDW <del>X</del>                              | f, d | ff00 0101 dfff ffff  | 1      | Z           | AND W with "f"                                   |
| CLR <del>X</del>                               | f    | ff00 0001 1fff ffff  | 1      | Z           | Clear "f"                                        |
| CLRW                                           |      | 0000 0001 0100 0000  | 1      | Z           | Clear W                                          |
| COM <del>X</del>                               | f, d | ff00 1001 dfff ff ff | 1      | Z           | Complement "f"                                   |
| DEC <del>X</del>                               | f, d | ff00 0011 dfff ffff  | 1      | Z           | Decrement "f"                                    |
| DEC <del>X</del> SZ                            | f, d | ff00 1011 dfff ffff  | 1 or 2 | -           | Decrement "f", skip if zero                      |
| INC <del>X</del>                               | f, d | ff00 1010 dfff ffff  | 1      | Z           | Increment "f"                                    |
| INC <del>X</del> SZ                            | f, d | ff00 1111 dfff ffff  | 1 or 2 | -           | Increment "f", skip if zero                      |
| IORW <del>X</del>                              | f, d | ff00 0100 dfff ffff  | 1      | Z           | OR W with "f"                                    |
| MOV <del>X</del>                               | f, d | ff00 1000 dfff ffff  | 1      | Z           | Move "f"                                         |
| MOV <del>X</del> W                             | f    | ff00 1000 0fff ffff  | 1      | Z           | Move "f" to W                                    |
| MOVW <del>X</del>                              | f    | ff00 0000 1fff ffff  | 1      | -           | Move W to "f"                                    |
| RL <del>X</del>                                | f, d | ff00 1101 dfff ffff  | 1      | C           | Rotate left "f" through carry                    |
| RR <del>X</del>                                | f, d | ff00 1100 dfff ffff  | 1      | C           | Rotate right "f" through carry                   |
| SUBW <del>X</del>                              | f, d | ff00 0010 dfff ffff  | 1      | C, DC, Z    | Subtract W from "f"                              |
| SWAP <del>X</del>                              | f, d | ff00 1110 dfff ffff  | 1      | -           | Swap nibbles in "f"                              |
| TST <del>X</del>                               | f    | ff00 1000 1fff ffff  | 1      | Z           | Test if "f" is zero                              |
| XORW <del>X</del>                              | f, d | ff00 0110 dfff ffff  | 1      | Z           | XOR W with "f"                                   |
| <b>Bit-Oriented File Register Instruction</b>  |      |                      |        |             |                                                  |
| BC <del>X</del>                                | f, b | ff11 00bb bfff ffff  | 1      | -           | Clear "b" bit of "f"                             |
| BS <del>X</del>                                | f, b | ff11 01bb bfff ffff  | 1      | -           | Set "b" bit of "f"                               |
| BT <del>X</del> SC                             | f, b | ff11 10bb bfff ffff  | 1 or 2 | -           | Test "b" bit of "f", skip if clear               |
| BT <del>X</del> SS                             | f, b | ff11 11bb bfff ffff  | 1 or 2 | -           | Test "b" bit of "f", skip if set                 |
| <b>Literal and Control Instruction</b>         |      |                      |        |             |                                                  |
| ADDLW                                          | k    | 0001 1100 kkkk kkkk  | 1      | C, DC, Z    | Add Literal "k" and W                            |
| ANDLW                                          | k    | 0001 1011 kkkk kkkk  | 1      | Z           | AND Literal "k" with W                           |
| L <del>C</del> ALL                             | k    | kk10 0kkk kkkk kkkk  | 2      | -           | Call subroutine "k"                              |
| CLRWD <del>T</del>                             |      | 0001 1110 0000 0100  | 1      | TO, PD      | Clear Watch Dog Timer                            |
| L <del>G</del> OTO                             | k    | kk10 1kkk kkkk kkkk  | 2      | -           | Jump to branch "k"                               |
| IORLW                                          | k    | 0001 1010 kkkk kkkk  | 1      | Z           | OR Literal "k" with W                            |
| MOVLW                                          | k    | 0001 1001 kkkk kkkK  | 1      | -           | Move Literal "k" to W                            |
| NOP                                            |      | 0000 0000 0000 0000  | 1      | -           | No operation                                     |
| RET                                            |      | 0000 0000 0100 0000  | 2      | -           | Return from subroutine                           |
| RETI                                           |      | 0000 0000 0110 0000  | 2      | -           | Return from interrupt                            |
| RETLW                                          | k    | 0001 1000 kkkk kkkk  | 2      | -           | Return with Literal in W                         |
| SLEEP                                          |      | 0001 1110 0000 0011  | 1      | TO, PD      | Go into Power-down mode, Clock oscillation stops |
| SUBLW                                          | k    | 0001 1111 kkkk kkkk  | 1      | C, DC, Z    | Subtract W from literal                          |
| TABRH                                          |      | 0000 0000 0101 1000  | 2      | -           | Lookup ROM high data to W                        |
| TABRL                                          |      | 0000 0000 0101 0000  | 2      | -           | Lookup ROM low data to W                         |
| XORLW                                          | k    | 0001 1101 kkkk kkkk  | 1      | Z           | XOR Literal "k" with W                           |

| <b>ADDLW</b>    | <b>Add Literal "k" and W</b>                                                                                      |                            |
|-----------------|-------------------------------------------------------------------------------------------------------------------|----------------------------|
| Syntax          | ADDLW k                                                                                                           |                            |
| Operands        | k : 00h ~ FFh                                                                                                     |                            |
| Operation       | $(W) \leftarrow (W) + k$                                                                                          |                            |
| Status Affected | C, DC, Z                                                                                                          |                            |
| OP-Code         | 0001 1100 kkkk kkkk                                                                                               |                            |
| Description     | The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register. |                            |
| Cycle           | 1                                                                                                                 |                            |
| Example         | ADDLW 0x15                                                                                                        | B : W =0x10<br>A : W =0x25 |

| <b>ADDWX</b>    | <b>Add W and "f"</b>                                                                                                                                               |                                                  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| Syntax          | ADDWX f [,d]                                                                                                                                                       |                                                  |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                                          |                                                  |
| Operation       | $(\text{destination}) \leftarrow (W) + (f)$                                                                                                                        |                                                  |
| Status Affected | C, DC, Z                                                                                                                                                           |                                                  |
| OP-Code         | ff00 0111 dfff ffff                                                                                                                                                |                                                  |
| Description     | Add the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'. |                                                  |
| Cycle           | 1                                                                                                                                                                  |                                                  |
| Example         | ADDWX FSR, 0                                                                                                                                                       | B : W =0x17, FSR =0xC2<br>A : W =0xD9, FSR =0xC2 |

| <b>ANDLW</b>    | <b>Logical AND Literal "k" with W</b>                                                                         |                            |
|-----------------|---------------------------------------------------------------------------------------------------------------|----------------------------|
| Syntax          | ANDLW k                                                                                                       |                            |
| Operands        | k : 00h ~ FFh                                                                                                 |                            |
| Operation       | $(W) \leftarrow (W) \text{ AND } k$                                                                           |                            |
| Status Affected | Z                                                                                                             |                            |
| OP-Code         | 0001 1011 kkkk kkkk                                                                                           |                            |
| Description     | The contents of W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register. |                            |
| Cycle           | 1                                                                                                             |                            |
| Example         | ANDLW 0x5F                                                                                                    | B : W =0xA3<br>A : W =0x03 |

| <b>ANDWX</b>    | <b>AND W with "f"</b>                                                                                                                              |                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| Syntax          | ANDWX f [,d]                                                                                                                                       |                                                  |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                          |                                                  |
| Operation       | $(\text{destination}) \leftarrow (W) \text{ AND } (f)$                                                                                             |                                                  |
| Status Affected | Z                                                                                                                                                  |                                                  |
| OP-Code         | ff00 0101 dfff ffff                                                                                                                                |                                                  |
| Description     | AND the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'. |                                                  |
| Cycle           | 1                                                                                                                                                  |                                                  |
| Example         | ANDWX FSR, 1                                                                                                                                       | B : W =0x17, FSR =0xC2<br>A : W =0x17, FSR =0x02 |



---

**BCX Clear "b" bit of "f"**


---

|                 |                                     |                                          |
|-----------------|-------------------------------------|------------------------------------------|
| Syntax          | BCX f [,b]                          |                                          |
| Operands        | f : 000h ~ 1FFh, b : 0 ~ 7          |                                          |
| Operation       | (f.b) ← 0                           |                                          |
| Status Affected | -                                   |                                          |
| OP-Code         | ff11 00bb bfff ffff                 |                                          |
| Description     | Bit 'b' in register 'f' is cleared. |                                          |
| Cycle           | 1                                   |                                          |
| Example         | BCX FLAG_REG, 7                     | B : FLAG_REG =0xC7<br>A : FLAG_REG =0x47 |

---

**BSX Set "b" bit of "f"**


---

|                 |                                 |                                          |
|-----------------|---------------------------------|------------------------------------------|
| Syntax          | BSX f [,b]                      |                                          |
| Operands        | f : 000h ~ 1FFh, b : 0 ~ 7      |                                          |
| Operation       | (f.b) ← 1                       |                                          |
| Status Affected | -                               |                                          |
| OP-Code         | ff11 01bb bfff ffff             |                                          |
| Description     | Bit 'b' in register 'f' is set. |                                          |
| Cycle           | 1                               |                                          |
| Example         | BSX FLAG_REG, 7                 | B : FLAG_REG =0x0A<br>A : FLAG_REG =0x8A |

---

**BTXSC Test "b" bit of "f", skip if clear(0)**


---

|                 |                                                                                                                                                                                                                      |  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Syntax          | BTXSC f [,b]                                                                                                                                                                                                         |  |
| Operands        | f : 000h ~ 1FFh, b : 0 ~ 7                                                                                                                                                                                           |  |
| Operation       | Skip next instruction if (f.b) =0                                                                                                                                                                                    |  |
| Status Affected | -                                                                                                                                                                                                                    |  |
| OP-Code         | ff11 10bb bfff ffff                                                                                                                                                                                                  |  |
| Description     | If bit 'b' in register 'f' is 1, then the next instruction is executed. If bit 'b' in register 'f' is 0, then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction. |  |
| Cycle           | 1 or 2                                                                                                                                                                                                               |  |
| Example         | LABEL1: BTXSC FLAG, 1      B : PC =LABEL1<br>TRUE: LGOTO SUB1      A : if FLAG.1 =0, PC =FALSE<br>FALSE: ...      if FLAG.1 =1, PC =TRUE                                                                             |  |

---

**BTXSS Test "b" bit of "f", skip if set(1)**


---

|                 |                                                                                                                                                                                                                      |  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| Syntax          | BTXSS f [,b]                                                                                                                                                                                                         |  |
| Operands        | f : 000h ~ 1FFh, b : 0 ~ 7                                                                                                                                                                                           |  |
| Operation       | Skip next instruction if (f.b) =1                                                                                                                                                                                    |  |
| Status Affected | -                                                                                                                                                                                                                    |  |
| OP-Code         | ff11 11bb bfff ffff                                                                                                                                                                                                  |  |
| Description     | If bit 'b' in register 'f' is 0, then the next instruction is executed. If bit 'b' in register 'f' is 1, then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction. |  |
| Cycle           | 1 or 2                                                                                                                                                                                                               |  |
| Example         | LABEL1: BTXSS FLAG, 1      B : PC =LABEL1<br>TRUE: LGOTO SUB1      A : if FLAG.1 =0, PC =TRUE<br>FALSE: ...      if FLAG.1 =1, PC =FALSE                                                                             |  |

---

**CLR X                      Clear "f"**


---

|                 |                                                                |                                                |
|-----------------|----------------------------------------------------------------|------------------------------------------------|
| Syntax          | CLR X f                                                        |                                                |
| Operands        | f : 000h ~ 1FFh                                                |                                                |
| Operation       | (f) ← 00h, Z ← 1                                               |                                                |
| Status Affected | Z                                                              |                                                |
| OP-Code         | ff00 0001 1fff ffff                                            |                                                |
| Description     | The contents of register 'f' are cleared and the Z bit is set. |                                                |
| Cycle           | 1                                                              |                                                |
| Example         | CLR X FLAG_REG                                                 | B : FLAG_REG =0x5A<br>A : FLAG_REG =0x00, Z =1 |

---

**CLR W                      Clear W**


---

|                 |                                         |                                  |
|-----------------|-----------------------------------------|----------------------------------|
| Syntax          | CLR W                                   |                                  |
| Operands        | -                                       |                                  |
| Operation       | (W) ← 00h, Z ← 1                        |                                  |
| Status Affected | Z                                       |                                  |
| OP-Code         | 0000 0001 0100 0000                     |                                  |
| Description     | W register is cleared and Z bit is set. |                                  |
| Cycle           | 1                                       |                                  |
| Example         | CLR W                                   | B : W =0x5A<br>A : W =0x00, Z =1 |

---

**CLR WDT                      Clear Watchdog Timer**


---

|                 |                                               |                                             |
|-----------------|-----------------------------------------------|---------------------------------------------|
| Syntax          | CLR WDT                                       |                                             |
| Operands        | -                                             |                                             |
| Operation       | WDT Timer ← 00h                               |                                             |
| Status Affected | TO, PD                                        |                                             |
| OP-Code         | 0001 1110 0000 0100                           |                                             |
| Description     | CLR WDT instruction clears the Watchdog Timer |                                             |
| Cycle           | 1                                             |                                             |
| Example         | CLR WDT                                       | B : WDT counter =?<br>A : WDT counter =0x00 |

---

**COM X                      Complement "f"**


---

|                 |                                                                                                                                                |                                           |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| Syntax          | COM X f [,d]                                                                                                                                   |                                           |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                      |                                           |
| Operation       | (destination) ← ( $\bar{f}$ )                                                                                                                  |                                           |
| Status Affected | Z                                                                                                                                              |                                           |
| OP-Code         | ff00 1001 dfff ffff                                                                                                                            |                                           |
| Description     | The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'. |                                           |
| Cycle           | 1                                                                                                                                              |                                           |
| Example         | COM X REG1, 0                                                                                                                                  | B : REG1 =0x13<br>A : REG1 =0x13, W =0xEC |



|                 |                                                                                                                                                                                                                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DECX</b>     | <b>Decrement "f"</b>                                                                                                                                                                                                                                                                                  |
| Syntax          | DECX f[,d]                                                                                                                                                                                                                                                                                            |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                                                                                                                                                                             |
| Operation       | (destination) ← (f) - 1                                                                                                                                                                                                                                                                               |
| Status Affected | Z                                                                                                                                                                                                                                                                                                     |
| OP-Code         | ff00 0011 dfff ffff                                                                                                                                                                                                                                                                                   |
| Description     | Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.                                                                                                                                                                  |
| Cycle           | 1                                                                                                                                                                                                                                                                                                     |
| Example         | DECX CNT, 1<br>B : CNT =0x01, Z =0<br>A : CNT =0x00, Z =1                                                                                                                                                                                                                                             |
| <b>DECXSZ</b>   | <b>Decrement "f", Skip if 0</b>                                                                                                                                                                                                                                                                       |
| Syntax          | DECXSZ f[,d]                                                                                                                                                                                                                                                                                          |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                                                                                                                                                                             |
| Operation       | (destination) ← (f) - 1, skip next instruction if result is 0                                                                                                                                                                                                                                         |
| Status Affected | -                                                                                                                                                                                                                                                                                                     |
| OP-Code         | ff00 1011 dfff ffff                                                                                                                                                                                                                                                                                   |
| Description     | The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead, making it a 2 cycle instruction. |
| Cycle           | 1 or 2                                                                                                                                                                                                                                                                                                |
| Example         | LABEL1: DECXSZ CNT, 1 B : PC =LABEL1<br>LGOTO LOOP A : CNT =CNT – 1<br>CONTINUE: if CNT =0, “LGOTO LOOP” is replace<br>with NOP<br>if CNT ≠0, “LGOTO LOOP” will be<br>executed                                                                                                                        |
| <b>INCX</b>     | <b>Increment "f"</b>                                                                                                                                                                                                                                                                                  |
| Syntax          | INCX f[,d]                                                                                                                                                                                                                                                                                            |
| Operands        | f : 000h ~ 1FFh                                                                                                                                                                                                                                                                                       |
| Operation       | (destination) ← (f) + 1                                                                                                                                                                                                                                                                               |
| Status Affected | Z                                                                                                                                                                                                                                                                                                     |
| OP-Code         | ff00 1010 dfff ffff                                                                                                                                                                                                                                                                                   |
| Description     | The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.                                                                                                                                            |
| Cycle           | 1                                                                                                                                                                                                                                                                                                     |
| Example         | INCX CNT, 1<br>B : CNT =0xFF, Z =0<br>A : CNT =0x00, Z =1                                                                                                                                                                                                                                             |



| INCXSZ          |                                                                                                                                                                                                                                                                                                  | Increment "f", Skip if 0    |  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|--|
| Syntax          | INCXSZ f [,d]                                                                                                                                                                                                                                                                                    |                             |  |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                                                                                                                                                                        |                             |  |
| Operation       | (destination) ← (f) + 1, skip next instruction if result is 0                                                                                                                                                                                                                                    |                             |  |
| Status Affected | -                                                                                                                                                                                                                                                                                                |                             |  |
| OP-Code         | ff00 1111 dfff ffff                                                                                                                                                                                                                                                                              |                             |  |
| Description     | The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead, making it a 2 cycle instruction. |                             |  |
| Cycle           | 1 or 2                                                                                                                                                                                                                                                                                           |                             |  |
| Example         | LABEL1: INCXSZ CNT, 1 B : PC =LABEL1<br>LGOTO LOOP A : CNT =CNT + 1<br>CONTINUE: if CNT =0, “LGOTO LOOP” is replace with NOP<br>if CNT ≠0, “LGOTO LOOP” will be executed                                                                                                                         |                             |  |
| IORLW           |                                                                                                                                                                                                                                                                                                  | Inclusive OR Literal with W |  |
| Syntax          | IORLW k                                                                                                                                                                                                                                                                                          |                             |  |
| Operands        | k : 00h ~ FFh                                                                                                                                                                                                                                                                                    |                             |  |
| Operation       | (W) ← (W) OR k                                                                                                                                                                                                                                                                                   |                             |  |
| Status Affected | Z                                                                                                                                                                                                                                                                                                |                             |  |
| OP-Code         | 0001 1010 kkkk kkkk                                                                                                                                                                                                                                                                              |                             |  |
| Description     | The contents of the W register are OR’ed with the eight-bit literal 'k'. The result is placed in the W register.                                                                                                                                                                                 |                             |  |
| Cycle           | 1                                                                                                                                                                                                                                                                                                |                             |  |
| Example         | IORLW 0x35 B : W =0x9A<br>A : W =0xBF, Z =0                                                                                                                                                                                                                                                      |                             |  |
| IORWX           |                                                                                                                                                                                                                                                                                                  | Inclusive OR W with "f"     |  |
| Syntax          | IORWX f [,d]                                                                                                                                                                                                                                                                                     |                             |  |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                                                                                                                                                                        |                             |  |
| Operation       | (destination) ← (W) OR (f)                                                                                                                                                                                                                                                                       |                             |  |
| Status Affected | Z                                                                                                                                                                                                                                                                                                |                             |  |
| OP-Code         | ff00 0100 dfff ffff                                                                                                                                                                                                                                                                              |                             |  |
| Description     | Inclusive OR the W register with register 'f'. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.                                                                                                                                      |                             |  |
| Cycle           | 1                                                                                                                                                                                                                                                                                                |                             |  |
| Example         | IORWX RESULT, 0 B : RESULT =0x13, W =0x91<br>A : RESULT =0x13, W =0x93, Z =0                                                                                                                                                                                                                     |                             |  |

|                 |                                                                                                                                                                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>LCALL</b>    | <b>Call subroutine "k"</b>                                                                                                                                                                                                                            |
| Syntax          | LCALL k                                                                                                                                                                                                                                               |
| Operands        | k : 0000h ~ 1FFFh                                                                                                                                                                                                                                     |
| Operation       | Operation: TOS $\leftarrow$ (PC) + 1, PC.12~0 $\leftarrow$ k                                                                                                                                                                                          |
| Status Affected | -                                                                                                                                                                                                                                                     |
| OP-Code         | kk10 0kkk kkkk kkkk                                                                                                                                                                                                                                   |
| Description     | LCALL Subroutine. First, return address (PC+1) is pushed onto the stack. The 13-bit immediate address is loaded into PC bits <12:0>. LCALL is a two-cycle instruction.                                                                                |
| Cycle           | 2                                                                                                                                                                                                                                                     |
| Example         | LABEL1 LCALL SUB1      B : PC =LABEL1<br>A : PC =SUB1, TOS =LABEL1 + 1                                                                                                                                                                                |
| <b>LGOTO</b>    | <b>Unconditional Branch</b>                                                                                                                                                                                                                           |
| Syntax          | LGOTO k                                                                                                                                                                                                                                               |
| Operands        | k : 0000h ~ 1FFFh                                                                                                                                                                                                                                     |
| Operation       | PC.12~0 $\leftarrow$ k                                                                                                                                                                                                                                |
| Status Affected | -                                                                                                                                                                                                                                                     |
| OP-Code         | kk10 1kkk kkkk kkkk                                                                                                                                                                                                                                   |
| Description     | LGOTO is an unconditional branch. The 13-bit immediate value is loaded into PC bits <12:0>. LGOTO is a two-cycle instruction.                                                                                                                         |
| Cycle           | 2                                                                                                                                                                                                                                                     |
| Example         | LABEL1 LGOTO SUB1      B : PC =LABEL1<br>A : PC =SUB1                                                                                                                                                                                                 |
| <b>MOVX</b>     | <b>Move f</b>                                                                                                                                                                                                                                         |
| Syntax          | MOVX f [,d]                                                                                                                                                                                                                                           |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                                                                                                                             |
| Operation       | (destination) $\leftarrow$ (f)                                                                                                                                                                                                                        |
| Status Affected | Z                                                                                                                                                                                                                                                     |
| OP-Code         | ff00 1000 dfff ffff                                                                                                                                                                                                                                   |
| Description     | The contents of register 'f' are moved to a destination dependent upon the status of d. If d=0, destination is W register. If d=1, the destination is file register f itself. d=1 is useful to test a file register, since status flag Z is affected. |
| Cycle           | 1                                                                                                                                                                                                                                                     |
| Example         | MOVX FSR,0      B : FSR =0xC2, W =?<br>A : FSR =0xC2, W =0xC2                                                                                                                                                                                         |
| <b>MOVXW</b>    | <b>Move "f" to W</b>                                                                                                                                                                                                                                  |
| Syntax          | MOVXW f                                                                                                                                                                                                                                               |
| Operands        | f : 000h ~ 1FFh                                                                                                                                                                                                                                       |
| Operation       | (W) $\leftarrow$ (f)                                                                                                                                                                                                                                  |
| Status Affected | Z                                                                                                                                                                                                                                                     |
| OP-Code         | ff00 1000 0fff ffff                                                                                                                                                                                                                                   |
| Description     | The contents of register 'f' are moved to W register.                                                                                                                                                                                                 |
| Cycle           | 1                                                                                                                                                                                                                                                     |
| Example         | MOVXW FSR      B : FSR =0xC2, W =?<br>A : FSR =0xC2, W =0xC2                                                                                                                                                                                          |

---

**MOVLW                      Move Literal to W**


---

|                 |                                                                                            |                         |
|-----------------|--------------------------------------------------------------------------------------------|-------------------------|
| Syntax          | MOVLW k                                                                                    |                         |
| Operands        | k : 00h ~ FFh                                                                              |                         |
| Operation       | (W) ← k                                                                                    |                         |
| Status Affected | -                                                                                          |                         |
| OP-Code         | 0001 1001 kkkk kkkk                                                                        |                         |
| Description     | The eight-bit literal 'k' is loaded into W register. The don't cares will assemble as 0's. |                         |
| Cycle           | 1                                                                                          |                         |
| Example         | MOVLW 0x5A                                                                                 | B : W =?<br>A : W =0x5A |

---

**MOVWX                      Move W to "f"**


---

|                 |                                            |                                                    |
|-----------------|--------------------------------------------|----------------------------------------------------|
| Syntax          | MOVWX f                                    |                                                    |
| Operands        | f : 000h ~ 1FFh                            |                                                    |
| Operation       | (f) ← (W)                                  |                                                    |
| Status Affected | -                                          |                                                    |
| OP-Code         | ff00 0000 1fff ffff                        |                                                    |
| Description     | Move data from W register to register 'f'. |                                                    |
| Cycle           | 1                                          |                                                    |
| Example         | MOVWX REG1                                 | B : REG1 =0xFF, W =0x4F<br>A : REG1 =0x4F, W =0x4F |

---

**NOP                          No Operation**


---

|                 |                     |   |
|-----------------|---------------------|---|
| Syntax          | NOP                 |   |
| Operands        | -                   |   |
| Operation       | No Operation        |   |
| Status Affected | -                   |   |
| OP-Code         | 0000 0000 0000 0000 |   |
| Description     | No Operation        |   |
| Cycle           | 1                   |   |
| Example         | NOP                 | - |

---

**RET                          Return from Subroutine**


---

|                 |                                                                                                                                                |             |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| Syntax          | RET                                                                                                                                            |             |
| Operands        | -                                                                                                                                              |             |
| Operation       | PC ← TOS                                                                                                                                       |             |
| Status Affected | -                                                                                                                                              |             |
| OP-Code         | 0000 0000 0100 0000                                                                                                                            |             |
| Description     | Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction. |             |
| Cycle           | 2                                                                                                                                              |             |
| Example         | RET                                                                                                                                            | A : PC =TOS |

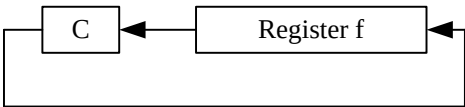
### RETI Return from Interrupt

|                 |                                                                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Syntax          | RETI                                                                                                                                          |
| Operands        | -                                                                                                                                             |
| Operation       | $PC \leftarrow TOS, GIE \leftarrow 1$                                                                                                         |
| Status Affected | -                                                                                                                                             |
| OP-Code         | 0000 0000 0110 0000                                                                                                                           |
| Description     | Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in to the PC. Interrupts are enabled. This is a two-cycle instruction. |
| Cycle           | 2                                                                                                                                             |
| Example         | RETI<br>A : PC =TOS, GIE =1                                                                                                                   |

### RETLW Return with Literal in W

|                 |                                                                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Syntax          | RETLW k                                                                                                                                                                 |
| Operands        | k : 00h ~ FFh                                                                                                                                                           |
| Operation       | $PC \leftarrow TOS, (W) \leftarrow k$                                                                                                                                   |
| Status Affected | -                                                                                                                                                                       |
| OP-Code         | 0001 1000 kkkk kkkk                                                                                                                                                     |
| Description     | The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction. |
| Cycle           | 2                                                                                                                                                                       |
| Example         | LCALL TABLE<br>:<br>TABLE ADDWX PCL, 1<br>RETLW k1<br>RETLW k2<br>:<br>RETLW kn<br>B : W =0x07<br>A : W =value of k8                                                    |

### RLX Rotate Left "f" through Carry

|                 |                                                                                                                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Syntax          | RLX f [,d]                                                                                                                                                                                        |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                                                                         |
| Operation       |                                                                                                                |
| Status Affected | C                                                                                                                                                                                                 |
| OP-Code         | ff00 1101 dfff ffff                                                                                                                                                                               |
| Description     | The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'. |
| Cycle           | 1                                                                                                                                                                                                 |
| Example         | RLX REG1, 0<br>B : REG1 =1110 0110, C =0<br>A : REG1 =1110 0110<br>W =1100 1100, C =1                                                                                                             |

### RRX Rotate Right "f" through Carry

|                 |                                                                                                                                                                                                    |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Syntax          | RRX f [,d]                                                                                                                                                                                         |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                                                                          |
| Operation       |                                                                                                                   |
| Status Affected | C                                                                                                                                                                                                  |
| OP-Code         | ff00 1100 dfff ffff                                                                                                                                                                                |
| Description     | The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. |
| Cycle           | 1                                                                                                                                                                                                  |
| Example         | RRX REG1, 0<br>B : REG1 =1110 0110, C =0<br>A : REG1 =1110 0110<br>W =0111 0011, C =0                                                                                                              |

### SLEEP Go into Power-down mode, Clock oscillation stops

|                 |                                                    |
|-----------------|----------------------------------------------------|
| Syntax          | SLEEP                                              |
| Operands        | -                                                  |
| Operation       | -                                                  |
| Status Affected | TO, PD                                             |
| OP-Code         | 001 1110 0000 0011                                 |
| Description     | Go into Power-down mode with the oscillator stops. |
| Cycle           | 1                                                  |
| Example         | SLEEP -                                            |

### SUBLW Subtract W from Literal

|                 |                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------|
| Syntax          | SUBLW k                                                                                                                      |
| Operands        | k : 00h ~ FFh                                                                                                                |
| Operation       | $(W) \leftarrow k - (W)$                                                                                                     |
| Status Affected | C, DC, Z                                                                                                                     |
| OP-Code         | 0001 1111 kkkk kkkk                                                                                                          |
| Description     | The W register is subtracted (2's complement method) from the eight-bit literal "k". The result is placed in the W register. |
| Cycle           | 1                                                                                                                            |
| Example         | SUBLW 0x15<br>B : W =0x25<br>A : W =0xF0                                                                                     |



**SUBWX**
**Subtract W from "f"**

|                 |                                                                                                                                                                             |  |                                                                        |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|------------------------------------------------------------------------|
| Syntax          | SUBWX f[,d]                                                                                                                                                                 |  |                                                                        |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                                                   |  |                                                                        |
| Operation       | (destination) $\leftarrow$ (f) - (W)                                                                                                                                        |  |                                                                        |
| Status Affected | C, DC, Z                                                                                                                                                                    |  |                                                                        |
| OP-Code         | ff00 0010 dfff ffff                                                                                                                                                         |  |                                                                        |
| Description     | Subtract (2's complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'. |  |                                                                        |
| Cycle           | 1                                                                                                                                                                           |  |                                                                        |
| Example         | SUBWX REG1, 1                                                                                                                                                               |  | B : REG1 =0x03, W =0x02, C=?, Z=?<br>A : REG1 =0x01, W =0x02, C=1, Z=0 |
|                 | SUBWX REG1, 1                                                                                                                                                               |  | B : REG1 =0x02, W =0x02, C=?, Z=?<br>A : REG1 =0x00, W =0x02, C=1, Z=1 |
|                 | SUBWX REG1, 1                                                                                                                                                               |  | B : REG1 =0x01, W =0x02, C=?, Z=?<br>A : REG1 =0xFF, W =0x02, C=0, Z=0 |

**SWAPX**
**Swap Nibbles in "f"**

|                 |                                                                                                                                                                |  |                                           |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--|-------------------------------------------|
| Syntax          | SWAPX f[,d]                                                                                                                                                    |  |                                           |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                                      |  |                                           |
| Operation       | (destination,7~4) $\leftarrow$ (f.3~0), (destination.3~0) $\leftarrow$ (f.7~4)                                                                                 |  |                                           |
| Status Affected | -                                                                                                                                                              |  |                                           |
| OP-Code         | ff00 1110 dfff ffff                                                                                                                                            |  |                                           |
| Description     | The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'. |  |                                           |
| Cycle           | 1                                                                                                                                                              |  |                                           |
| Example         | SWAPX REG1, 0                                                                                                                                                  |  | B : REG1 =0xA5<br>A : REG1 =0xA5, W =0x5A |
|                 |                                                                                                                                                                |  |                                           |

**TABRH**
**Return DPTR high byte to W**

|                 |                                                                                                                                    |  |                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------|--|------------------------|
| Syntax          | TABRH                                                                                                                              |  |                        |
| Operands        | -                                                                                                                                  |  |                        |
| Operation       | (W) $\leftarrow$ ROM[DPTR] high byte content, (TABR) $\leftarrow$ ROM[DPTR] high byte content, Where DPTR = {DPH[max:8], DPL[7:0]} |  |                        |
| Status Affected | -                                                                                                                                  |  |                        |
| OP-Code         | 0000 0000 0101 1000                                                                                                                |  |                        |
| Description     | The W and TABR register is loaded with high byte of ROM[DPTR]. This is a two-cycle instruction.                                    |  |                        |
| Cycle           | 2                                                                                                                                  |  |                        |
| Example         | MOVLW (TAB1&0xFF)                                                                                                                  |  |                        |
|                 | MOVWX DPL                                                                                                                          |  | ;Where DPL is register |
|                 | MOVLW (TAB1>>8)&0xFF                                                                                                               |  |                        |
|                 | MOVWX DPH                                                                                                                          |  | ;Where DPH is register |
|                 | TABRL                                                                                                                              |  | ;W =0x89, TABR=0x89    |
|                 | TABRH                                                                                                                              |  | ;W =0x37, TABR=0x37    |
|                 |                                                                                                                                    |  |                        |
|                 | ORG 0234H                                                                                                                          |  |                        |
|                 | TAB1:                                                                                                                              |  |                        |
|                 | DT 0x3789, 0x2277                                                                                                                  |  | ;ROM data 16 bits      |



| <b>TABRL</b>    | <b>Return DPTR low byte to W</b>                                                                              |                |                        |
|-----------------|---------------------------------------------------------------------------------------------------------------|----------------|------------------------|
| Syntax          | TABRL                                                                                                         |                |                        |
| Operands        | -                                                                                                             |                |                        |
| Operation       | (W) ← ROM[DPTR] low byte content, (TABR) ← ROM[DPTR] low byte content,<br>Where DPTR = {DPH[max:8], DPL[7:0]} |                |                        |
| Status Affected | -                                                                                                             |                |                        |
| OP-Code         | 0000 0000 0101 0000                                                                                           |                |                        |
| Description     | The W and TABR register is loaded with low byte of ROM[DPTR]. This is a two-cycle instruction.                |                |                        |
| Cycle           | 2                                                                                                             |                |                        |
| Example         | MOVLW                                                                                                         | (TAB1&0xFF)    |                        |
|                 | MOVWX                                                                                                         | DPL            | ;Where DPL is register |
|                 | MOVLW                                                                                                         | (TAB1>>8)&0xFF |                        |
|                 | MOVWX                                                                                                         | DPH            | ;Where DPH is register |
|                 | TABRL                                                                                                         |                | ;W =0x89, TABR=0x89    |
|                 | TABRH                                                                                                         |                | ;W =0x37, TABR=0x37    |
|                 | ORG 0234H                                                                                                     |                |                        |
|                 | TAB1:                                                                                                         |                |                        |
|                 | DT                                                                                                            | 0x3789, 0x2277 | ;ROM data 16 bits      |

| <b>TSTX</b>     | <b>Test if "f" is zero</b>                                  |
|-----------------|-------------------------------------------------------------|
| Syntax          | TSTX f                                                      |
| Operands        | f : 000h ~ 1FFh                                             |
| Operation       | Set Z flag if (f) is 0                                      |
| Status Affected | Z                                                           |
| OP-Code         | ff00 1000 1fff ffff                                         |
| Description     | If the content of register 'f' is 0, Zero flag is set to 1. |
| Cycle           | 1                                                           |
| Example         | TSTX REG1<br>B : REG1 =0, Z =?<br>A : REG1 =0, Z =1         |

| <b>XORLW</b>    | <b>Exclusive OR Literal with W</b>                                                                                |                              |
|-----------------|-------------------------------------------------------------------------------------------------------------------|------------------------------|
| Syntax          | XORLW k                                                                                                           |                              |
| Operands        | k : 00h ~ FFh                                                                                                     |                              |
| Operation       | (W) ← (W) XOR k                                                                                                   |                              |
| Status Affected | Z                                                                                                                 |                              |
| OP-Code         | 0001 1101 kkkk kkkk                                                                                               |                              |
| Description     | The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register. |                              |
| Cycle           | 1                                                                                                                 |                              |
| Example         | XORLW 0xAF                                                                                                        | B : W = 0xB5<br>A : W = 0x1A |

| <b>XORWX</b>    | <b>Exclusive OR W with "f"</b>                                                                                                                                              |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Syntax          | XORWX f [,d]                                                                                                                                                                |
| Operands        | f : 000h ~ 1FFh, d : 0, 1                                                                                                                                                   |
| Operation       | (destination) $\leftarrow$ (W) XOR (f)                                                                                                                                      |
| Status Affected | Z                                                                                                                                                                           |
| OP-Code         | ff00 0110 dfff ffff                                                                                                                                                         |
| Description     | Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'. |
| Cycle           | 1                                                                                                                                                                           |
| Example         | <div>XORWX REG1, 1</div> <div>           B : REG1 =0xAF, W =0xB5<br/>           A : REG1 =0x1A, W =0xB5         </div>                                                      |

## ELECTRICAL CHARACTERISTICS

All of the parameters are based on the characteristics of tested samples.

### 1. Absolute Maximum Ratings

(T<sub>A</sub> = 25°C)

| Parameter                       | Rating                                       | Unit |
|---------------------------------|----------------------------------------------|------|
| Supply voltage                  | V <sub>SS</sub> -0.3 to V <sub>SS</sub> +5.5 | V    |
| Input voltage                   | V <sub>SS</sub> -0.3 to V <sub>CC</sub> +0.3 |      |
| Output voltage                  | V <sub>SS</sub> -0.3 to V <sub>CC</sub> +0.3 |      |
| Output current high per 1 PIN   | -25                                          | mA   |
| Output current high per all PIN | -80                                          |      |
| Output current low per 1 PIN    | +30                                          |      |
| Output current low per all PIN  | +150                                         |      |
| Maximum operating voltage       | 5.5                                          | V    |
| Operating temperature           | -40 to +105                                  | °C   |
| Storage temperature             | -65 to +150                                  |      |

### 2. DC Characteristics

(T<sub>A</sub> = 25°C, V<sub>CC</sub> = 5.0V, unless otherwise specified)

| Parameter                           | Symbol           | Conditions                                                           |                                                   | Min.               | Typ. | Max.               | Unit |
|-------------------------------------|------------------|----------------------------------------------------------------------|---------------------------------------------------|--------------------|------|--------------------|------|
| Operating Voltage                   | V <sub>cc</sub>  | Fsys = 20 MHz (FXT)                                                  |                                                   | 2.6                | —    | 5.5                | V    |
|                                     |                  | Fsys = 16 MHz (FIRC)(RDCTL=4ns)<br>(PWMCKS=FIRC*1)(-40°C ~ 105°C)    |                                                   | 2.3                | —    | 5.5                | V    |
|                                     |                  | Fsys = 8 MHz (FIRC/2) (RDCTL=4ns)<br>(PWMCKS=FIRC*1) (-40°C ~ 105°C) |                                                   | 1.6                | —    | 5.5                | V    |
| Input High Voltage                  | V <sub>IH</sub>  | All Input                                                            | V <sub>CC</sub> = 3.0~5.0V                        | 0.6V <sub>CC</sub> | —    | V <sub>CC</sub>    | V    |
| Input Low Voltage                   | V <sub>IL</sub>  | All Input                                                            | V <sub>CC</sub> = 3.0~5.0V                        | V <sub>SS</sub>    | —    | 0.2V <sub>CC</sub> | V    |
| I/O port<br>Source Current          | I <sub>OH</sub>  | All I/O pin                                                          | V <sub>CC</sub> = 5.0V,<br>V <sub>OH</sub> = 4.5V | 6.4                | 12.8 | —                  | mA   |
|                                     |                  |                                                                      | V <sub>CC</sub> = 3.0V,<br>V <sub>OH</sub> = 2.7V | 2.7                | 5.4  | —                  |      |
| I/O port<br>Sink Current            | I <sub>OL</sub>  | All I/O pin except<br>PA7<br>(HSINK=1)                               | V <sub>CC</sub> = 5.0V,<br>V <sub>OL</sub> = 0.5V | 31                 | 62   | —                  | mA   |
|                                     |                  |                                                                      | V <sub>CC</sub> = 3.0V,<br>V <sub>OL</sub> = 0.3V | 14                 | 28   | —                  |      |
|                                     |                  | All I/O pin<br>(HSINK=0)                                             | V <sub>CC</sub> = 5.0V,<br>V <sub>OL</sub> = 0.5V | 18                 | 36   | —                  | mA   |
|                                     |                  |                                                                      | V <sub>CC</sub> = 3.0V,<br>V <sub>OL</sub> = 0.3V | 8                  | 16   | —                  |      |
| Input Leakage Current<br>(pin high) | I <sub>ILH</sub> | All Input                                                            | V <sub>IN</sub> = V <sub>CC</sub>                 | —                  | —    | 1                  | μA   |
| Input Leakage Current<br>(pin low)  | I <sub>ILL</sub> | All Input                                                            | V <sub>IN</sub> = 0V                              | —                  | —    | −1                 | μA   |

| Parameter                                     | Symbol           | Conditions                                                   | Min.                   | Typ. | Max.  | Unit |
|-----------------------------------------------|------------------|--------------------------------------------------------------|------------------------|------|-------|------|
| Power Supply Current<br>(No Load)<br>(ATD On) | I <sub>CC</sub>  | FAST mode<br>FXT 20 MHz<br>POR/LVR On                        | V <sub>CC</sub> = 5.0V | –    | 4.9   | –    |
|                                               |                  |                                                              | V <sub>CC</sub> = 3.0V | –    | 2.6   | –    |
|                                               |                  | FAST mode<br>FIRC 16 MHz                                     | V <sub>CC</sub> = 5.0V | –    | 4.3   | –    |
|                                               |                  |                                                              | V <sub>CC</sub> = 3.0V | –    | 2.5   | –    |
|                                               |                  | FAST mode<br>FIRC 8 MHz                                      | V <sub>CC</sub> = 5.0V | –    | 3     | –    |
|                                               |                  |                                                              | V <sub>CC</sub> = 3.0V | –    | 1.7   | –    |
|                                               |                  | FAST mode<br>FIRC 4 MHz                                      | V <sub>CC</sub> = 5.0V | –    | 2     | –    |
|                                               |                  |                                                              | V <sub>CC</sub> = 3.0V | –    | 1.3   | –    |
|                                               |                  | FAST mode<br>FIRC 2 MHz                                      | V <sub>CC</sub> = 5.0V | –    | 1.5   | –    |
|                                               |                  |                                                              | V <sub>CC</sub> = 3.0V | –    | 0.89  | –    |
|                                               |                  | SLOW mode<br>SXT 32 KHz<br>FIRC STOP<br>POR/LVR On           | V <sub>CC</sub> = 5.0V | –    | 0.09  | –    |
|                                               |                  |                                                              | V <sub>CC</sub> = 3.0V | –    | 0.054 | –    |
|                                               |                  | SLOW mode<br>SIRC div1<br>FIRC STOP<br>POR/LVR On            | V <sub>CC</sub> = 5.0V | –    | 0.083 | –    |
|                                               |                  |                                                              | V <sub>CC</sub> = 3.0V | –    | 0.055 | –    |
|                                               |                  | SLOW mode<br>SIRC div1<br>FIRC STOP<br>POR/LVR Off<br>ATD On | V <sub>CC</sub> = 5.0V | –    | 0.026 | –    |
|                                               |                  |                                                              | V <sub>CC</sub> = 3.0V | –    | 0.014 | –    |
| Pull-up Resistor                              | R <sub>UP</sub>  | V <sub>IN</sub> = 0 V<br>Ports A, B, D                       | V <sub>CC</sub> = 5.0V | –    | 36    | –    |
|                                               |                  |                                                              | V <sub>CC</sub> = 3.0V | –    | 37.5  | –    |
|                                               |                  |                                                              |                        |      |       |      |
|                                               |                  |                                                              |                        |      |       |      |
| POR Voltage                                   | V <sub>POR</sub> | T <sub>A</sub> = 25°C                                        | 1.48                   | 1.63 | 1.78  | V    |

### 3. Clock Timing

The value of this parameter is based on the characteristics of tested samples.

| Parameter          | Condition                                                   | Min.  | Typ. | Max.  | Unit |
|--------------------|-------------------------------------------------------------|-------|------|-------|------|
| FIRC Frequency (*) | T <sub>A</sub> = -40°C ~ 105°C V <sub>CC</sub> = 3.0 ~ 5.0V | -4%   | 16   | +3%   | MHz  |
|                    | T <sub>A</sub> = -40°C ~ 105°C V <sub>CC</sub> = 4.0 V      | -2.5% | 16   | +2%   |      |
|                    | T <sub>A</sub> = 0°C ~ 70°C V <sub>CC</sub> = 4.0 V         | -2%   | 16   | +1.5% |      |
|                    | T <sub>A</sub> = 25°C V <sub>CC</sub> = 3.5 ~ 5.0 V         | -1%   | 16   | +1%   |      |
| SIRC Frequency     | T <sub>A</sub> = 25°C V <sub>CC</sub> = 5.0 V               | -2%   | 65.5 | +2%   | KHz  |

(\*) FIRC frequency can be divided by 1/2/4/8.

### 4. Reset Timing Characteristics

(T<sub>A</sub> = 25°C)

| Parameter             | Conditions                               | Min. | Typ. | Max. | Unit          |
|-----------------------|------------------------------------------|------|------|------|---------------|
| RESET Input Low width | Input $V_{CC} = 5.0 \text{ V} \pm 10 \%$ | –    | 15.3 | –    | $\mu\text{s}$ |
| WDT time              | $V_{CC} = 5.0 \text{ V}$ , WDTPSC = 11b  | –    | 2001 | –    | ms            |
| WKT time              | $V_{CC} = 5.0 \text{ V}$ , WKTPSC = 11b  | –    | 125  | –    | ms            |
| CPU start up time     | $V_{CC} = 5.0 \text{ V}$                 | –    | 34.8 | –    | ms            |

## 5. LVR Circuit Characteristics

( $T_A = 25^\circ\text{C}$ )

| Parameter                  | Symbol         | Condition                | LVRS | Min. | Typ. | Max. | Unit          |
|----------------------------|----------------|--------------------------|------|------|------|------|---------------|
| LVR Voltage                | $LVR_{th}$     | $T_A = 25^\circ\text{C}$ | 0001 | –    | 1.73 | –    | V             |
|                            |                |                          | 0010 | –    | 1.85 | –    |               |
|                            |                |                          | 0011 | –    | 1.98 | –    |               |
|                            |                |                          | 0100 | –    | 2.11 | –    |               |
|                            |                |                          | 0101 | –    | 2.23 | –    |               |
|                            |                |                          | 0110 | –    | 2.36 | –    |               |
|                            |                |                          | 0111 | –    | 2.49 | –    |               |
|                            |                |                          | 1000 | –    | 2.61 | –    |               |
|                            |                |                          | 1001 | –    | 2.74 | –    |               |
|                            |                |                          | 1010 | –    | 2.87 | –    |               |
|                            |                |                          | 1011 | –    | 2.99 | –    |               |
|                            |                |                          | 1100 | –    | 3.12 | –    |               |
|                            |                |                          | 1101 | –    | 3.25 | –    |               |
|                            |                |                          | 1110 | –    | 3.37 | –    |               |
|                            |                |                          | 1111 | –    | 3.50 | –    |               |
| LVR Hysteresis Window      | $V_{HYS\_LVR}$ | $T_A = 25^\circ\text{C}$ |      | –    | 0    | –    | mV            |
| Low Voltage Detection time | $T_{LVR}$      | $T_A = 25^\circ\text{C}$ |      | 100  | –    | –    | $\mu\text{s}$ |

## 6. LVD Circuit Characteristics

( $T_A = 25^\circ\text{C}$ )

| Parameter   | Symbol     | Condition                | LVDS | Min.    | Typ. | Max. | Unit |
|-------------|------------|--------------------------|------|---------|------|------|------|
| LVD Voltage | $LVD_{th}$ | $T_A = 25^\circ\text{C}$ | 0000 | disable |      |      | V    |
|             |            |                          | 0001 | –       | 1.73 | –    |      |
|             |            |                          | 0010 | –       | 1.85 | –    |      |
|             |            |                          | 0011 | –       | 1.98 | –    |      |
|             |            |                          | 0100 | –       | 2.11 | –    |      |
|             |            |                          | 0101 | –       | 2.23 | –    |      |
|             |            |                          | 0110 | –       | 2.36 | –    |      |
|             |            |                          | 0111 | –       | 2.49 | –    |      |
|             |            |                          | 1000 | –       | 2.61 | –    |      |
|             |            |                          | 1001 | –       | 2.74 | –    |      |
|             |            |                          | 1010 | –       | 2.87 | –    |      |
|             |            |                          | 1011 | –       | 2.99 | –    |      |
|             |            |                          | 1100 | –       | 3.12 | –    |      |
|             |            |                          | 1101 | –       | 3.25 | –    |      |
|             |            |                          | 1110 | –       | 3.37 | –    |      |
|             |            |                          | 1111 | –       | 3.50 | –    |      |

|                            |                |                            |  |     |    |   |               |
|----------------------------|----------------|----------------------------|--|-----|----|---|---------------|
| LVD Hysteresis Window      | $V_{HYS\_LVD}$ | LVDHYS = 0                 |  | –   | 0  | – | mV            |
|                            |                | LVDHYS = 1                 |  | –   | 80 | – |               |
| Low Voltage Detection time | $T_{LVD}$      | $T_A = 25^{\circ}\text{C}$ |  | 100 | –  | – | $\mu\text{s}$ |

## 7. ADC Electrical Characteristics

( $T_A = 25^{\circ}\text{C}$ ,  $V_{CC} = 3.0\text{V}$  to  $5.5\text{V}$ ,  $V_{SS} = 0\text{V}$ )

| Parameter                                                                                                                                  | Conditions                                                                               | Min.     | Typ.         | Max.      | Units         |
|--------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|----------|--------------|-----------|---------------|
| Total Accuracy                                                                                                                             | $V_{CC} = 5.0\text{V}$ , $V_{SS} = 0\text{V}$ , $F_{ADC} = 1\text{ MHz}$                 | –        | $\pm 3$      | –         | LSB           |
| Integral Non-Linearity                                                                                                                     |                                                                                          | –        | $\pm 3.2$    | –         |               |
| Differential Non-Linearity                                                                                                                 |                                                                                          | –        | $\pm 1$      | $\pm 4$   |               |
| Max Input Clock freq. ( $F_{ADC}$ )                                                                                                        | Source impedance ( $R_s < 10\text{K ohm}$ )                                              | –        | –            | 2         | MHz           |
|                                                                                                                                            | Source impedance ( $R_s < 20\text{K ohm}$ )                                              | –        | –            | 1         |               |
|                                                                                                                                            | Source impedance ( $R_s < 50\text{K ohm}$ )                                              | –        | –            | 0.5       |               |
|                                                                                                                                            | Source is VBG (ADCHS=01110b)                                                             | –        | –            | 2         |               |
| Conversion Time                                                                                                                            | $F_{ADC} = 1\text{ MHz}$                                                                 | –        | 42           | –         | $\mu\text{s}$ |
| Conversion Current                                                                                                                         | $V_{CC}=5\text{V}$ , ADVREFS=00b, ADVREF1P2=0                                            | –        | 0.45         | –         | mA            |
|                                                                                                                                            | $V_{CC}=4\text{V}$ , ADVREFS=01b                                                         | –        | 0.6          | –         |               |
| BandGap Voltage Reference (1.2V $V_{BG}$ )<br>ADC reference voltage ( $V_{REF}$ )<br>(ADVREFS=00b, ADVREF1P2=1b)<br>(No power disturbance) | $25^{\circ}\text{C}$ , $V_{CC} = 3.0\text{V} \sim 5.0\text{V}$                           | -1%      | 1.20         | +1%       | V             |
|                                                                                                                                            | $25^{\circ}\text{C} \sim 105^{\circ}\text{C}$ , $V_{CC} = 3.0\text{V} \sim 5.0\text{V}$  | -1%      | 1.20         | +1.5%     | V             |
|                                                                                                                                            | $-20^{\circ}\text{C} \sim 105^{\circ}\text{C}$ , $V_{CC} = 3.0\text{V} \sim 5.0\text{V}$ | -2%      | 1.20         | +1.5%     | V             |
| BandGap Voltage Reference (2.56V $V_{BG}$ )<br>ADC reference voltage ( $V_{REF}$ )<br>(ADVREFS=01b)<br>(No power disturbance)              | $25^{\circ}\text{C}$ , $V_{CC} = 3.0\text{V} \sim 5.5\text{V}$                           | -1.2%    | 2.56         | +1.2%     | V             |
|                                                                                                                                            | $-20^{\circ}\text{C} \sim 105^{\circ}\text{C}$ , $V_{CC} = 3.0\text{V} \sim 5.5\text{V}$ | -2.5%    | 2.56         | +2%       | V             |
| BandGap Voltage Reference (2V $V_{BG}$ )<br>ADC reference voltage ( $V_{REF}$ )<br>(ADVREFS=11b)<br>(No power disturbance)                 | $25^{\circ}\text{C}$ , $V_{CC} = 3.0\text{V} \sim 5.5\text{V}$                           | -1.2%    | 2            | +1.2%     | V             |
|                                                                                                                                            | $-20^{\circ}\text{C} \sim 105^{\circ}\text{C}$ , $V_{CC} = 3.0\text{V} \sim 5.5\text{V}$ | -2.5%    | 2            | +2%       | V             |
| $V_{CC}/4$ reference voltage                                                                                                               | $25^{\circ}\text{C}$ , $V_{CC} = 3.0\text{V} \sim 5.5\text{V}$                           | -1%      | $0.25V_{CC}$ | +1%       | V             |
| Input Voltage                                                                                                                              | –                                                                                        | $V_{SS}$ | –            | $V_{REF}$ | V             |

## 8. EEPROM Characteristics

| Parameter                  | Conditions                                                                         | Min. | Typ. | Max. | Units  |
|----------------------------|------------------------------------------------------------------------------------|------|------|------|--------|
| Write Voltage $V_{EEWR}$   | $-40^{\circ}\text{C} \sim 105^{\circ}$<br>$F_{sys}=FRC/1$ , $V_{CC}/47\mu\text{F}$ | 2.5  |      | 5.5  | V      |
| Read Voltage $V_{EERD}$    | $-40^{\circ}\text{C} \sim 105^{\circ}$<br>$F_{sys}=FRC/1$ , $V_{CC}/47\mu\text{F}$ | 2.5  |      | 5.5  | V      |
| Write Endurance $N_{EE}^*$ | $V_{CC} = 2.5 \sim 5.5\text{V}$ , $-40^{\circ}\text{C} \sim 105^{\circ}\text{C}$   | 20K  | –    | –    | cycles |
|                            | $V_{CC} = 3.0 \sim 5.5\text{V}$ , $-40^{\circ}\text{C} \sim 105^{\circ}\text{C}$   | 30K  | –    | –    |        |
|                            | $V_{CC} = 2.5 \sim 5.5\text{V}$ , $-20^{\circ}\text{C} \sim 85^{\circ}\text{C}$    | 30K  | –    | –    |        |
| Write Time $T_{EEWR}$      | $V_{CC} = 5.0\text{V}$ , $25^{\circ}\text{C}$ , WDT disable                        |      | 1.5  |      | ms     |
|                            | $V_{CC} = 2.5\text{V}$ , $25^{\circ}\text{C}$ , WDT disable                        |      | 4    |      |        |
|                            | $V_{CC} = 3.0\text{V}$ , $105^{\circ}\text{C}$ , WDT disable                       |      | 15   |      |        |
| Data Retention $Y_{RET}$   |                                                                                    | 10   |      |      | Year   |

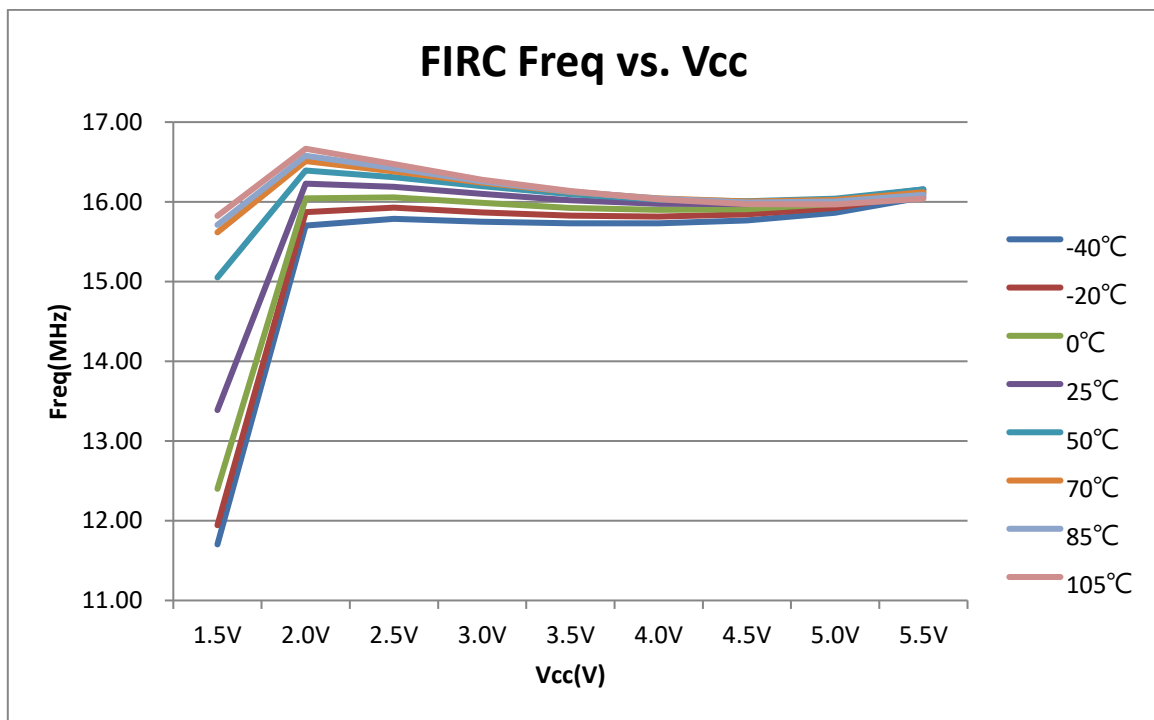
\*: The value of this parameter is based on the characteristics of tested samples.

## 9. Comparator Characteristics

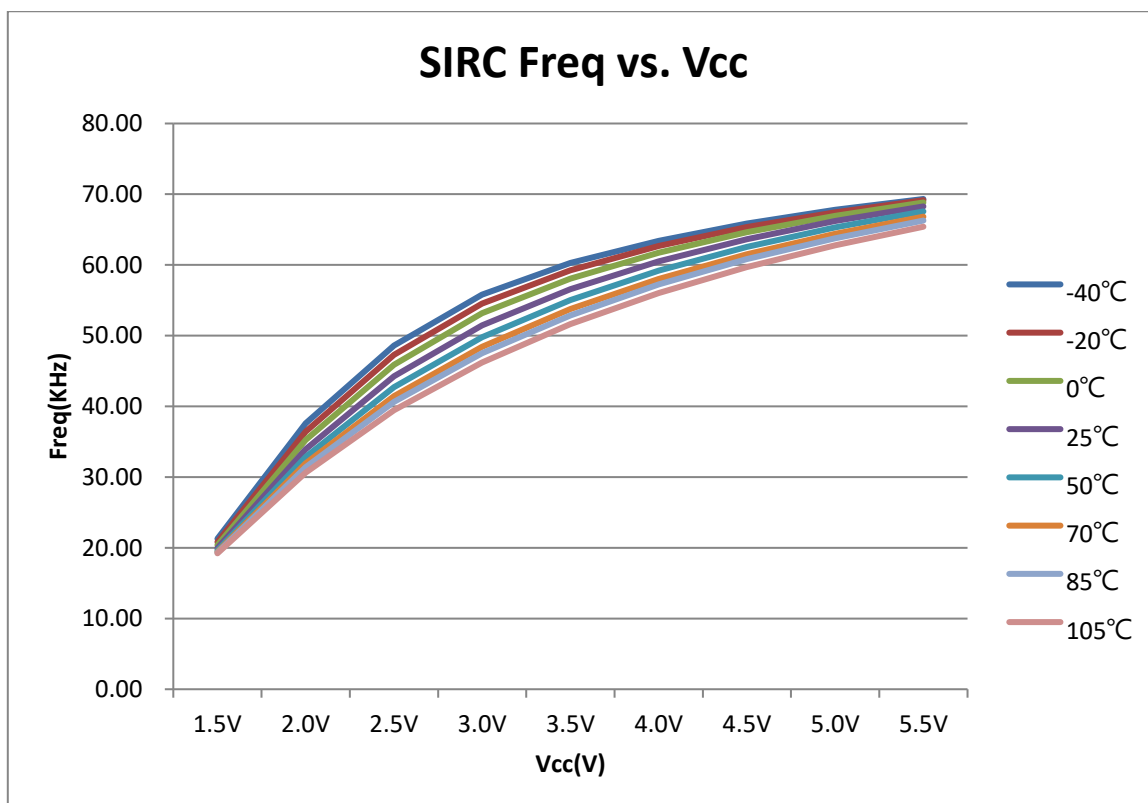
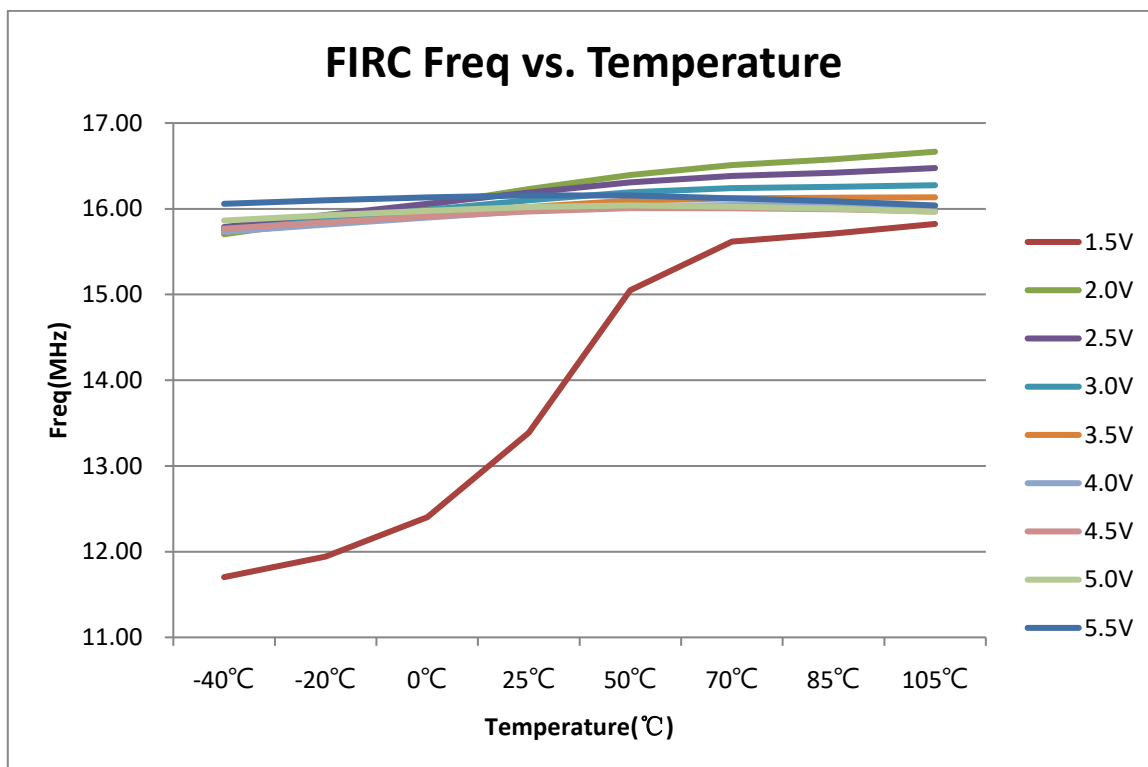
( $T_A = 25^\circ\text{C}$ ,  $V_{CC} = 3.0\text{V}$  to  $5.5\text{V}$ ,  $V_{SS} = 0\text{V}$ )

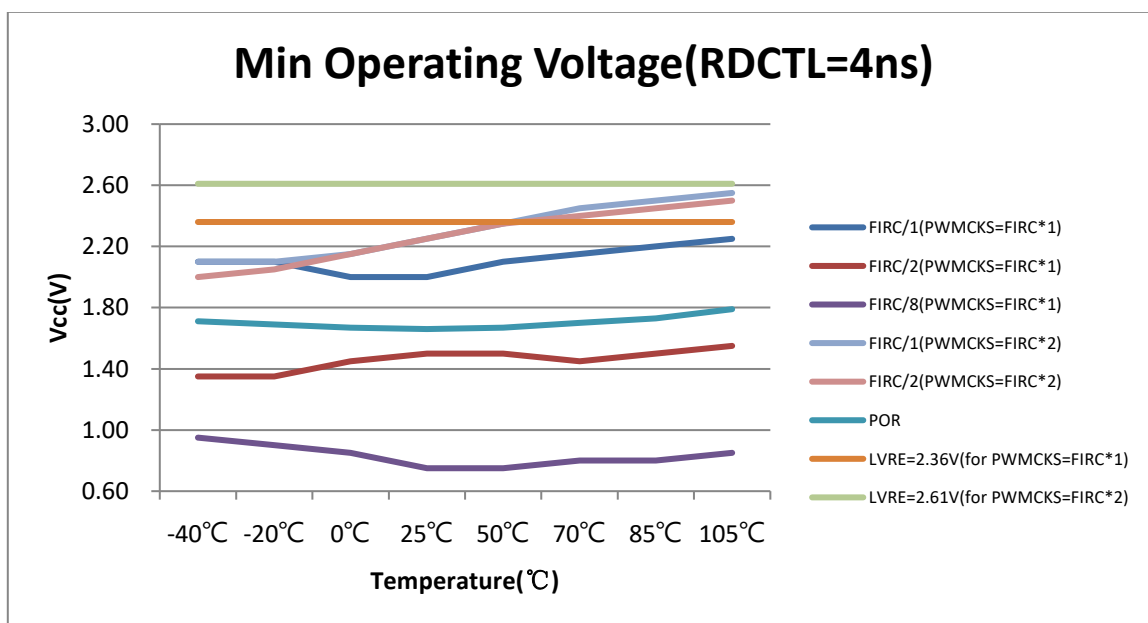
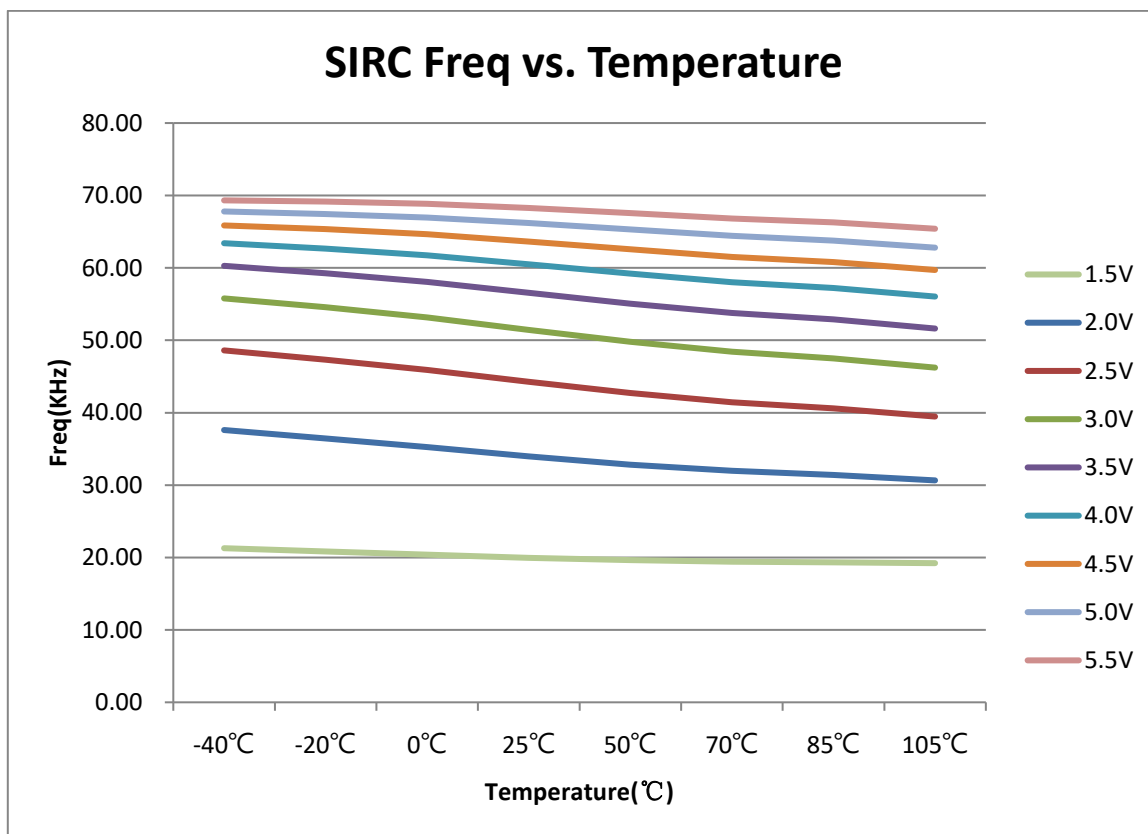
| Parameter         | Conditions             | Min. | Typ. | Max.         | Units         |
|-------------------|------------------------|------|------|--------------|---------------|
| Power supply      | –                      | 2.2  | –    | 5.5          | V             |
| Quiescent Current | $V_{CC} = 5.0\text{V}$ | –    | 100  | –            | $\mu\text{A}$ |
| DAC Current       | $V_{CC} = 5.0\text{V}$ | 60   | –    | 220          | $\mu\text{A}$ |
| $V_{OS\_CMP}$     | $V_{CC} = 5.0\text{V}$ | -15  | –    | 15           | mV            |
| $V_{CM\_CMP}$     | $V_{CC} = 5.0\text{V}$ | 0    | –    | $V_{CC}-0.5$ | V             |
| $V_{HYS\_CMP}$    | $V_{CC} = 5.0\text{V}$ | -    | 25   | -            | mV            |

## 10. Characteristics Graphs

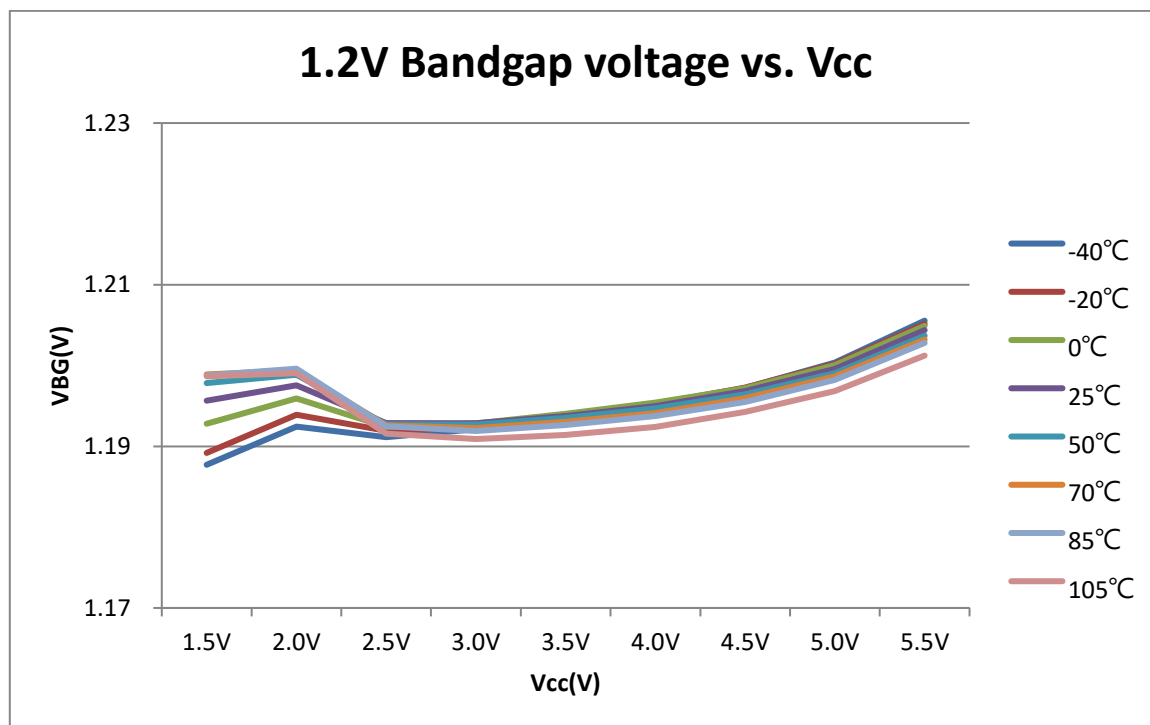








Note: The user must switch RDCTL to “4ns” to enhance the performance of minimal operating voltage

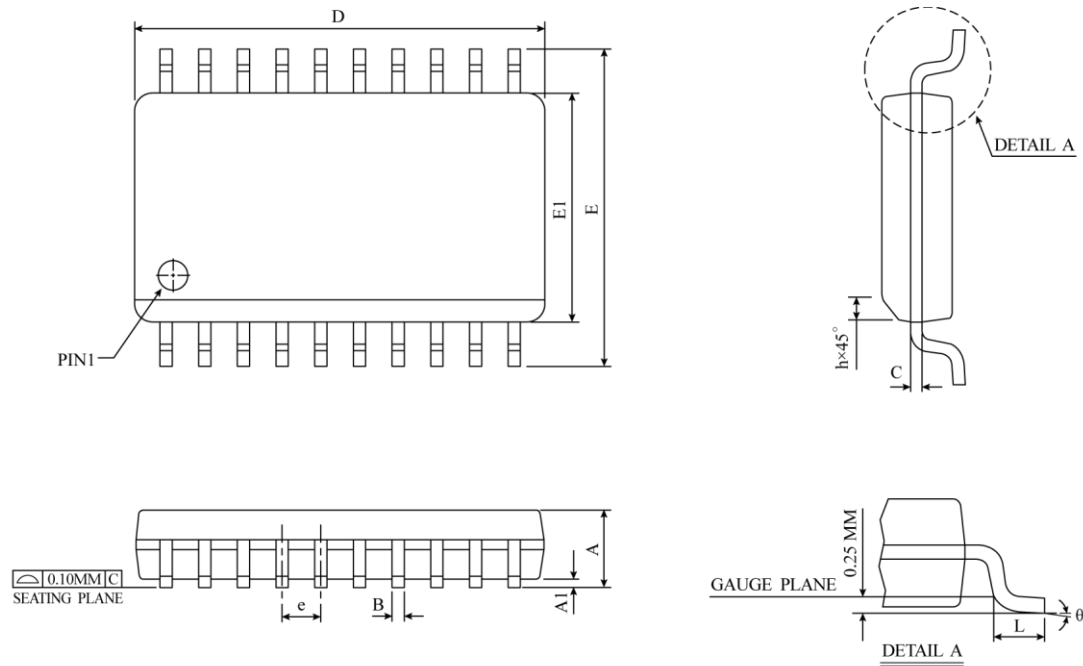


## PACKAGING INFORMATION

Please note that the package information provided is for reference only. Since this information is frequently updated, users can contact Sales to consult the latest package information and stocks.

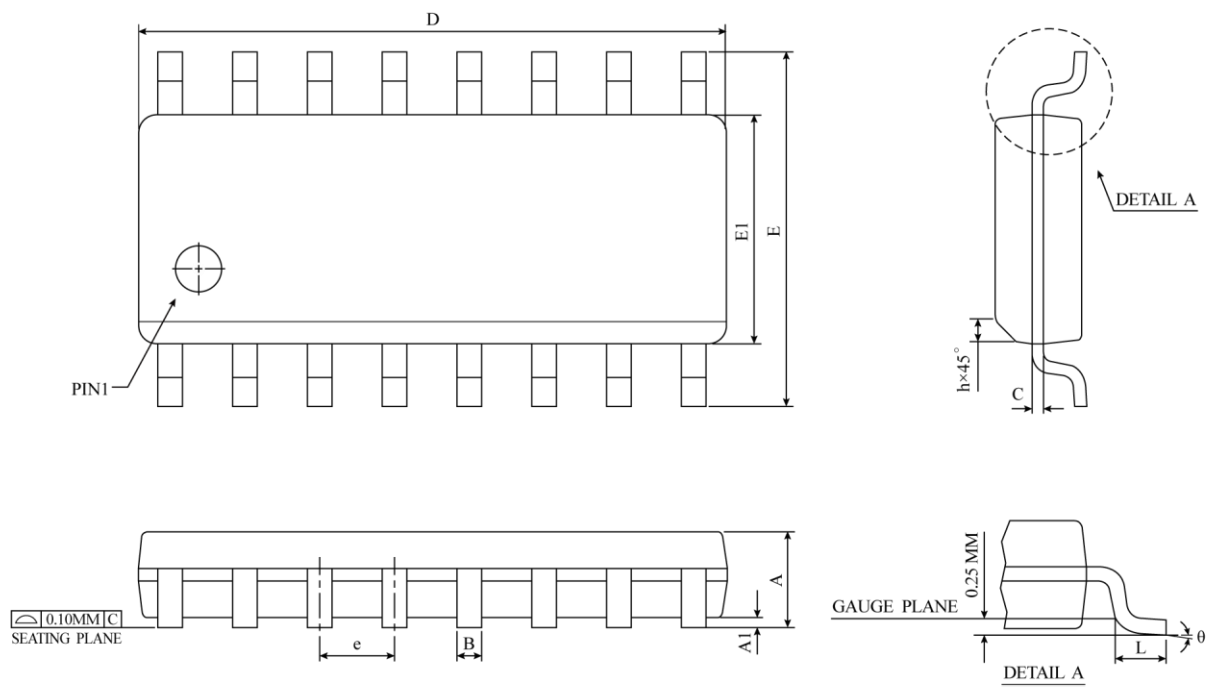
The ordering information:

| Ordering number | Package                                     |
|-----------------|---------------------------------------------|
| TM56E64223S     | SOP 20-pin (300 mil)                        |
| TM56E64222S     | SOP 16-pin (150 mil)                        |
| TM56E6422ES     | SOP 14-pin (150 mil)                        |
| TM56E64221S     | SOP 8-pin (150 mil)                         |
| TM56E64223T     | TSSOP 20-pin (173 mil)                      |
| TM56E64223Q     | QFN 20-pin (3x3x0.74-0.4 mm)<br>(L=0.25 mm) |

**SOP-20 (300 mil) Package Dimension**


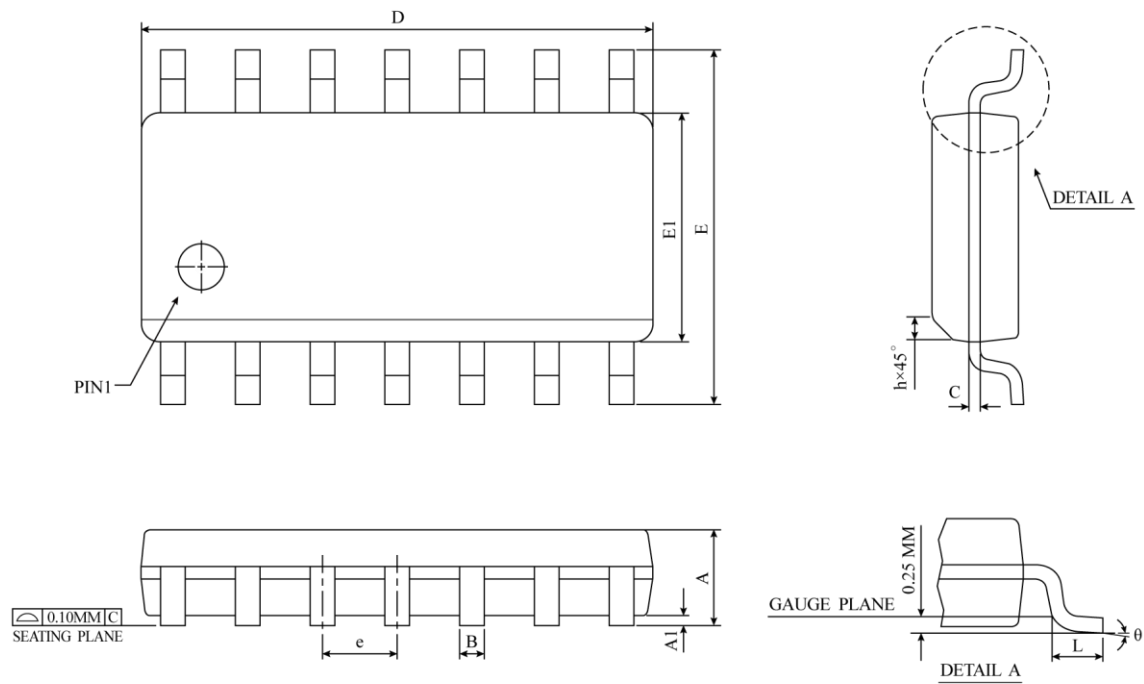
| SYMBOL | DIMENSION IN MM |       |       | DIMENSION IN INCH |        |        |
|--------|-----------------|-------|-------|-------------------|--------|--------|
|        | MIN             | NOM   | MAX   | MIN               | NOM    | MAX    |
| A      | 2.35            | 2.50  | 2.65  | 0.0926            | 0.0985 | 0.1043 |
| A1     | 0.10            | 0.20  | 0.30  | 0.0040            | 0.0079 | 0.0118 |
| B      | 0.33            | 0.42  | 0.51  | 0.0130            | 0.0165 | 0.0200 |
| C      | 0.23            | 0.28  | 0.32  | 0.0091            | 0.0108 | 0.0125 |
| D      | 12.60           | 12.80 | 13.00 | 0.4961            | 0.5040 | 0.5118 |
| E      | 10.00           | 10.33 | 10.65 | 0.3940            | 0.4425 | 0.4910 |
| E1     | 7.40            | 7.50  | 7.60  | 0.2914            | 0.2953 | 0.2992 |
| e      | 1.27 BSC        |       |       | 0.050 BSC         |        |        |
| h      | 0.25            | 0.50  | 0.75  | 0.0100            | 0.0195 | 0.0290 |
| L      | 0.40            | 0.84  | 1.27  | 0.0160            | 0.0330 | 0.0500 |
| θ      | 0°              | 4°    | 8°    | 0°                | 4°     | 8°     |
| JEDEC  | MS-013 (AC)     |       |       |                   |        |        |

△ \* NOTES : DIMENSION "D" DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.  
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL  
NOT EXCEED 0.15 MM ( 0.006 INCH ) PER SIDE.

**SOP-16 (150 mil) Package Dimension**


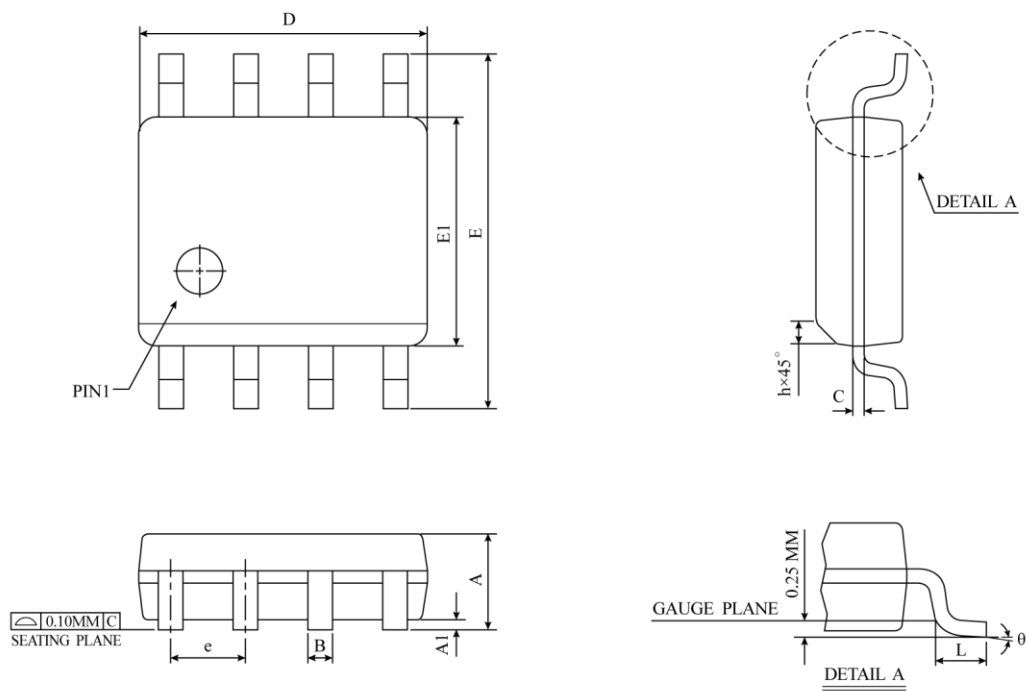
| SYMBOL | DIMENSION IN MM |      |       | DIMENSION IN INCH |        |        |
|--------|-----------------|------|-------|-------------------|--------|--------|
|        | MIN             | NOM  | MAX   | MIN               | NOM    | MAX    |
| A      | 1.35            | 1.55 | 1.75  | 0.0532            | 0.0610 | 0.0688 |
| A1     | 0.10            | 0.18 | 0.25  | 0.0040            | 0.0069 | 0.0098 |
| B      | 0.33            | 0.42 | 0.51  | 0.0130            | 0.0165 | 0.0200 |
| C      | 0.19            | 0.22 | 0.25  | 0.0075            | 0.0087 | 0.0098 |
| D      | 9.80            | 9.90 | 10.00 | 0.3859            | 0.3898 | 0.3937 |
| E      | 5.80            | 6.00 | 6.20  | 0.2284            | 0.2362 | 0.2440 |
| E1     | 3.80            | 3.90 | 4.00  | 0.1497            | 0.1536 | 0.1574 |
| e      | 1.27 BSC        |      |       | 0.050 BSC         |        |        |
| h      | 0.25            | 0.38 | 0.50  | 0.0099            | 0.0148 | 0.0196 |
| L      | 0.40            | 0.84 | 1.27  | 0.0160            | 0.0330 | 0.0500 |
| θ      | 0°              | 4°   | 8°    | 0°                | 4°     | 8°     |
| JEDEC  | MS-012 (AC)     |      |       |                   |        |        |

△ \* NOTES : DIMENSION "D" DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.  
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL  
NOT EXCEED 0.15 MM ( 0.006 INCH ) PER SIDE.

**SOP-14 (150 mil) Package Dimension**


| SYMBOL | DIMENSION IN MM |      |      | DIMENSION IN INCH |        |        |
|--------|-----------------|------|------|-------------------|--------|--------|
|        | MIN             | NOM  | MAX  | MIN               | NOM    | MAX    |
| A      | 1.35            | 1.55 | 1.75 | 0.0532            | 0.0610 | 0.0688 |
| A1     | 0.10            | 0.18 | 0.25 | 0.0040            | 0.0069 | 0.0098 |
| B      | 0.33            | 0.42 | 0.51 | 0.0130            | 0.0165 | 0.0200 |
| C      | 0.19            | 0.22 | 0.25 | 0.0075            | 0.0087 | 0.0098 |
| D      | 8.55            | 8.65 | 8.75 | 0.3367            | 0.3410 | 0.3444 |
| E      | 5.80            | 6.00 | 6.20 | 0.2284            | 0.2362 | 0.2440 |
| E1     | 3.80            | 3.90 | 4.00 | 0.1497            | 0.1536 | 0.1574 |
| e      | 1.27 BSC        |      |      | 0.050 BSC         |        |        |
| h      | 0.25            | 0.38 | 0.50 | 0.0099            | 0.0148 | 0.0196 |
| L      | 0.40            | 0.84 | 1.27 | 0.0160            | 0.0330 | 0.0500 |
| θ      | 0°              | 4°   | 8°   | 0°                | 4°     | 8°     |
| JEDEC  | MS-012 (AB)     |      |      |                   |        |        |

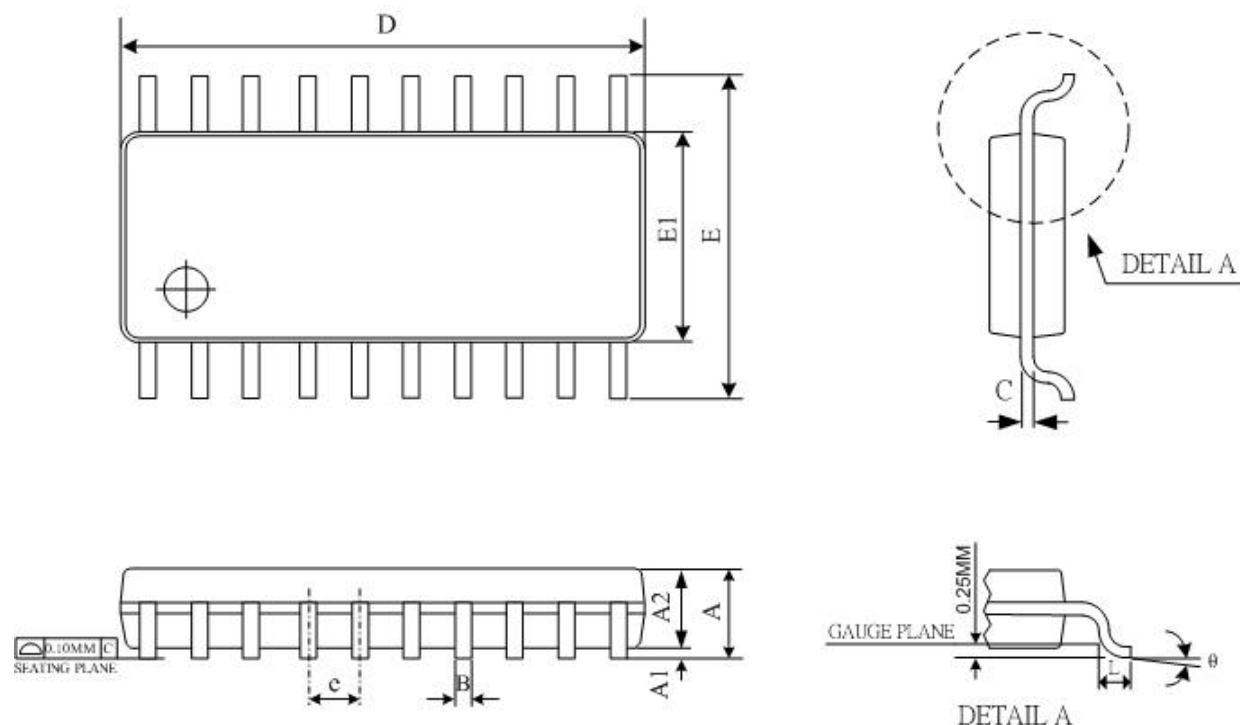
△ \*NOTES : DIMENSION "D" DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.  
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL  
NOT EXCEED 0.15 MM ( 0.006 INCH ) PER SIDE.

**SOP-8 (150 mil) Package Dimension**


| SYMBOL | DIMENSION IN MM |      |      | DIMENSION IN INCH |        |        |
|--------|-----------------|------|------|-------------------|--------|--------|
|        | MIN             | NOM  | MAX  | MIN               | NOM    | MAX    |
| A      | 1.35            | 1.55 | 1.75 | 0.0532            | 0.0610 | 0.0688 |
| A1     | 0.10            | 0.18 | 0.25 | 0.0040            | 0.0069 | 0.0098 |
| B      | 0.33            | 0.42 | 0.51 | 0.0130            | 0.0165 | 0.0200 |
| C      | 0.19            | 0.22 | 0.25 | 0.0075            | 0.0087 | 0.0098 |
| D      | 4.80            | 4.90 | 5.00 | 0.1890            | 0.1939 | 0.1988 |
| E      | 5.80            | 6.00 | 6.20 | 0.2284            | 0.2362 | 0.2440 |
| E1     | 3.80            | 3.90 | 4.00 | 0.1497            | 0.1536 | 0.1574 |
| e      | 1.27 BSC        |      |      | 0.050 BSC         |        |        |
| h      | 0.25            | 0.38 | 0.50 | 0.0099            | 0.0148 | 0.0196 |
| L      | 0.40            | 0.84 | 1.27 | 0.0160            | 0.0330 | 0.0500 |
| θ      | 0°              | 4°   | 8°   | 0°                | 4°     | 8°     |
| JEDEC  | MS-012 (AA)     |      |      |                   |        |        |

△ \* NOTES : DIMENSION "D" DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.  
 MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL  
 NOT EXCEED 0.15 MM ( 0.006 INCH ) PER SIDE.



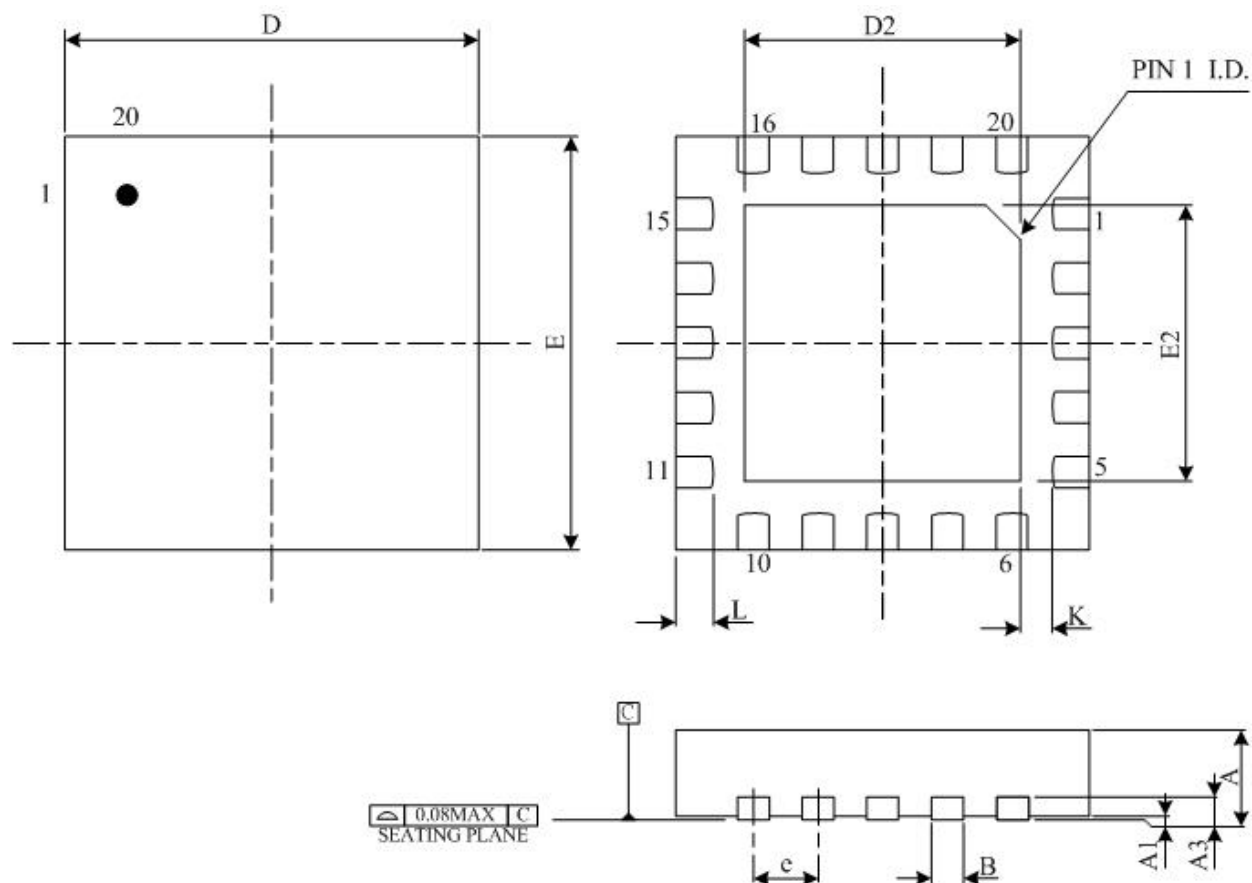
**TSSOP-20 (173 mil) Package Dimension**


| SYMBOL | DIMENSION IN MM |      |      | DIMENSION IN INCH |       |       |
|--------|-----------------|------|------|-------------------|-------|-------|
|        | MIN             | NOM  | MAX  | MIN               | NOM   | MAX   |
| A      | -               | -    | 1.2  | -                 | -     | 0.047 |
| A1     | 0.05            | 0.10 | 0.15 | 0.002             | 0.004 | 0.006 |
| A2     | 0.8             | 0.93 | 1.05 | 0.031             | 0.036 | 0.041 |
| B      | 0.19            | -    | 0.3  | 0.007             | -     | 0.012 |
| D      | 6.4             | 6.5  | 6.6  | 0.252             | 0.256 | 0.260 |
| E      | 6.25            | 6.4  | 6.55 | 0.246             | 0.252 | 0.258 |
| E1     | 4.3             | 4.4  | 4.5  | 0.169             | 0.173 | 0.177 |
| e      | 0.65 BSC        |      |      | 0.026 BSC         |       |       |
| L      | 0.45            | 0.60 | 0.75 | 0.018             | 0.024 | 0.030 |
| θ      | 0 °             |      | 8 °  | 0 °               |       | 8 °   |
| JEDEC  | MO-153 AC REV.F |      |      |                   |       |       |

**Notes :**

- 1.DIMENSION "D" DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS. MOLD FLASH, PROTRUSIONS OR GATE BURRS SHALL NOT EXCEED 0.15 PER SIDE.
- 2.DIMENSION "E1" DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSION. INTERLEAD FLASH OR PROTRUSION SHALL NOT EXCEED 0.25 PER SIDE.
- 3.DIMENSION "B" DOES NOT INCLUDE DAMBAR PROTRUSION.ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08MM TOTAL IN EXCESS OF THE "B" DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OF THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD IS 0.07MM.

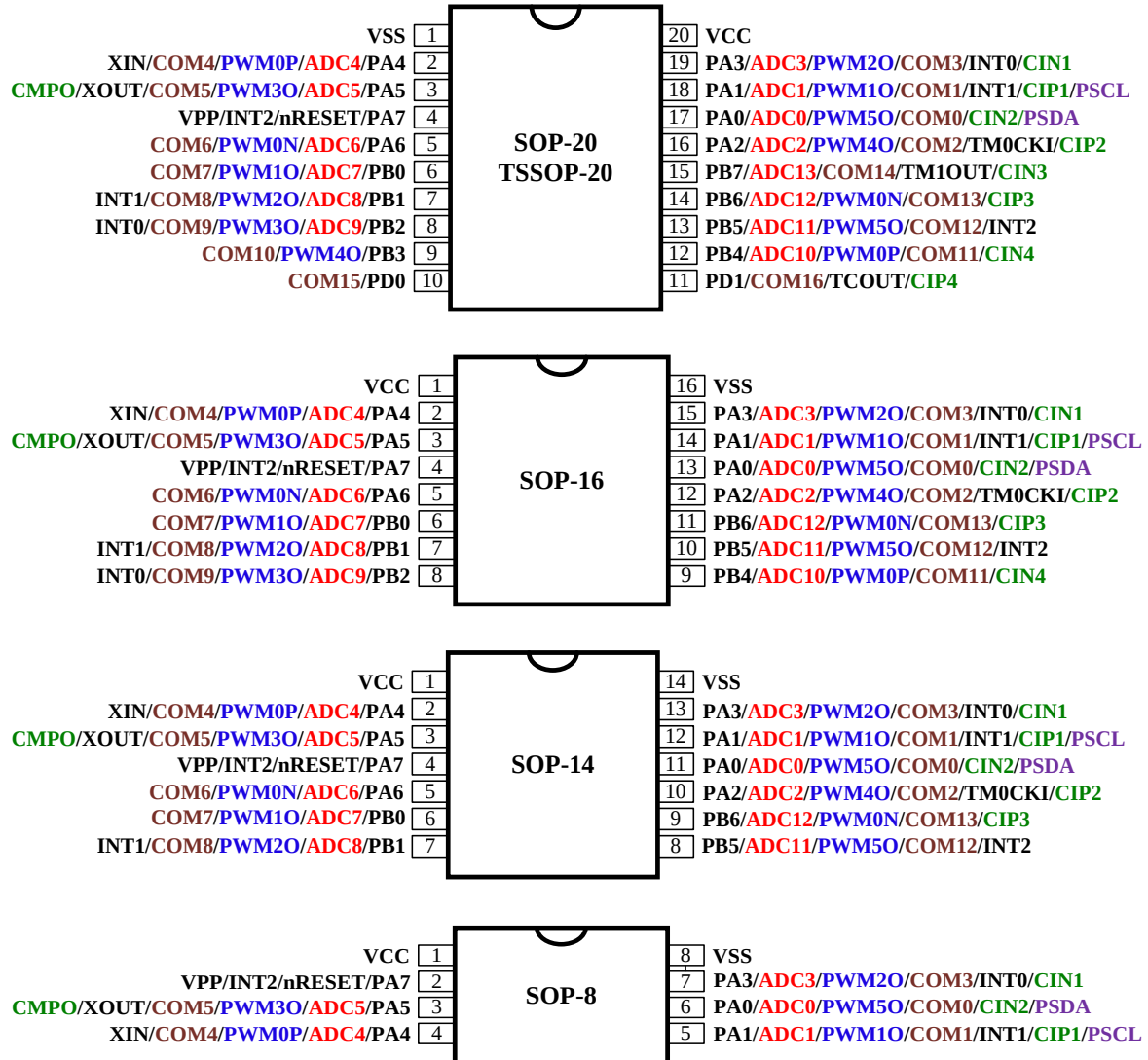
### QFN-20 (3x3x0.75-0.4 mm) (L=0.25 mm) Package Dimension

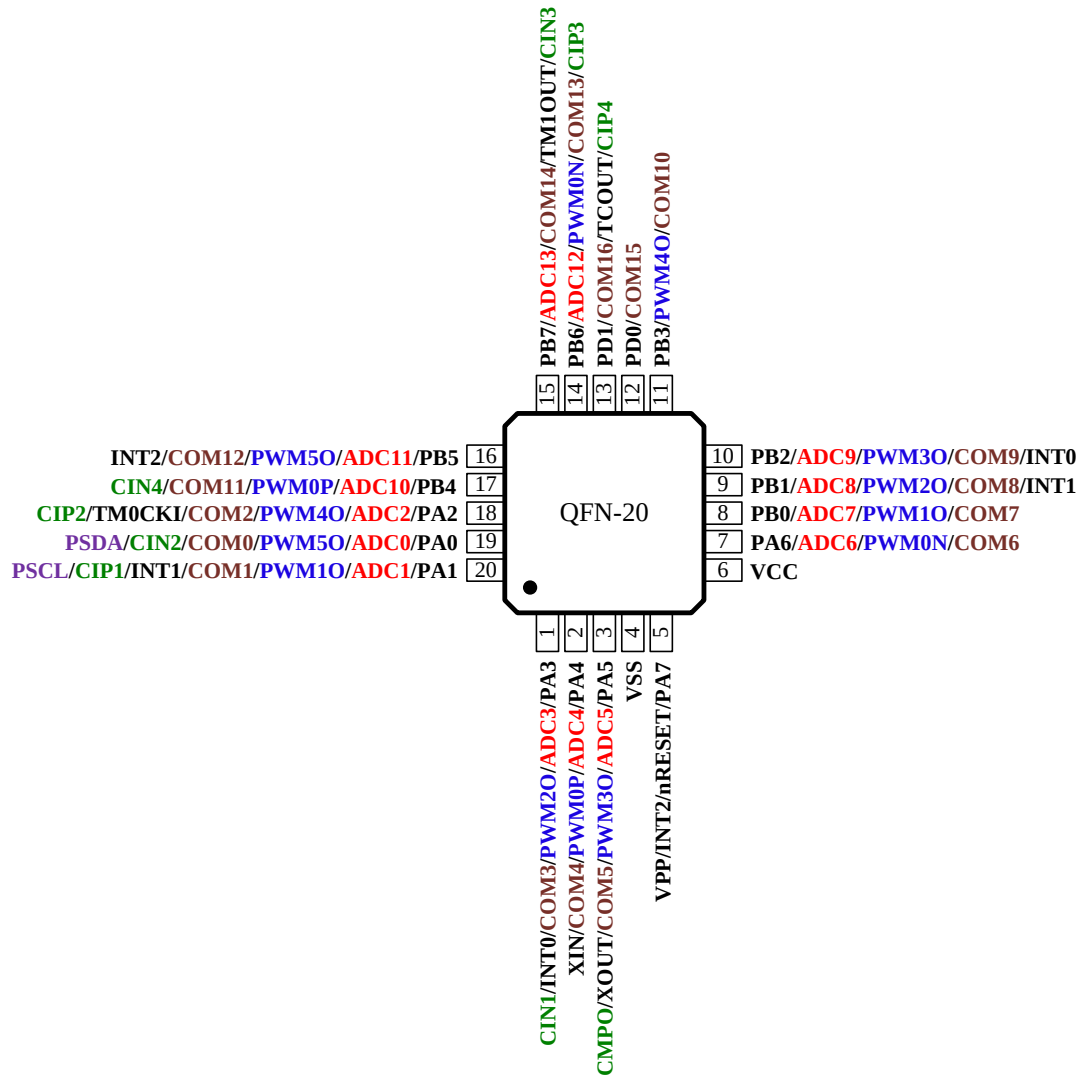


| SYMBOL | DIMENSION IN MM |      |      | DIMENSION IN INCH |       |       |
|--------|-----------------|------|------|-------------------|-------|-------|
|        | MIN             | NOM  | MAX  | MIN               | NOM   | MAX   |
| A      | 0.70            | 0.75 | 0.80 | 0.028             | 0.030 | 0.031 |
| A1     | 0.00            | 0.02 | 0.05 | 0.00              | 0.001 | 0.002 |
| A3     | 0.203 REF       |      |      | 0.008 REF         |       |       |
| B      | 0.15            | 0.20 | 0.25 | 0.006             | 0.008 | 0.010 |
| D      | 3 BSC           |      |      | 0.118 BSC         |       |       |
| E      | 3 BSC           |      |      | 0.118 BSC         |       |       |
| D2     | 1.80            | 1.90 | 2.00 | 0.071             | 0.075 | 0.079 |
| E2     | 1.80            | 1.90 | 2.00 | 0.071             | 0.075 | 0.079 |
| e      | 0.40 BSC        |      |      | 0.016 BSC         |       |       |
| L      | 0.15            | 0.25 | 0.35 | 0.006             | 0.010 | 0.014 |
| K      | 0.30 REF        |      |      | 0.012 REF         |       |       |
| JEDEC  | MO-220          |      |      |                   |       |       |

## APPENDIX: PIN ASSIGNMENT DIAGRAM

Software initialization is necessary for the pads that are not bonded.





## AMENDMENT HISTORY

| Version | Date      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0.9     | Apr, 2024 | <ol style="list-style-type: none"> <li>1. Update Characteristics Graphs</li> <li>2. Update EEPROM Characteristics, DC Characteristics, Clock Timing, Reset Timing Characteristics, LVR/LVD Circuit Characteristics, ADC Electrical Characteristics, EEPTE</li> <li>3. Update Package Types in FEATURES</li> <li>4. Update FAMILY OVERVIEW</li> <li>5. Update EEP write/erase cycle as 40K@25°C/5V</li> <li>6. Delete TBD</li> <li>7. Delete the description about endurance in the section of EEPROM.</li> <li>8. Update Ordering number</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 0.91    | May, 2024 | <ol style="list-style-type: none"> <li>1. Delete items of wafer and dice in Ordering Information</li> <li>2. Update POR curve</li> <li>3. Update PWM0 Block Diagram for PWM0PMSK/PWM0NMSK/PWM0MSKE</li> <li>4. Modify the description of ADVREFS</li> <li>5. Add PSDA and PSCL into the chapters of PIN ASSIGNMENT DIAGRAM, PIN DESCRIPTION and PIN SUMMARY</li> <li>6. Replace “non-overlap” with “dead-zone(non-overlap)”</li> <li>7. Modify the description of EEPIF</li> <li>8. Add the section of “The Precaution of EEPROM Programming”</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 0.92    | Aug, 2024 | <ol style="list-style-type: none"> <li>1. Replace “dead zone” with “dead-zone(non-overlap)” in the sections of “6.5.1.1 Normal Mode” and “6.5.1.2 Half-Bridge Mode”</li> <li>2. Add description: VPP could be VCC, VSS or floating when EEPROM is programming.</li> <li>3. Add comment about ATD</li> <li>4. Update the link of “MOVX”</li> <li>5. Modify the example for writing EEPROM</li> <li>6. Modify the method to clear interrupt flag</li> <li>7. Add comment: EEPTO will be cleared when EEPEN is disabled</li> <li>8. Delete the rows before version 0.9 in the table of “AMENDMENT HISTORY”</li> <li>9. Fix typos in the example of C language of table read</li> <li>10. Add comments for PORSEL, ATDOFF, SCKTYPE and FCKTYPE.</li> <li>11. Add the relative description of “External Crystal/Resonator oscillator”</li> <li>12. Add description for LCD.</li> <li>13. Change FIRC frequency from -5%~+2% to -4%~+3%</li> <li>14. Change the condition of FIRC frequency from “25°C, 3~5V” to “25°C, 3.5~5V”</li> <li>15. Change FIRC frequency from “-3%~+1.5%” to “-2.5%~+2%” in the condition of “-40°C ~ 105°C, 4V”</li> <li>16. Add comment for “Clock Timing”: The value of this parameter is based on the characteristics of tested samples.</li> <li>17. Add comment for ELECTRICAL CHARACTERISTICS: All of the parameters are based on the characteristics of tested samples.</li> <li>18. Add comments for SIRC: 65.5KHz±2%@25°C/V<sub>CC</sub>=5V</li> <li>19. Add LVCTL into the section of “Low Voltage Reset (LVR)” and the chapter of “Interrupt”</li> </ol> |
| 0.93    | Dec, 2024 | <ol style="list-style-type: none"> <li>1. Fix typo: The reset value of LVDHYS is 0</li> <li>2. Add the specification of ADC conversion current</li> <li>3. Add condition into “BandGap Voltage Reference” and “ADC reference voltage”: No power disturbance</li> <li>4. Change EEPROM write time as “typ 4ms@V<sub>CC</sub>=2.5V/25°C, 1.5ms@V<sub>CC</sub>=5V/25°C” in the section of “1.3 EEPROM”</li> <li>5. Add “PWM trigger ADC” into the block diagram of “Analog-to-Digital Converter”</li> <li>6. Add CMPTRIG and ADVREF1P2 into the block diagram of Comparator</li> <li>7. Add comment: ATD off(recommend for EFT issue)</li> <li>8. Add “LCD 1/2 bias” into the table of “PIN DESCRIPTIONS”.</li> <li>9. Fix typo: replace CIN with CIN1 for PA3 in the table of “PIN SUMMARY”</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 0.94    | Jan, 2025 | <ol style="list-style-type: none"> <li>1. Add “pin change” into the description in the section of STOP mode Setting.</li> <li>2. Add the figures of “PA7 Structure” and “Constraint on PA7”, and update the figure of “General Pin Structure”</li> <li>3. Delete the specification of FIRC Frequency @4V/25°C in the chapter of ELECTRICAL CHARACTERISTICS</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

|      |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0.95 | Apr, 2025 | <ol style="list-style-type: none"> <li>1. Modify the description about TM0IF below Timer0 Block Diagram</li> <li>2. Fix typo: 16-channel ADC</li> <li>3. Re-divide the paragraphs of section 6.7 Comparator</li> <li>4. Fix typo for the value of ADCTL2 in the example of 6.6.1 Normal Trigger ADC</li> <li>5. Add a simple explanation of pin-change wakeup</li> <li>6. To change “pin change”, “wake up” and “wakeup” as pin-change, wake-up and wake-up respectively in somewhere of this document.</li> <li>7. Add the function of pin-change wake-up into the system block diagram.</li> <li>8. Replace “I<sup>2</sup>C SCL for program” and “I<sup>2</sup>C SDA for program” with “clock for programmer” and “data for programmer” respectively.</li> <li>9. Add supplementary explanation about pin-change wake-up into the table of family overview</li> <li>10. Add Precaution for EEPROM writing</li> <li>11. Add “clr INTIE/INTIE1” into the example of writing EEPROM.</li> <li>12. Change Wake-up as “Wake-up Interrupt” in the table of “PIN SUMMARY”</li> <li>13. Change the maximum input voltage in the table of “ADC Electrical Characteristics” as <math>V_{REF}</math></li> <li>14. Add comments into the parameter column of the table of “ADC Electrical Characteristics”</li> <li>15. Add port D into the condition of <math>R_{UP}</math></li> <li>16. Modify Timer0/1 block diagram</li> <li>17. Fix something about ADVREFS</li> <li>18. Change V<sub>bg</sub> from 2.48V to 2.56V</li> <li>19. Change POR in the table of “FAMILY OVERVIEW” from 1.53V to 1.63V</li> <li>20. Change ADVREFS in the figure of ADC block diagram from 2.48V to 2.56V</li> </ol> |
| 1.00 | Aug, 2025 | <ol style="list-style-type: none"> <li>1. Change the size of some figures in the “PWM: 16 bits PWM” section.</li> <li>2. Delete the some “MTP” characters</li> <li>3. Modify the system block diagram</li> <li>4. Move “AMENDMENT HISTORY” to the tail of this document</li> <li>5. Delete the model name in the chapter of “APPENDIX: PIN ASSIGNMENT DIAGRAM”</li> <li>6. Add “APPENDIX: PIN ASSIGNMENT DIAGRAM”</li> <li>7. Modify some description in the Program ROM(PROM) section</li> <li>8. Modify the table of PIN SUMMARY</li> <li>9. Modify some description in “Programming Counter (PC) and Stack” section.</li> <li>10. delete 65.5KHz in the “System Clock” section</li> <li>11. Add the chapter of “PRECAUTIONS”.</li> <li>12. delete 85KHz in the example of Timer1 section</li> <li>13. Modify the description about LVR</li> <li>14. Modify the description about the size of ROM, RAM and EEPROM in the table of “FAMILY OVERVIEW”</li> <li>15. Modify the description of LVDS</li> <li>16. Modify the description of package types in the chapter of “FEATURES”.</li> <li>17. Add the description about pin-change into the chapter of “PRECAUTION”</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |