# TM57FA40/40A

## DATA SHEET

## Rev 2.3

# AMENDMENT HISTORY

| Version | Date | Description |
|---------|------|-------------|
| V1.0 | May, 2009 | New release |
| V1.1 | Sept, 2009 | 1. Modify system clock mode for fast crystal 455 KHz ~ 12 MHz.<br>2. Add 8 DIP/SOP in packet types in Features section.<br>3. Add 8 SOP, 8 DIP figure in Pin Assignment section.<br>4. In SYSCFG section, bit 11-10, for 01 setting, modify the LVR threshold to 2.9V.<br>5. In SYSCFG section, bit 9-8, modify the Fast crystal to 455 KHz ~ 12 MHz.<br>6. Modify CLRWDT description to Clear Watchdog/Wakeup Timer.<br>7. Add the figures in Clock Timing in Electrical Characteristics section.<br>8. Add figures for 8-DIP and 8-SOP Package Dimension. |
| V1.2 | Jan, 2010 | 1. Remove 8 DIP/SOP in packet types in Features section.<br>2. Remove 8 SOP, 8 DIP figures in Pin Assignment section.<br>3. Modify the default of nPAPU, nPBPU, nPDPU in R-Plane.<br>4. Modify GOTO / CALL range to 0xFFF.<br>5. Modify the Operating temperature becomes -40 to +85℃.<br>6. Modify the temperature in clock timing in Electrical Characteristics section becomes -40 to +85℃.<br>7. Add FSR.7 fixed 0 in F-Plane.<br>8. Modify PA1 set by rising/falling edge. |
| V1.3 | Dec, 2010 | 1. Add more description about /Borrow and /Digit Borrow in ALU and Working (W) Register section.<br>2. Add Internal RC mode description and figure in System Clock Oscillator section.<br>3. Modify the equation of the frequency calculation in Buzzer Output section.<br>4. Modify the system block diagram.<br>5. Modify the ordering information. |
| V1.4 | Oct, 2011 | 1. Modify the package type data.<br>2. Add description about LVR selection in Reset Section.<br>3. Modify the data in DC Characteristics. |
| V1.5 | Dec, 2011 | Add Ordering Information table in the Packaging Information section. |
| V1.6 | Jan, 2012 | Modify Ordering Information table in the Packaging Information section. |
| V1.7 | Jan, 2012 | 1. Add the Electrical Characteristics specs in the Features section.<br>2. Add description in Reset section.<br>3. Merge the information about LVR Circuit Characteristics into DC Characteristics table.<br>4. Modify the Low Voltage Detection Time data in Reset Timing Characteristics. |
| V1.8 | Apr, 2012 | 1. Modify the document, add serial number for TM57FA40A.<br>2. Add description in Features section about the system clock mode and power on reset.<br>3. Modify figures in the Pin Assignment section. |

| | | 4. Modify the description in Interrupt section. |
|---|---|---|
| | | 5. Modify the description in Reset section. |
| | | 6. Modify the SYSCFG table. |
| | | 7. Add output high/low current in DC Characteristics section. |
| | | 8. Add description for TM57FA40A about power supply current in DC Characteristics section. |
| | | 9. Add description for TM57FA40A in the ordering information. |
| V1.9 | Jul, 2012 | Modify Absolute Maximum Ratings data in Electrical Characteristics section. |
| V2.0 | Jun, 2013 | 1. Add supported EV board on ICE. |
| | | 2. Add Pin Summary. |
| | | 3. Modify Ordering Information. |
| V2.1 | Aug, 2013 | Modify Ordering Information. |
| V2.2 | Feb, 2015 | Modify Pin Assignment. |
| V2.3 | Aug, 2018 | Add SOP/DIP8 information |

# CONTENTS

## FEATURES

1. ROM: 4K x 14 bits Flash ROM

2. RAM: 184 x 8 bits

3. STACK: 6 Levels

4. I/O ports: Three Programmable I/O ports (Max. 18 pins)

5. Timer0/Counter: 8-bit timer/counter with divided by 1~256 pre-scale option

6. Timer1: 8-bit auto-reloadable timer with divided by 1~256 pre-scale option

7. Two 8+2 bit PWM channels capable of 1024 duty resolution

8. 12-bit ADC with 8 channels input

9. Buzzer output

10. Watchdog/Wakeup Timer: On chip Timer based on internal RC oscillation, 11~132 ms wakeup time

11. Reset: Power On Reset, Watchdog Reset, Low Voltage Reset, External pin Reset

12. System Clock Mode:

    - Slow Crystal (SXT): 32 KHz

    - Fast Crystal (FXT): 455 KHz ~12 MHz

    - Internal RC (IRC): 4 MHz

    - External RC (XRC)

13. 2-Level Low Voltage Reset: 2.1V/2.9V (Can be disabled)

14. Operation Voltage: Low Voltage Reset Level to 5.5V

    - fosc = 4 MHz, 2.4V ~ 5.5V

    - fosc = 8 MHz, 2.4V ~ 5.5V

    - fosc = 12 MHz, 3.0V ~ 5.5V

15. Instruction set: 36 Instructions

16. Interrupts: Three pin interrupts, Timer0/Timer1 interrupt and Wakeup Timer interrupt

17. Power Down mode support

18. Package Types: 8-DIP/SOP, 16-DIP/SOP, 20-DIP/SOP

| Device | Icc in Stop Mode | Note |
|---|---|---|
| TM57FA40 | 0.1 uA@2.2~5V | Power-on Reset disable in Stop mode |
| TM57FA40A | 3 uA@3V, 5 uA@5V | Power-on Reset enable in Stop mode |

19. Supported EV board on ICE

    EV board: EV2795

## BLOCK DIAGRAM



**TM57FA40/40A Block Diagram**

# PIN ASSIGNMENT

```
           VSS ─┤1        8├─ Vcc
    Xrc/Xin/PA4 ─┤2  TM57FA40  7├─ PA1/ADC1/INT1
      Xout/PA3 ─┤3  TM57FA40A  6├─ PA2/ADC2/T0I
VPP/nRESET/INT2/PA7 ─┤4  DIP-8/SOP-8  5├─ PA0/PWM0
```

```
           VSS ─┤1        16├─ Vcc
    Xrc/Xin/PA4 ─┤2        15├─ PA6/ADC0/INT0
      Xout/PA3 ─┤3  TM57FA40  14├─ PA1/ADC1/INT1
VPP/nRESET/INT2/PA7 ─┤4  TM57FA40A  13├─ PA2/ADC2/T0I
    T1OUT/PD0 ─┤5        12├─ PB1/ADC3
   BUZZER/PD1 ─┤6  DIP-16  11├─ PD7/ADC4
          PD2 ─┤7  SOP-16  10├─ PA5/ADC5
          PD3 ─┤8         9├─ PA0/PWM0
```

```
           VSS ─┤1        20├─ Vcc
    Xrc/Xin/PA4 ─┤2        19├─ PA6/ADC0/INT0
      Xout/PA3 ─┤3        18├─ PA1/ADC1/INT1
VPP/nRESET/INT2/PA7 ─┤4  TM57FA40  17├─ PA2/ADC2/T0I
    T1OUT/PD0 ─┤5  TM57FA40A  16├─ PB1/ADC3
   BUZZER/PD1 ─┤6        15├─ PD7/ADC4
          PD2 ─┤7  DIP-20  14├─ PA5/ADC5
          PD3 ─┤8  SOP-20  13├─ PA0/PWM0
          PD4 ─┤9        12├─ PB0/ADC7/PWM1
          PD5 ─┤10       11├─ PD6/ADC6/TCOUT
```

## PIN DESCRIPTION

| Name | In/Out | Pin Description |
|---|---|---|
| PA2–PA0 | I/O | Bit-programmable I/O port for Schmitt-trigger input, CMOS "**push-pull**" output or "**pseudo-open-drain**" output. Pull-up resistors are assignable by software. |
| PA6-PA3 PB1-PB0 PD7-PD0 | I/O | Bit-programmable I/O port for Schmitt-trigger input, CMOS "**push-pull**" output or "**open-drain**" output. Pull-up resistors are assignable by software. |
| PA7 | I | Schmitt-trigger input |
| nRESET | I | External active low reset |
| Xin, Xout | – | Crystal/Resonator oscillator connection for system clock |
| Xrc | | External RC oscillator connection for system clock |
| VCC, VSS | P | Power input pin and ground |
| VPP | I | PROM programming high voltage input |
| INT0~2 | I | External interrupt input |
| T0I | I | Timer0's input in counter mode |
| T1OUT | O | Timer1 match output, T1OUT toggles when Timer1 overflow occurs. |
| BUZZER | O | BUZZER output |
| TCOUT | O | Instruction cycle clock divided by N output. Where N is 1, 2, 4, 8. The instruction clock frequency is system clock frequency divided by two. |
| PWM0/PWM1 | O | 10-bit PWM output |
| ADC7~0 | I | A/D channels input |

# PIN SUMMARY

| 20-SOP/DIP | 16-SOP/DIP | Pin Name | Type | Weak Pull-up | Ext. Interrupt | O.D | P.O.D | P.P | Function After Reset | PWM | Touch Key | ADC | MISC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | VSS | P | | | | | | | | | | |
| 2 | 2 | Xrc/Xin/PA4 | I/O | ○ | | ○ | | ○ | SYS | | | | |
| 3 | 3 | Xout/PA3 | I/O | ○ | | ○ | | ○ | SYS | | | | |
| 4 | 4 | VPP/nRESET/INT2/PA7 | I | ○ | ○ | | | | SYS | | | | nRESET |
| 5 | 5 | T1OUT/PD0 | I/O | ○ | | ○ | | ○ | PD0 | | | | T1OUT |
| 6 | 6 | BUZZER/PD1 | I/O | ○ | | ○ | | ○ | PD1 | | | | BUZZER |
| 7 | 7 | PD2 | I/O | ○ | | ○ | | ○ | PD2 | | | | |
| 8 | 8 | PD3 | I/O | ○ | | ○ | | ○ | PD3 | | | | |
| 9 | - | PD4 | I/O | ○ | | ○ | | ○ | PD4 | | | | |
| 10 | - | PD5 | I/O | ○ | | ○ | | ○ | PD5 | | | | |
| 11 | - | PD6/ADC6/TCOUT | I/O | ○ | | ○ | | ○ | PD6 | | | ○ | TCOUT |
| 12 | - | PB0/ADC7/PWM1 | I/O | ○ | | ○ | | ○ | PB0 | ○ | | ○ | |
| 13 | 9 | PA0/PWM0 | I/O | ○ | | | ○ | ○ | PA0 | ○ | | | |
| 14 | 10 | PA5/ADC5 | I/O | ○ | | ○ | | ○ | PA5 | | | ○ | |
| 15 | 11 | PD7/ADC4 | I/O | ○ | | ○ | | ○ | PD7 | | | ○ | |
| 16 | 12 | PB1/ADC3 | I/O | ○ | | ○ | | ○ | PB1 | | | ○ | |
| 17 | 13 | PA2/ADC2/T0I | I/O | ○ | | | ○ | ○ | PA2 | | | ○ | T0I |
| 18 | 14 | PA1/ADC1/INT1 | I/O | ○ | ○ | | ○ | ○ | PA1 | | | ○ | |
| 19 | 15 | PA6/ADC0/INT0 | I/O | ○ | ○ | ○ | | ○ | PA6 | | | ○ | |
| 20 | 16 | VCC | P | | | | | | | | | | |

Symbol：P.P.　＝ Push-Pull Output
　　　　　P.O.D. ＝ Pseudo Open Drain
　　　　　O.D.　＝ Open Drain
　　　　　SYS　＝ by SYSCFG bit

# FUNCTIONAL DESCRIPTION

## 1. CPU Core

### 1.1 Clock Scheme and Instruction Cycle

The system clock is internally divided by two to generate Q1 state and Q2 state for each instruction cycle. The Programming Counter (PC) is updated at Q1 and the instruction is fetched from program ROM and latched into the instruction register in Q2. It is then decoded and executed during the following Q1-Q2 cycle. Branch instructions take two cycles since the fetch instruction is 'flushed' from the pipeline, while the new instruction is being fetched and then executed.
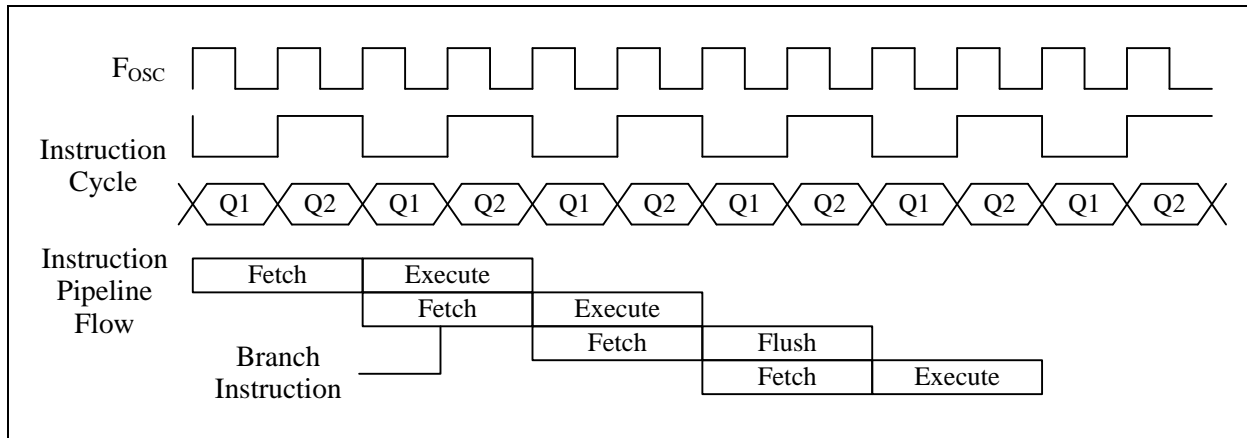
## 1.2 Addressing Mode

There are two Data Memory Planes in CPU, R-Plane and F-Plane. The registers in R-Plane are write-only. The "MOVWR" instruction copy the W-register's content to R-Plane registers by direct addressing mode.

The lower locations of F-Plane are reserved for the SFR. Above the SFR is General Purpose Data Memory, implemented as static RAM. F-Plane can be addressed directly or indirectly. Indirect Addressing is made by INDF register. The INDF register is not a physical register. Addressing INDF actually addresses the register whose address contains in the FSR register (FSR is a pointer). The first half of F-Plane is bit-addressable, while the second half of F-Plane is not bit-addressable.
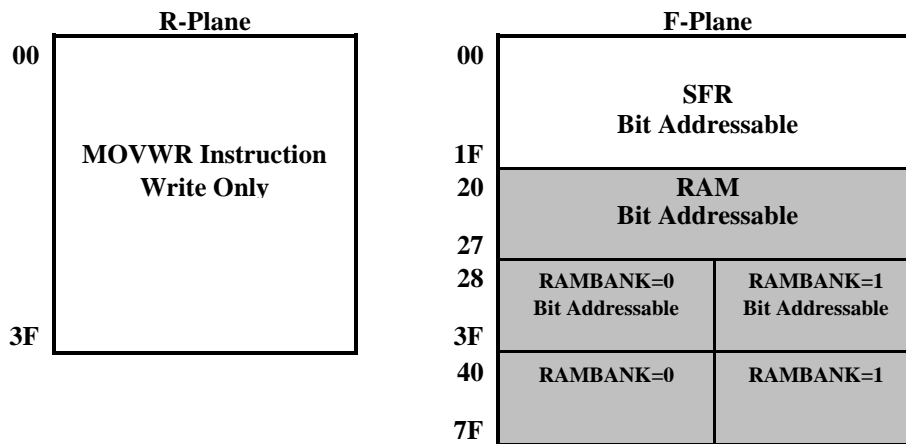


## 1.3 Programming Counter (PC) and Stack

The Programming Counter is 12-bit wide capable of addressing a 4K x 14 program ROM. 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 the followings.

The Reset Vector (000h) and the Interrupt Vector (001h) are provided for PC initialization and Interrupt. For CALL/GOTO instructions, PC loads 12 bits address from instruction word. For RET/RETI/RETLW instructions, PC retrieves its content from the top level STACK. For the other instructions updating PC [7:0], the PC [11:8] keeps unchanged. The STACK is 12-bit wide and 6-level in depth. The CALL instruction and Hardware interrupt will push STACK level in order. While the RET/RETI/RETLW instruction pops the STACK level in order.

## 1.4 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 STATUS Register

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 BCF, BSF and MOVWF 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 |
| R/W | – | – | R/W | R | R | R/W | R/W | R/W |

| Bit | Description | |
|---|---|---|
| 7-6 | Not Used | |
| 5 | **RAMBANK:** RAM Bank Selection<br>  0: RAM Bank0<br>  1: RAM Bank1 | |
| 4 | **TO**: Time Out<br>  0: after Power On Reset, LVR Reset, or CLRWDT/SLEEP instruction<br>  1: WDT time out occurs | |
| 3 | **PD**: Power Down<br>  0: after Power On Reset, LVR Reset, or CLRWDT instruction<br>  1: after SLEEP instruction | |
| 2 | **Z**: Zero Flag<br>  0: the result of a logic operation is not zero<br>  1: the result of a logic operation is zero | |
| 1 | **DC**: Decimal Carry Flag or Decimal/Borrow Flag | |
| | ADD instruction | SUB instruction |
| | 1: a carry from the low nibble bits of the result occurs<br>0: no carry | 1: no borrow<br>0: a borrow from the low nibble bits of the result occurs |
| 0 | **C**: Carry Flag or Borrow Flag | |
| | ADD instruction | SUB instruction |
| | 1: a carry occurs from the MSB<br>0: no carry | 1: no borrow<br>0: a borrow occurs from the MSB |

## 1.6 Interrupt

The device has 1 level, 1 vector and six interrupt sources. Each interrupt source has its own enable control bit. An interrupt event will set its individual pending flag, no matter its interrupt enable control bit is 0 or 1. Because device has only 1 vector, there is not an interrupt priority register. The interrupt priority is determined by F/W.

If the corresponding interrupt enable bit has been set (INTE), it would trigger CPU to service the interrupt. CPU accepts interrupt in the end of current executed instruction cycle. In the mean while, a "CALL 001" 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.
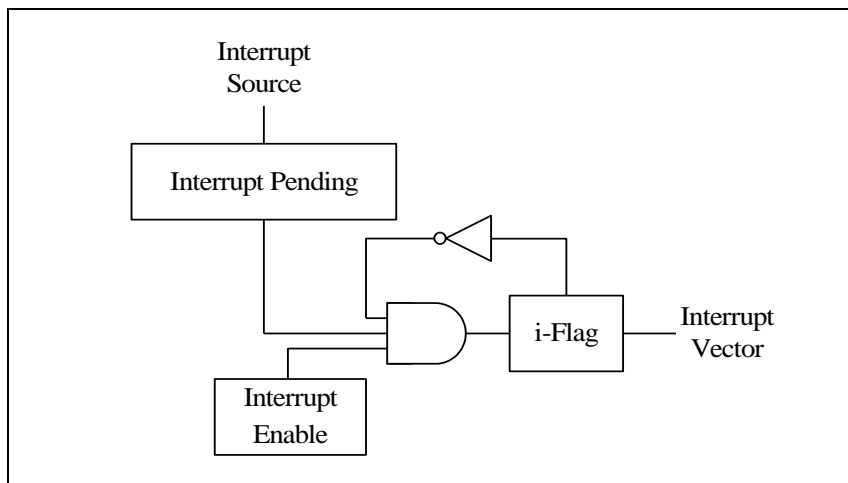
## 2. Chip Operation Mode

### 2.1 Reset

The device can be RESET in four ways.

- Power-On-Reset

- Low Voltage Reset (LVR)

- External Pin Reset (PA7)

- Watchdog Reset (WDT)

After Power-On-Reset, all system and peripheral control registers are then set to their default hardware Reset values. The clock source, LVR level and chip operation mode are selected by the SYSCFG register value.

The Low Voltage Reset features static reset when supply voltage is below a threshold level. There are two threshold levels can be selected. The LVR's operation mode is defined by the SYSCFG register. There are two voltage selections for the LVR threshold level, one is higher level which is suitable for application with VCC is more than 3.3V, while another one is suitable for application with VCC is less than 3.3V.

See the following LVR Selection Table; user must also consider the lowest operating voltage of operating frequency.

LVR Selection Table:

| LVR Threshold Level | Consider the operating voltage to choose LVR |
|---|---|
| LVR2.9 | 5.5V > VCC > 3.3V |
| LVR2.1 | VCC is wide voltage range |

The External Pin Reset and Watchdog Reset can be disabled or enabled by the SYSCFG register. These two resets also set all the control registers to their default reset value. The TO/PD flag is not affected by these resets.

**2.2 System Configuration Register (SYSCFG)**

The System Configuration Register (SYSCFG) is located at Flash INFO area. The SYSCFG determines the option for initial condition of MCU. It is written by FLASH Writer only. User can select clock source, LVR threshold voltage and chip operation mode by SYSCFG register. The 13th bit of SYSCFG is code protection selection bit. If this bit is 1, the data in FLASH will be protected when user reads FLASH.

| Bit | 13~0 | |
|---|---|---|
| **Default Value** | **00_0000_000x_xxxx** | |
| Bit | Description | |
| 13 | **PROTECT**: Code Protection Selection | |
| | 1 | Enable |
| | 0 | Disable |
| 12 | **IVCPD**: IVC*/LVR Power Down in Sleep mode | |
| | 1 | IVC/LVR OFF in Sleep mode |
| | 0 | IVC/LVR ON in Sleep mode |
| 11-10 | **LVR**: LV reset mode | |
| | 11 | 2.1V, always enable |
| | 10 | Invalid |
| | 01 | 2.9V, always enable |
| | 00 | LVR disable |
| 9-8 | **CLK**: Clock Source Selection | |
| | 11 | Fast Xtal (FXT) (455 KHz~12 MHz) |
| | 10 | Slow Xtal (SXT) (32 KHz) |
| | 01 | Internal RC (IRC) (4 MHz) |
| | 00 | External RC (XRC) |
| 7 | **RESET**: External pin Reset Enable | |
| | 1 | Enable External pin Reset |
| | 0 | Disable External pin Reset |
| 6 | **WDT**: WDT Reset Enable | |
| | 1 | Enable WDT Reset, Disable WKT Timer |
| | 0 | Disable WDT Reset, Enable WKT Timer |
| 4-0 | Reserved by tenx | |

* IVC is the chip built-in 3.3V regulator for internal circuit.

## 2.3 FLASH Program ROM

The FLASH Program ROM of this device is 4K words, with an extra INFO area to store the SYSCFG. The FLASH ROM can be written multi-times and can be read as long as the PROTECT bit of SYSCFG is not set. The SYSCFG can be read no matter PROTECT is set or cleared, but can be written only when PROTECT is not set or FLASH ROM is erased. That is, un-protect the PROTECT bit needs the erased FLASH ROM.

| | |
|---|---|
| 000 | **Reset Vector** |
| 001 | **Interrupt Vector** |
| | **User Code** |
| FFF | |
| | **SYSCFG (INFO area)** |
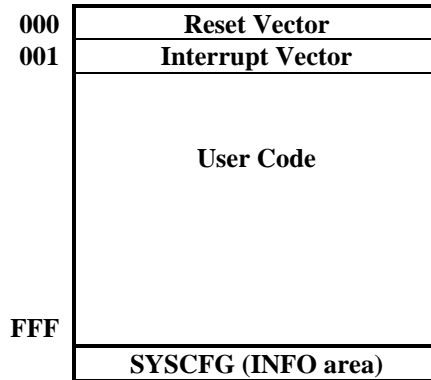
## 2.4 Power-Down Mode

The Power-down mode is activated by SLEEP instruction. During the Power-down mode, the system clock and peripherals stop to minimize power consumption, while the WDT/WKT Timer is working or not depends on F/W setting. The Power down mode can be terminated by Reset, or enabled Interrupts (External pins and WKT interrupt). In the power down mode, user can enable or disable IVC according to the standby current requirement. Enabled IVC can provide the chip internal circuit more stable 3.3V power.

# 3. Peripheral Functional Block

## 3.1 Watchdog (WDT) / Wakeup (WKT) Timer

The WDT and WKT share the same internal RC Timer. The overflow period of WDT/WKT can be selected from 11 ms to 132 ms. The WDT/WKT is cleared by the CLRWDT instruction. If the Watchdog Reset is enabled (WDTE=1), the WDT generates the chip reset signal, even in sleep mode, otherwise, the WKT only generates overflow time out interrupt. The WDT/WKT works in both normal mode and sleep mode. If WDTE=0 and WKTIE=0 (Wakeup interrupt disable), the internal RC Timer stops for power saving.
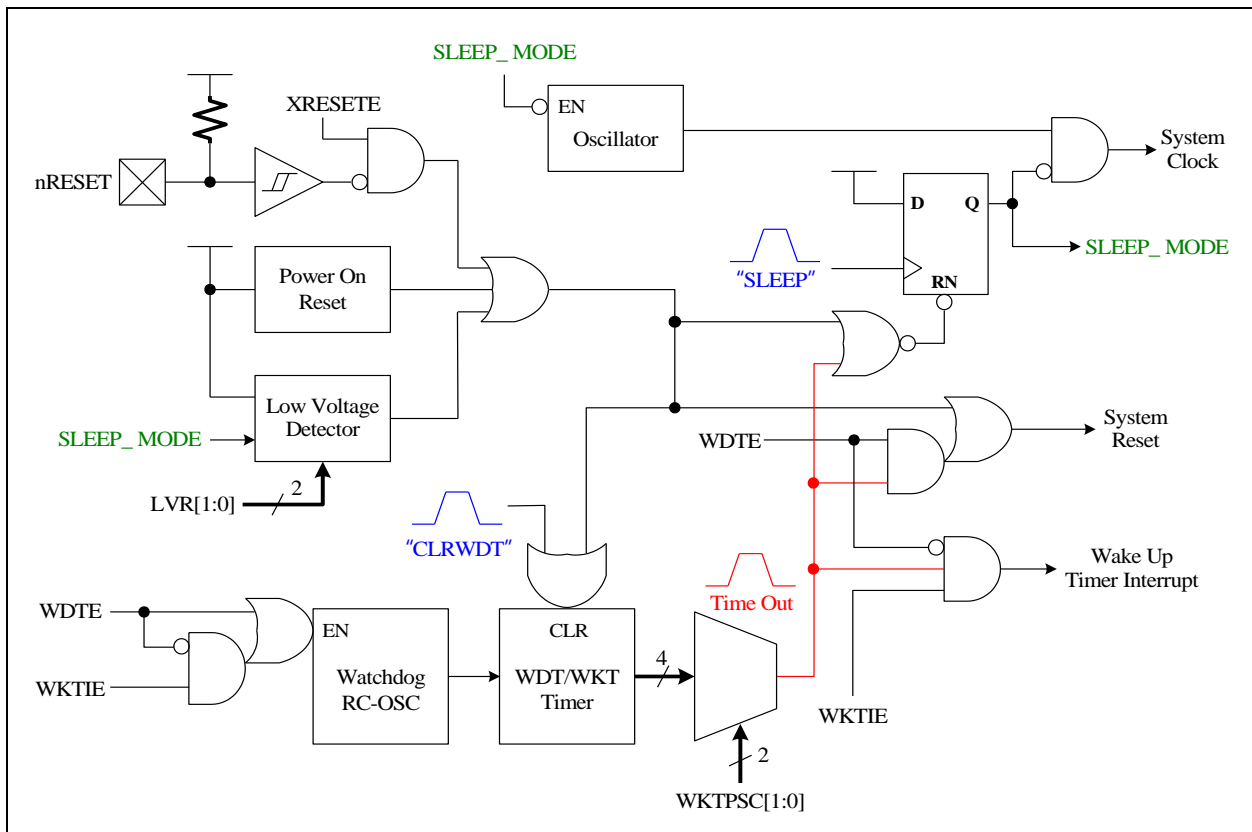
## 3.2 Timer0: 8-bit Timer/Counter with Pre-scale (PSC)

The Timer0 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatically rolls over based on the pre-scaled clock source, which can be the instruction cycle or T0I input. The Timer0 increase rate is determined by "Timer0 Pre-Scale" (TM0PSC) register in R-Plane. The Timer0 can generate interrupt (TM0I) when it rolls over.
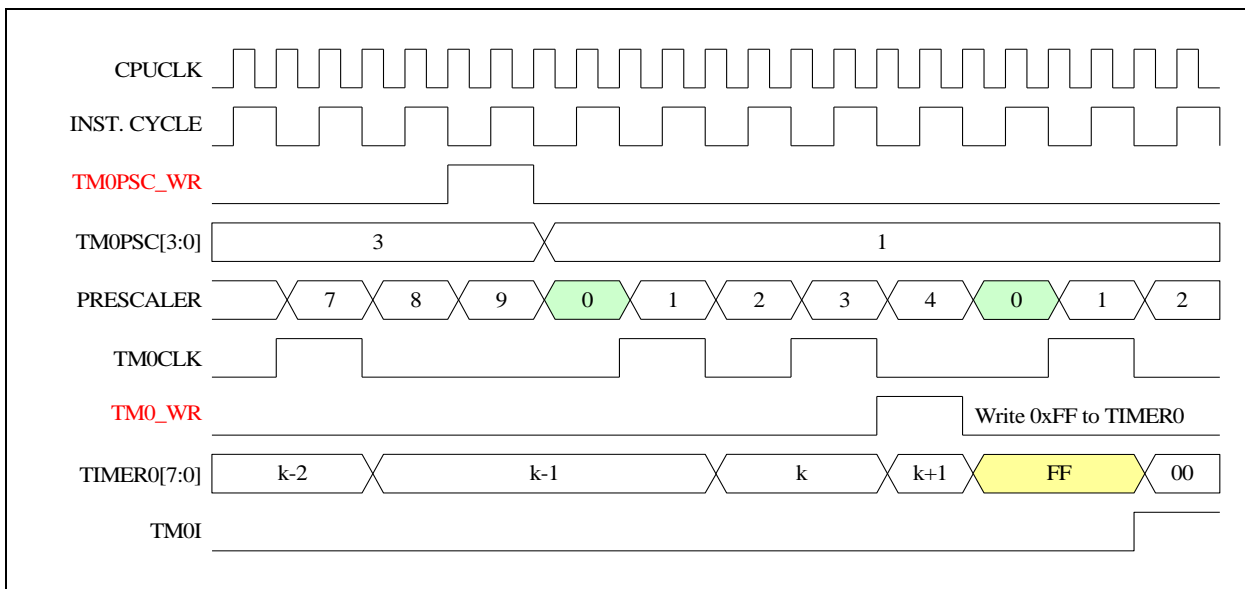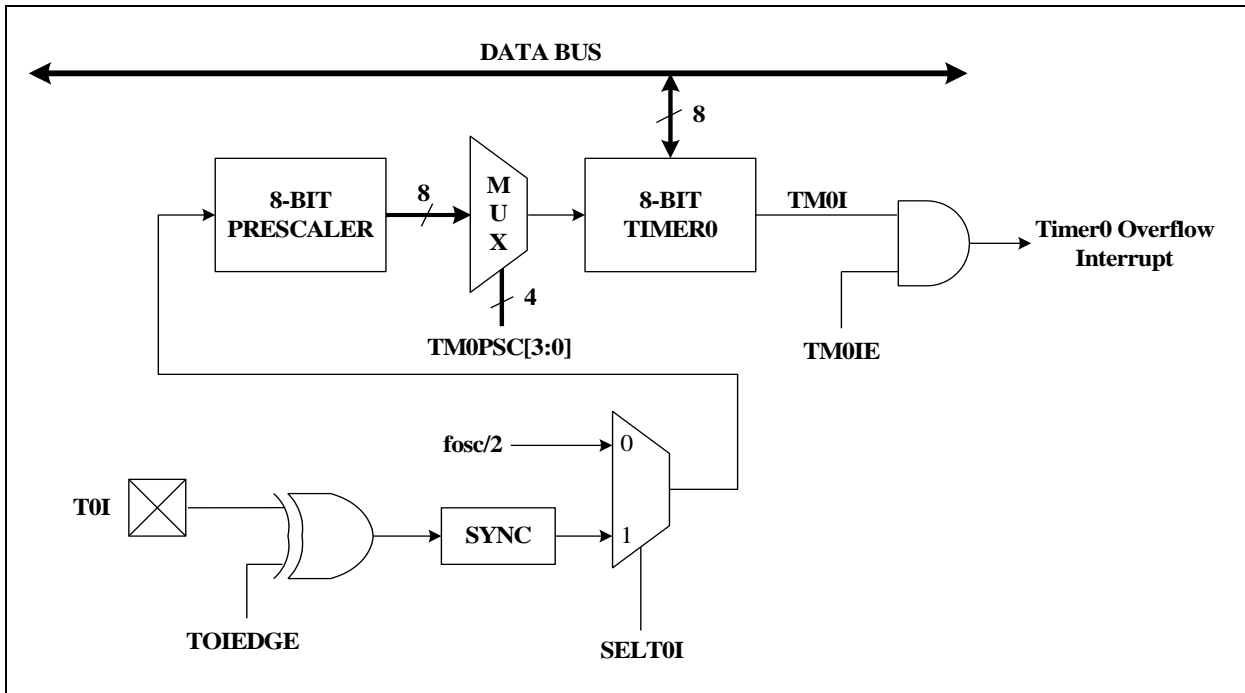
### 3.3 Timer1: 8-bit Timer with Pre-scale (PSC)

The Timer1 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer1 increases itself periodically and automatically reloads a new "offset value" (TM1RELD) while it rolls over based on the pre-scaled instruction clock. The Timer1 increase rate is determined by "Timer1 Pre-Scale" (TM1PSC) register in R-Plane. The Timer1 can generate interrupt (TM1I) and T1OUT toggle signal when it rolls over.
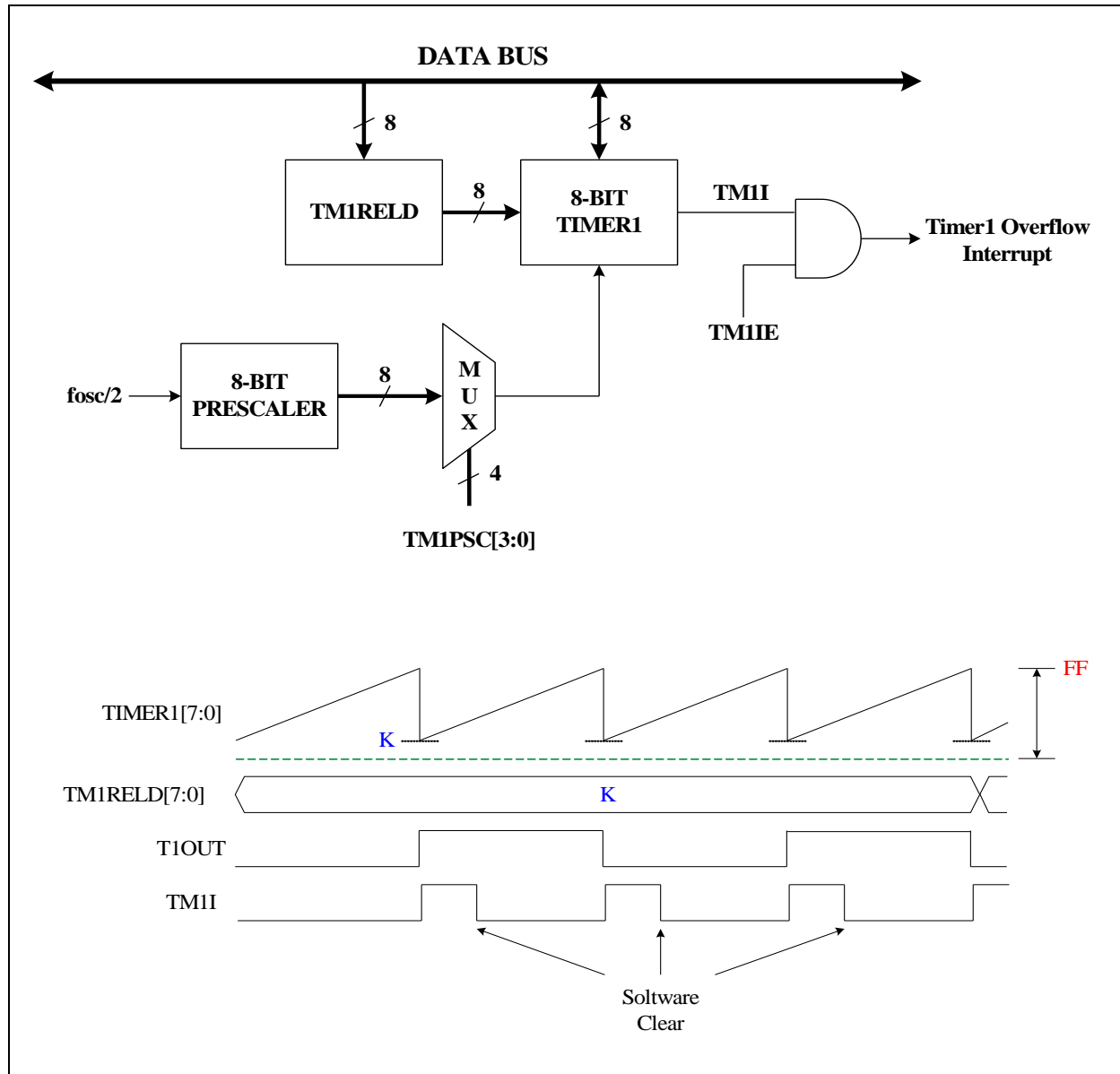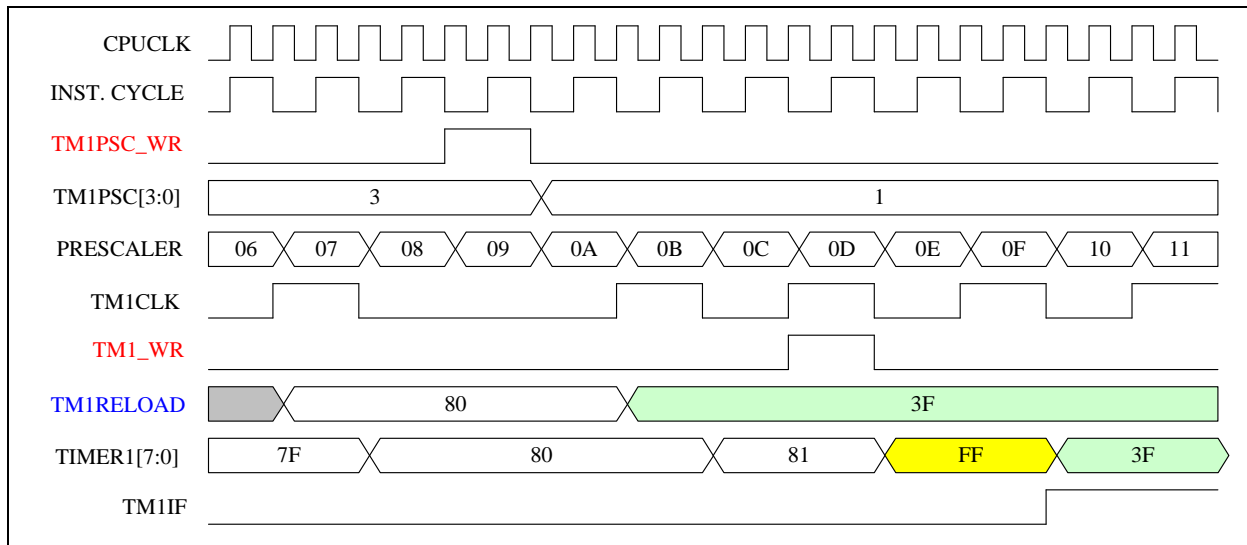
## 3.4 8+2 bit PWM

PWM0 and PWM1 have the same structure. The PWM can generate fix frequency waveform with 1024 duty resolution based on System Clock. A spread LSB technique allows PWM to run its frequency at "System Clock divided by 256" instead of "System Clock divided by 1024", which means the PWM is 4 times faster than normal. The advantage of higher PWM frequency is that the post RC filter can transform the PWM signal to more stable DC voltage level. The PWM output signal reset to low level whenever the 8-bit base counter matches the 8-bit MSB of PWM duty register (PWMDUTY). When the base counter rolls over, the 2-bit LSB of PWM duty register decide whether to set the PWM output signal high immediately or set it high after one clock cycle delay.
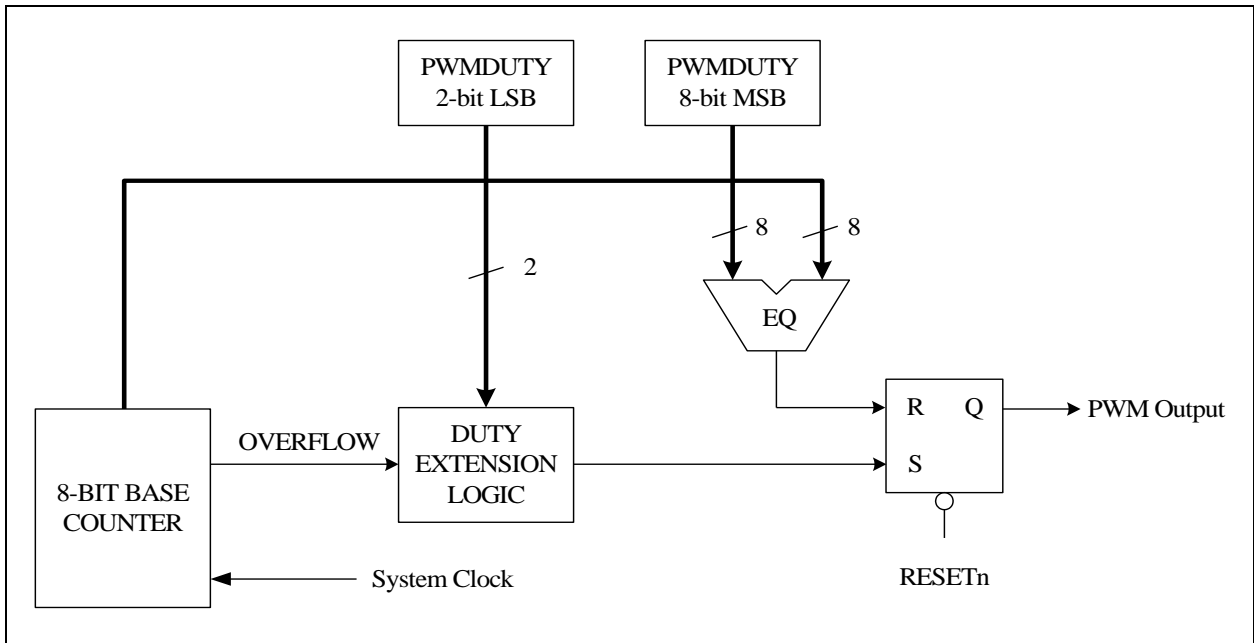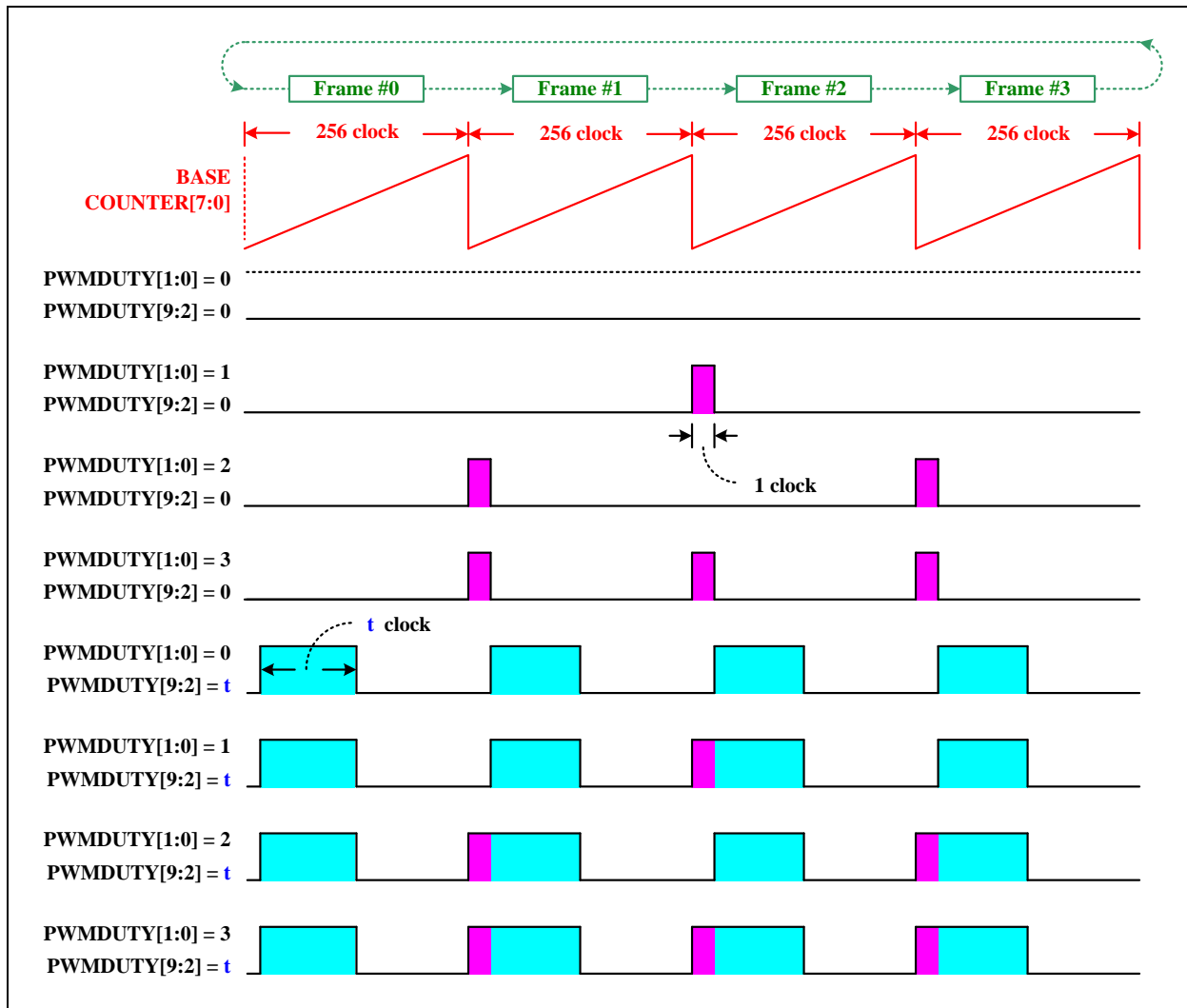
PWM example code:

```
        movlw       01111111b
        movwf       0ch             ;set PWM0DUTY[9:2]=8'b01111111
        movlw       11000000b
        movwf       0dh             ;set PWM0DUTY[1:0]=2'b11
        movlw       01000000b
        movwr       0bh             ;enable PWM0 output to PA0  (PWM0_OUT)
        :
        :
        movlw       00h
        movwr       0bh             ;disable PWM0 (PWM0_OUT)
```

### 3.5 12-bit ADC
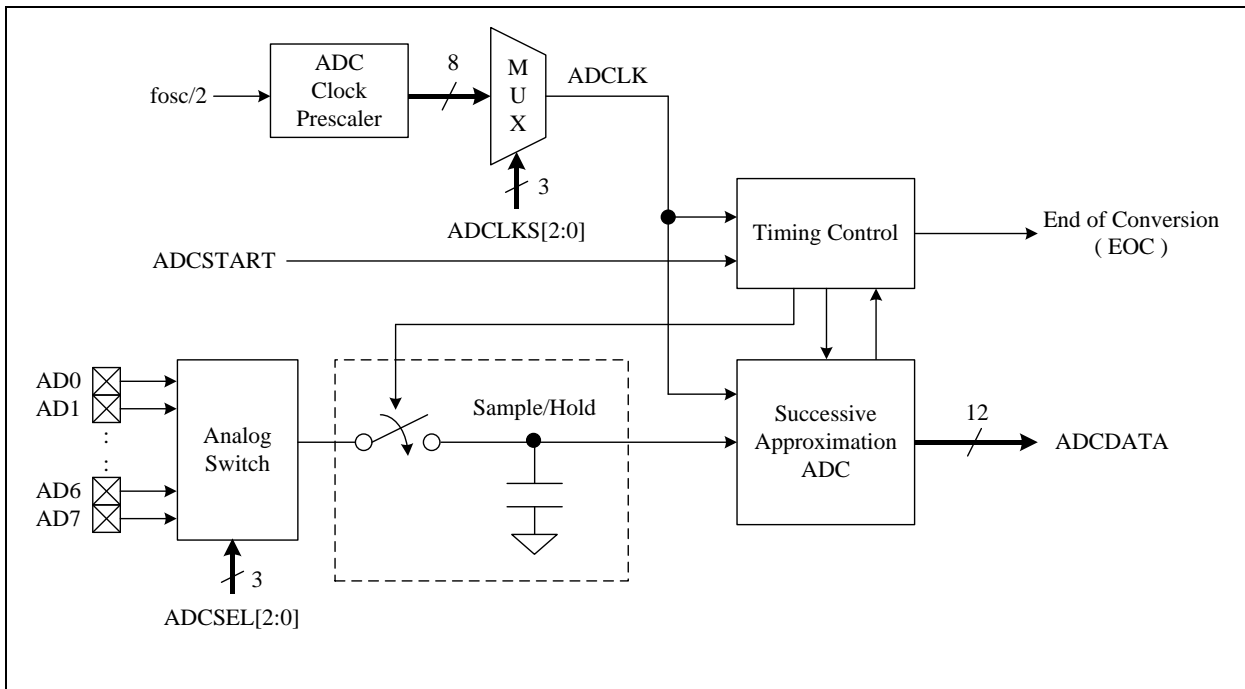


The 12-bit ADC (Analog to Digital Converter) consists of an 8-channel analog input multiplexer, control register, clock generator, 12-bit successive approximation register, and output data register. To use the ADC, user needs to set ADCLKS to choose a proper ADC clock frequency, which must be less than 2 MHz. User then launches the ADC conversion by setting the ADCSTART control bit. After end of conversion, H/W automatically clears the ADCSTAT bit. User can poll this bit to know the conversion status. The nADC_IE control register is used for ADC pin type setting, user can write the corresponding bit to "0" when the pin is used as an ADC input. The setting can disable the pin logical input path to save power consumption.
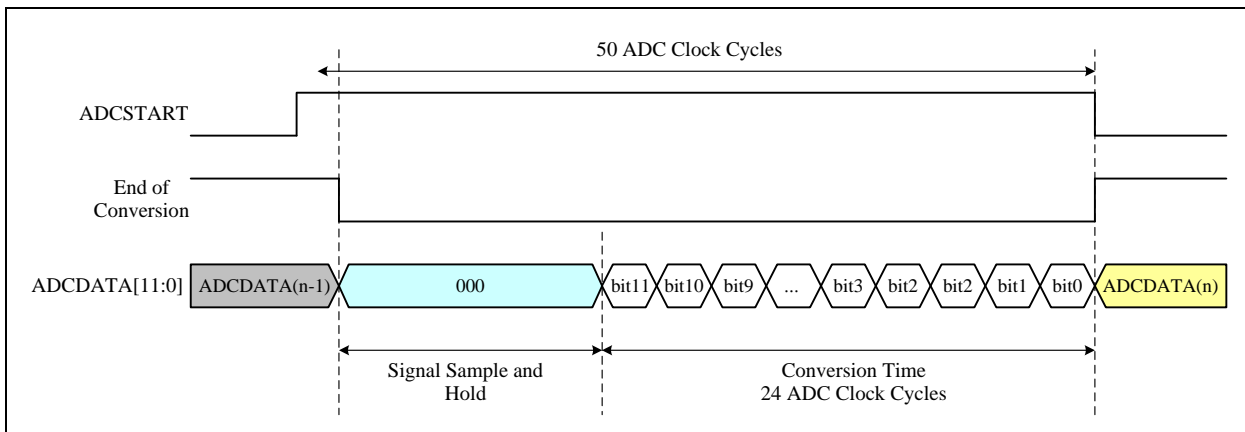
ADC example code:

```
        movlw    00000111b
        movwf    11h          ;ADC channel select,ADC7(PB0) (ADCSEL)

        movlw    00000001h
        movwr    09h          ;disable PB0 pull up resistor      (nPBPU)

        movlw    01111111b
        movwr    12h          ;set ADC7 input enable (nADC_IE)

        movlw    00010000b
        movwr    0ch          ;set ADC clock is instruction cycle / 64 (ADCCLKS)

        bsf      11h,3        ;start  ADC conversion (ADCSTART)


ADC_LOOP:

        btfsc    11h,3
        goto     ADC_LOOP     ;wait ADCSTART go LOW

        :                     ;read ADCQ[11:0] (ADCDQ)
        :
```

## 3.6 System Clock Oscillator

System Clock can be operated in four different oscillation modes, which is selected by setting the CLKS in the SYSCFG register. In Slow/Fast Crystal mode, a crystal or ceramic resonator is connected to the Xin and Xout pins to establish oscillation. In external RC mode, the external resistor and capacitor determine the oscillation frequency. In the internal RC mode, the on chip oscillator generates 4 MHz system clock. In this mode, PCB Layout may have strong effect on the stability of Internal Clock Oscillator. Since power noise degrades the performance of Internal Clock Oscillator, placing power supply bypass capacitors 1 uF and 0.1 uF very close to VCC/VSS pins improves the stability of clock and the overall system.
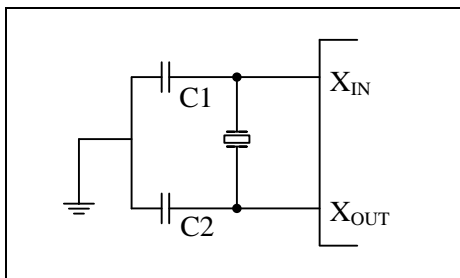
External Oscillator Circuit
(Crystal or Ceramic)
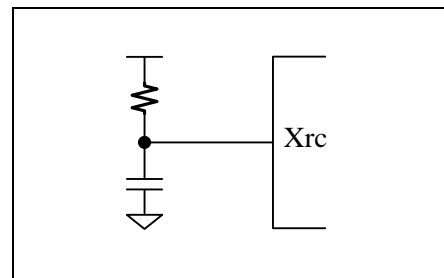
External RC Oscillator
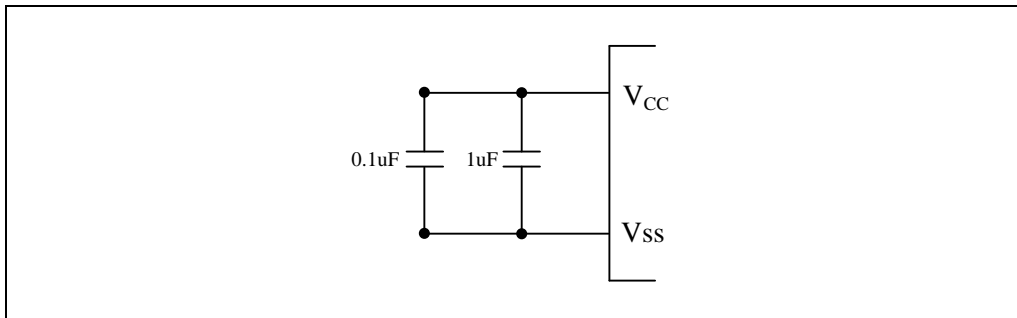
Internal RC Mode

## 3.7 BUZZER Output

The Buzzer driver consists of 6-bit counter and a clock divider. It generates 50% duty square waveform with wide frequency range. To use the Buzzer function, user needs to set both the Buzzer enable control bit (BUZ_EN) and the Buzzer output pin enable control bit (BUZ_OUT).
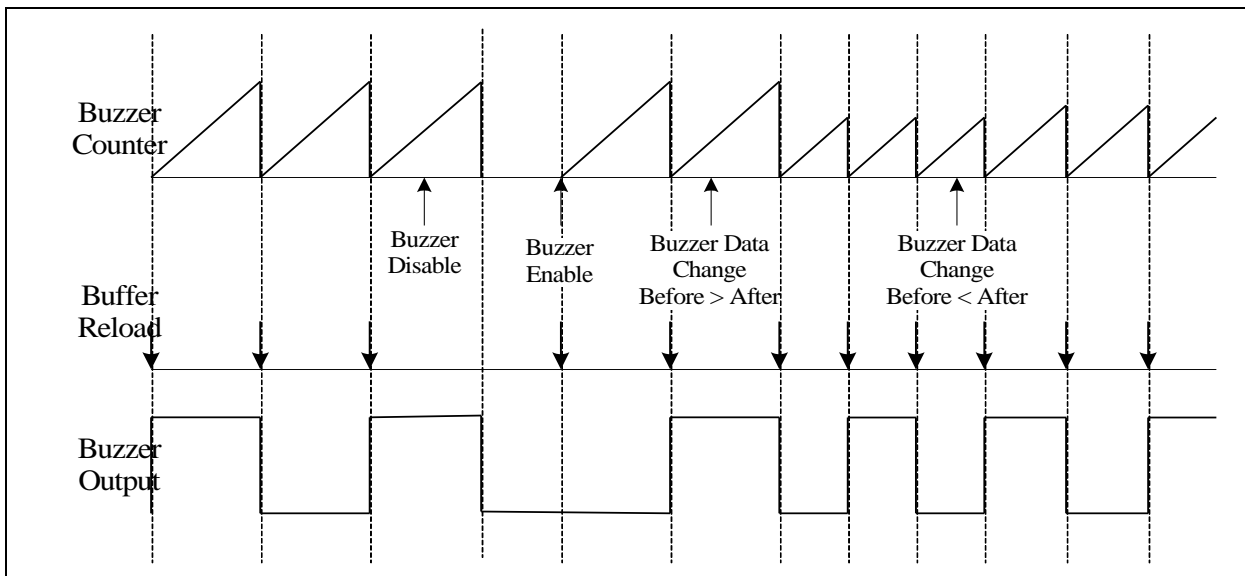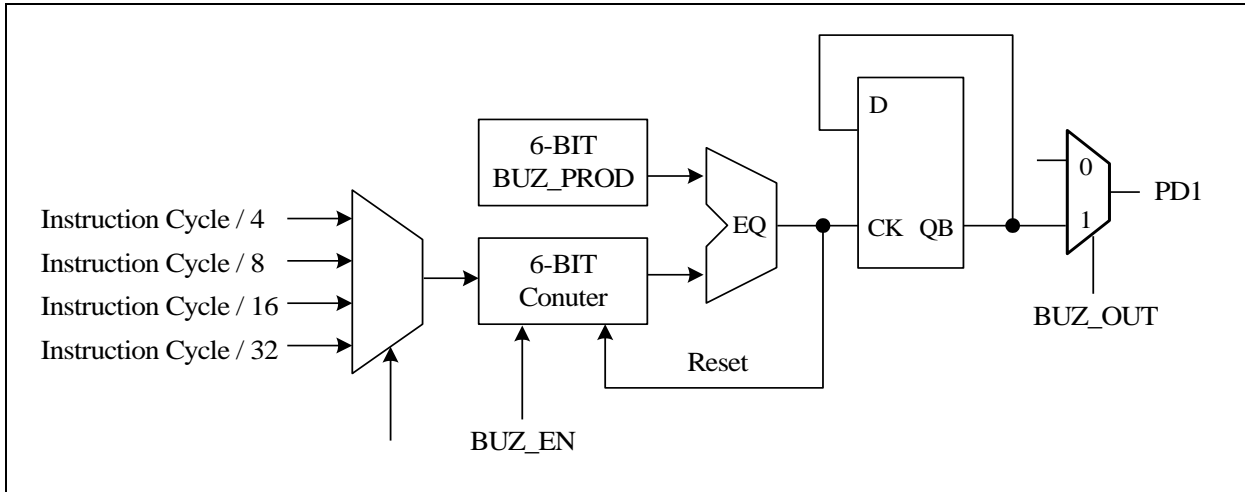
BUZ_PROD[5:0] determines output frequency. Frequency calculation is as follows.

$$F_{BZ} = (f_{OSC}/2) / (\text{Instruction Cycle Divider}) / (\text{BUZ\_PROD} + 1) / 2$$

Output frequency calculation

CPU Clock (fosc) = 8192 KHz
Instruction Cycle = fosc/2 = 8192 KHz/2 = 4096 KHz
Prescaler Ratio (BUZ_PSC) = 2'b11 (Instruction Cycle Divider = 32)
Period Data (BUZ_PROD) = 9

FBZ = (8192 KHz/2) / 32 / (9+1) / 2 = 6.4 (KHz)

BUZZER example code:

```
movlw       10000000b
movwr       0bh              ; enable BUZZER output to PD1 (BUZ_OUT)

movlw       11001001b        ; (fosc/2)/32 (BUZ_PSC)
movwr       10h              ; Period=9 (BUZ_PROD)

movlw       80h
movwr       0ch              ; enable BUZZER counting (BUZ_EN)
:
:
movlw       00h
movwr       0ch              ; disable BUZZER counting (BUZ_EN)
```

# 4. I/O Port

## 4.1 PA0-2

These pins can be used as Schmitt-trigger input, CMOS push-pull output or "pseudo-open-drain" output. The pull-up resistor is assignable to each pin by S/W setting. To use the pin in Schmitt-trigger input mode, S/W needs to set the PAE=0 and PAD=1. To use the pin in **pseudo-open-drain** mode, S/W sets the PAE=0. The benefit of pseudo-open-drain structure is that the output rise time can be much faster than pure open-drain structure. S/W sets PAE=1 to use the pin in CMOS push-pull output mode. Reading the pin data (PAD) has different meaning. In "Read-Modify-Write" instruction, CPU actually reads the output data register. In the other instructions, CPU reads the pin state. The so-called "Read-Modify-Write" instruction includes BSF, BCF and all instructions using F-Plane as destination.
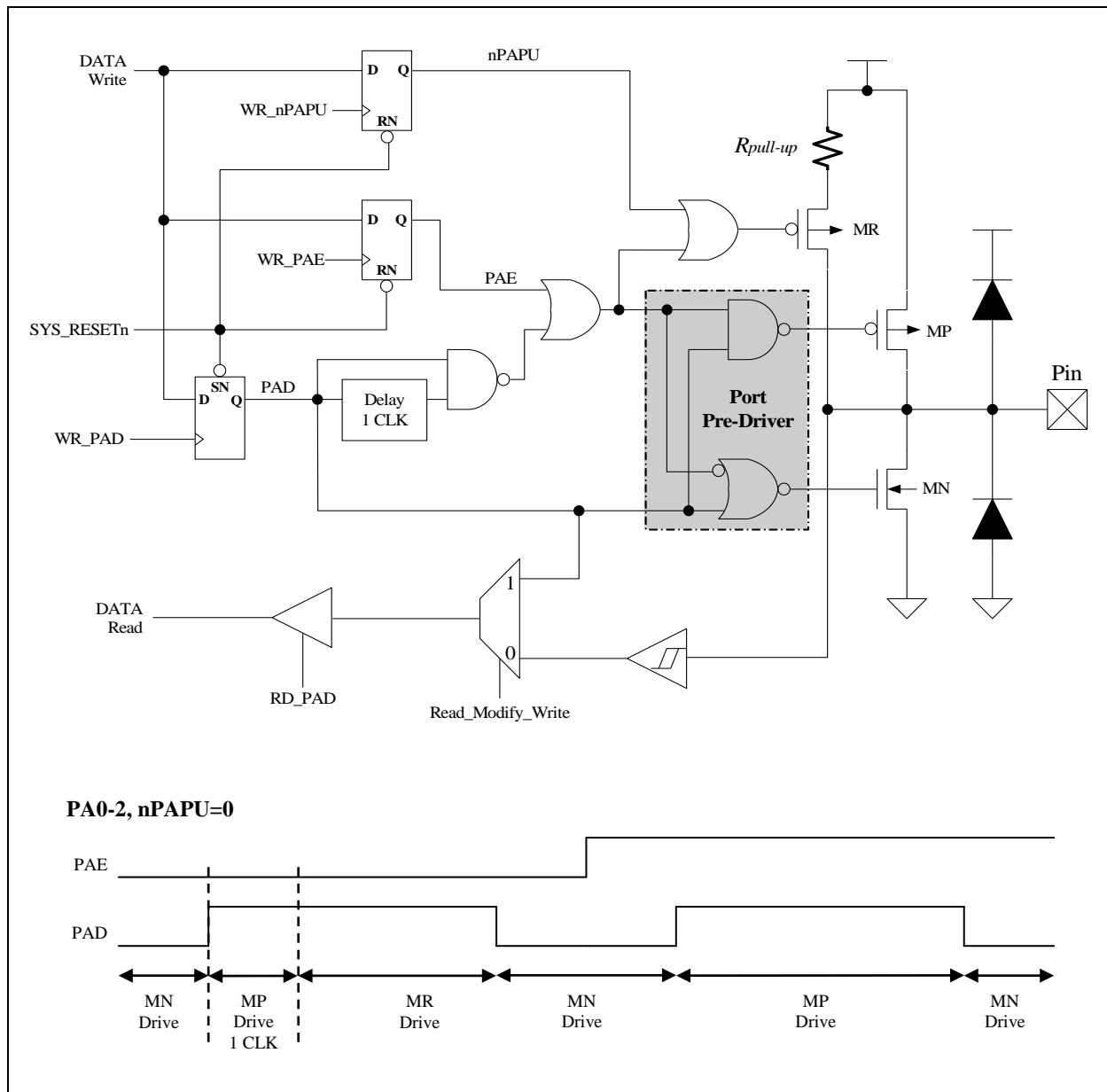
## 4.2 PA3-6, PB0-1, PD0-7

These pins are almost the same as PA0-2, except they do not support pseudo-open-drain mode. They can be used in pure **open-drain** mode, instead.
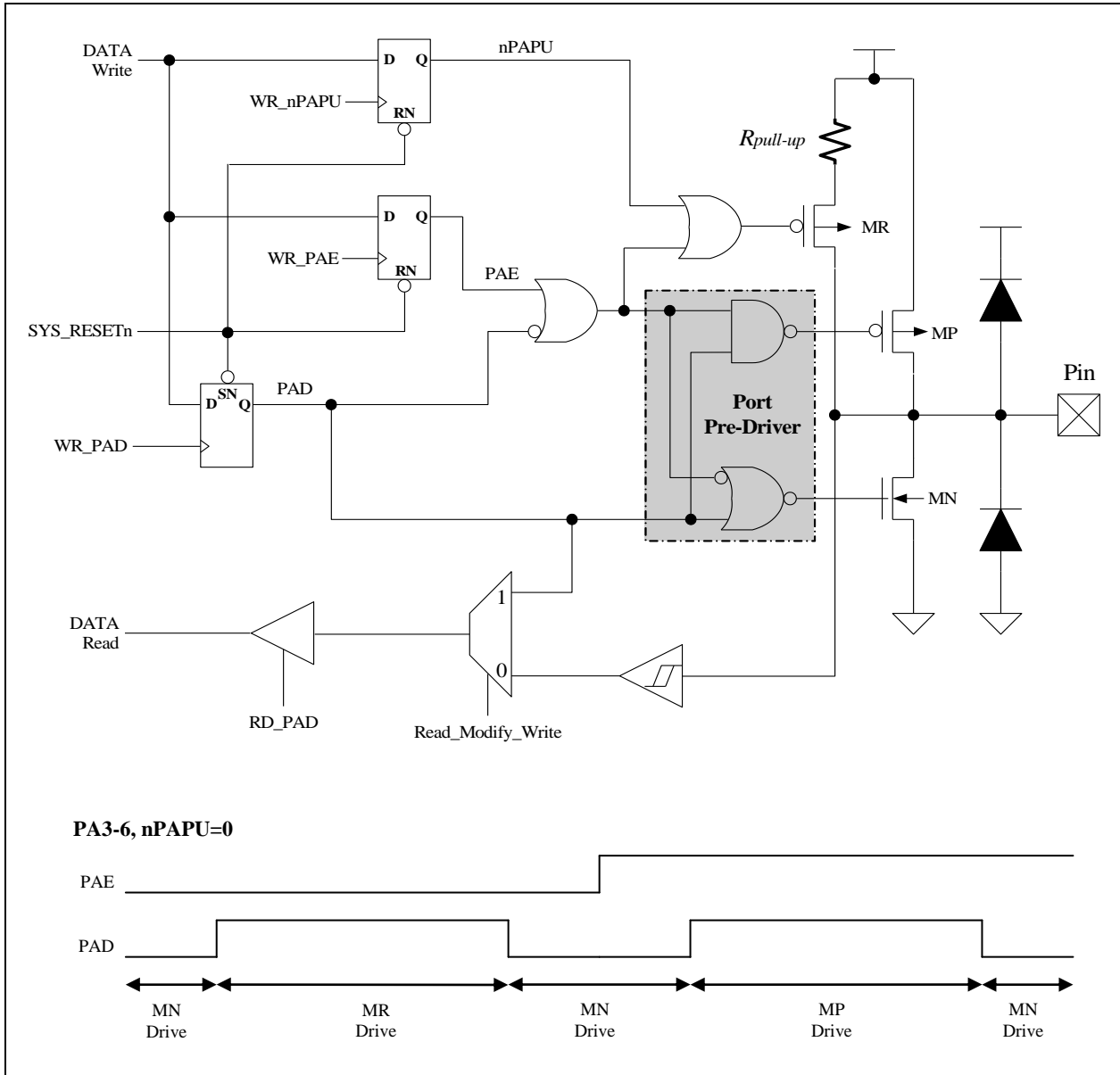
## 4.3 PA7

PA7 can be only used in Schmitt-trigger input mode. The pull-up resistor is always connected to this pin.
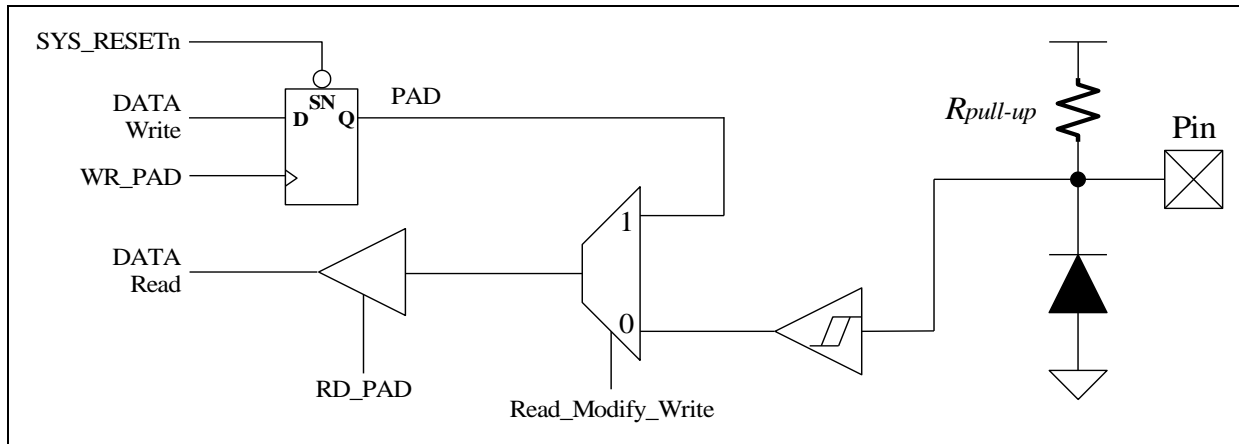
# MEMORY MAP

## F-Plane

| Name | Address | R/W | Rst | Description |
|------|---------|-----|-----|-------------|
| 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 |
| TIMER0 | 01.7~0 | R/W | 0 | Timer0 content |
| PC | 02.7~0 | R/W | 0 | Programming Counter [7~0] |
| RAMBANK | 03.5 | R/W | 0 | RAM Bank Selection |
| TO | 03.4 | R | 0 | WDT time out flag, cleared by 'SLEEP', 'CLRWDT' instruction |
| PD | 03.3 | R | 0 | Sleep mode flag, cleared by 'CLRWDT' instruction |
| ZFLAG | 03.2 | R/W | 0 | Zero Flag |
| DCFLAG | 03.1 | R/W | 0 | Decimal Carry Flag |
| CFLAG | 03.0 | R/W | 0 | Carry Flag |
| FSR | 04.7 | R | 0 | Fixed 0 |
| | 04.6~0 | R/W | - | File Select Register, indirect address mode pointer |
| PA[7] | 05.7 | R | - | PA7 pin state |
| PAD[6:0] | 05.6~0 | R | - | Port A pin or "data register" state |
| | | W | 7F | Port A output data register |
| PBD[1:0] | 06.1~0 | R | - | Port B pin or "data register" state |
| | | W | 3 | Port B output data register |
| PDD[7:0] | 07.7~0 | R | - | Port D pin or "data register" state |
| | | W | FF | Port D output data register |
| TM1I | 09.5 | R | - | Timer1 interrupt event pending flag, set by H/W while Timer1 overflows |
| | | W | 0 | write 0: clear this flag; write 1: no action |
| TM0I | 09.4 | R | - | Timer0 interrupt event pending flag, set by H/W while Timer0 overflows |
| | | W | 0 | write 0: clear this flag; write 1: no action |
| WKTI | 09.3 | R | - | WKT interrupt event pending flag, set by H/W while WKT time out |
| | | W | 0 | write 0: clear this flag; write 1: no action |
| XINT2 | 09.2 | R | - | INT2 pin (PA7) interrupt event pending flag, set by H/W at INT2 pin's falling edge |
| | | W | 0 | write 0: clear this flag; write 1: no action |
| XINT1 | 09.1 | R | - | INT1 pin (PA1) interrupt event pending flag, set by H/W at INT1 pin's falling or rising edge |
| | | W | 0 | write 0: clear this flag; write 1: no action |
| XINT0 | 09.0 | R | - | INT0 pin (PA6) interrupt event pending flag, set by H/W at INT0 pin's falling edge |
| | | W | 0 | write 0: clear this flag; write 1: no action |
| TIMER1 | 0a.7~0 | R/W | 0 | Timer1 content |
| PWM0DUTY | 0c.7~0 | R/W | 0 | PWM0 duty 8-bit MSB |
| | 0d.7~6 | R/W | 0 | PWM0 duty 2-bit LSB |
| PWM1DUTY | 0e.7~0 | R/W | 0 | PWM1 duty 8-bit MSB |
| | 0f.7~6 | R/W | 0 | PWM1 duty 2-bit LSB |
| ADCQ | 10.7~0 | R | - | ADC conversion result ADCQ[11:4] |
| | 11.7~4 | R | - | ADC conversion result ADCQ[3:0] |
| ADCSTART | 11.3 | R | - | H/W clears this bit after ADC end of conversion |
| | | W | 0 | S/W sets this bit to start ADC conversion |
| ADCSEL | 11.2~0 | R/W | 0 | ADC channel select; 0: ADC0, 1: ADC1,…,7: ADC7 |
| RAM | 20~27 | R/W | - | Internal RAM – Common Area |
| | 28~7F | R/W | - | Internal RAM – RAM Bank0 |
| | 28~7F | R/W | - | Internal RAM – RAM Bank1 |

## R-Plane

| Name | Address | R/W | Rst | Discription |
|---|---|---|---|---|
| TCOUT_PSC | 02.7~6 | W | 0 | TCOUT Pre-Scale<br>00: TCOUT output clock is "Instruction Cycle" divided by 1<br>01: TCOUT output clock is "Instruction Cycle" divided by 2<br>10: TCOUT output clock is "Instruction Cycle" divided by 4<br>11: TCOUT output clock is "Instruction Cycle" divided by 8 |
| T0IEDGE | 02.5 | W | 0 | 0: T0I (PA2) rising edge to increase Timer0/PSC count<br>1: T0I (PA2) falling edge to increase Timer0/PSC count |
| SELT0I | 02.4 | W | 0 | 0: Timer0/PSC clock source is "Instruction Cycle"<br>1: Timer0/PSC clock source is T0I pin |
| TM0PSC | 02.3~0 | W | 0 | Timer0 Pre-Scale<br>    0000: Timer0 input clock is "Instruction Cycle" divided by 1<br>    0001: Timer0 input clock is "Instruction Cycle" divided by 2<br>    ~<br>    0111: Timer0 input clock is "Instruction Cycle" divided by 128<br>    1000: Timer0 input clock is "Instruction Cycle" divided by 256 |
| PWRDOWN | 03 | W |  | write this register to enter Power-Down Mode |
| CLRWDT | 04 | W |  | write this register to clear WDT/WKT |
| PAE | 05.6~3 | W | 0 | Each bit controls its corresponding pin, if the bit is<br>0: the pin is open-drain output or Schmitt-trigger input<br>1: the pin is CMOS push-pull output |
|  | 05.2~0 | W | 0 | Each bit controls its corresponding pin, if the bit is<br>0: the pin is pseudo-open-drain output or Schmitt-trigger input<br>1: the pin is CMOS push-pull output |
| PBE | 06.1~0 | W | 0 | Each bit controls its corresponding pin, if the bit is<br>0: the pin is open-drain output or Schmitt-trigger input<br>1: the pin is CMOS push-pull output |
| PDE | 07.7~0 | W | 0 | Each bit controls its corresponding pin, if the bit is<br>0: the pin is open-drain output or Schmitt-trigger input<br>1: the pin is CMOS push-pull output |
| nPAPU | 08.6~0 | W | 7F | Each bit controls its corresponding pin, if the bit is<br>0: the pin pull up resistor is enabled, except<br>    a. the pin's output data register (PAD) is 0<br>    b. the pin's CMOS push-pull mode is chosen (PAE=1)<br>    c. the pin is working for Crystal or external RC oscillation<br>    d. PA0 is working for PWM0 output<br>1: the pin pull up resistor is disabled |
| nPBPU | 09.1~0 | W | 3 | Each bit controls its corresponding pin, if the bit is<br>0: the pin pull up resistor is enabled, except<br>    a. the pin's output data register (PBD) is 0<br>    b. the pin's CMOS push-pull mode is chosen (PBE=1)<br>    c. PB0 is working for PWM1 output<br>1: the pin pull up resistor is disabled |
| nPDPU | 0a.7~0 | W | FF | Each bit controls its corresponding pin, if the bit is<br>0: the pin pull up resistor is enabled, except<br>    a. the pin's output data register (PDD) is 0<br>    b. the pin's CMOS push-pull mode is chosen (PDE=1)<br>    c. the pin is working for T1OUT/BUZZER/TCOUT output<br>1: the pin pull up resistor is disabled |

| Name | Address | R/W | Rst | Discription |
|---|---|---|---|---|
| BUZ_OUT | 0b.7 | W | 0 | 0: disable BUZZER output to PD1 pin<br>1: enable BUZZER output to PD1 pin |
| PWM0_OUT | 0b.6 | W | 0 | 0: disable PWM0 output to PA0 pin<br>1: enable PWM0 output to PA0 pin |
| PWM1_OUT | 0b.5 | W | 0 | 0: disable PWM1 output to PB0 pin<br>1: enable PWM1 output to PB0 pin |
| INT1EDGE | 0b.4 | W | 0 | 0: INT1 pin (PA1) falling edge to trigger interrupt event<br>1: INT1 pin (PA1) rising edge to trigger interrupt event |
| TC_OUT | 0b.3 | W | 0 | 0: disable Instruction Clock divider output to PD6 pin<br>1: enable Instruction Clock divider output to PD6 pin |
| TM1_OUT | 0b.2 | W | 0 | 0: disable Timer1 match out (T1OUT) to PD0<br>1: enable Timer1 match out (T1OUT) to PD0 |
| WKTPSC | 0b.1~0 | W | 11 | WDT/WKT period (VCC=5V)<br>  00: 11 ms<br>  01: 23 ms<br>  10: 45 ms<br>  11: 90 ms |
| BUZ_EN | 0c.7 | W | 0 | 0: disable BUZZER timer counting<br>1: enable BUZZER timer counting |
| ADCLKS | 0c.6~4 | W | 0 | 000: ADC clock is "Instruction Cycle" divided by 128<br>001: ADC clock is "Instruction Cycle" divided by 64<br> ~<br>111: ADC clock is "Instruction Cycle" divided by 1 |
| TM1PSC | 0c.3~0 | W | 0 | 0000: Timer1 input clock is "Instruction Cycle" divided by 1<br>0001: Timer1 input clock is "Instruction Cycle" divided by 2<br> ~<br>0111: Timer1 input clock is "Instruction Cycle" divided by 128<br>1000: Timer1 input clock is "Instruction Cycle" divided by 256 |
| TM1RELD | 0d.7~0 | W | 0 | Timer1 reload offset value while it rolls over |
| TM1IE | 0e.5 | W | 0 | Timer1 interrupt enable, 1=enable, 0=disable |
| TM0IE | 0e.4 | W | 0 | Timer0 interrupt enable, 1=enable, 0=disable |
| WKTIE | 0e.3 | W | 0 | Wakeup Timer interrupt enable, 1=enable, 0=disable |
| XINT2E | 0e.2 | W | 0 | INT2 pin (PA7) interrupt enable, 1=enable, 0=disable |
| XINT1E | 0e.1 | W | 0 | INT1 pin (PA1) interrupt enable, 1=enable, 0=disable |
| XINT0E | 0e.0 | W | 0 | INT0 pin (PA6) interrupt enable, 1=enable, 0=disable |
| TESTREG | 0f.1~0 | W | 0 | Test mode register, for manufacturer only, user does not write it |
| BUZ_PSC | 10.7~6 | W | 0 | BUZZER Clock Prescaler<br>00: BUZZER clock is "Instruction Cycle" divided by 4<br>01: BUZZER clock is "Instruction Cycle" divided by 8<br>10: BUZZER clock is "Instruction Cycle" divided by 16<br>11: BUZZER clock is "Instruction Cycle" divided by 32 |
| BUZ_PROD | 10.5~0 | W | 0 | BUZZER Period Data. BUZZER output is BUZZER Clock divided by BUZ_PROD |
| IVC_REG | 11.4~3 | W | 0 | Built-in regulator control in sleep mode<br>00: VCC > 4.5 V<br>01: 4.5 V > VCC > 3.6 V<br>10: 3.6 V > VCC |
| ADC_TRIM | 11.2~0 | W | 0 | Test mode register, for manufacturer only, user does not write it |
| nADC_IE | 12.7~0 | W | ff | Each bit controls its corresponding ADC7~0 enable pin, if the bit is<br>0: the corresponding pin is ADC input<br>1: the corresponding pin is digital input |

# INSTRUCTION SET

Each instruction is a 14-bit word divided into an OPCODE, 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" or "r" 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 | F-Plane Register File Address |
| r | R-Plane 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 |
| DC | Decimal Carry 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 |
|---|---|---|---|---|---|
| **Byte-Oriented File Register Instruction** | | | | | |
| ADDWF | f,d | 00 0111 dfff ffff | 1 | C, DC, Z | Add W and "f" |
| ANDWF | f,d | 00 0101 dfff ffff | 1 | Z | AND W with "f" |
| CLRF | f | 00 0001 1fff ffff | 1 | Z | Clear "f" |
| CLRW | | 00 0001 0100 0000 | 1 | Z | Clear W |
| COMF | f,d | 00 1001 dfff ffff | 1 | Z | Complement "f" |
| DECF | f,d | 00 0011 dfff ffff | 1 | Z | Decrement "f" |
| DECFSZ | f,d | 00 1011 dfff ffff | 1 or 2 | - | Decrement "f", skip if zero |
| INCF | f,d | 00 1010 dfff ffff | 1 | Z | Increment "f" |
| INCFSZ | f,d | 00 1111 dfff ffff | 1 or 2 | - | Increment "f", skip if zero |
| IORWF | f,d | 00 0100 dfff ffff | 1 | Z | OR W with "f" |
| MOVFW | f | 00 1000 0fff ffff | 1 | - | Move "f" to W |
| MOVWF | f | 00 0000 1fff ffff | 1 | - | Move W to "f" |
| MOVWR | r | 00 0000 00rr rrrr | 1 | - | Move W to "r" |
| RLF | f,d | 00 1101 dfff ffff | 1 | C | Rotate left "f" through carry |
| RRF | f,d | 00 1100 dfff ffff | 1 | C | Rotate right "f" through carry |
| SUBWF | f,d | 00 0010 dfff ffff | 1 | C, DC, Z | Subtract W from "f" |
| SWAPF | f,d | 00 1110 dfff ffff | 1 | - | Swap nibbles in "f" |
| TESTZ | f | 00 1000 1fff ffff | 1 | Z | Test if "f" is zero |
| XORWF | f,d | 00 0110 dfff ffff | 1 | Z | XOR W with "f" |
| **Bit-Oriented File Register Instruction** | | | | | |
| BCF | f,b | 01 000b bbff ffff | 1 | - | Clear "b" bit of "f" |
| BSF | f,b | 01 001b bbff ffff | 1 | - | Set "b" bit of "f" |
| BTFSC | f,b | 01 010b bbff ffff | 1 or 2 | - | Test "b" bit of "f", skip if clear |
| BTFSS | f,b | 01 011b bbff ffff | 1 or 2 | - | Test "b" bit of "f", skip if set |
| **Literal and Control Instruction** | | | | | |
| ADDLW | k | 01 1100 kkkk kkkk | 1 | C, DC, Z | Add Literal "k" and W |
| ANDLW | k | 01 1011 kkkk kkkk | 1 | Z | AND Literal "k" with W |
| CALL | k | 10 kkkk kkkk kkkk | 2 | - | Call subroutine "k" |
| CLRWDT | | 00 0000 0000 0100 | 1 | TO, PD | Clear Watchdog/Wakeup Timer |
| GOTO | k | 11 kkkk kkkk kkkk | 2 | - | Jump to branch "k" |
| IORLW | k | 01 1010 kkkk kkkk | 1 | Z | OR Literal "k" with W |
| MOVLW | k | 01 1001 kkkk kkkk | 1 | - | Move Literal "k" to W |
| NOP | | 00 0000 0000 0000 | 1 | - | No operation |
| RET | | 00 0000 0100 0000 | 2 | - | Return from subroutine |
| RETI | | 00 0000 0110 0000 | 2 | - | Return from interrupt |
| RETLW | k | 01 1000 kkkk kkkk | 2 | - | Return with Literal in W |
| SLEEP | | 00 0000 0000 0011 | 1 | TO, PD | Go into standby mode, Clock oscillation stops |
| XORLW | k | 01 1111 kkkk kkkk | 1 | Z | XOR Literal "k" with W |

| **ADDLW** | **Add Literal "k" and W** | |
|---|---|---|
| Syntax | ADDLW k | |
| Operands | k : 00h ~ FFh | |
| Operation | (W) ← (W) + k | |
| Status Affected | C, DC, Z | |
| OP-Code | 01 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 |
| | | A : W = 0x25 |

| **ADDWF** | **Add W and "f"** | |
|---|---|---|
| Syntax | ADDWF f [,d] | |
| Operands | f : 00h ~ 7Fh d : 0, 1 | |
| Operation | (Destination) ← (W) + (f) | |
| Status Affected | C, DC, Z | |
| OP-Code | 00 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 | ADDWF FSR, 0 | B : W = 0x17, FSR = 0xC2 |
| | | A : W = 0xD9, FSR = 0xC2 |

| **ANDLW** | **Logical AND Literal "k" with W** | |
|---|---|---|
| Syntax | ANDLW k | |
| Operands | k : 00h ~ FFh | |
| Operation | (W) ← (W) 'AND' (f) | |
| Status Affected | Z | |
| OP-Code | 01 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 |
| | | A : W = 0x03 |

| **ANDWF** | **AND W with "f"** | |
|---|---|---|
| Syntax | ANDWF f [,d] | |
| Operands | f : 00h ~ 7Fh d : 0, 1 | |
| Operation | (Destination) ← (W) 'AND' (f) | |
| Status Affected | Z | |
| OP-Code | 00 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 | ANDWF FSR, 1 | B : W = 0x17, FSR = 0xC2 |
| | | A : W = 0x17, FSR = 0x02 |

| **BCF** | **Clear "b" bit of "f"** | |
|---|---|---|
| Syntax | BCF  f [,b] | |
| Operands | f : 00h ~ 3Fh   b : 0 ~ 7 | |
| Operation | (f.b) ← 0 | |
| Status Affected | - | |
| OP-Code | 01 000b bbff ffff | |
| Description | Bit 'b' in register 'f' is cleared. | |
| Cycle | 1 | |
| Example | BCF FLAG_REG, 7 | B : FLAG_REG = 0xC7 |
| | | A : FLAG_REG = 0x47 |

| **BSF** | **Set "b" bit of "f"** | |
|---|---|---|
| Syntax | BSF  f [,b] | |
| Operands | f : 00h ~ 3Fh   b : 0 ~ 7 | |
| Operation | (f.b) ← 1 | |
| Status Affected | - | |
| OP-Code | 01 001b bbff ffff | |
| Description | Bit 'b' in register 'f' is set. | |
| Cycle | 1 | |
| Example | BSF FLAG_REG, 7 | B : FLAG_REG = 0x0A |
| | | A : FLAG_REG = 0x8A |

| **BTFSC** | **Test "b" bit of "f", skip if clear(0)** | |
|---|---|---|
| Syntax | BTFSC  f [,b] | |
| Operands | f : 00h ~ 3Fh   b : 0 ~ 7 | |
| Operation | Skip next instruction if  (f.b) = 0 | |
| Status Affected | - | |
| OP-Code | 01 010b bbff 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  BTFSC  FLAG, 1 | B : PC = LABEL1 |
| | TRUE    GOTO SUB1 | A : if FLAG.1 = 0, PC = FALSE |
| | FALSE   ... | if FLAG.1 = 1, PC = TRUE |

| **BTFSS** | **Test "b" bit of "f", skip if set(1)** | |
|---|---|---|
| Syntax | BTFSS  f [,b] | |
| Operands | f : 00h ~ 3Fh   b : 0 ~ 7 | |
| Operation | Skip next instruction if  (f.b) = 1 | |
| Status Affected | - | |
| OP-Code | 01 011b bbff 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 | | |
| Example | LABEL1  BTFSS  FLAG, 1 | B : PC = LABEL1 |
| | TRUE    GOTO SUB1 | A : if FLAG.1 = 0, PC = TRUE |
| | FALSE   ... | if FLAG.1 = 1, PC = FALSE |

## CALL            Call subroutine "k"

| | |
|---|---|
| Syntax | CALL k |
| Operands | K : 00h ~ FFFh |
| Operation | Operation: TOS ← (PC)+ 1, PC.11~0 ← k |
| Status Affected | - |
| OP-Code | 10 kkkkk kkkk kkkk |
| Description | Call Subroutine. First, return address (PC+1) is pushed onto the stack. The 12-bit immediate address is loaded into PC bits <11:0>. CALL is a two-cycle instruction. |
| Cycle | 2 |
| Example | LABEL1    CALL SUB1        B : PC = LABEL1 |
| |                                   A : PC = SUB1, TOS = LABEL1+1 |

## CLRF            Clear "f"

| | |
|---|---|
| Syntax | CLRF f |
| Operands | f : 00h ~ 7Fh |
| Operation | (f) ← 00h, Z ← 1 |
| Status Affected | Z |
| OP-Code | 00 0001 1fff ffff |
| Description | The contents of register 'f' are cleared and the Z bit is set. |
| Cycle | 1 |
| Example | CLRF FLAG_REG        B : FLAG_REG = 0x5A |
| |                                   A : FLAG_REG = 0x00, Z = 1 |

## CLRW            Clear W

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

## CLRWDT            Clear Watchdog Timer

| | |
|---|---|
| Syntax | CLRWDT |
| Operands | - |
| Operation | WDT/WKT Timer ← 00h |
| Status Affected | TO, PD |
| OP-Code | 00 0000 0000 0100 |
| Description | CLRWDT instruction clears the Watchdog/Wakeup Timer |
| Cycle | 1 |
| Example | CLRWDT                   B : WDT counter = ? |
| |                                   A : WDT counter = 0x00 |

| COMF | Complement "f" |
|---|---|
| Syntax | COMF f [,d] |
| Operands | f : 00h ~ 7Fh, d : 0, 1 |
| Operation | (destination) ← ($\bar{f}$) |
| Status Affected | Z |
| OP-Code | 00 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 | COMF REG1,0          B : REG1 = 0x13 |
|  |                              A : REG1 = 0x13, W = 0xEC |

| DECF | Decrement "f" |
|---|---|
| Syntax | DECF f [,d] |
| Operands | f : 00h ~ 7Fh, d : 0, 1 |
| Operation | (destination) ← (f) - 1 |
| Status Affected | Z |
| OP-Code | 00 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 | DECF CNT, 1          B : CNT = 0x01, Z = 0 |
|  |                              A : CNT = 0x00, Z = 1 |

| DECFSZ | Decrement "f", Skip if 0 |
|---|---|
| Syntax | DECFSZ f [,d] |
| Operands | f : 00h ~ 7Fh, d : 0, 1 |
| Operation | (destination) ← (f) - 1, skip next instruction if result is 0 |
| Status Affected | - |
| OP-Code | 00 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  DECFSZ CNT, 1          B : PC = LABEL1 |
|  |             GOTO LOOP                    A : CNT = CNT – 1 |
|  |             CONTINUE                        if CNT=0, PC = CONTINUE |
|  |                                                     if CNT≠0, PC = LABEL1+1 |

| **GOTO** | **Unconditional Branch** | |
| --- | --- | --- |
| Syntax | GOTO k | |
| Operands | k : 00h ~ FFFh | |
| Operation | PC.11~0 ← k | |
| Status Affected | - | |
| OP-Code | 11 kkkk kkkk kkkk | |
| Description | GOTO is an unconditional branch. The 12-bit immediate value is loaded into PC bits <11:0>. GOTO is a two-cycle instruction. | |
| Cycle | 2 | |
| Example | LABEL1    GOTO SUB1 | B : PC = LABEL1 |
| | | A : PC = SUB1 |

| **INCF** | **Increment "f"** | |
| --- | --- | --- |
| Syntax | INCF f [,d] | |
| Operands | f : 00h ~ 7Fh | |
| Operation | (destination) ← (f) + 1 | |
| Status Affected | Z | |
| OP-Code | 00 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 | INCF CNT, 1 | B : CNT = 0xFF, Z = 0 |
| | | A : CNT = 0x00, Z = 1 |

| **INCFSZ** | **Increment "f", Skip if 0** | |
| --- | --- | --- |
| Syntax | INCFSZ f [,d] | |
| Operands | f : 00h ~ 7Fh, d : 0, 1 | |
| Operation | (destination) ← (f) + 1, skip next instruction if result is 0 | |
| Status Affected | - | |
| OP-Code | 00 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    INCFSZ CNT, 1 | B : PC = LABEL1 |
| | GOTO LOOP | A : CNT = CNT + 1 |
| | CONTINUE | if CNT=0, PC = CONTINUE |
| | | if CNT≠0, PC = LABEL1+1 |

| **IORLW** | **Inclusive OR Literal with W** | |
| --- | --- | --- |
| Syntax | IORLW k | |
| Operands | k : 00h ~ FFh | |
| Operation | (W) ← (W) OR k | |
| Status Affected | Z | |
| OP-Code | 01 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 |
| | | A : W = 0xBF, Z = 0 |

| **IORWF** | **Inclusive OR W with "f"** | |
|---|---|---|
| Syntax | IORWF f [,d] | |
| Operands | f : 00h ~ 7Fh, d : 0, 1 | |
| Operation | (destination) ← (W) OR k | |
| Status Affected | Z | |
| OP-Code | 00 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 | IORWF RESULT, 0 | B : RESULT = 0x13, W = 0x91 |
| | | A : RESULT = 0x13, W = 0x93, Z = 0 |


| **MOVFW** | **Move "f" to W** | |
|---|---|---|
| Syntax | MOVFW f | |
| Operands | f : 00h ~ 7Fh | |
| Operation | (W) ← (f) | |
| Status Affected | - | |
| OP-Code | 00 1000 0fff ffff | |
| Description | The contents of register f are moved to W register. | |
| Cycle | 1 | |
| Example | MOVF FSR, 0 | B : W = ? |
| | | A : W ← f,  if W = 0 Z = 1 |


| **MOVLW** | **Move Literal to W** | |
|---|---|---|
| Syntax | MOVLW k | |
| Operands | k : 00h ~ FFh | |
| Operation | (W) ← k | |
| Status Affected | - | |
| OP-Code | 01 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 = ? |
| | | A : W = 0x5A |


| **MOVWF** | **Move W to "f"** | |
|---|---|---|
| Syntax | MOVWF f | |
| Operands | f : 00h ~ 7Fh | |
| Operation | (f) ← (W) | |
| Status Affected | - | |
| OP-Code | 00 0000 1fff ffff | |
| Description | Move data from W register to register 'f'. | |
| Cycle | 1 | |
| Example | MOVWF REG1 | B : REG1 = 0xFF, W = 0x4F |
| | | A : REG1 = 0x4F, W = 0x4F |

| **MOVWR** | **Move W to "r"** | |
|---|---|---|
| Syntax | MOVWR r | |
| Operands | r : 00h ~ 3Fh | |
| Operation | (r) ← (W) | |
| Status Affected | - | |
| OP-Code | 00 0000 00rr rrrr | |
| Description | Move data from W register to register 'r'. | |
| Cycle | 1 | |
| Example | MOVWR REG1 | B : REG1 = 0xFF, W = 0x4F |
| | | A : REG1 = 0x4F, W = 0x4F |

| **NOP** | **No Operation** | |
|---|---|---|
| Syntax | NOP | |
| Operands | - | |
| Operation | No Operation | |
| Status Affected | - | |
| OP-Code | 00 0000 0000 0000 | |
| Description | No Operation | |
| Cycle | 1 | |
| Example | NOP | - |

| **RETI** | **Return from Interrupt** | |
|---|---|---|
| Syntax | RETI | |
| Operands | - | |
| Operation | PC ← TOS, GIE ← 1 | |
| Status Affected | - | |
| OP-Code | 00 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 | RETFIE | A : PC = TOS, GIE = 1 |

| **RETLW** | **Return with Literal in W** | |
|---|---|---|
| Syntax | RETLW k | |
| Operands | k : 00h ~ FFh | |
| Operation | PC ← TOS, (W) ← k | |
| Status Affected | - | |
| OP-Code | 01 1000 kkkk kkkk | |
| Description | The W register is loaded with the 8-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 | CALL TABLE | B : W = 0x07 |
| | : | A : W = value of k8 |
| | TABLE ADDWF PCL,1 | |
| |     RETLW k1 | |
| |     RETLW k2 | |
| |      : | |
| |      RETLW kn | |

| **RET** | **Return from Subroutine** | |
| --- | --- | --- |
| Syntax | RET | |
| Operands | - | |
| Operation | PC ← TOS | |
| Status Affected | - | |
| OP-Code | 00 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 | RETURN | A : PC = TOS |

| **RLF** | **Rotate Left f through Carry** | |
| --- | --- | --- |
| Syntax | RLF f [,d] | |
| Operands | f : 00h ~ 7Fh, d : 0, 1 | |
| Operation |  | |
| Status Affected | C | |
| OP-Code | 00 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 | RLF REG1,0 | B : REG1 = 1110 0110, C = 0 <br> A : REG1 = 1110 0110 <br> W　　 = 1100 1100, C = 1 |

| **RRF** | **Rotate Right "f" through Carry** | |
| --- | --- | --- |
| Syntax | RRF f [,d] | |
| Operands | f : 00h ~ 7Fh, d : 0, 1 | |
| Operation |  | |
| Status Affected | C | |
| OP-Code | 00 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 | RRF REG1,0 | B : REG1 = 1110 0110, C = 0 <br> A : REG1 = 1110 0110 <br> W　　 = 0111 0011, C = 0 |

| **SLEEP** | **Go into standby mode, Clock oscillation stops** | |
|---|---|---|
| Syntax | SLEEP | |
| Operands | - | |
| Operation | - | |
| Status Affected | TO, PD | |
| OP-Code | 00 0000 0000 0011 | |
| Description | Go into SLEEP mode with the oscillator stopped. | |
| Cycle | 1 | |
| Example | SLEEP | - |

| **SUBWF** | **Subtract W from "f"** | |
|---|---|---|
| Syntax | SUBWF f [,d] | |
| Operands | f : 00h ~ 7Fh, d : 0, 1 | |
| Operation | (W) ← (f) – (W) | |
| Status Affected | C, DC, Z | |
| OP-Code | 00 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 | SUBWF REG1,1 | B : REG1 = 3, W = 2, C = ?, Z = ? |
| | | A : REG1 = 1, W = 2, C = 1, Z = 0 |
| | SUBWF REG1,1 | B : REG1 = 2, W = 2, C = ?, Z = ? |
| | | A : REG1 = 0, W = 2, C = 1, Z = 1 |
| | SUBWF REG1,1 | B : REG1 = 1, W = 2, C = ?, Z = ? |
| | | A : REG1 = FFh, W = 2, C = 0, Z = 0 |

| **SWAPF** | **Swap Nibbles in "f"** | |
|---|---|---|
| Syntax | SWAPF f [,d] | |
| Operands | f : 00h ~ 7Fh, d : 0, 1 | |
| Operation | (destination,7~4) ← (f.3~0), (destination.3~0) ← (f.7~4) | |
| Status Affected | - | |
| OP-Code | 00 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 | SWAPF REG, 0 | B : REG1 = 0xA5 |
| | | A : REG1 = 0xA5, W = 0x5A |

| **TESTZ** | **Test if "f" is zero** | |
|---|---|---|
| Syntax | TESTZ f | |
| Operands | f : 00h ~ 7Fh | |
| Operation | Set Z flag if (f) is 0 | |
| Status Affected | Z | |
| OP-Code | 00 1000 1fff ffff | |
| Description | If the content of register 'f' is 0, Zero flag is set to 1. | |
| Cycle | 1 | |
| Example | TESTZ REG1 | B : REG1 = 0, Z = ? |
| | | A : REG1 = 0, Z = 1 |

| **XORLW** | **Exclusive OR Literal with W** | |
|---|---|---|
| Syntax | XORLW k | |
| Operands | k : 00h ~ FFh | |
| Operation | (W) ← (W)  XOR  k | |
| Status Affected | Z | |
| OP-Code | 01 1111 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 |
| | | A : W = 0x1A |

| **XORWF** | **Exclusive OR W with "f"** | |
|---|---|---|
| Syntax | XORWF f [,d] | |
| Operands | f : 00h ~ 7Fh, d : 0, 1 | |
| Operation | (destination) ← (W)  XOR  (f) | |
| Status Affected | Z | |
| OP-Code | 00 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 | XORWF REG 1 | B : REG = 0xAF, W = 0xB5 |
| | | A : REG = 0x1A, W = 0xB5 |

# ELECTRICAL CHARACTERISTICS

## 1. Absolute Maximum Ratings (T$_A$ = 25℃)

| Parameter | Rating | Unit |
|---|---|---|
| Supply voltage | V$_{SS}$ - 0.3 to V$_{SS}$ + 6.5 | V |
| Input voltage | V$_{SS}$ - 0.3 to V$_{CC}$ + 0.3 | |
| Output voltage | V$_{SS}$ - 0.3 to V$_{CC}$ + 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 +85 | ℃ |
| Storage temperature | -65 to +150 | |

## 2. DC Characteristics ($T_A$ = 25 C, $V_{CC}$ = 2.0V to 5.5V)

| Parameter | Sym | Conditions | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| Input High Voltage | $V_{IH}$ | All Input, except PA7 | $V_{CC}$ = 5V | $0.5V_{CC}$ | | | V |
| | | | $V_{CC}$ = 3V | $0.55V_{CC}$ | | | V |
| | | PA7 | $V_{CC}$ = 3~5V | $0.6V_{CC}$ | | | V |
| Input Low Voltage | $V_{IL}$ | All Input, except PA7 | $V_{CC}$ = 5V | | | $0.3V_{CC}$ | V |
| | | | $V_{CC}$ = 3V | | | $0.25V_{CC}$ | V |
| | | PA7 | $V_{CC}$ = 3~5V | | | $0.4V_{CC}$ | V |
| Output High Current | $I_{OH}$ | All Output | $V_{CC}$ = 5V,$V_{OH}$=4.5V | 3 | 7 | | mA |
| | | | $V_{CC}$ = 3V,$V_{OH}$=2.7V | 2 | 4 | | |
| Output Low Current | $I_{OL}$ | All Output | $V_{CC}$ = 5V,$V_{OL}$=0.5V | 9 | 18 | | mA |
| | | | $V_{CC}$ = 3 V,$V_{OL}$=0.3V | 4 | 8 | | |
| Input Leakage Current (pin high) | $I_{ILH}$ | All Input | $V_{IN}$ = $V_{CC}$ | – | – | 1 | uA |
| Input Leakage Current (pin low) | $I_{ILL}$ | All Input | $V_{IN}$ = 0V | – | – | –1 | uA |
| Power Supply Current | $I_{CC}$ | Run 10 MHz No Load | $V_{CC}$ = 4.5 to 5.5V | – | 5 | | mA |
| | | Run 4 MHz No Load | $V_{CC}$ = 2.0V | | 1 | | |
| | | Stop mode No Load (TM57FA40A) | $V_{CC}$ = 5V, IVC_REG = 0, IVC disable | – | 5 | 7 | uA |
| | | | $V_{CC}$ = 5V, IVC enable | | | 200 | |
| | | | $V_{CC}$ = 2.6V IVC_REG = 2, IVC disable | | 5 | 7 | |
| | | Stop mode No Load (TM57FA40) | $V_{CC}$ = 5V, IVC_REG = 0, IVC disable | – | 0.1 | 1 | |
| | | | $V_{CC}$ = 5V, IVC enable | | | 200 | |
| | | | $V_{CC}$ = 2.6V IVC_REG = 2, IVC disable | | 0.1 | 1 | |
| System Clock Frequency | $f_{OSC}$ | $V_{CC}$ > LVR$_{th}$ | $V_{CC}$ = 3.0V | – | | 12 | MHz |
| | | | $V_{CC}$ = 2.6V | | | 10 | |
| | | | $V_{CC}$ = 2.4V | | | 4~8 | |
| LVR reference Voltage | | $V_{LVR}$ | | 1.8 | 2.1 | 2.3 | V |
| | | | | 2.6 | 2.9 | 3.2 | |
| LVR Hysteresis Voltage | | $V_{HYST}$ | | – | ±0.1 | – | V |
| Low Voltage Detection time | | $t_{LVR}$ | | 100 | – | – | μs |
| Pull-Up Resistor | $R_P$ | $V_{IN}$ = 0 V Ports A/B/D | $V_{CC}$ = 5V | | 130 | | k |
| | | | $V_{CC}$ = 3V | | 300 | | |
| | | $V_{IN}$ = 0 V PA7 | $V_{CC}$ = 5V | | 80 | | k |
| | | | $V_{CC}$ = 3V | | 90 | | |

**3. Clock Timing** (T$_A$ = -40 C to +85 C)

| Parameter | Condition | | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| External RC Frequency | V$_{CC}$ = 3V | R = 5K | C = 33 pF | – | 1.8 | – | MHz |
| | | R = 10K | C = 100 pF | – | 0.73 | – | |
| | | R = 100K | C = 300 pF | – | 0.04 | – | |
| | V$_{CC}$ = 5V | R = 5K | C = 33 pF | – | 2.5 | – | |
| | | R = 10K | C = 100 pF | – | 0.66 | – | |
| | | R = 100K | C = 300 pF | – | 0.03 | – | |
| Internal RC Frequency | V$_{CC}$ = 4.75 to 5.25V | | | Typ-2% | 3.9 | Typ+2% | |
| | V$_{CC}$ = 2.8 to 3.2V | | | Typ+4% | 3.83 | Typ+4% | |
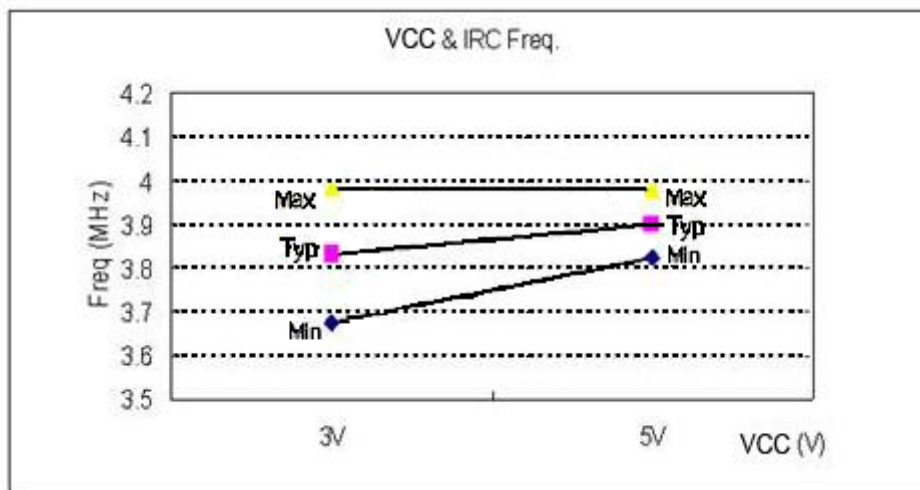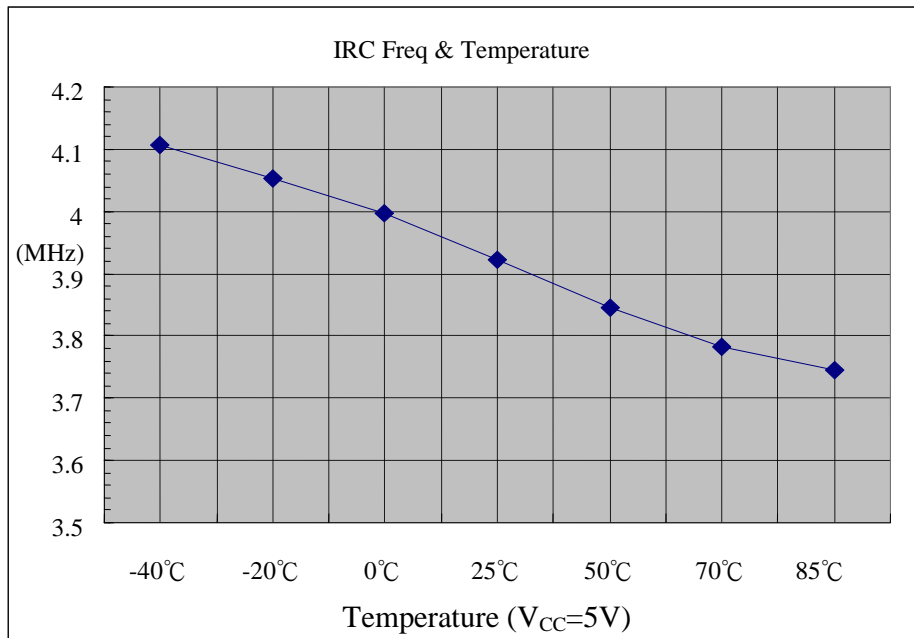
**4. Reset Timing Characteristics** ($T_A = 25°C$, $V_{CC} = 2.0V$ to 5.5V)

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| RESET Input Low width | Input $V_{CC} = 5V \pm 10\%$ | 3 | – | – | μs |
| WDT wakeup time | $V_{CC} = 5V$, WKTPSC = 11 | 80 | 90 | 100 | ms |
| | $V_{CC} = 3V$, WKTPSC = 11 | 108 | 120 | 132 | |
| CPU start up time | $V_{CC} = 5V$ | – | 3.5 | - | ms |

**5. ADC Electrical Characteristics** ($T_A = 25°C$, $V_{CC} = 2.0V$ to 5.5V, VSS = 0V)

| Parameter | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Total Accuracy | $V_{CC} = 5.12V$, VSS = 0V | – | ± 2.5 | ± 4 | LSB |
| Integral Non-Linearity | | – | ± 3.2 | ± 5 | |
| Max Input Clock ($f_{ADC}$) | – | – | – | 2 | MHz |
| Conversion Time | $f_{ADC} = 2$ MHz | – | 25 | – | μs |
| Input Voltage | – | VSS | – | $V_{CC}$ | V |

# PACKAGING INFORMATION

The ordering information:

| Ordering number | Package |
|---|---|
| TM57FA40-MTP<br>TM57FA40A-MTP | Wafer / Dice blank chip |
| TM57FA40-COD<br>TM57FA40A-COD | Wafer / Dice with code |
| TM57FA40-MTP-03<br>TM57FA40A-MTP-03 | DIP 16-pin (300 mil) |
| TM57FA40-MTP-16<br>TM57FA40A-MTP-16 | SOP 16-pin (150 mil) |
| TM57FA40-MTP-05<br>TM57FA40A-MTP-05 | DIP 20-pin (300 mil) |
| TM57FA40-MTP-21<br>TM57FA40A-MTP-21 | SOP 20-pin (300 mil) |
| TM57FA40-MTP-01<br>TM57FA40A-MTP-01 | DIP 8-pin (300 mil) |
| TM57FA40-MTP-14<br>TM57FA40A-MTP-14 | SOP 8-pin (150 mil) |

## 20-DIP Package Dimension
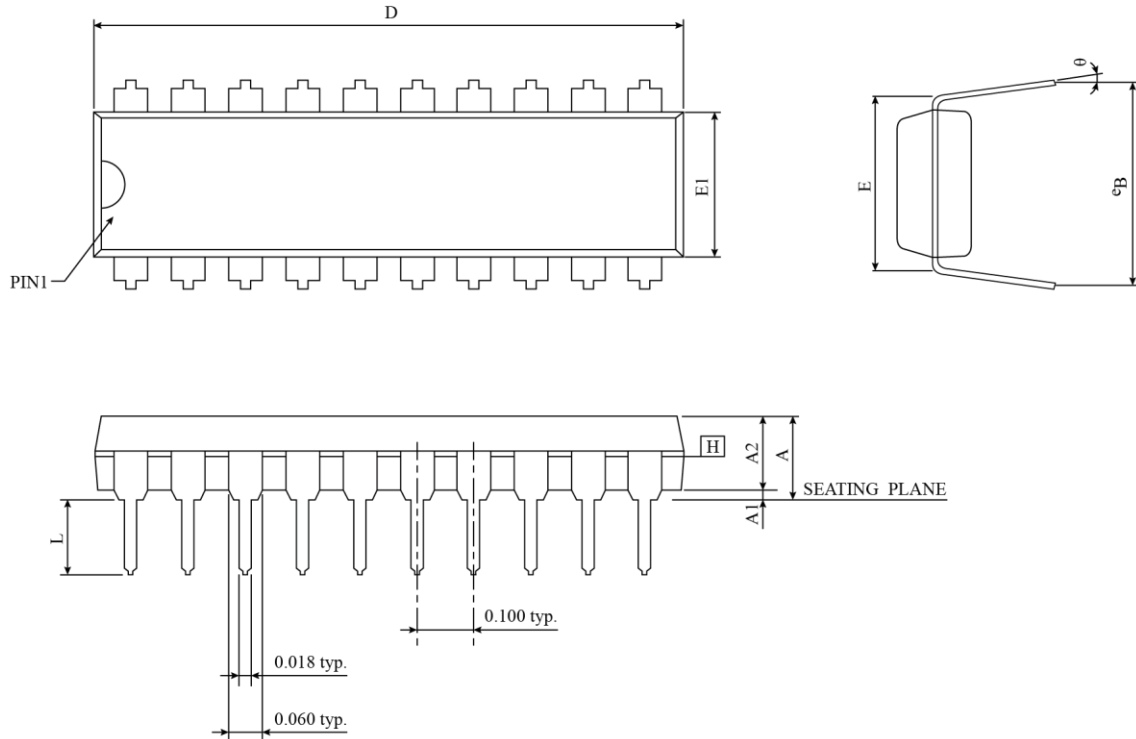


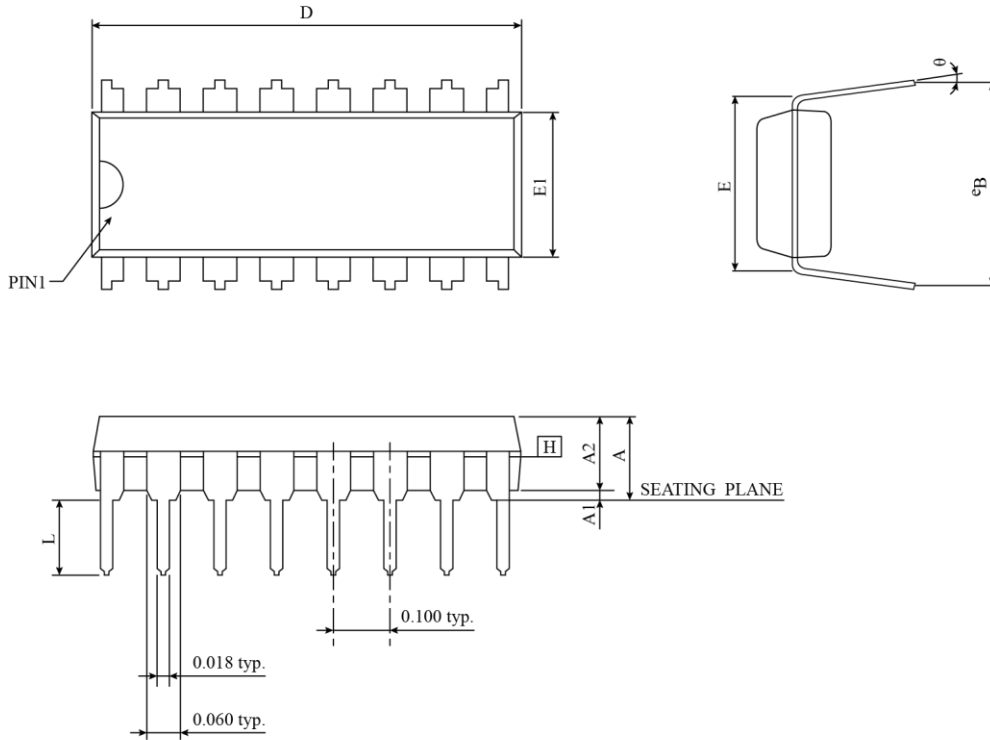| SYMBOL | DIMENSION IN MM | | DIMENSION IN INCH | |
|---|---|---|---|---|
| | MIN | MAX | MIN | MAX |
| A | - | 4.445 | - | 0.175 |
| A1 | 0.381 | - | 0.015 | - |
| A2 | 3.175 | 3.429 | 0.125 | 0.135 |
| D | 25.705 | 26.416 | 1.012 | 1.040 |
| E | 7.620 | 7.874 | 0.300 | 0.310 |
| E1 | 6.223 | 6.477 | 0.245 | 0.255 |
| L | 3.048 | 3.556 | 0.120 | 0.140 |
| $^eB$ | 8.509 | 9.525 | 0.335 | 0.375 |
| θ | 0° | 15° | 0° | 15° |
| JEDEC | MS-001 (AD) | | | |

NOTES：

1. "D"，"E1" DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS. MOLD FLASH OR PROTRUSIONS SHALL NOTEXCEED .010 INCH.

2. eB IS MEASURED AT THE LEAD TIPS WITH THE LEADS UNCONSTRAINED.

3. POINTED OR ROUNDED LEAD TIPS ARE PREFERRED TO EASE INSERTION.

4. DISTANCE BETWEEN LEADS INCLUDING DAM BAR PROTRUSIONS TO BE .005 INCH MININUM.

5. DATUM PLANE [H] COINCIDENT WITH THE BOTTOM OF LEAD, WHERE LEAD EXITS BODY.

## 20-SOP Package Dimension
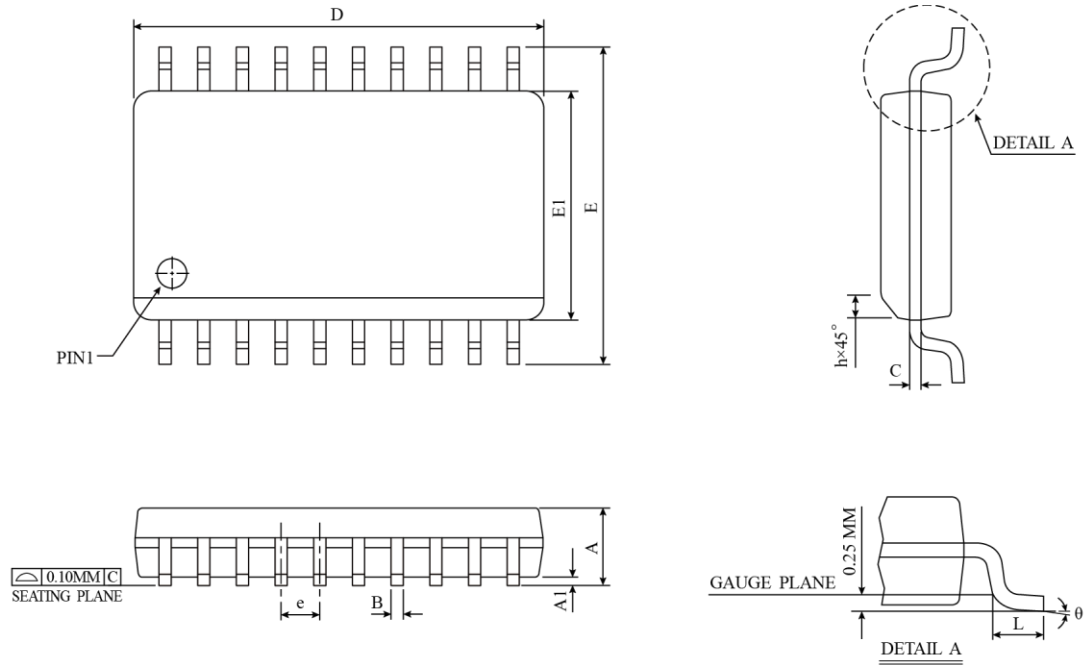


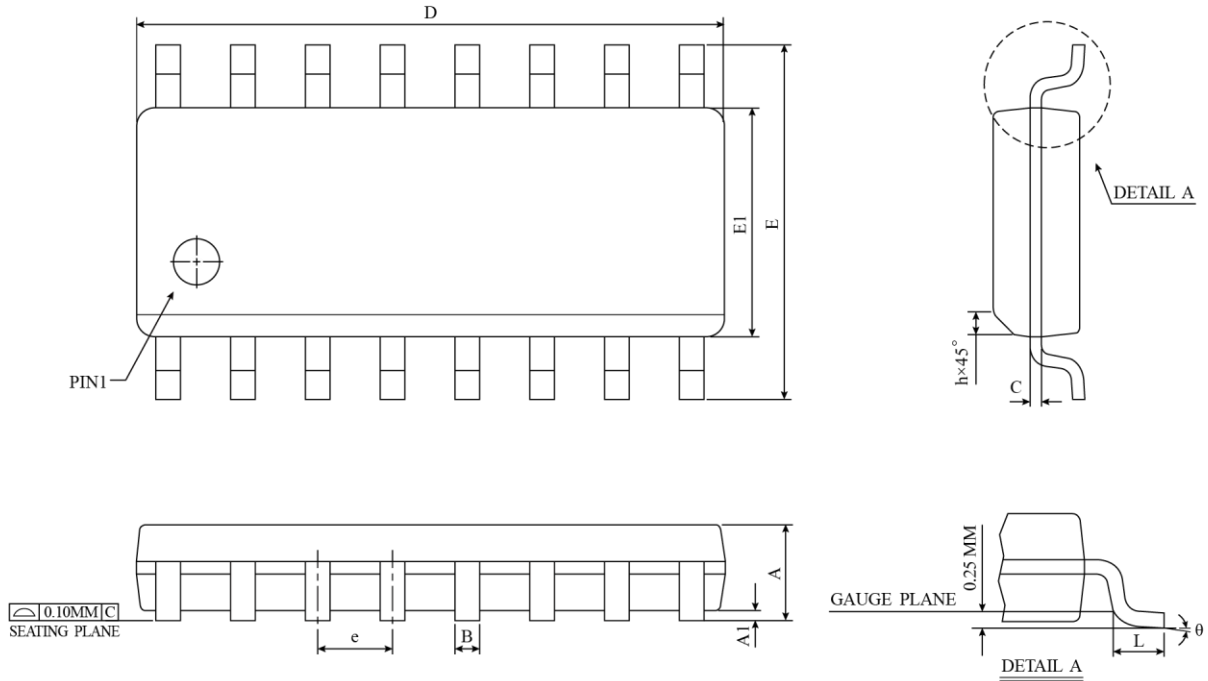| SYMBOL | DIMENSION IN MM | | DIMENSION IN INCH | |
|---|---|---|---|---|
| | MIN | MAX | MIN | MAX |
| A | 2.35 | 2.65 | 0.0926 | 0.1043 |
| A1 | 0.10 | 0.30 | 0.0040 | 0.0118 |
| B | 0.33 | 0.51 | 0.013 | 0.020 |
| C | 0.23 | 0.32 | 0.0091 | 0.0125 |
| D | 12.60 | 13.00 | 0.4961 | 0.5118 |
| E | 10.00 | 10.65 | 0.394 | 0.491 |
| E1 | 7.40 | 7.60 | 0.2914 | 0.2992 |
| e | 1.27 BSC | | 0.050 BSC | |
| h | 0.25 | 0.75 | 0.010 | 0.029 |
| L | 0.40 | 1.27 | 0.016 | 0.050 |
| θ | $0^\circ$ | $8^\circ$ | $0^\circ$ | $8^\circ$ |
| 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.

## 16-DIP Package Dimension



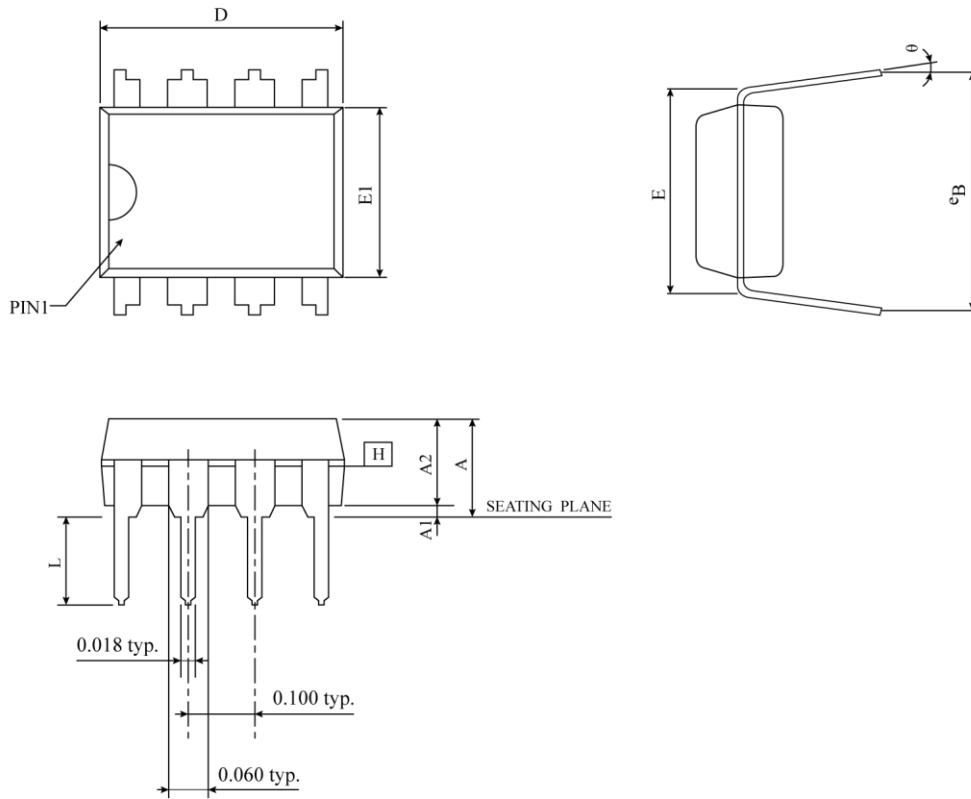| SYMBOL | DIMENSION IN MM | | DIMENSION IN INCH | |
|---|---|---|---|---|
| | MIN | MAX | MIN | MAX |
| A | - | 4.369 | - | 0.172 |
| A1 | 0.381 | 0.965 | 0.015 | 0.038 |
| A2 | 3.175 | 3.429 | 0.125 | 0.135 |
| D | 18.669 | 19.685 | 0.735 | 0.775 |
| E | 7.620 BSC | | 0.300 BSC | |
| E1 | 6.223 | 6.477 | 0.245 | 0.255 |
| L | 2.921 | 3.810 | 0.115 | 0.150 |
| eB | 8.509 | 9.525 | 0.335 | 0.375 |
| θ | $0°$ | $15°$ | $0°$ | $15°$ |
| JEDEC | MS-001 (BB) | | | |

NOTES：

1. "D" , "E1" DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS. MOLD FLASH OR PROTRUSIONS SHALL NOTEXCEED .010 INCH.
2. eB IS MEASURED AT THE LEAD TIPS WITH THE LEADS UNCONSTRAINED.
3. POINTED OR ROUNDED LEAD TIPS ARE PREFERRED TO EASE INSERTION.
4. DISTANCE BETWEEN LEADS INCLUDING DAM BAR PROTRUSIONS TO BE .005 INCH MININUM.
5. DATUM PLANE Ⓗ COINCIDENT WITH THE BOTTOM OF LEAD, WHERE LEAD EXITS BODY.

## 16-SOP Package Dimension



| SYMBOL | DIMENSION IN MM | | DIMENSION IN INCH | |
|---|---|---|---|---|
| | MIN | MAX | MIN | MAX |
| A | 1.35 | 1.75 | 0.0532 | 0.0688 |
| A1 | 0.10 | 0.25 | 0.0040 | 0.0098 |
| B | 0.33 | 0.51 | 0.013 | 0.020 |
| C | 0.19 | 0.25 | 0.0075 | 0.0098 |
| D | 9.80 | 10.00 | 0.3859 | 0.3937 |
| E | 5.80 | 6.20 | 0.2284 | 0.2440 |
| E1 | 3.80 | 4.00 | 0.1497 | 0.1574 |
| e | 1.27 BSC | | 0.050 BSC | |
| h | 0.25 | 0.50 | 0.0099 | 0.0196 |
| L | 0.40 | 1.27 | 0.016 | 0.050 |
| θ | $0^{\circ}$ | $8^{\circ}$ | $0^{\circ}$ | $8^{\circ}$ |
| 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.

## 8-DIP (300mil) Package Dimension



| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | - | - | 5.334 | - | - | 0.210 |
| A1 | 0.381 | - | - | 0.015 | - | - |
| A2 | 3.175 | 3.302 | 3.429 | 0.125 | 0.130 | 0.135 |
| D | 9.017 | 9.586 | 10.160 | 0.355 | 0.378 | 0.400 |
| E | 7.620 BSC | | | 0.300 BSC | | |
| E1 | 6.223 | 6.350 | 6.477 | 0.245 | 0.250 | 0.255 |
| L | 2.921 | 3.366 | 3.810 | 0.115 | 0.133 | 0.150 |
| $e_B$ | 8.509 | 9.017 | 9.525 | 0.335 | 0.355 | 0.375 |
| θ | $0°$ | $7.5°$ | $15°$ | $0°$ | $7.5°$ | $15°$ |
| JEDEC | MS-001 (BA) | | | | | |

NOTES：

1. `D″, `E1″ DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS. MOLD FLASH OR PROTRUSIONS SHALL NOTEXCEED .010 INCH.

2. eB IS MEASURED AT THE LEAD TIPS WITH THE LEADS UNCONSTRAINED.

3. POINTED OR ROUNDED LEAD TIPS ARE PREFERRED TO EASE INSERTION.

4. DISTANCE BETWEEN LEADS INCLUDING DAM BAR PROTRUSIONS TO BE .005 INCH MININUM.

5. DATUM PLANE H COINCIDENT WITH THE BOTTOM OF LEAD, WHERE LEAD EXITS BODY.

## 8-SOP (150mil) Package Dimension
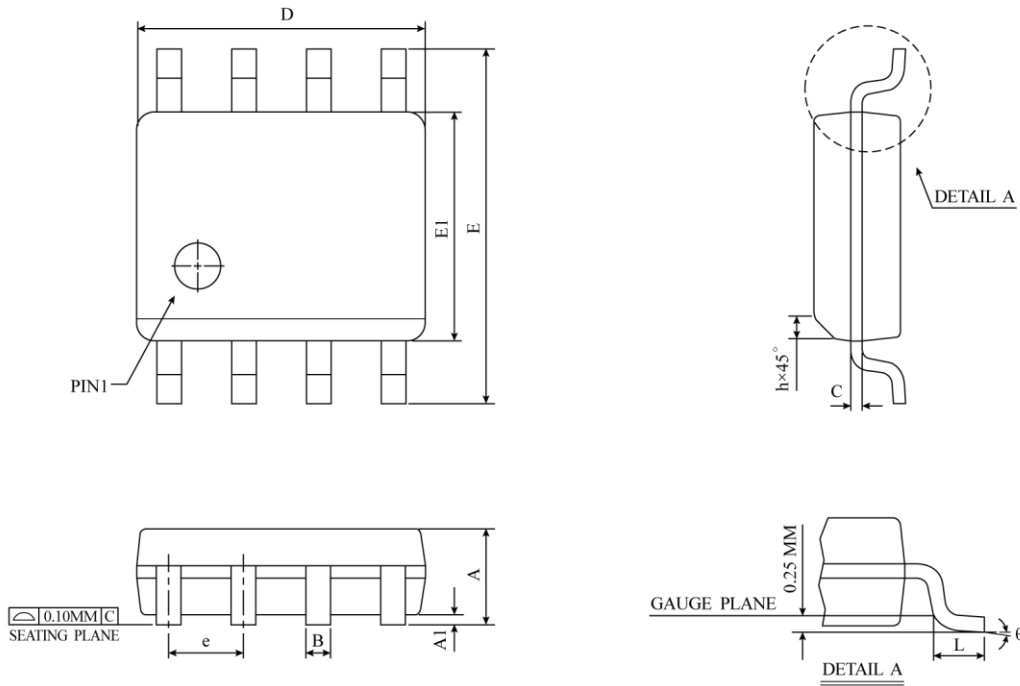


| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | 1.35 | 1.55 | 1.75 | 0.0532 | 0.0610 | 0.0688 |
| A1 | 0.10 | 0.18 | 0.25 | 0.0040 | 0.0069 | 0.0098 |
| B | 0.33 | 0.42 | 0.51 | 0.0130 | 0.0165 | 0.0200 |
| C | 0.19 | 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.