TM57PA15

DATA SHEET

Rev D0.91

AMENDMENT HISTORY

Version	Date	Description
D0.90	Mar, 2014	New release
D0.91	Dec, 2015	1. P12, LVR table update 2. P50, DC Characteristics update 3. P54, New LVR vs Temperature in Characteristic Graphs

CONTENTS

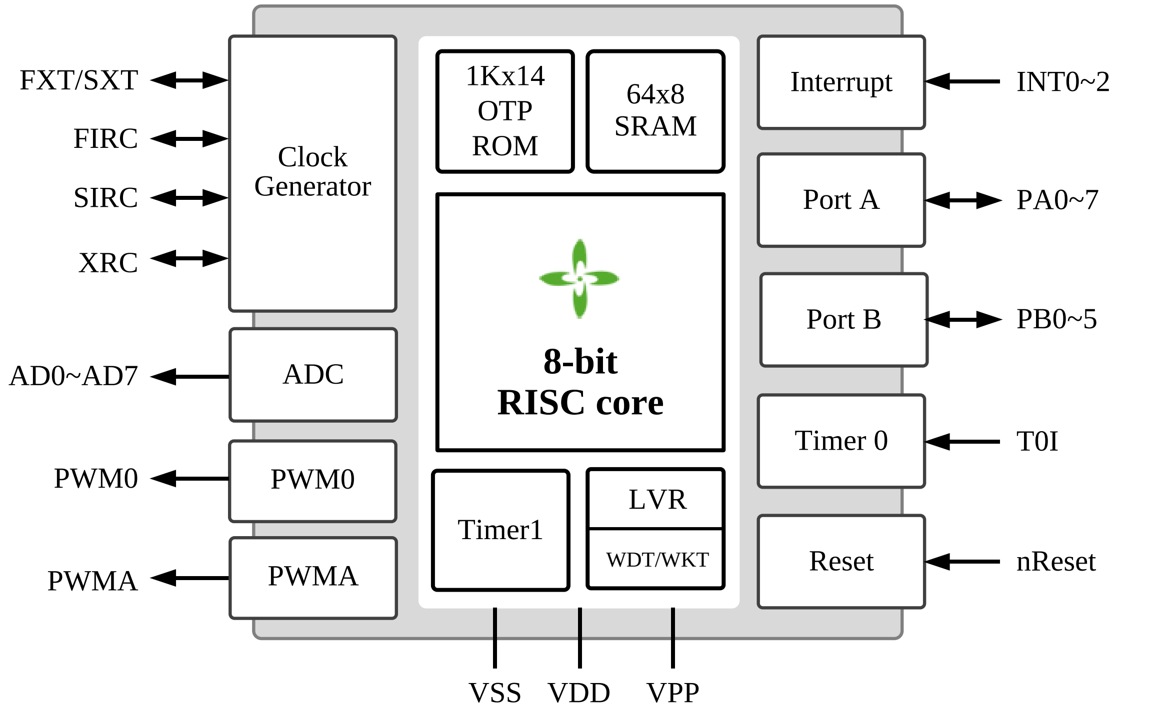
FEATURES

1. ROM: 1K x 14 bits OTP or 512 x 14 bits TTP™ (Two Time Programmable ROM)
2. RAM: 64 x 8 bits
3. STACK: 5 Levels
4. I/O ports: Two-bit programmable I/O ports (Max. 14 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. PWM0: 8-bit with 1~8 pre-scale, Interrupt / Period-adjustment / Duty-adjustment / Clear and Hold
8. PWMA: (8+2) Period-adjustment /Duty-adjustment / Clear and Hold
9. 12-bit ADC with 8 channels input
10. Watchdog/Wakeup Timer: On-chip Timer based on internal RC oscillator, 20~160 ms wakeup time
11. Reset: Power On Reset, Watchdog Reset, Low Voltage Reset, External pin Reset
12. Dual System Clock: (CPUCLK DIV: 1/2/4/16)

    Fast Clock:
    - Fast Internal RC: 8 MHz
    - Fast Crystal: 455 KHz ~24 MHz

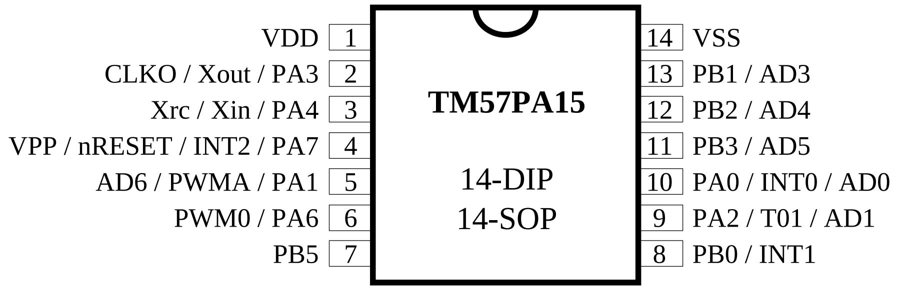
    Slow Clock:
    - Slow Crystal: 32 KHz
    - Slow Internal RC: 140KHz (default)
13. 2-Level Low Voltage Reset: 2.0V/2.8V (Can be disabled)
14. Operation Voltage: Low Voltage Reset Level to 5.5V
    - Fsys = 4 MHz, 2.4V ~ 5.5V
    - Fsys = 8 MHz, 2.5V ~ 5.5V
    - Fsys = 12 MHz, 2.7V ~ 5.5V
    - Fsys = 16 MHz, 3.1V ~ 5.5V
15. Instruction set: 38 Instructions
16. Interrupts: Three pin interrupts, Timer0/Timer1/ADC interrupt and Wakeup Timer interrupt
17. Power Down mode support
18. Package Types: 14-DIP/SOP, 16-DIP/SOP

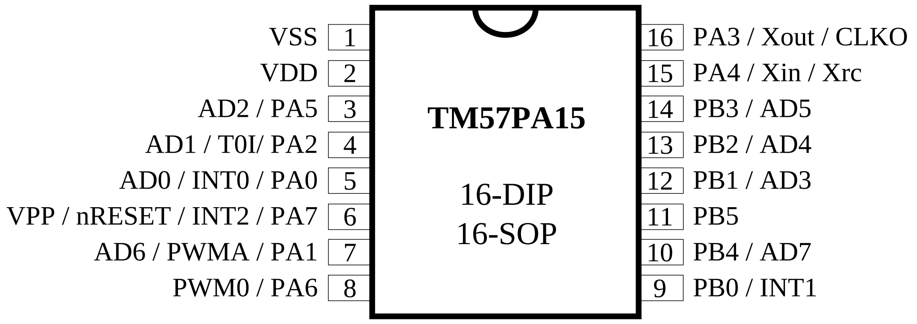
BLOCK DIAGRAM



TM57PA15 Block Diagram

PIN ASSIGNMENT





PIN DESCRIPTION

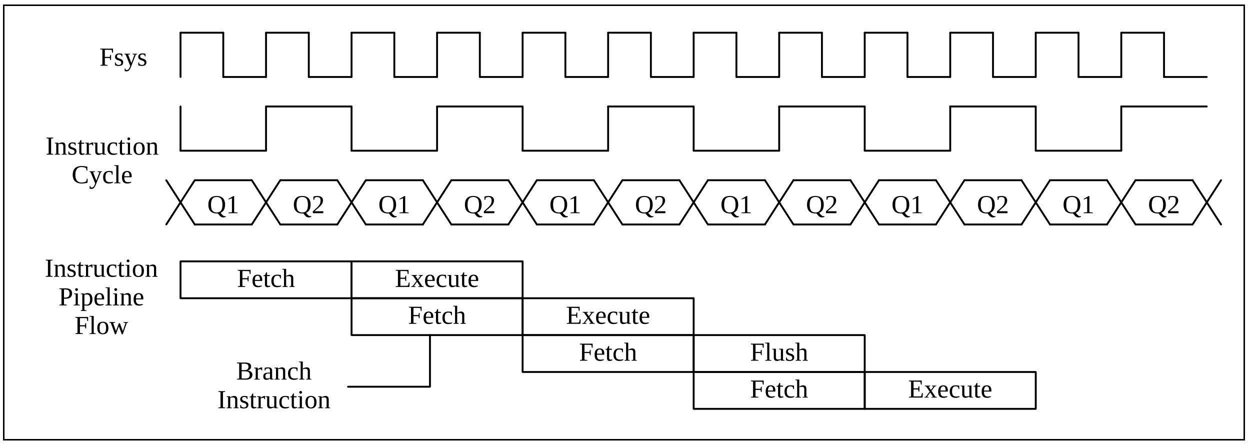
Name	In/Out	Pin Description
PA0–PA2	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or “pseudo-open-drain” output. Pull-up resistors are assignable by software.
PA3–PA6	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or open drain output. Pull-up resistors are assignable by software.
PA7	I	Schmitt-trigger input
PB0–PB5	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.
nRESET	I	External active low reset
Xin, Xout	–	Crystal/Resonator oscillator connection for system clock.
Xrc	–	External RC oscillator connection for system clock
CLKO	O	CPU Instruction clock output for external/internal RC mode
VDD, VSS	P	Voltage input pin and ground
VPP	I	PROM programming high voltage input
INT0–INT2	I	External interrupt input
AD0-AD7	I	ADC signal input
PWMA-PWM0	O	PWM output
T0I	I	Clock input to Timer0

FUNCTIONAL DESCRIPTION

1. CPU Core

1.1 Clock Scheme and Instruction Cycle

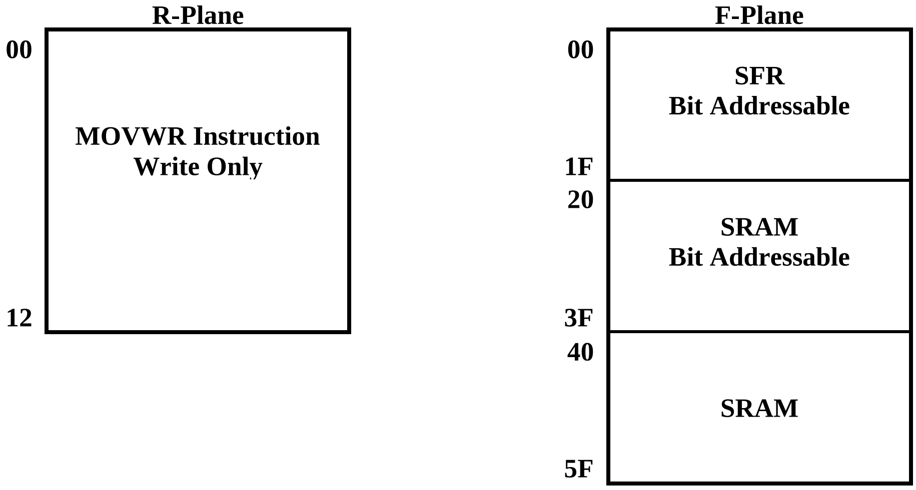
The system clock is internally divided by two to generate Q1 state and Q2 state for each instruction cycle. The Programming Counter (PC) is updated at Q1 and the instruction is fetched from program ROM and latched into the instruction register in Q2. It is then decoded and executed during the following 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 copies 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 is contained 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 10-bit wide capable of addressing a 1K 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 10 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 [9:8] keeps unchanged. The STACK is 10-bit wide and 5-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.

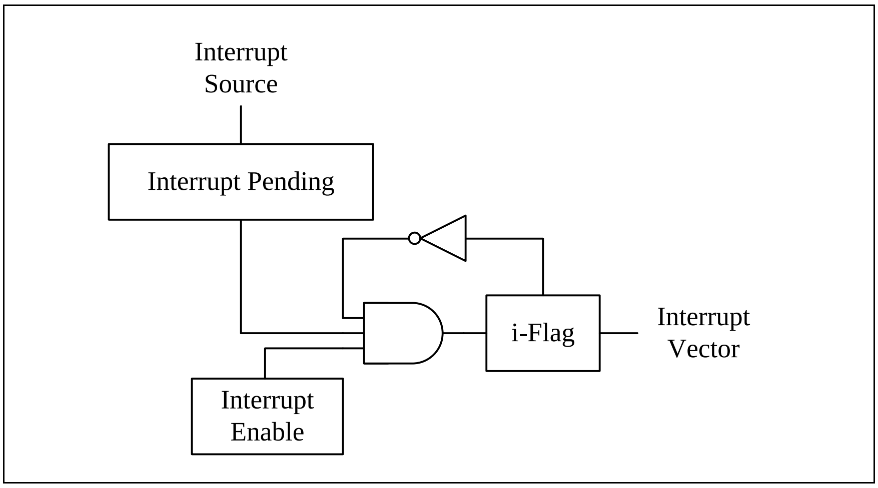
STATUS	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset Value	**0**	**–**	**–**	**0**	**0**	**0**	**0**	**0**
R/W	**R**	**R/W**	**–**	**R**	**R**	**R/W**	**R/W**	**R/W**

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td rowspan="1">7</td><td colspan="2">LVD: Low Voltage Detector Flag
LVD threshold is 2.2V/3.0V when LVR is 2.0V/2.8V.
0:V_{DD} voltage is more than LVD threshold, or LVR is disabled.
1: V_{DD} voltage is less than LVD threshold.</td></tr>
<tr><td>6</td><td colspan="2">GB1: General purpose bit</td></tr>
<tr><td>5</td><td colspan="2">Not Used</td></tr>
<tr><td>4</td><td colspan="2">TO: Time Out
0: after Power On Reset, LVR Reset, or CLRWDT/SLEEP instruction
1: WDT time out occurs</td></tr>
<tr><td>3</td><td colspan="2">PD: Power Down
0: after Power On Reset, LVR Reset, or CLRWDT instruction
1: after SLEEP instruction</td></tr>
<tr><td>2</td><td colspan="2">Z: Zero Flag
0: the result of a logic operation is not zero
1: the result of a logic operation is zero</td></tr>
<tr><td rowspan="3">1</td><td colspan="2">DC: Decimal Carry Flag or Decimal/Borrow Flag</td></tr>
<tr><td>ADD instruction</td><td>SUB instruction</td></tr>
<tr><td>1: a carry from the low nibble bits of the result occurs
0: no carry</td><td>1: no borrow
0: a borrow from the low nibble bits of the result occurs</td></tr>
<tr><td rowspan="3">0</td><td colspan="2">C: Carry Flag or Borrow Flag</td></tr>
<tr><td>ADD instruction</td><td>SUB instruction</td></tr>
<tr><td>1: a carry occurs from the MSB
0: no carry</td><td>1: no borrow
0: a borrow occurs from the MSB</td></tr>
</table>

1.6 Interrupt

The TM57PA15 has 1 level, 1 vector and five 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 TM57PA15 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 (INTIE), it will 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 TM57PA15 can be RESET in four ways.

- Power-On-Reset
- Low Voltage Reset (LVR)
- External Pin Reset
- 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 V_{DD} is more than 3.3V, while another one is suitable for application with V_{DD} 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
LVR2.8	5.5V > V_{DD} > 3.3V or V_{DD} =5.0V
LVR2.0	V_{DD} is wide voltage range

Different Fsys 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 enter dead-band and error occur.

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.

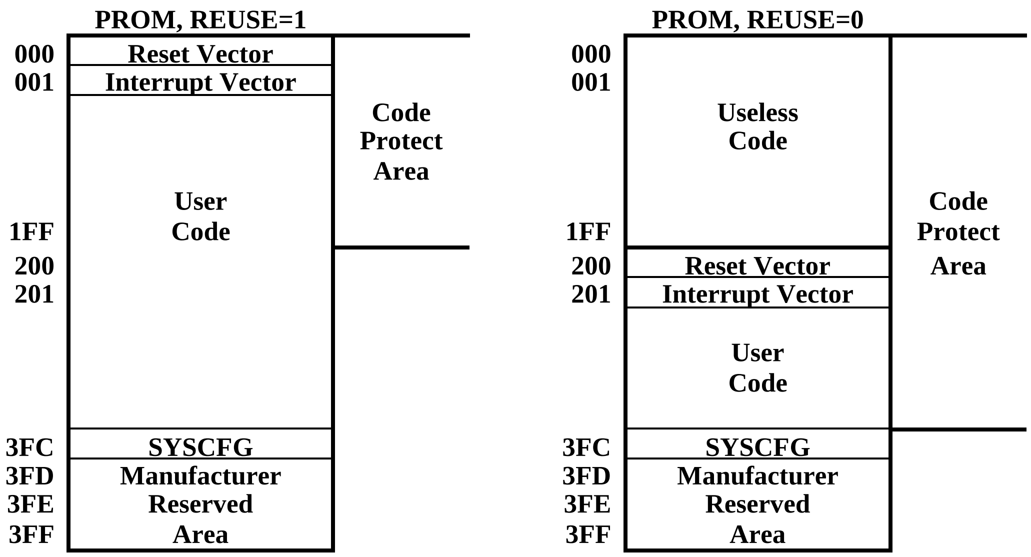
2.2 System Configuration Register (SYSCFG)

The System Configuration Register (SYSCFG) is located at ROM address 3FCh. The SYSCFG determines the option for initial condition of MCU. It is written by PROM Writer only. User can select clock source, LVR threshold voltage and chip operation mode by SYSCFG register. The default value of SYSCFG is 3FFFh. The 13th bit of SYSCFG is code protection selection bit. If this bit is 0, when user reads PROM, the data in PROM will be protected.

Bit	**13~0**	
Default Value	**11_1111_111X_XXXX**	
Bit	Description	
13	**PROTECT**: Code protection selection	
	1	Disable
	0	Enable
12	**REUSE**: PROM Re-use control	
	1	Disable (First time program)
	0	Enable (second time program)
11	**LVR**: LV reset mode	
	11	LVR = 2.0V, LVD = 2.2V, always enable.
	10	LVR = 2.0V, LVD = 2.2V, disable in STOP mode.
	01	LVR = 2.8V, LVD = 3.0V, always enable.
	00	LVR, LVD always disable.
9-8	Reserved	
7	**XRSTE**: External pin Reset Enable	
	1	Enable
	0	Disable , PA7 as input pin
6	**WDTE**: WDT Reset Enable	
	1	Enable WDT Reset (WDT), Disable Wakeup Timer (WKT)
	0	Disable WDT Reset (WDT), Enable Wakeup Timer (WKT)
4-0	**IRCF**: Internal RC Frequency adjustment control	

2.3 PROM Re-use

The PROM of this device is 1K words. For some F/W program, the program size could be less than 512 words. To fully utilize the PROM, the device allows users to reuse the PROM. This feature is named as Two Time Programmable (TTP) ROM. While the first half of PROM is occupied by a useless program code and the second half of the PROM remains blank, users can re-write the PROM with the updated program code into the second half of the PROM. In the Re-use mode, the Reset Vector and Interrupt Vector are re-allocated at the beginning of the PROM's second half by the Assembly Compiler. Users simply choose the "REUSE" option in the ICE tool interface, and then the Compiler will move the object code to proper location. That is, the user's program still has reset vector at address 000h, but the compiled object code has reset vector at 200h. In the SYSCFG, if PROTECT=0 and REUSE=1, the Code protection area is first half of PROM. This allows the Writer tool to write then verify the Code during the Re-use Code programming. After the Re-use Code being written into the PROM's second half, user should write "REUSE" control bit to "0". In the mean while, the Code protection area becomes the whole PROM except the Reserved Area.

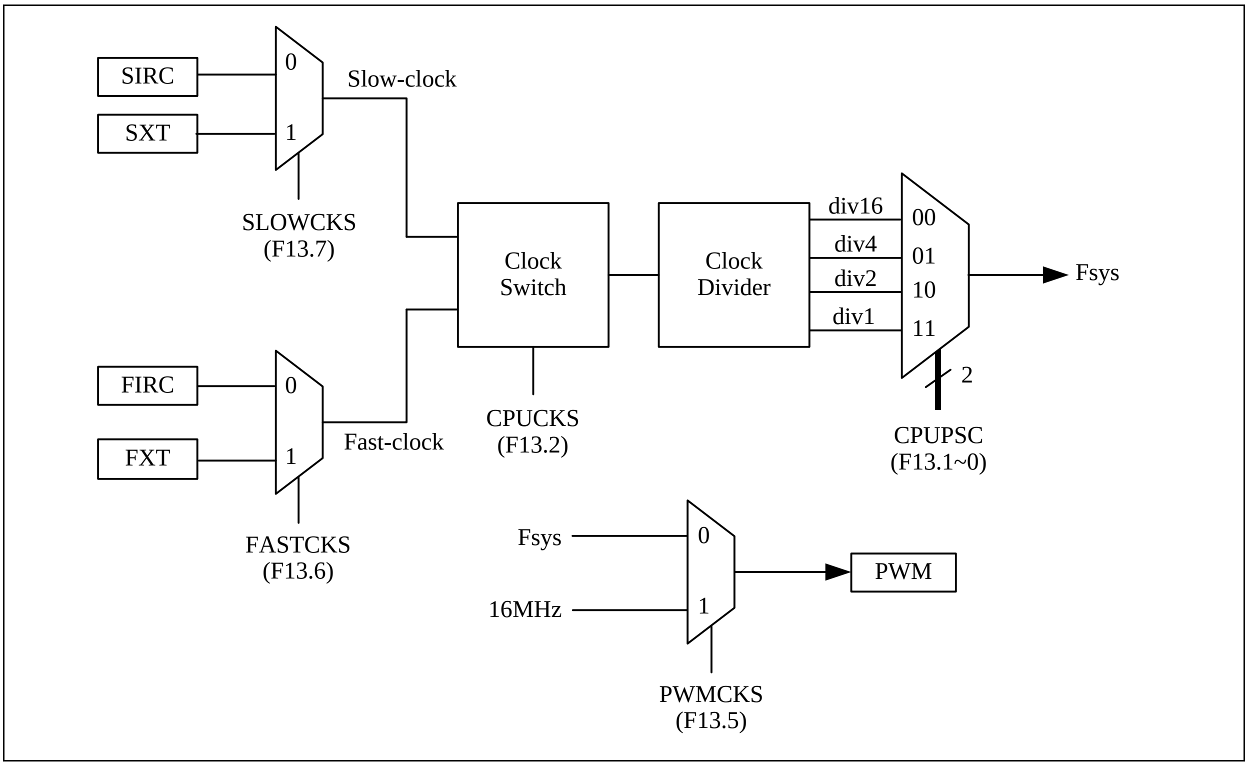


2.4 Power-Down Mode

The Power-down mode is also known as STOP Mode. Its 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, enabled Interrupts (External pin and WKT interrupts), PA1-6 low level or level charge or PB1-5 pin low level wakeup.

2.5 Dual System Clock

TM57PA15 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 source. Refer to the Figure as below.



FAST Mode:

TM57PA15 enters FAST mode by setting the CPUCKS (F13.2). In FAST mode, TM57PA15 can select FXT or FIRC as its system clock source by setting FASTCKS (F13.6). However, change Fast-clock type under FAST mode is not allowed. User should let TM57PA15 enter SLOW mode first, change FASTCKS, then back to FAST mode.

In this mode, the program is executed using Fast-clock as system clock source. The Timer0 block is driven by Fast-clock. PWM can be driven by Fast-clock or FIRC 16 MHz by setting PWMCKS (F13.5).

SLOW Mode:

After power on or reset, TM57PA15 enters SLOW mode, the default Slow-clock is SIRC. User can select SXT or SIRC as its System clock by setting SLOWCKS (F13.7). However, change Slow-clock type under SLOW mode is not allowed. User should let TM57PA15 enter FAST mode first, change SLOWCKS, then back to SLOW mode.

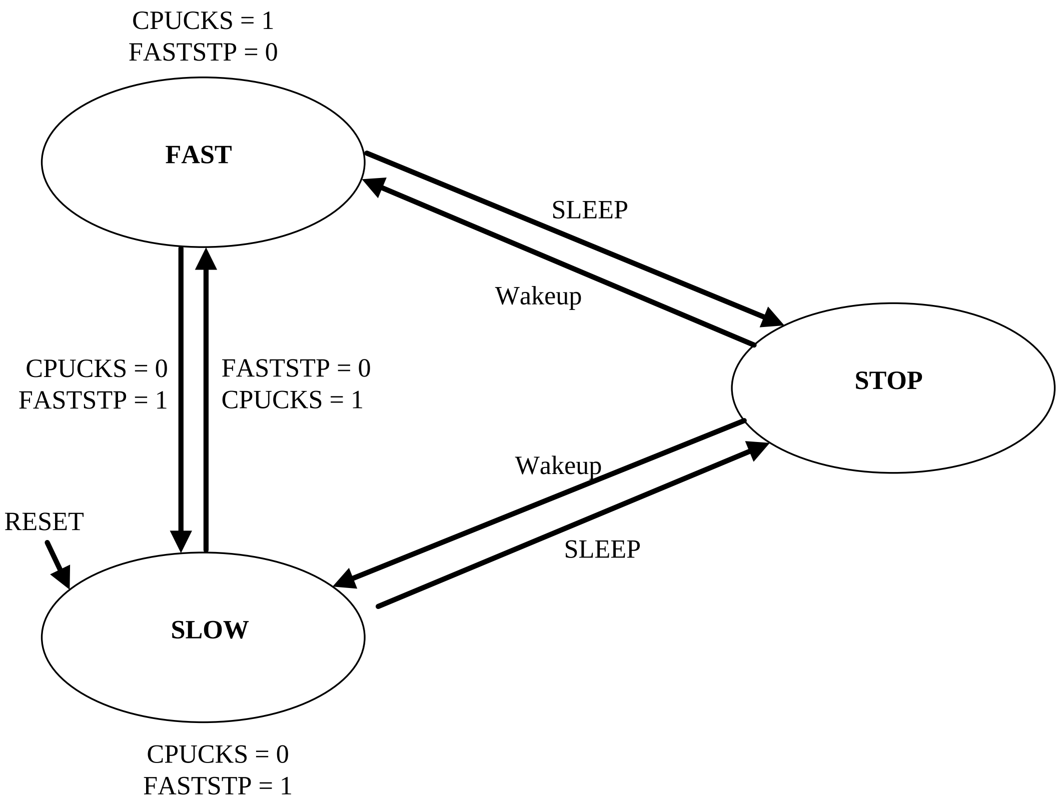
STOP Mode:

TM57PA15 will enter the “STOP Mode” after executing the SLEEP instruction. In STOP mode, all blocks will be turned off and no clocks are generated.

2.6 Dual System Clock Modes Transition

TM57PA15 is operated in one of the three modes: FAST Mode, SLOW Mode, and STOP Mode.

Modes Transition Diagram:



Note:

- SLEEP denotes SLEEP instruction
- Wakeup denotes wake-up events, such as External pin, interrupts or pin wake-up
- CPUCKS (F13.2), FASTSTP (F13.3)

CPU Mode & Clock Functions Table:

Mode	Oscillator	Fsys	Fast-clock	Slow-clock	TM0	PWM0/1	Wakeup event
FAST	FIRC, FXT	Fast-clock	Run	Run	Run	Run	X
SLOW	SIRC, SXT	Slow-clock	Stop	Run	Run	Run	X
STOP	Stop	Stop	Stop	Stop	Stop	Stop	IO

FAST Mode transits to SLOW Mode:

The source clock of Slow-clock can be chosen by SLOWCKS (F13.7). If SLOWCKS is set, the source clock of Slow-clock is Slow Crystal (SXT), otherwise is Slow Internal RC (SIRC). The following steps are suggested to be executed by order when FAST mode transits to SLOW mode:

(1) Select Slow-clock type (SXT: SLOWCKS=1, SIRC: SLOWCKS=0)

(2) Switch system clock source to Slow-clock (CPUCKS = 0)

(3) Stop Fast-clock (FASTSTP = 1)

◇Example: Switch operating mode from FAST mode to SLOW mode with SXT

```
BSF     SLOWCKS     ; Select SXT as Slow-clock source
BCF     CPUCKS      ; Switch system clock source to Slow-clock
BSF     FASTSTP     ; Stop Fast-clock
```

SLOW Mode transits to FAST Mode:

The source clock of Fast-clock can be chosen by FASTCKS (F13.6). If FASTCKS is set, the source clock of Fast-clock is Fast Crystal (FXT), otherwise is Fast Internal RC (FIRC). The following steps are suggested to be executed by order when SLOW mode transits to FAST mode:

(1) Select Fast-clock type (FXT: FASTCKS=1, FIRC: FASTCKS=0)

(2) Enable Fast-clock (FASTSTP = 0)

(3) Switch system clock source to Fast-clock (CPUCKS = 1)

◇Example: Switch operating mode from SLOW mode to FAST mode with FXT

```
BSF     FASTCKS     ; Select FXT as Fast-clock source
BCF     FASTSTP     ; Enable Fast-clock
BSF     CPUCKS      ; Switch system clock source to Fast-clock
```

F13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCTL	SLOWCKS	FASTCKS	PWMCKS	-	FASTSTP	CPUCKS	CPUPSC	
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	
Reset	0	0	0	-	0	0	1	1

F13.7 **SLOWCKS**: Slow-clock type select
0: SIRC 140KHz
1: SXT

F13.6 **FASTCKS**: Fast-clock type select
0: FIRC 8MHz
1: FXT

F13.5 **PWMCKS**: PWM clock source select
0: CPUCLK
1: IRC 16MHz

F13.3 **FASTSTP**: Fast-clock Enable / Disable
0: enable
1: disable

F13.2 **CPUCKS**: System clock source select
0: Slow-clock
1: Fast-clock

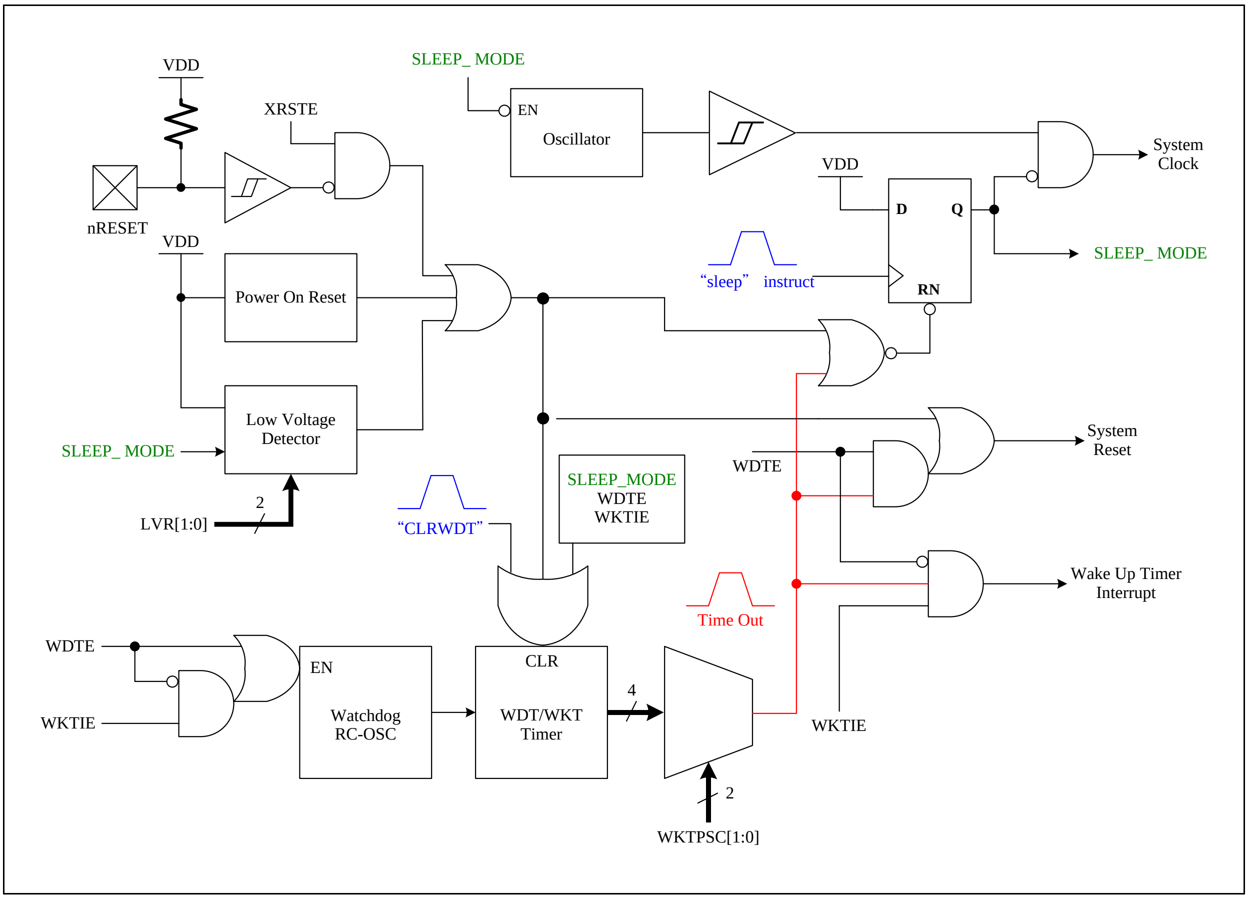
F13.1~0 **CPUPSC**: System clock source prescaler. System clock source
00: divided by 16
01: divided by 4
10: divided by 2
11: divided by 1

Warning: The CLKCTL (F13) can't be set directly for CPU modes transition. It may cause the transition fails. Please refer the mentioned steps for transition in this chapter.

3. Peripheral Functional Block

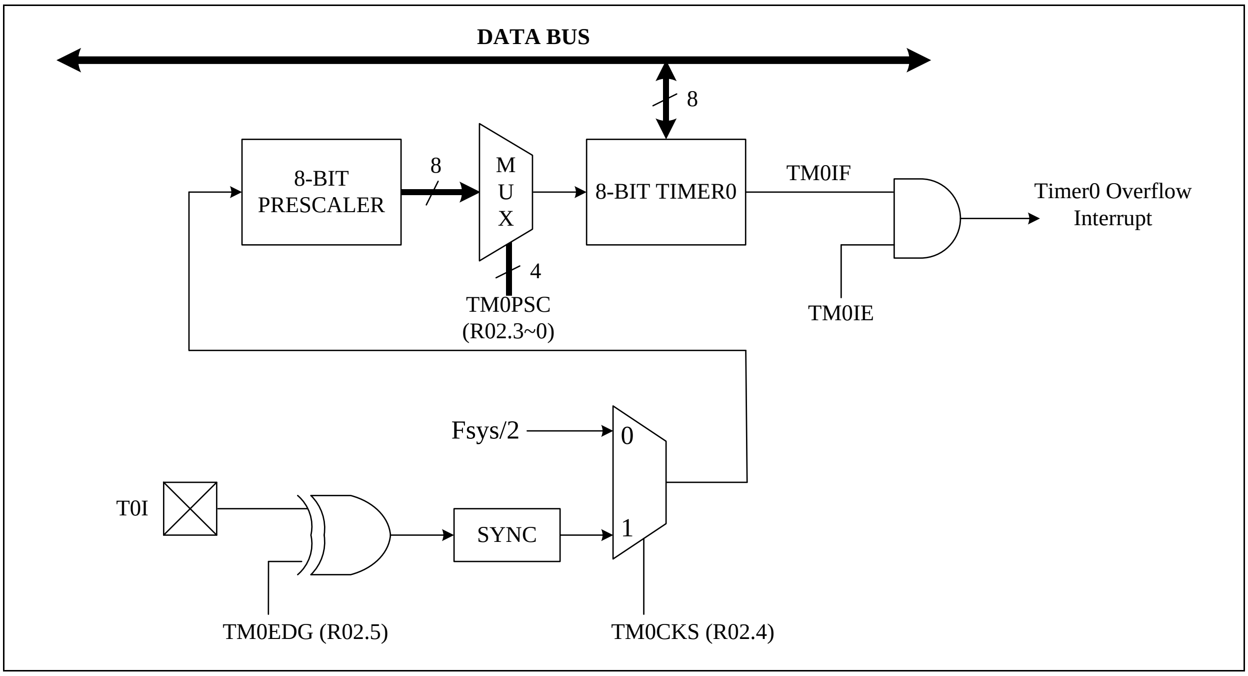
3.1 Watchdog (WDT) / Wakeup (WKT) Timer

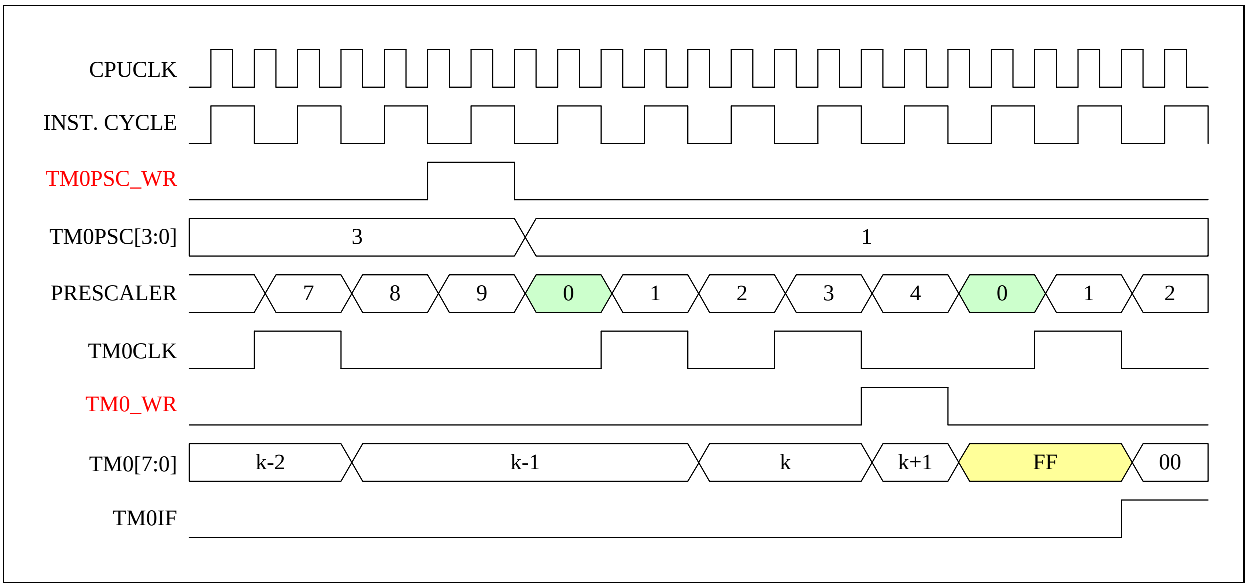
The WDT and WKT share the same internal RC Timer. The overflow period of WDT/WKT can be selected from 20 ms to 160 ms. The WDT/WKT is cleared by the CLRWDT instruction. If the Watchdog Reset is enabled (WDTE=1), the WDT generates the chip reset signal, otherwise, the WKT only generates overflow time out interrupt. The WDT/WKT works in both normal mode and stop mode. During stop mode, user can further choose to enable or disable the WDT/WKT by "WKTIE". If WKTIE=0 in stop mode (no matter WDTE is 1 or 0), the internal RC timer stops for power saving. In other words, user keeps the WDT/WKT alive in stop mode by setting WKTIE=1.



3.2 8-bit Timer/Counter (Timer0) with Pre-scale (PSC)

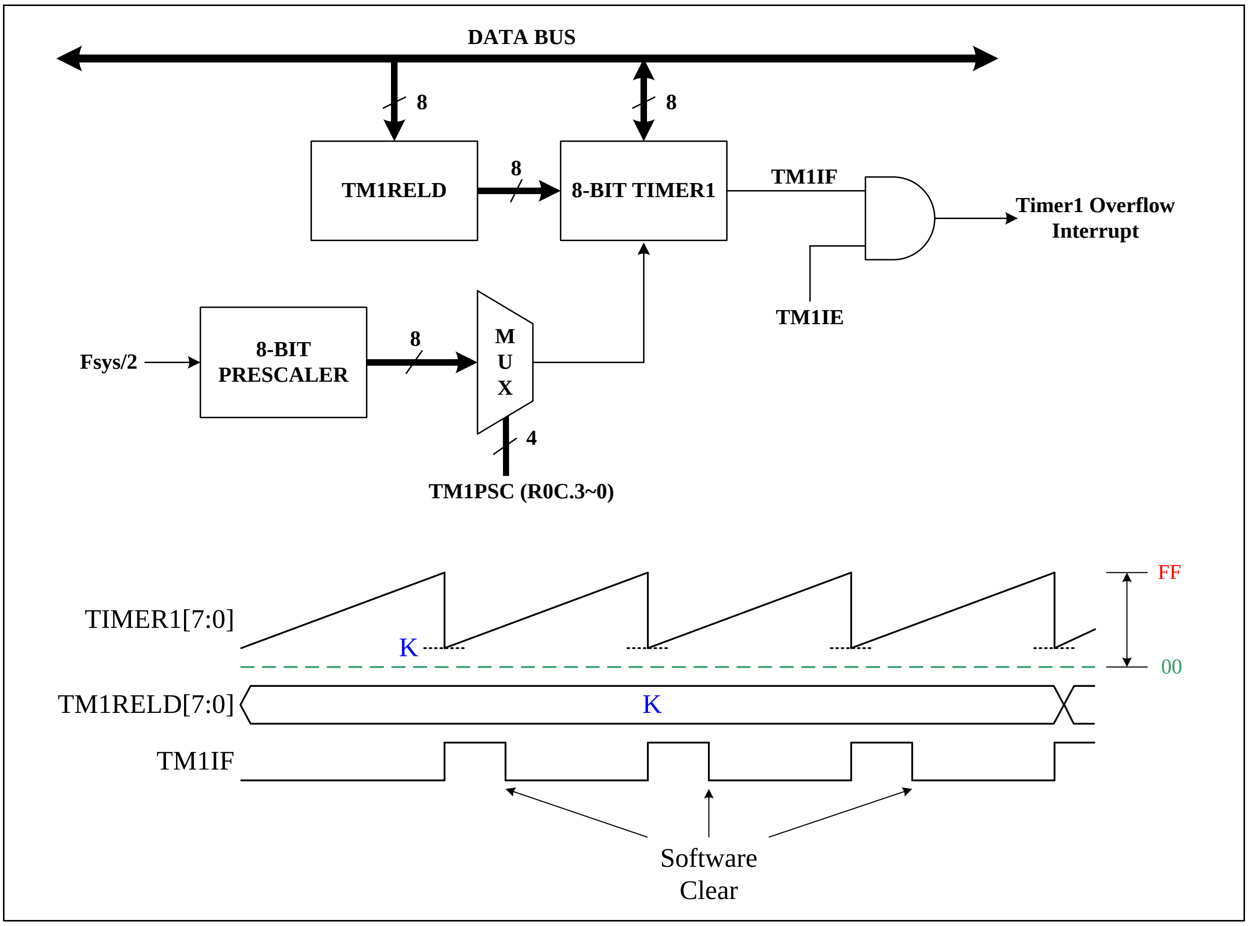
The Timer0 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatically rolls over based on the pre-scaled clock source, which can be the instruction cycle or T0I input. The Timer0 increase rate is determined by "Timer0 Pre-Scale" (TM0PSC) register in R-Plane. The Timer0 can generate interrupt (TM0IF) when it rolls over.

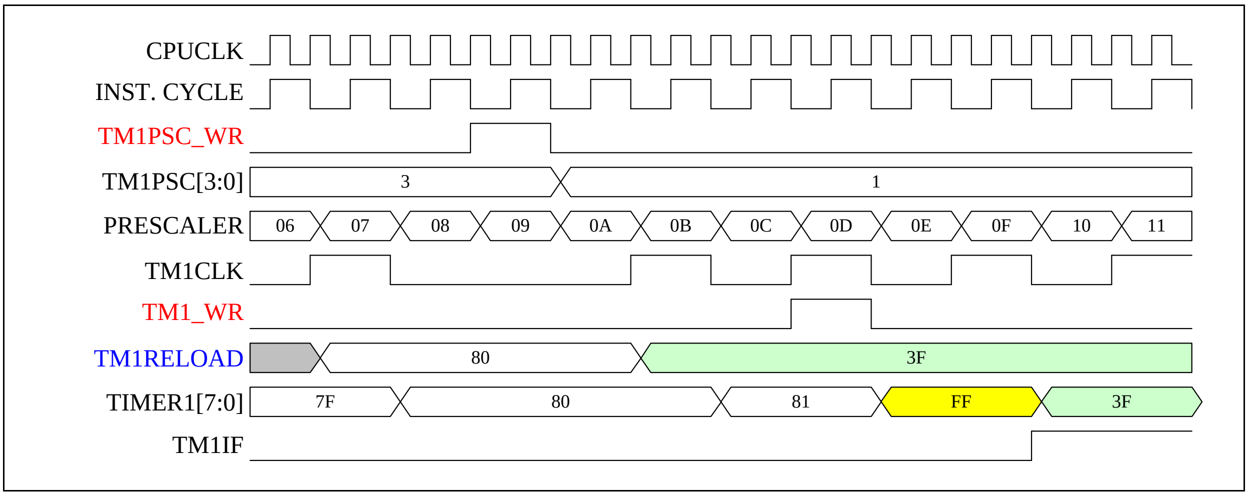




3.3 Timer1: 8-bit Timer with Pre-scale (PSC)

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 (TM1IF) when it rolls over.

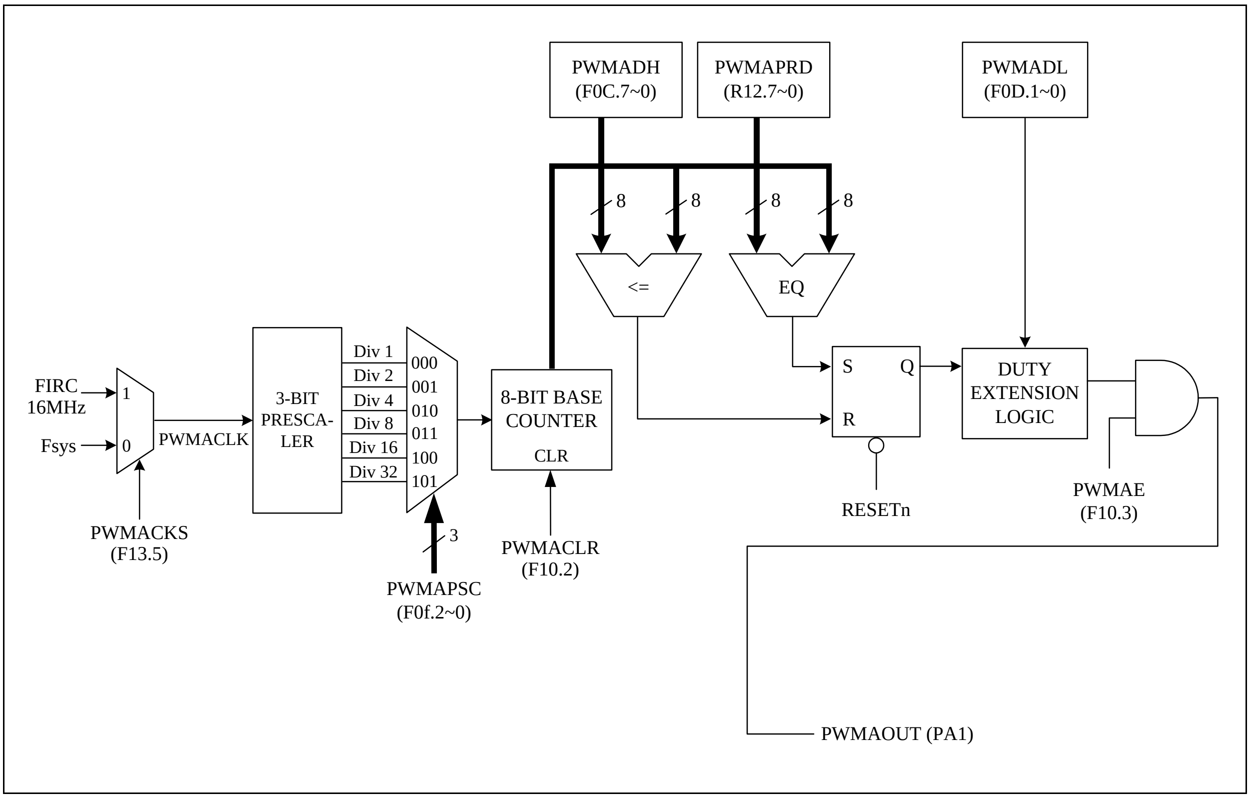




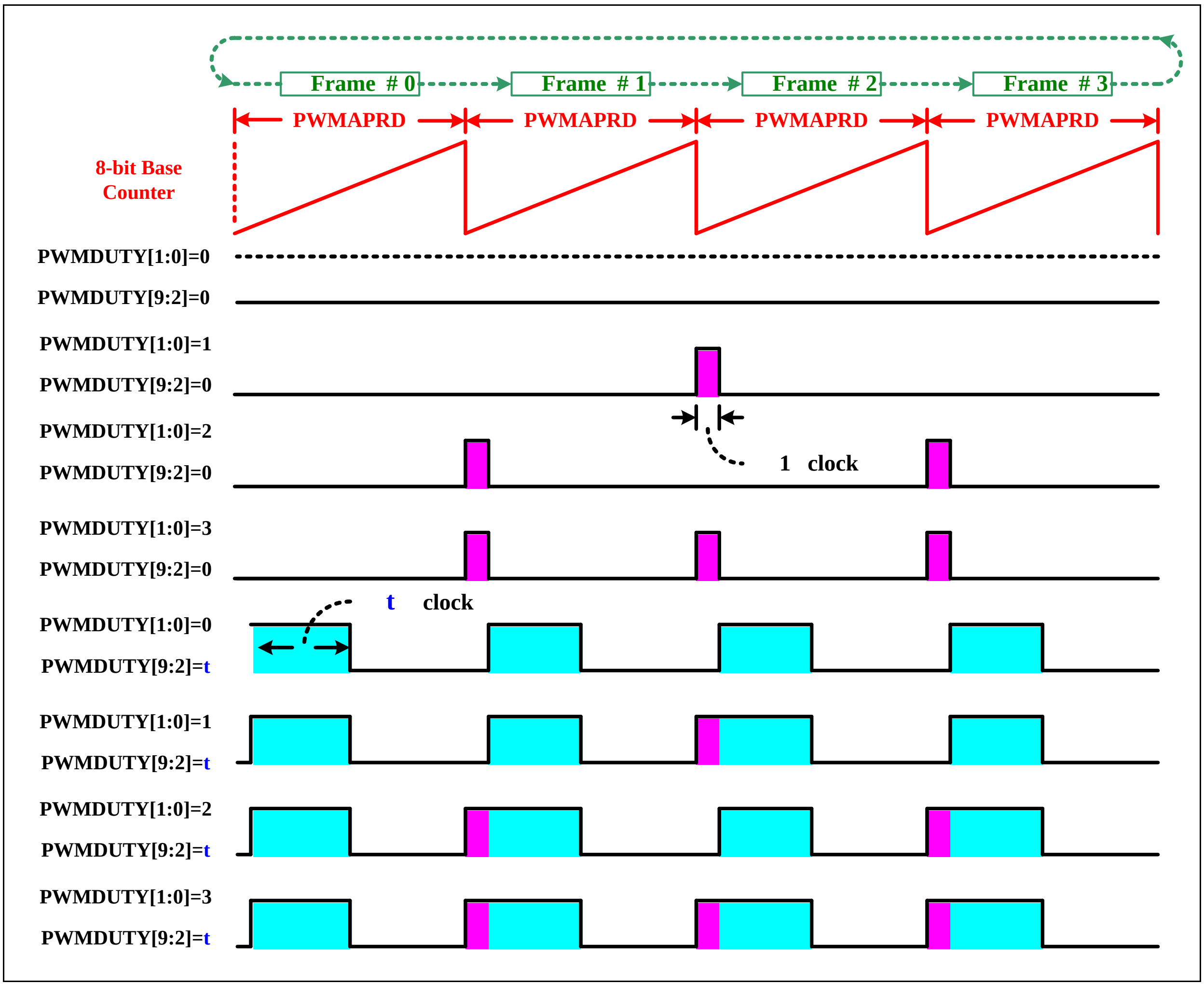
3.4 PWMA: (8+2) bits PWM

The PWMA can generate fix frequency waveform with 1024 duty resolution based on PWMCLK, which can select Fsys or FIRC 16 MHz, decided by PWMCKS (F13.5). A spread LSB technique allows PWMA to run its frequency at “PWMCLK divided by 256” instead of “PWMCLK 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 PWMADH (F11.7~0). When the base counter rolls over, the 2-bit LSB of PWM duty register PWMADL (F10.1~0) decides whether to set the PWMA output signal high immediately or set it high after one clock cycle delay.

The PWMA period can be set by writing period value to PWMAPRD register (R12). Note that changing the PWMAPRD will immediately change the PWMAPRD values, which are different from PWMADH /PWMADL which has buffer to update the duty at the end of current period. The Programmer must pay attention to the current time to change PWMAPRD by observing the following figure. There is a digital comparator that compares the PWMA counter and PWMAPRD, if PWMA counter is larger than PWMAPRD after setting the PWMAPRD, a fault long PWM cycle will be generated because PWMA counter must count to overflow then keep counting to PWMAPRD to finish the cycle.



PWMA Block Diagram



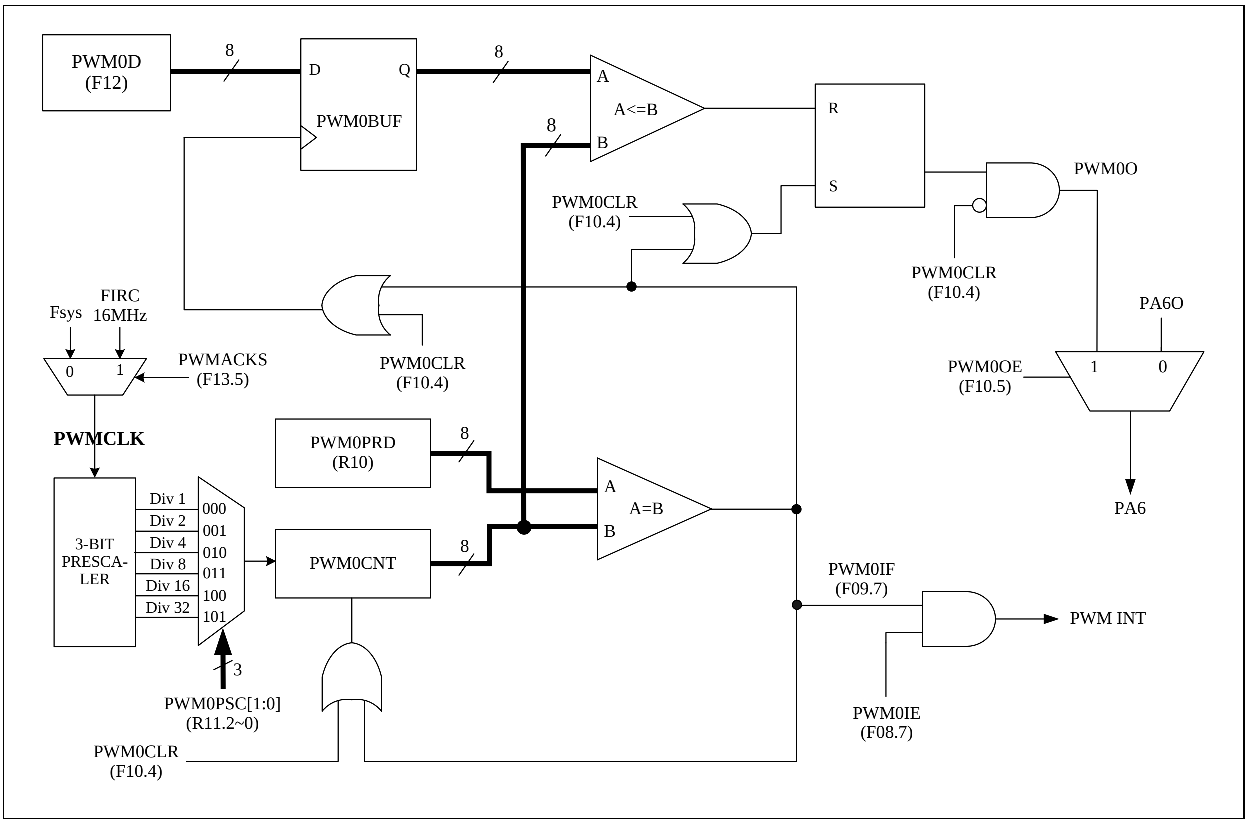
PWMA 8+2 Timing Diagram

3.5 PWM0: 8 bit PWM

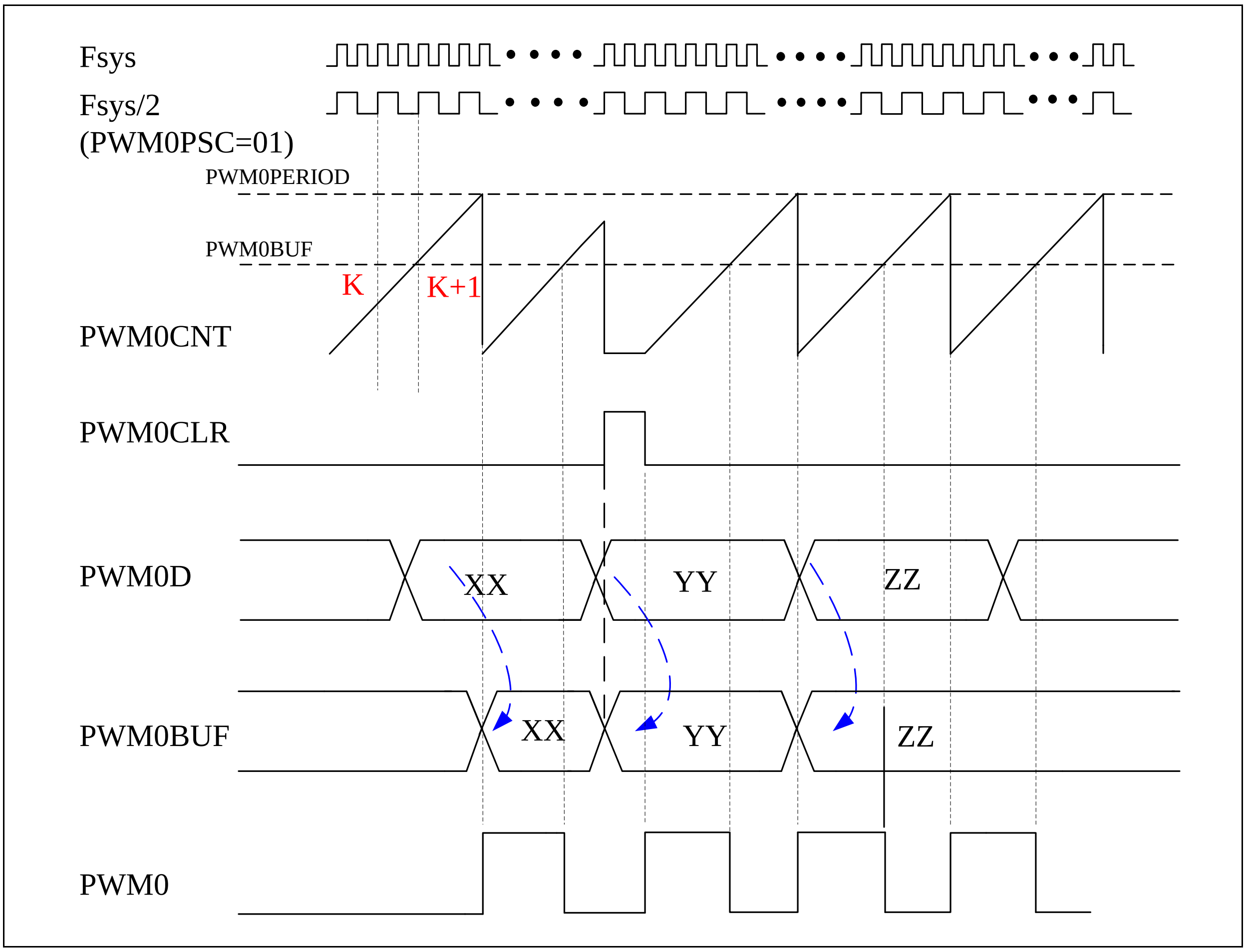
The chip has a built-in 8-bit PWM generator. The source clock comes from PWMCLK divided by 1, 2, 4, and 8. The PWM0 duty cycle can be changed with writing to PWM0D, writing to PWM0D will not change the current PWM duty until the current PWM period complete. When finish current PWM period, the new value of PWM0D will update to the PWM0BUF.

The PWM0 will be output to PB0 if PWM0OE is set to 1 and PWM0NOE = 0. The complement of PWM0, PWM0N, will be output to PB0 if PWM0OE is set to 1 and PWN0NOE = 1. Also, the PWM period complete will generate an interrupt when PWM0IE is set to 1. Setting the PWM0CLR bit will clear the PWM0 counter and load the PWM0D to PWM0BUF, PWM0CLR bit must be cleared so that the PWM0 counter can count.

Note that the default value of PWM0CLR bit is ‘1’.

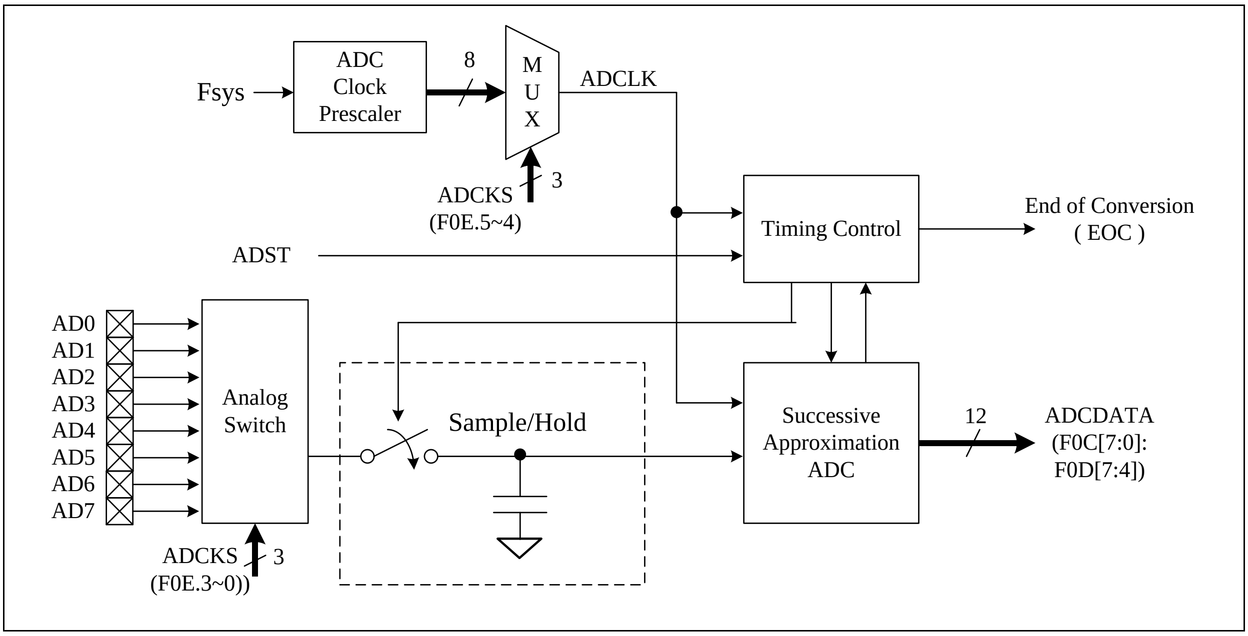


The next two Figures show the PWM0 waveforms. When PWM0CLR bit is set to ‘1’, the PWM0 output is cleared to ‘0’ no matter what its current status is. Once the PWM0CLR bit is cleared to ‘0’, the PWM0 output is set to ‘1’ to begin a new PWM cycle. PWM0 output will be ‘0’ when PWM0CNT is greater than or equal to PWM0BUF. PWM0CNT keeps counting up when equals to PWM0PRD, the PWM0 output is set to ‘1’ again.

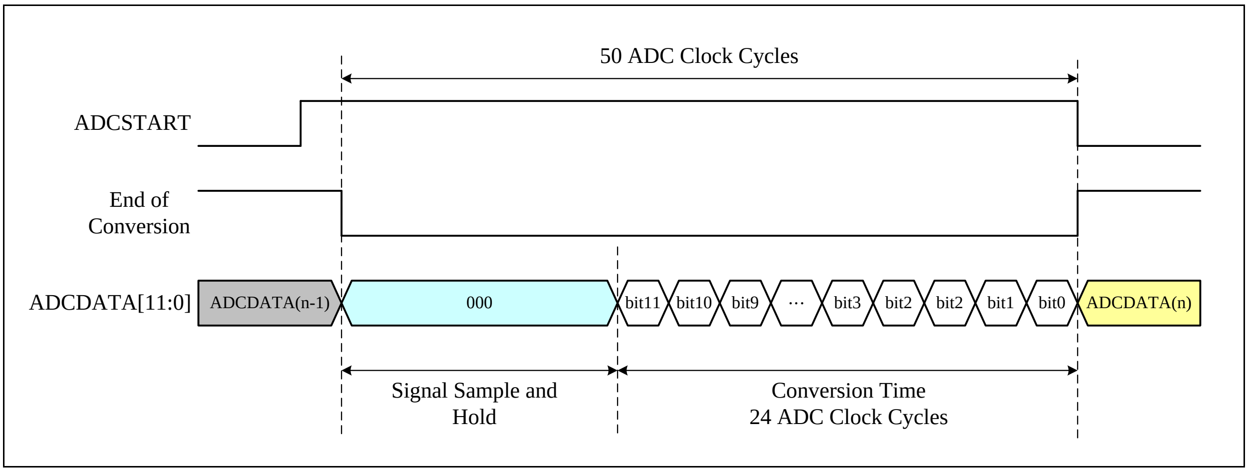


PWM0 Timing (PWM0CLR after PWM0CNT over PWM0BUF)

3.6 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 the end of conversion, H/W automatically clears the ADCSTAT bit. User can poll this bit to know the conversion status. The ADPIN 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.

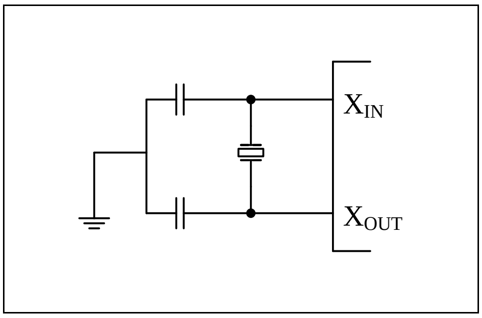


ADC example code:

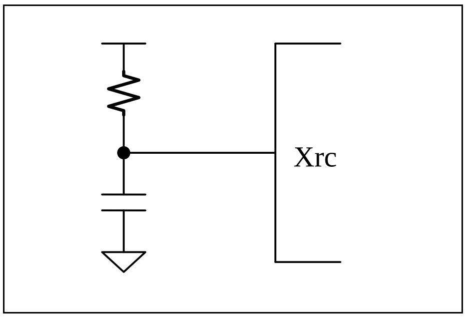
```
        MOVLW    00100010B
        MOVWF    0EH          ;ADC channel select ADC2 (PA5)
                              ;Set ADC clock is PWMCLK / 64  (ADCKS)
        MOVLW    00100000B
        MOVWR    08H          ;Disable PA pull up resistor (PAPUN)
        MOVLW    00000100B
        MOVWR    17H          ;Set ADC2(PA5) input enable (ADPINE)
        BSF      0EH,7        ;Start ADC conversion (ADST)
ADC_LOOP:
        BTFSC    0EH,7
        GOTO     ADC_LOOP     ;Wait ADST go LOW
        :
        :                     ;read ADCDATA[11:0] (ADCDATA)
```

3.7 System Clock Oscillator

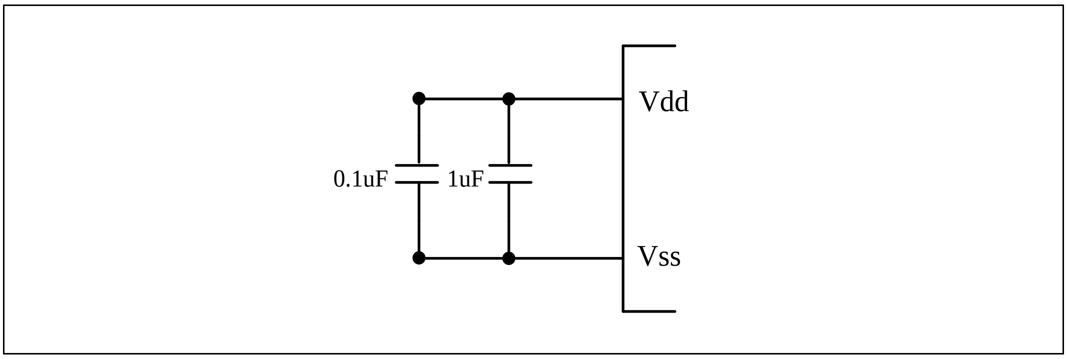
System Clock can be operated in four different oscillation modes, which is selected by setting the CLKS in the SYSCFG register. In Slow/Fast Crystal mode, a crystal or ceramic resonator is connected to the Xin and Xout pins to establish oscillation. In external RC mode, the external resistor and capacitor determine the oscillation frequency. In the internal RC mode, the on chip oscillator generates 4 MHz or 12 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 V_{DD} /V_{SS} pins improves the stability of clock and the overall system.



External Oscillator Circuit (Crystal or Ceramic)



External RC Oscillator

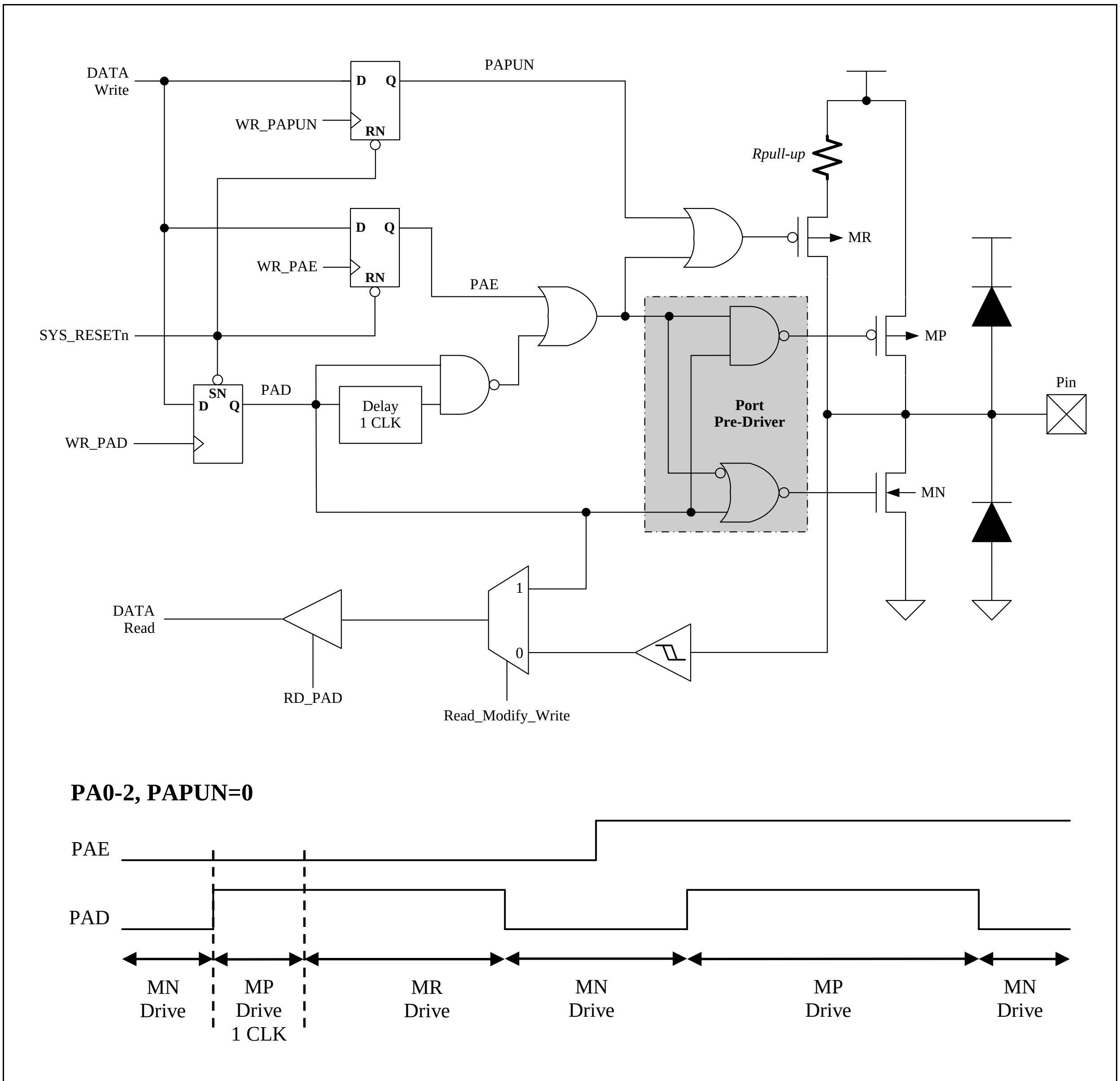


Internal RC Mode

4. I/O Port

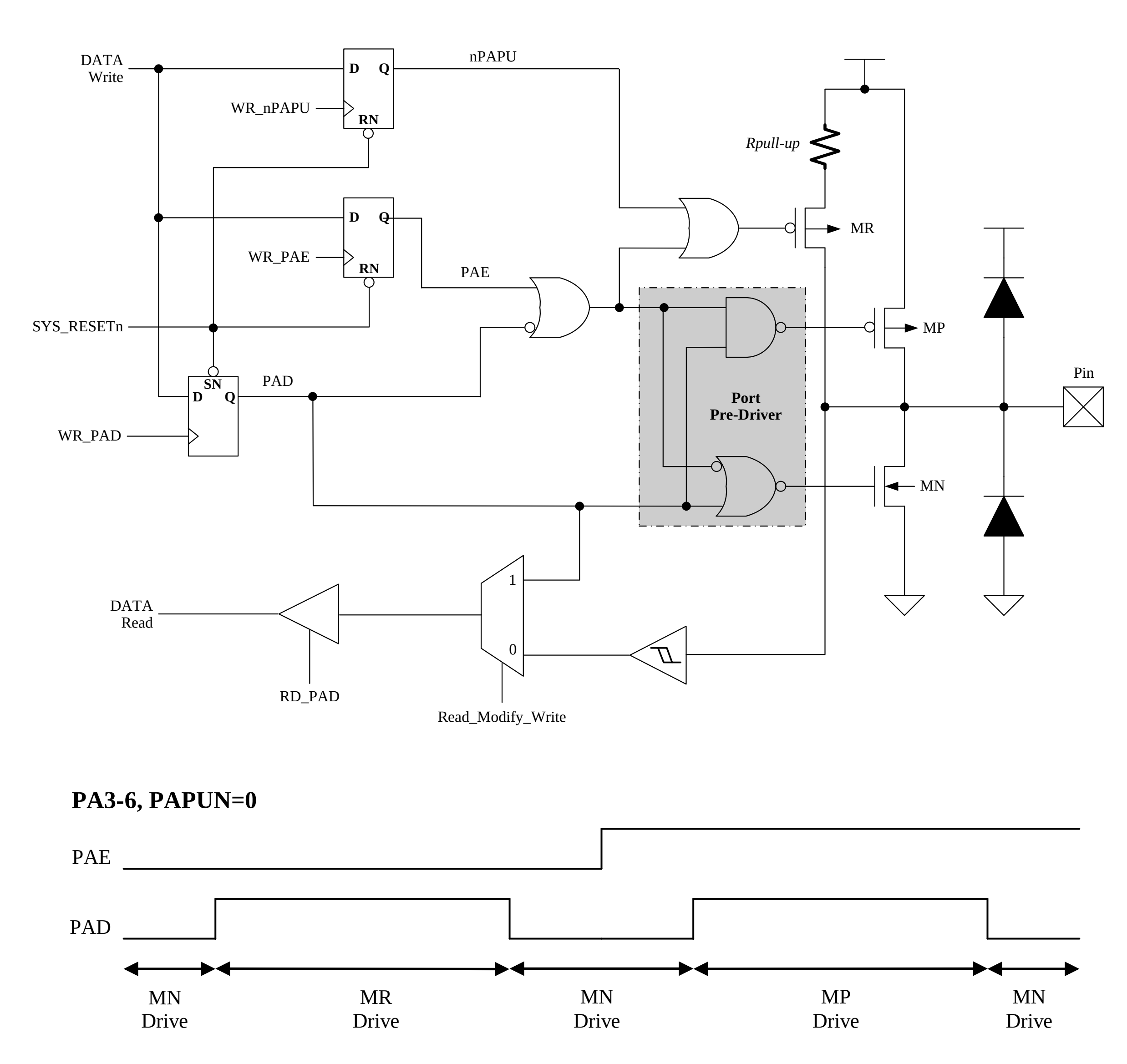
4.1 PA0-2

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.



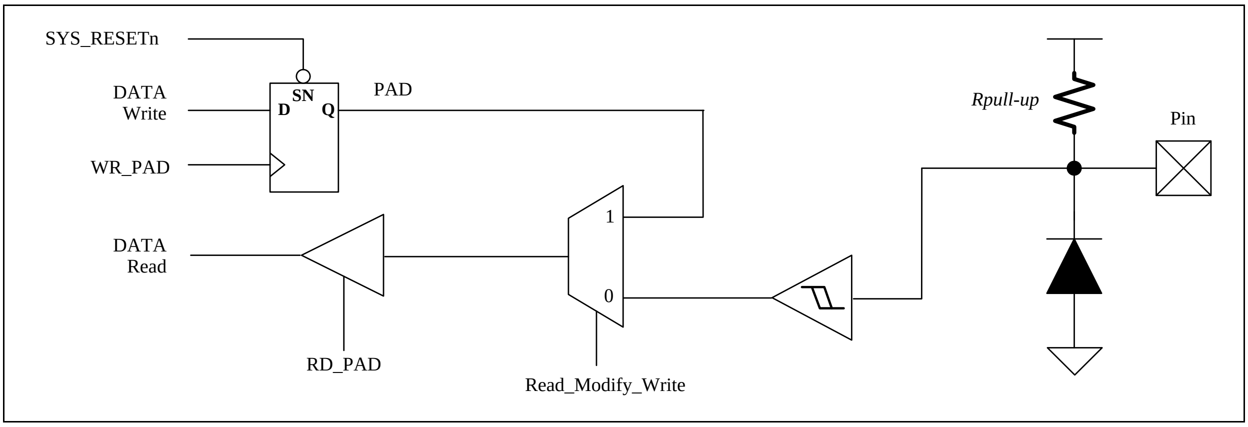
4.2 PA3-6 & PB0-5

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
(F00) INDF				
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
(F01) TM0				
TM0	01.7~0	R/W	0	Timer0 content
(F02) PCL				
PCL	02.7~0	R/W	0	Programming Counter [7~0]
(F03) STATUS				
LVDF	03.7	R	0	LVR Flag
GB1	03.6	R/W		General purpose bit
TO	03.4	R	0	WDT time out flag
PD	03.3	R	0	Sleep mode flag
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
(F04) FSR				
FSR	04.6~0	R/W	-	File Select Register, indirect address mode pointer
(F05) PAD				
PAD7	05.7	R	-	PA7 pin state
PAD	05.6~0	R	-	Port A pin or “data register” state
		W	7F	Port A output data register
(F06) PBD				
PBD	06.5~0	R	-	Port B pin or “data register” state
		W	3F	Port B output data register
(F08) INTIE				
PWM0IE	08.7	R/W	0	PWM0 interrupt enable, 1=enable, 0=disable
ADCIE	08.6	R/W	0	ADC interrupt enable, 1=enable, 0=disable
TM1IE	08.5	R/W	0	Timer1 interrupt enable, 1=enable, 0=disable
TM0IE	08.4	R/W	0	Timer0 interrupt enable, 1=enable, 0=disable
WKTIE	08.3	R/W	0	Wakeup Timer interrupt enable, 1=enable, 0=disable If WKTIE=0, the WDT/WKT stops in Sleep mode
INT2IE	08.2	R/W	0	INT2 (PA7) pin interrupt enable, 1=enable, 0=disable
INT1IE	08.1	R/W	0	INT1 (PB0) pin interrupt enable, 1=enable, 0=disable
INT0IE	08.0	R/W	0	INT0 (PA0) pin interrupt enable, 1=enable, 0=disable
(F09) INTIF				
PWM0IF	09.7	R	-	PWM0 interrupt flag, set by H/W while PWM0 end of period
		W	0	write 0: clear this flag; write 1: no action
ADCIF	09.6	R	-	ADC interrupt flag, set by H/W while ADC end of conversion
		W	0	write 0: clear this flag; write 1: no action

Name	Address	R/W	Rst	Description
TM1IF	09.5	R	-	Timer1 interrupt event pending flag, set by H/W while Timer1 overflow
		W	0	write 0: clear this flag; write 1: no action
TM0IF	09.4	R	-	Timer0 interrupt event pending flag, set by H/W while Timer0 overflow
		W	0	write 0: clear this flag; write 1: no action
WKTIF	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
INT2IF	09.2	R	-	INT2 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
INT1IF	09.1	R	-	INT1 interrupt event pending flag, set by H/W at INT1 pin's falling edge
		W	0	write 0: clear this flag; write 1: no action
INT0IF	09.0	R	-	INT0 interrupt event pending flag, set by H/W at INT0 pin's f/r edge
		W	0	write 0: clear this flag; write 1: no action
(F0A) TM1				
TM1	0a.7~0	R/W	0	Timer1 content
(F0C) ADH				
ADH	0c.7~0	R		ADC conversion data 8-bit MSB
(F0D) ADH				
ADL	0d.7~4	R		ADC conversion data 4-bit LSB
(F0E) ADCTL				
ADST	0e.7	R	-	H/W clears this bit after ADC end of conversion
		W	0	S/W sets this bit to start ADC conversion
ADCKS	0e.5~4	R/W	0	ADC clock frequency selection: 00: PWMCLK / 16 01: PWMCLK / 32 10: PWMCLK / 64 11: PWMCLK / 128
ADCHS	0e.3~0	R/W	0	ADC channel select; 0:AD0, 1:AD1,…,7:AD7
(F0F) PWMAPSC				
PWMAPSC	0f.2~0	R/W	0	PWMA prescaler , PWMCLK divided by 000:1,001:2,010:4,011:8,100:16,101:32,110:64,111:128
(F10) MF10				
TM0STP	10.6	R/W	0	Stop TM0 timer counting
PWM0OE	10.5	R/W	0	PWM0 positive output to PA6 pin
PWM0CLR	10.4	R/W	1	PWM0 clear and hold
PWMAOE	10.3	R/W	0	PWMA positive output to pin
PWMACLR	10.2	R/W	1	PWMA clear and hold
PWMADL	10.1~0	R/W	0	PWMA duty 2-bit LSB
(F11) PWMADH				
PWMADH	11.7~0	R/W	0	PWMA duty 8-bit MSB
(F12) PWM0D				
PWM0D	12.7~0	R/W	0	PWM0 duty 8-bit
(F13) CLKCTL				
SLOWCKS	13.7	R/W	0	Slow Clock Type. 0=SIRC 140KHz , 1=SXT

Name	Address	R/W	Rst	Description
FASTCKS	13.6	R/W	0	Fast Clock Type. 0=FIRC 8MHz, 1=FXT
PWMCKS	13.5	R/W	0	1: IRC 16 MHz as PWMCLK 0: CPUCLK as PWMCLK
SLOWSTP	13.4	R/W	0	1:Stop Slow Clock in Sleep Mode
FASTSTP	13.3	R/W	0	1:Stop Fast Clock
CPUCKS	13.2	R/W	0	1: Select Fast Clock 0: slow clock
CPUPSC	13.1~0	R/W	11	CPUCLK Prescaler, 0:divide 16, 1:divide 4, 2:divide 2, 3:divide 1
(F17) DPH				
DPH	17.1~0	R/W	0	DPTR high byte
SRAM	20~5F	R/W	-	Internal RAM

R-Plane

Name	Address	R/W	Rst	Description
(R02) TM0CTL				
TM0EDG	02.5	W	0	0: T0I (PA2) rising edge to increase Timer0/PSC count 1: T0I (PA2) falling edge to increase Timer0/PSC count
TM0CKS	02.4	W	0	0: Timer0/PSC clock source is "Instruction Cycle" 1: Timer0/PSC clock source is T0I pin
TM0PSC	02.3~0	W	0	Timer0 Pre-Scale 0000: Timer0 input clock is "Instruction Cycle" divided by 1 0001: Timer0 input clock is "Instruction Cycle" divided by 2 ~ 0111: Timer0 input clock is "Instruction Cycle" divided by 128 1000: Timer0 input clock is "Instruction Cycle" divided by 256
(R03) PWRDN				
PWRDN	03	W		write this register to enter Power-Down Mode
(R04) WDTCRL				
WDTCLR	04	W		write this register to clear WDT/WKT
(R05) PAE				
PAE	05.6~3	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is **open-drain** output or Schmitt-trigger input 1: the pin is CMOS push-pull output
	05.2~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is **pseudo-open-drain** output or Schmitt-trigger input 1: the pin is CMOS push-pull output
(R06) PBE				
PBE	06.5~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is **open-drain** output or Schmitt-trigger input 1: the pin is CMOS push-pull output
(R08) PAPUN				
PAPUN	08.6~0	W	7F	Each bit controls its corresponding pin, if the bit is 0: the pin pull up resistor is enable, except a. the pin's output data register (PAD) is 0 b. the pin's CMOS push-pull mode is chosen (PAE=1) c. the pin is working for Crystal or external RC oscillation 1: the pin pull up resistor is disable
(R09) PBPUN				
PBPUN	09.5~0	W	3F	Each bit controls its corresponding pin, if the bit is 0: the pin pull up resistor is enable a. the pin's output data register (PBD) is 0 b. the pin's CMOS push-pull mode is chosen (PBE=1) 1: the pin pull up resistor is disable
(R0B) MR0B				
INT0EDG	0b.4	W	0	0: INT0 (PA0) pin falling edge to trigger interrupt event 1: INT0 (PA0) pin rising edge to trigger interrupt event
TCOE	0b.3	W	0	0: No Instruction Clock output to PA3 pin 1: Instruction Clock output to PA3 pin for external/internal RC mode
WKTPSC	0b.1~0	W	11	WDT/WKT typical period (V_{DD} =5V) 00: WDT/WKT period is 13 ms 01: WDT/WKT period is 25 ms 10: WDT/WKT period is 50 ms 11: WDT/WKT period is 100 ms

Name	Address	R/W	Rst	Description
(R0C) TM1PSC				
TM1PSC	0c.3~0	W	0	0000: Timer1 input clock is "Instruction Cycle" divided by 1 0001: Timer1 input clock is "Instruction Cycle" divided by 2 ~ 0111: Timer1 input clock is "Instruction Cycle" divided by 128 1000: Timer1 input clock is "Instruction Cycle" divided by 256
(R0D) TM1RLD				
TM1RLD	0d.7~0	W	0	Timer1 reloads offset value while it rolls over
(R0E) MR0E				
CKHLDE	0e.6	W	0	Clock Hold Enable
EMIIMPRV	0e.5	W	1	EMI IMPRV Enable
(R10) PWM0PRD				
PWM0PRD	10.7~0	W	ff	PWM0 period
(R11) PWM0PSC				
PWM0PSC	11.2~0	W	0	PWM0 prescaler, PWMCLK divided by 000: 1, 001: 2, 010: 4, 011: 8, 100: 16, 101: 32, 110: 64, 111: 128
(R12) PWMPRD				
PWMAPRD	12.7~0	W	ff	PWMA period
(R13) PAWK				
PAWKEN	13.6~1	W	0	Enable PA6~PA1 pin wake up
PAWKMODE	13.0	W	0	PA6~PA1 wake up mode 0: Low level 1: level change
(R17) ADPINE				
ADPINE	17.7~0	W	0	Each bit controls its corresponding ADC7~0 enable pin, if the bit is 0: the corresponding pin is I/O pin 1: the corresponding pin is ADC pin
(R18) PBWKEN				
PBWKEN	18.5~1	W	0	Enable PB5~PB1 pin low level wake up

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 used by the instruction. The destination designator specifies where the result of the operation is 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 00kk kkkk kkkk**	2	-	Call subroutine "k"
CLRWDT		**00 0000 0000 0100**	1	TO, PD	Clear WDT/WKT Timer
GOTO	k	**11 00kk 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 ~ 5Fh 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' (k)
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 ~ 5Fh 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 ‘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 BTFSC FLAG, 1 TRUE GOTO SUB1 FALSE ...	B : PC = LABEL1 A : if FLAG.1 = 0, PC = 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	1 or 2	
Example	LABEL1 BTFSS FLAG, 1 TRUE GOTO SUB1 FALSE ...	B : PC = LABEL1 A : if FLAG.1 = 0, PC = TRUE if FLAG.1 = 1, PC = FALSE

CALL — Call subroutine "k"

Syntax	CALL k	
Operands	K : 00h ~ 3FFh	
Operation	Operation: TOS ← (PC)+ 1, PC.9~0 ← k	
Status Affected	-	
OP-Code	10 00kk kkkk kkkk	
Description	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The 10-bit immediate address is loaded into PC bits <9:0>. 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 ~ 5Fh	
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 Timer.	
Cycle	1	
Example	CLRWDT	B : WDT counter = ? A : WDT counter = 0x00

COMF — Complement "f"

Syntax	COMF f [,d]
Operands	f : 00h ~ 5Fh, d : 0, 1
Operation	(destination) ← ($\overline{f}$)
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 ~ 5Fh, 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 ~ 5Fh, 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 GOTO LOOP CONTINUE B : PC = LABEL1 A : CNT = CNT – 1 if CNT=0, PC = CONTINUE if CNT≠0, PC = LABEL1+1

GOTO — Unconditional Branch

Syntax	GOTO k
Operands	k : 00h ~ 3FFh
Operation	PC.9~0 ← k
Status Affected	-
OP-Code	11 00kk kkkk kkkk
Description	GOTO is an unconditional branch. The 10-bit immediate value is loaded into PC bits <9:0>. 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 ~ 5Fh
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 ~ 5Fh, 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 GOTO LOOP CONTINUE B : PC = LABEL1 A : CNT = CNT + 1 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 is 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 ~ 5Fh, 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 ~ 5Fh
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 ~ 5Fh	
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

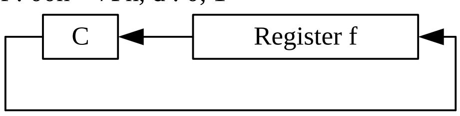
MOVWR	**Move W to "r"**	
Syntax	MOVWR r	
Operands	r : 00h ~ 12h	
Operation	(r) ← (W)	
Status Affected	-	
OP-Code	00 0000 00rr rrrr	
Description	Move data from W register to register 'r'.	
Cycle	1	
Example	MOVWR REG1	B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F

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

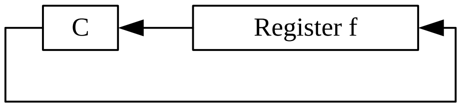
RETI	**Return from Interrupt**	
Syntax	RETI	
Operands	-	
Operation	PC ← TOS, GIE ← 1	
Status Affected	-	
OP-Code	00 0000 0110 0000	
Description	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in to the PC. Interrupts are enabled. This is a two-cycle instruction.	
Cycle	2	
Example	RETI	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 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	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	RET	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
		A : REG1 = 1110 0110
		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 A : REG1 = 1110 0110 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 stops.
Cycle	1
Example	SLEEP — -

SUBWF — Subtract W from "f"

Syntax	SUBWF f [,d]
Operands	f : 00h ~7Fh, d : 0, 1
Operation	(destination) ← (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 REG1, 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°C)

Parameter	Rating	Unit
Supply voltage	V_{SS} - 0.3 to V_{SS} + 6.5	V
Input voltage	V_{SS} - 0.3 to V_{DD} + 0.3	
Output voltage	V_{SS} - 0.3 to V_{DD} + 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	°C
Storage temperature	-65 to +150	

2. DC Characteristics (T_A = 25℃, V_{DD} = 5.0V, unless otherwise specified)

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Input High Voltage	V_{IH}	All Input, except PA7	V_{DD} = 5V	$0.44V_{DD}$			V
			V_{DD} = 3V	$0.5V_{DD}$			V
		PA7	V_{DD} = 5V	$0.6V_{DD}$			V
			V_{DD} = 3V	$0.63V_{DD}$			V
Input Low Voltage	V_{IL}	All Input, except PA7	V_{DD} = 5V			$0.26V_{DD}$	V
			V_{DD} = 3V			$0.3V_{DD}$	V
		PA7	V_{DD} = 5V			$0.36V_{DD}$	V
			V_{DD} = 3V			$0.33V_{DD}$	V
Output High Voltage (NOTE 1)	V_{OH}	All Output	V_{DD} = 5V, I_{OH}=7 mA	4.5			V
			V_{DD} = 3V, I_{OH}=4 mA	2.7			V
Output Low Voltage	V_{OL}	All Output	V_{DD} = 5V, I_{OL}=20 mA			0.5	V
			V_{DD} = 3V, I_{OL}=10 mA			0.3	V
Input Leakage Current (pin high)	I_{ILH}	All Input	V_{IN} = V_{DD}	–	–	1	uA
Input Leakage Current (pin low)	I_{ILL}	All Input	V_{IN} = 0 V	–	–	–1	uA
Output Leakage Current (pin high)	I_{OLH}	All Output	V_{OUT} = V_{DD}	–	–	2	uA
Output Leakage Current (pin low)	I_{OLL}	All Output	V_{OUT} = 0V	–	–	–2	uA
Power Supply Current	I_{DD}	Run 8 MHz, No Load	V_{DD} = 4.5 to 5.5V	–	3.4		mA
		Run 4 MHz, No Load	V_{DD} =3.0V		0.9		
		Stop mode, No Load	V_{DD} = 4.5 to 5.5V	–		1	uA
			V_{DD} = 3.0V			1	
System Clock Frequency	f_{OSC}	VDD > LVR_{th}	V_{DD} = 5V	–	–	24	MHz
			V_{DD} = 3V			12	
			V_{DD} = 2.2V			8	
LVR reference Voltage		V_{LVR}		1.85	2.0	2.2	V
				2.75	2.85	3.2	V
LVR Hysteresis Voltage		V_{HYST}		–	±0.1	–	V
Low Voltage Detection time		t_{LVR}		10	–	–	μs
Pull-Up Resistor	R_P	V_{IN} = 0V Ports A/B	V_{DD} = 5V V_{DD} = 3V		65 115		k
		V_{IN} = 0 V PA7	V_{DD} = 5V V_{DD} = 3V		120 60		k

NOTE : 1. while strong MP drives

3. Clock Timing (T_A = -40℃ to +85℃)

Parameter	Condition	Min	Typ	Max	Unit
Internal RC Frequency	V_{DD} = 4.75 to 5.25V (T_A = 25℃)	7.75	8	8.25	MHz
	V_{DD} = 2.8 to 3.2V (T_A = 25℃)	7.6	8	8.4	
	V_{DD} = 2.8 to 5.25V (T_A = -40℃~85℃)	7.5	8	8.5	

4. Reset Timing Characteristics (T_A = 25℃, V_{DD} = 2.0V to 5.5V)

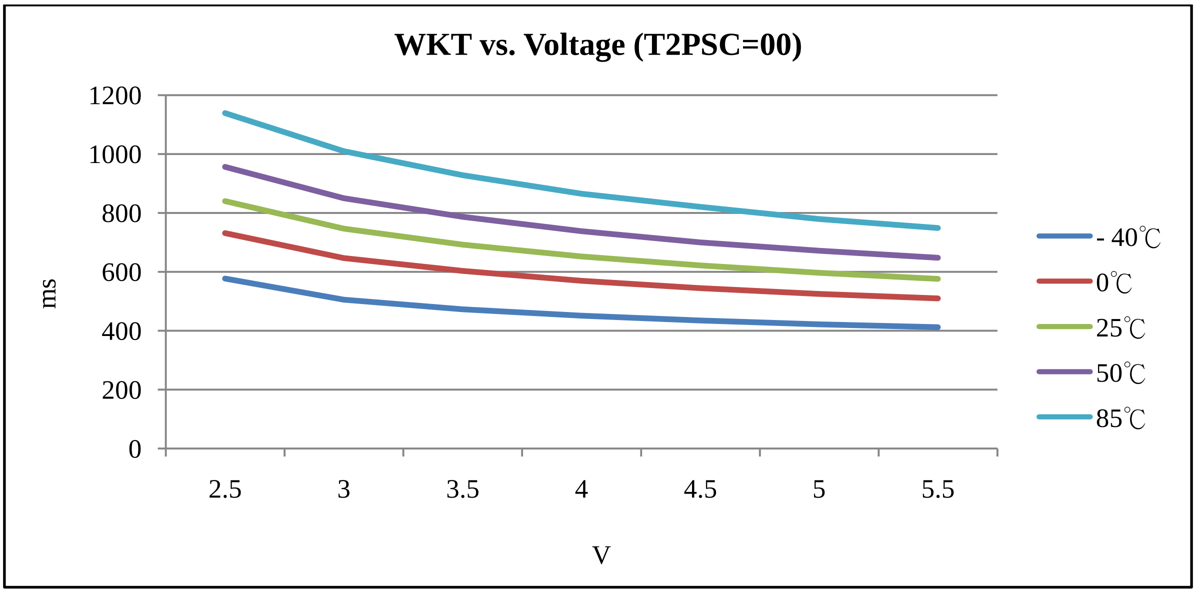
Parameter	Conditions	Min	Typ	Max	Unit
RESET Input Low width	Input V_{DD} = 5V ± 10 %	3	–	–	μs
WDT wakeup time	V_{DD} = 5V, WKTPSC = 00	–	19	–	ms
	V_{DD} = 3V, WKTPSC = 00	–	24	–	
CPU start up time	V_{DD} = 5V	–	19	–	ms
	V_{DD} = 5V		24	–	

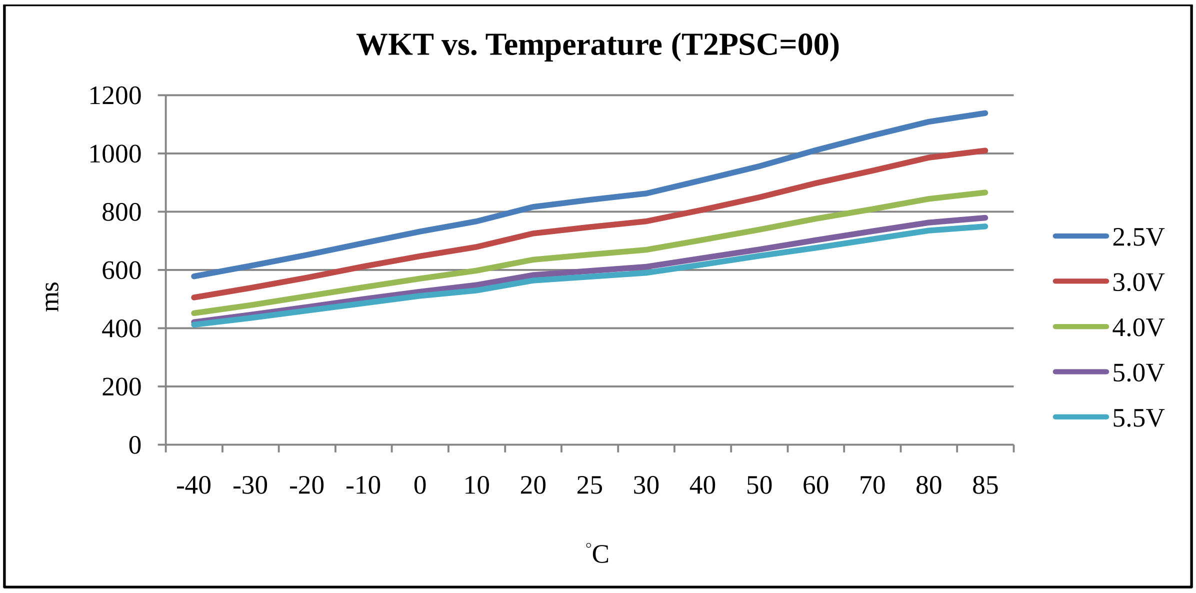
5. ADC Electrical Characteristics (T_A = 25℃, V_{DD} = 2.0V to 5.5V, V_{SS} = 0V)

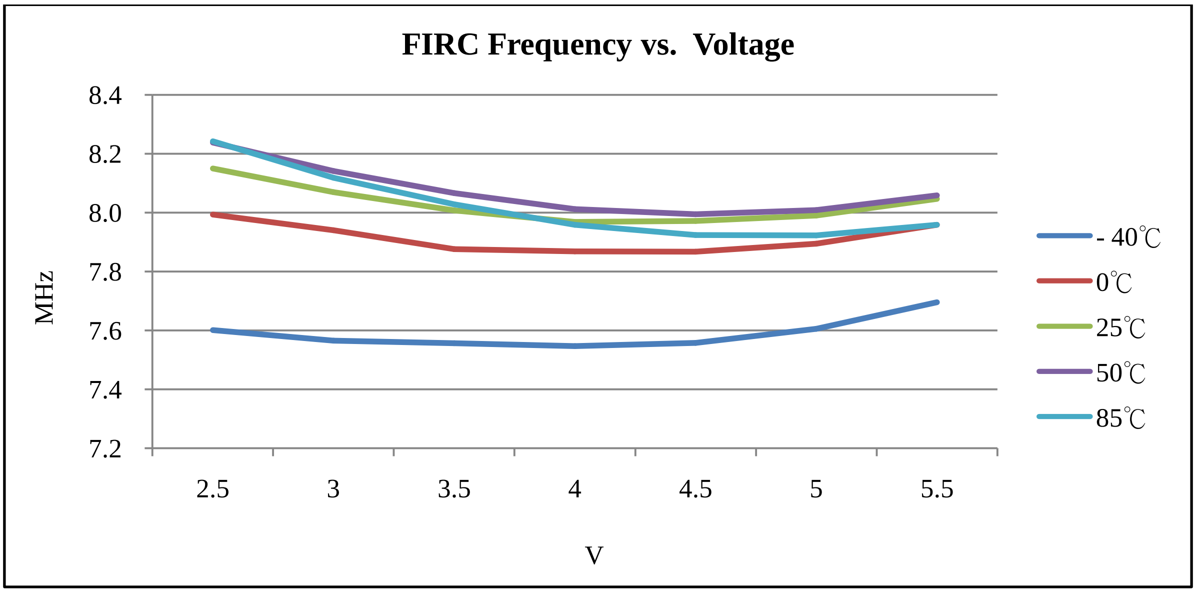
Parameter	Conditions	Min	Typ	Max	Units
Total Accuracy	V_{DD} = 5.12V, V_{SS} = 0 V	–	± 2.5	± 8	LSB
Integral Non-Linearity		–	± 3.2	± 5	
Max Input Clock (f_{ADC})	–	–	–	2	MHz
Conversion Time	f_{ADC} = 2 MHz	–	25	–	μs
Input Voltage	–	V_{SS}	–	V_{DD}	V

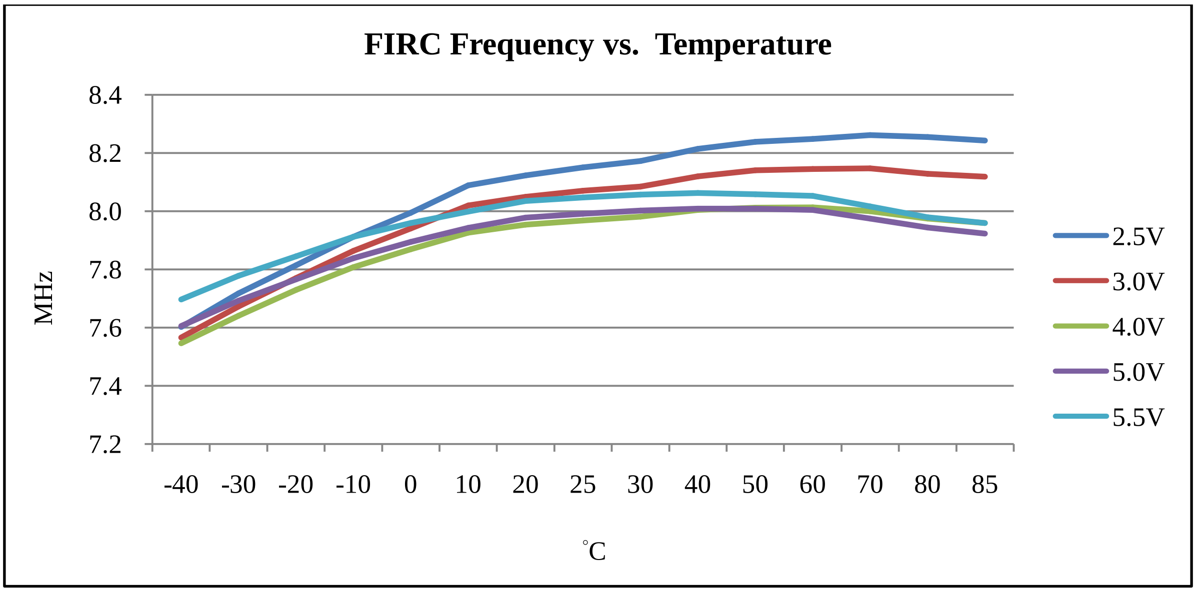
6.

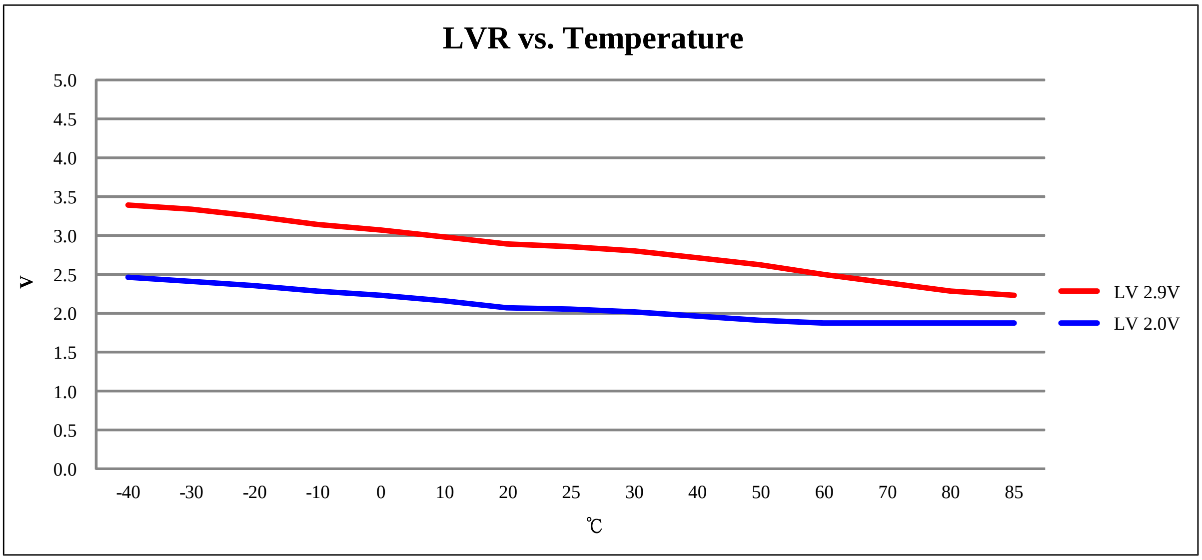
7. Characteristic Graphs






FIRC Frequency vs. Voltage
8.4
8.2
8.0
7.8
7.6
7.4
7.2
MHz
2.5
3
3.5
4
4.5
5
5.5
V
- 40℃
0℃
25℃
50℃
85℃


FIRC Frequency vs. Temperature
8.4
8.2
8.0
7.8
7.6
7.4
7.2
MHz
-40
-30
-20
-10
0
10
20
25
30
40
50
60
70
80
85
˚C
2.5V
3.0V
4.0V
5.0V
5.5V

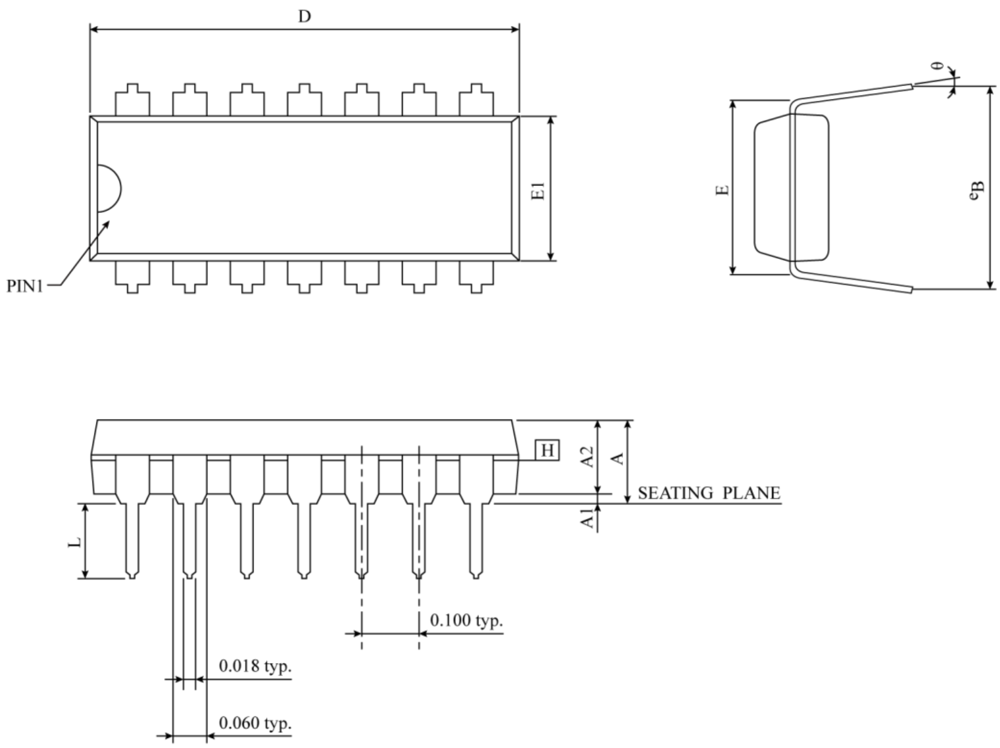

LVR vs. Temperature
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
V
-40
-30
-20
-10
0
10
20
25
30
40
50
60
70
80
85
℃
LV 2.9V
LV 2.0V

PACKAGING INFORMATION

The ordering information:

Ordering number	Package
TM57PA15-OTP	Wafer / Dice blank chip
TM57PA15-COD	Wafer / Dice with code
TM57PA15-OTP-02	DIP 14-pin (300 mil)
TM57PA15-OTP-15	SOP 14-pin (150 mil)
TM57PA15-OTP-03	DIP 16-pin (300 mil)
TM57PA15-OTP-16	SOP 16-pin (150 mil)

14-DIP Package Dimension

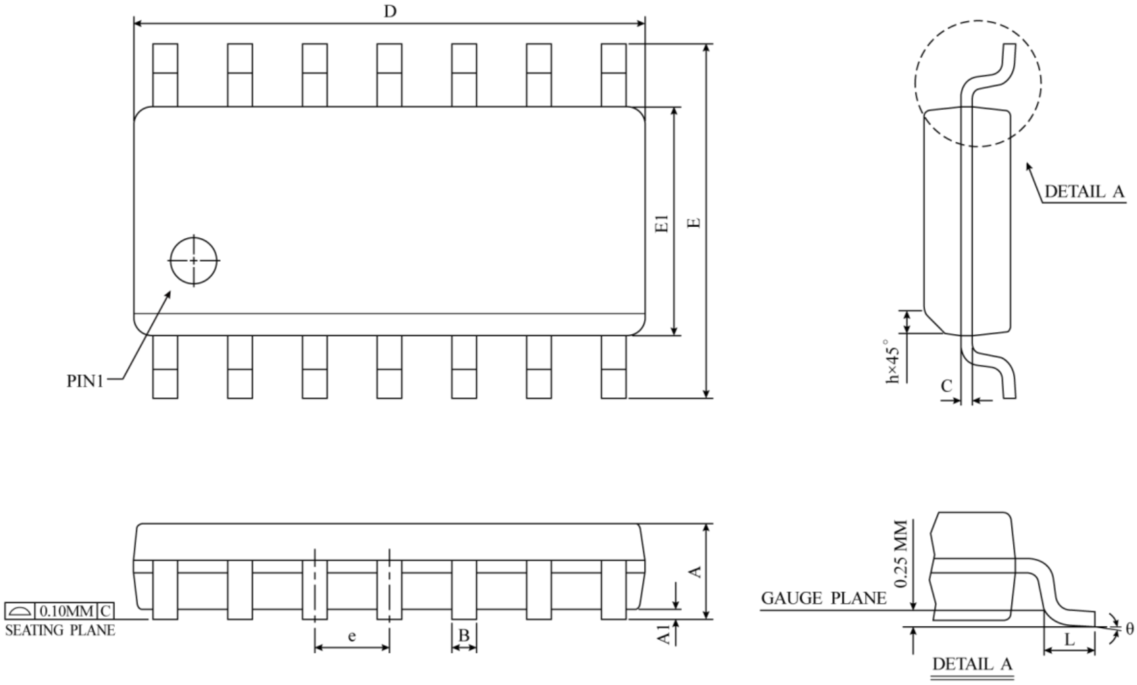


SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	5.334	-	-	0.210
A1	0.381	-	-	0.015	-	-
A2	3.175	3.302	3.429	0.125	0.130	0.135
D	18.669	19.177	19.685	0.735	0.755	0.775
E	7.620 BSC			0.300 BSC		
E1	6.223	6.350	6.477	0.245	0.250	0.255
L	2.921	3.366	3.810	0.115	0.133	0.150
eB	8.509	9.017	9.525	0.335	0.355	0.375
θ	0°	7.5°	15°	0°	7.5°	15°
JEDEC	MS-001 (AA)					

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.

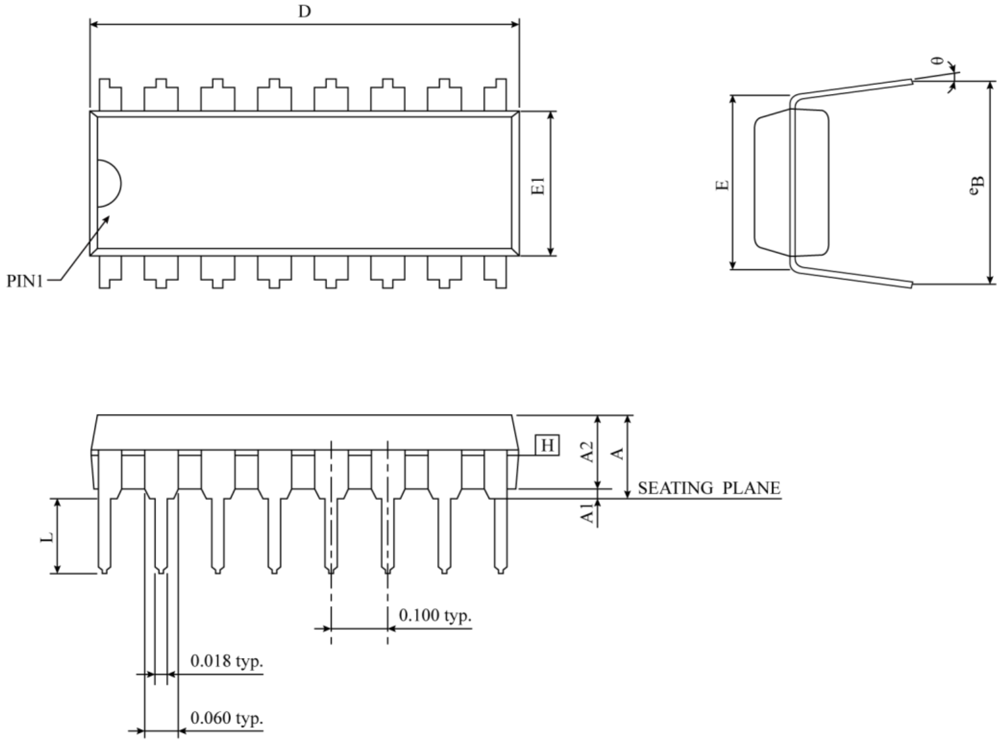
14-SOP Package Dimension



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.

16-DIP Package Dimension

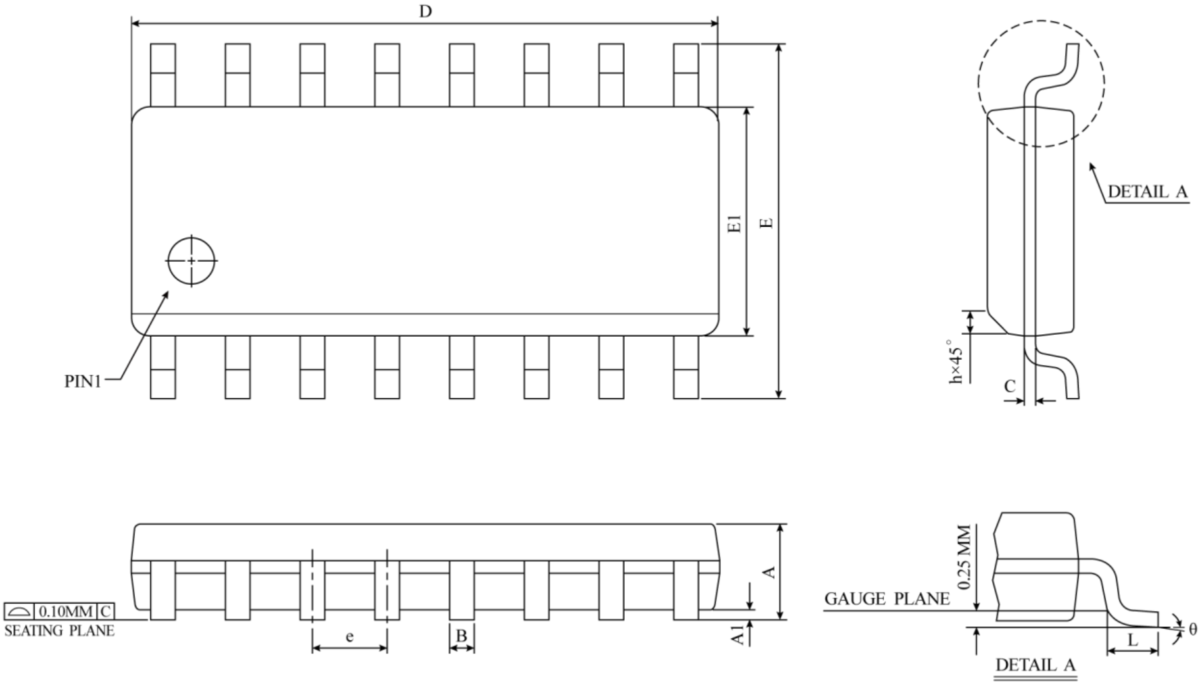


SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	4.369	-	-	0.172
A1	0.381	0.673	0.965	0.015	0.027	0.038
A2	3.175	3.302	3.429	0.125	0.130	0.135
D	18.669	19.177	19.685	0.735	0.755	0.775
E	7.620 BSC			0.300 BSC		
E1	6.223	6.350	6.477	0.245	0.250	0.255
L	2.921	3.366	3.810	0.115	0.133	0.150
eB	8.509	9.017	9.525	0.335	0.355	0.375
θ	0°	7.5°	15°	0°	7.5°	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 [H] COINCIDENT WITH THE BOTTOM OF LEAD, WHERE LEAD EXITS BODY.

16-SOP 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.