十速科技股份有限公司
tenx technology inc.

Advance Information

TM57PA40

8 Bit Microcontroller

User Manual

Tenx reserves the right to change or discontinue this product without notice.

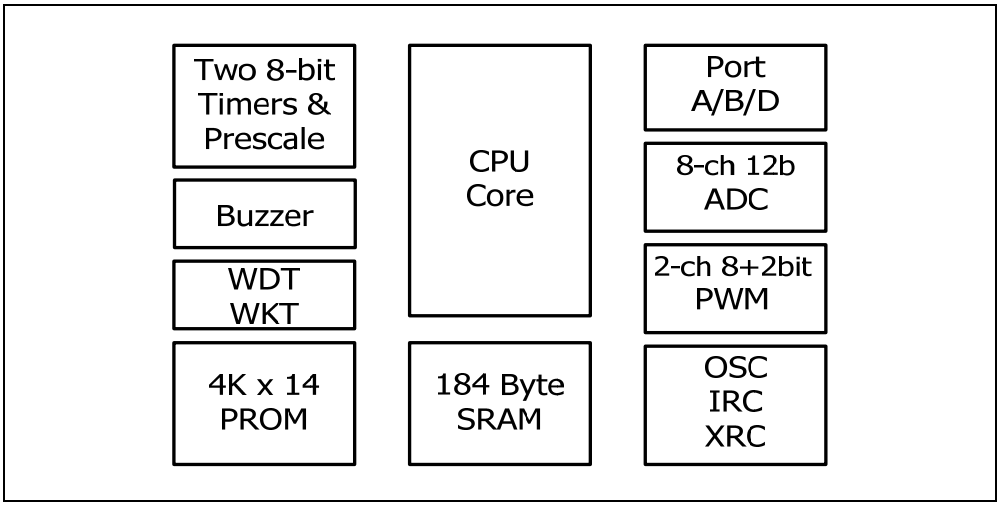
tenx technology inc.

INDEX

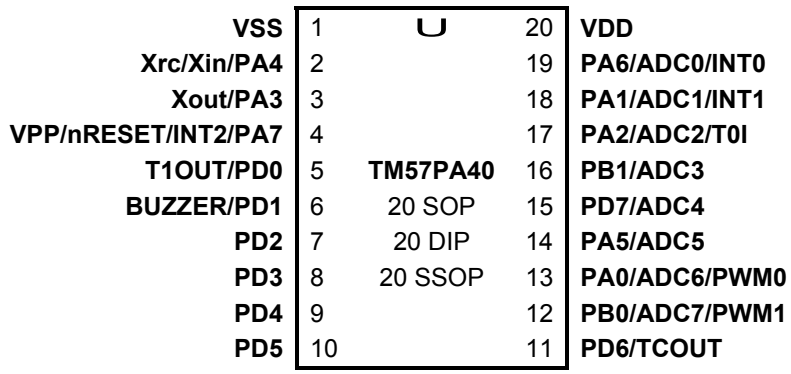
FEATURES

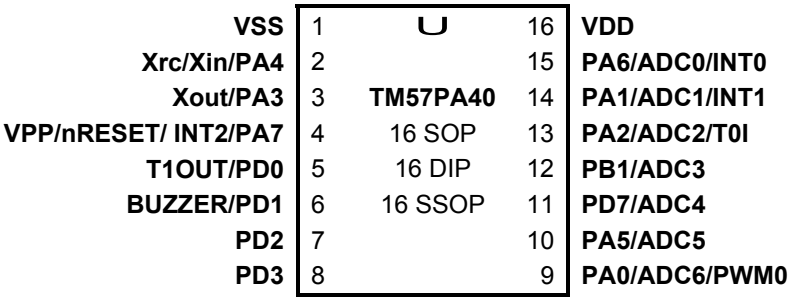
1. ROM: 4K x 14 bits OTP or 2K x 14 bits TTP™ (Two Time Programmable ROM)
2. RAM: 184 x 8 bits
3. STACK: 6 Levels
4. I/O ports: Three Programmable I/O ports (Max. 18 pins)
5. Timer0/Counter: 8-bit timer/counter with divided by 1~256 pre-scale option
6. Timer1: 8-bit auto-reloadable timer with divided by 1~256 pre-scale option
7. Two 8+2 bit PWM channels capable of 1024 duty resolution
8. 12-bit ADC with 8 channel input
9. Buzzer output
10. Watchdog/Wakeup Timer: On chip Timer based on internal RC oscillation, 13~140mS wakeup time
11. Reset: Power On Reset, Watchdog Reset, Low Voltage Reset, External pin Reset
12. System Clock Mode:
    - Slow Crystal: 32KHz ~ 455KHz
    - Fast Crystal: 1MHz ~24MHz
    - Internal RC: 4MHz
    - External RC
13. Operation Voltage: Low Voltage Reset Level to 5.5V
14. Instruction set: 36 Instructions
15. Interrupts: Three pin interrupts, Timer0/Timer1 interrupt and Wakeup Timer interrupt
16. Power Down mode support
17. Package Types: 16 DIP/SOP/SSOP, 20 DIP/SOP/SSOP

BLOCK DIAGRAM



PIN ASSIGNMENT





PIN DESCRIPTION

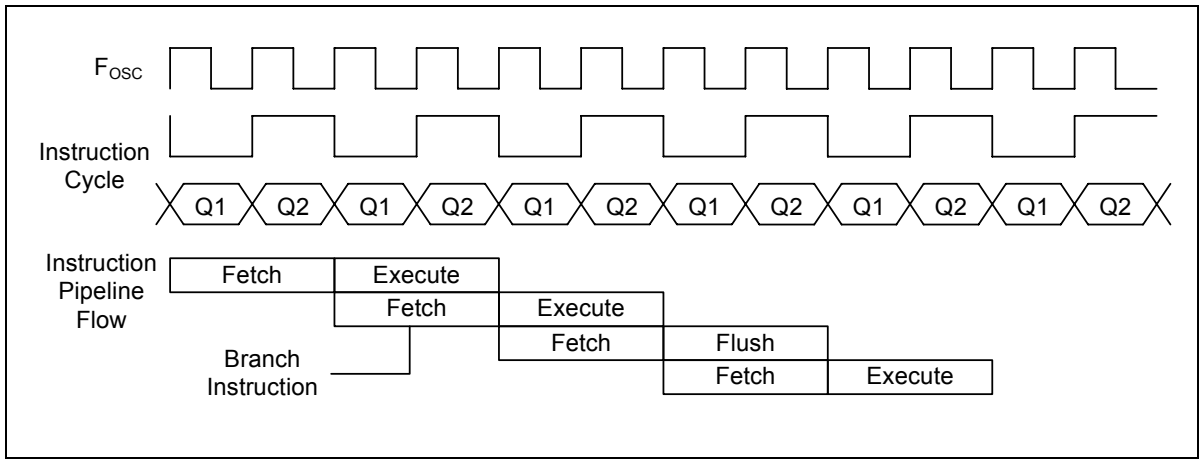
Name	In/Out	Pin Description
PA2–PA0	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS "**push-pull**" output or "**pseudo-open-drain**" output. Pull-up resistors are assignable by software.
PA6-PA3 PB1-PB0 PD7-PD0	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS "**push-pull**" 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
VDD, 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 converter input

FUNCTIONAL DESCRIPTION

1. CPU Core

1.1 Clock Scheme and Instruction Cycle

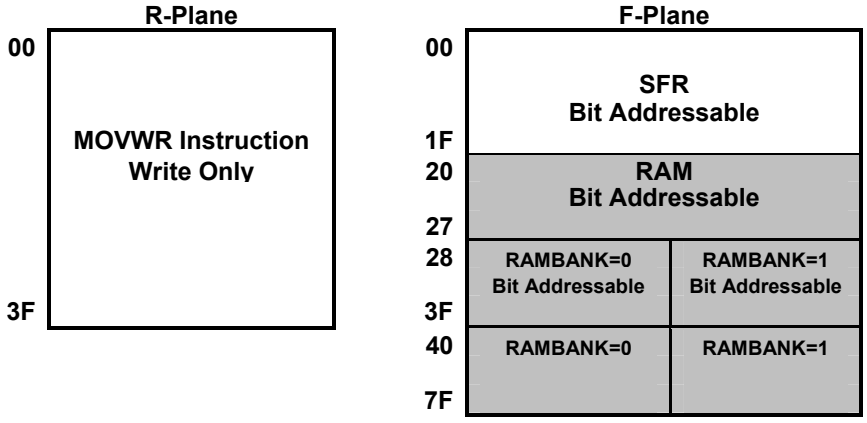
The system clock is internally divided by two to generate Q1 state and Q2 state for each instruction cycle. The Programming Counter (PC) is updated at Q1 and the instruction is fetched from program ROM and latched into the instruction register in Q2. It is then decoded and executed during the following 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 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 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 bits 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.

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 be used to alter the STATUS Register because these instructions do not affect those bits.

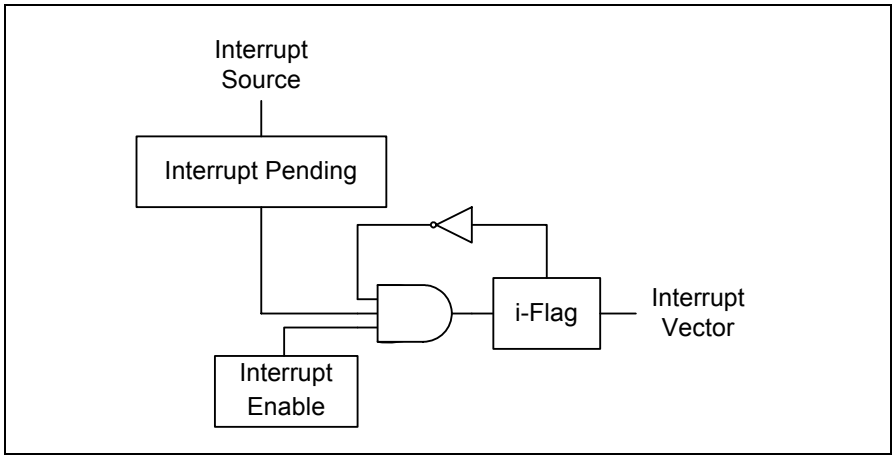
STATUS	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset Value	–	–	0	0	0	0	0	0
R/W	–	–	R/W	R	R	R/W	R/W	R/W

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

1.6 Interrupt

The TM57PA40 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 TM57PA40 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 trigged. F/W must clear the interrupt event register while serves the interrupt routine.



2. Chip Operation Mode

2.1 Reset

The TM57PA40 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. And 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.

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.

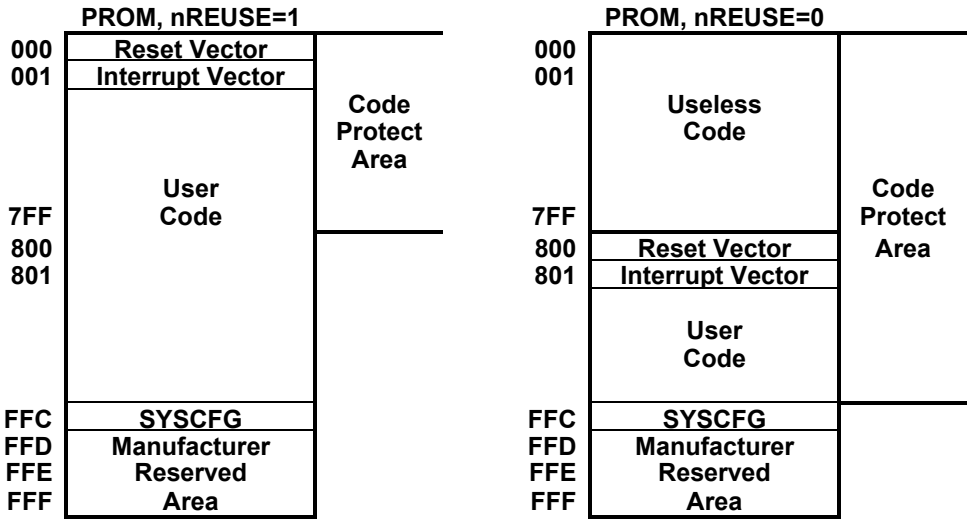
2.2 System Configuration Register (SYSCFG)

The System Configuration Register (SYSCFG) is located at ROM address FFCh. 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 13th bit of SYSCFG is code protection selection bit. If this bit is 0, the data in PROM will be protected, when user read PROM.

Bit	13~0	
Default Value	11_1111_111x_xxxx	
Bit	Description	
13	**nPROTECT**: Code Protection Selection	
	1	No protect
	0	Code protection
12	**nREUSE**: PROM Re-use Control	
	1	Not Re-use
	0	Re-use
11-10	**LVR**: LV reset mode	
	11	LVR threshold is 2.1V, always enable
	10	Invalid
	01	LVR threshold is 3.2V, always enable
	00	LVR disable
9-8	**CLKS**: Clock Source Selection	
	11	Fast Xtal (1MHz~24MHz)
	10	Slow Xtal (32KHz~455KHz)
	01	Internal RC (4MHz)
	00	External RC
7	**XRESETE**: External pin Reset Enable	
	1	Enable External pin Reset
	0	Disable External pin Reset
6	**WDTE**: WDT Reset Enable	
	1	Enable WDT Reset, Disable WKT Timer
	0	Disable WDT Reset, Enable WKT Timer
4-0	**IRCF**: Internal RC Frequency adjustment control	

2.3 PROM Re-use

The PROM of this device is 4K words. For some F/W program, the program size could be less than 2K 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, 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 800h. In the SYSCFG, if nPROTECT=0 and nREUSE=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 "nREUSE" control bit to "0". In the mean while, the Code protection area becomes the whole PROM except the Reserved Area.



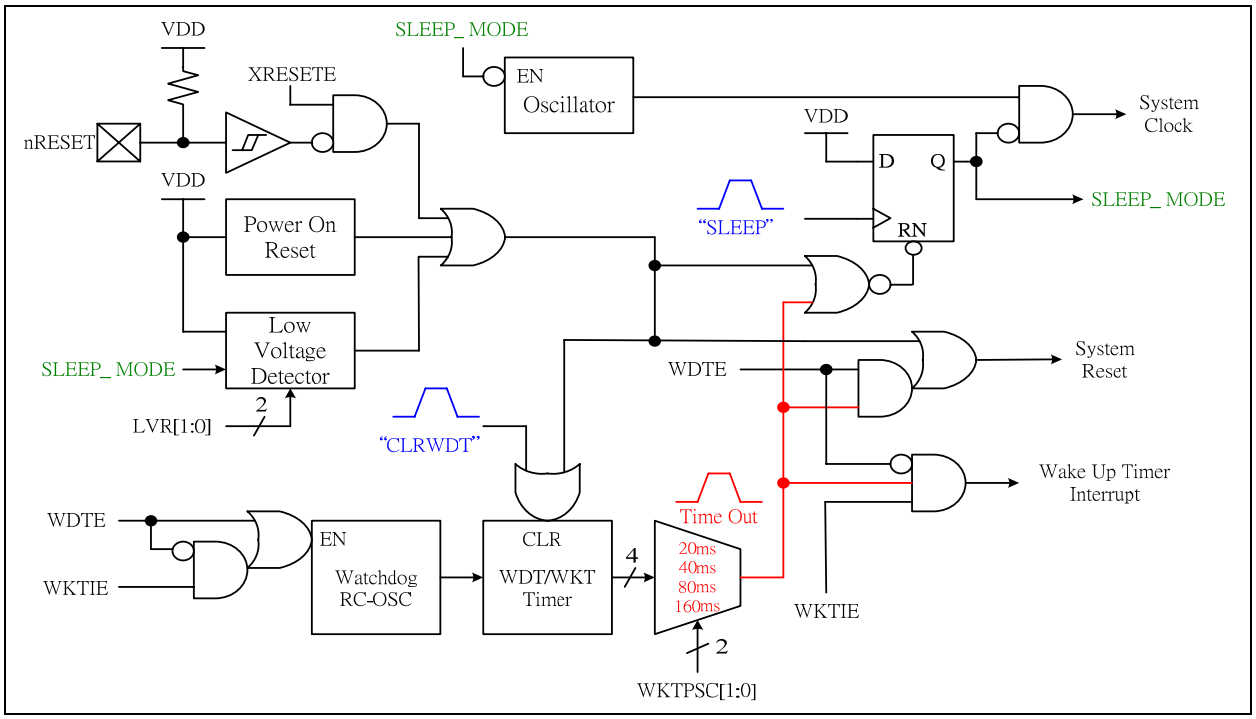
2.4 Power-Down Mode

The Power-down mode is activated by SLEEP instruction. During the Power-down mode, the system clock and peripherals stops 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).

3. Peripheral Functional Block

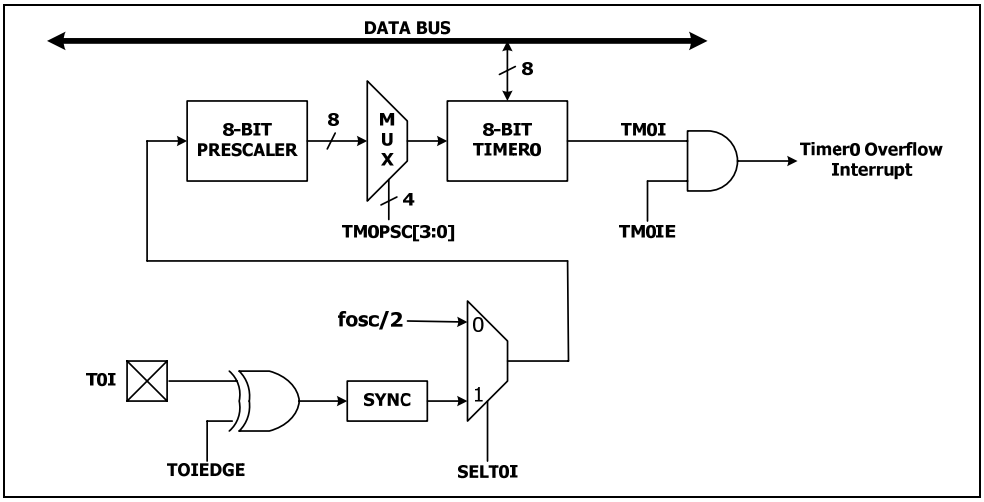
3.1 Watchdog(WDT) / Wakeup(WKT) Timer

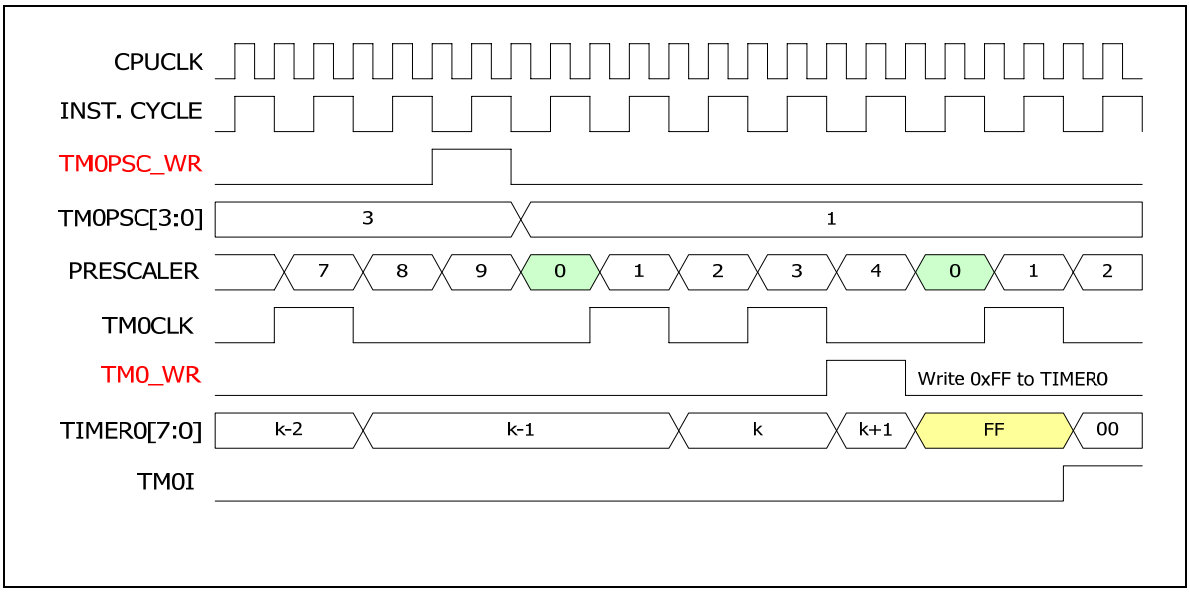
The WDT and WKT share the same internal RC Timer. The overflow period of WDT/WKT can be selected from 13mS to 140mS. The WDT/WKT is cleared by the CLRWDT instruction. If the Watchdog Reset is enabled (WDTE=1), the WDT generates the chip reset signal, otherwise, the WKT only generates overflow time out interrupt. The WDT/WKT works in both normal mode and 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)

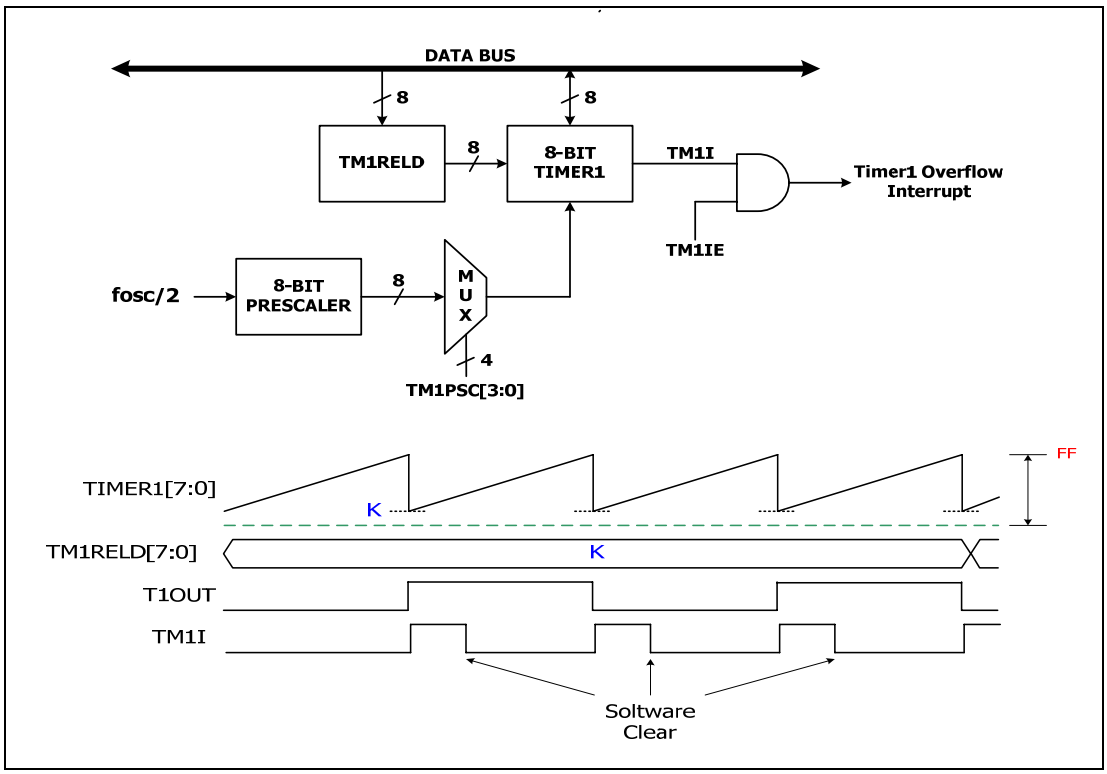
The Timer0 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatic roll over base on the pre-scaled clock source, which can be the instruction cycle or T0I input. The Timer0 increase rate is determined by “Timer0 Pre-Scale” (TM0PSC) register in R-Plane. The Timer0 can generate interrupt (TM0I) when it rolls over.

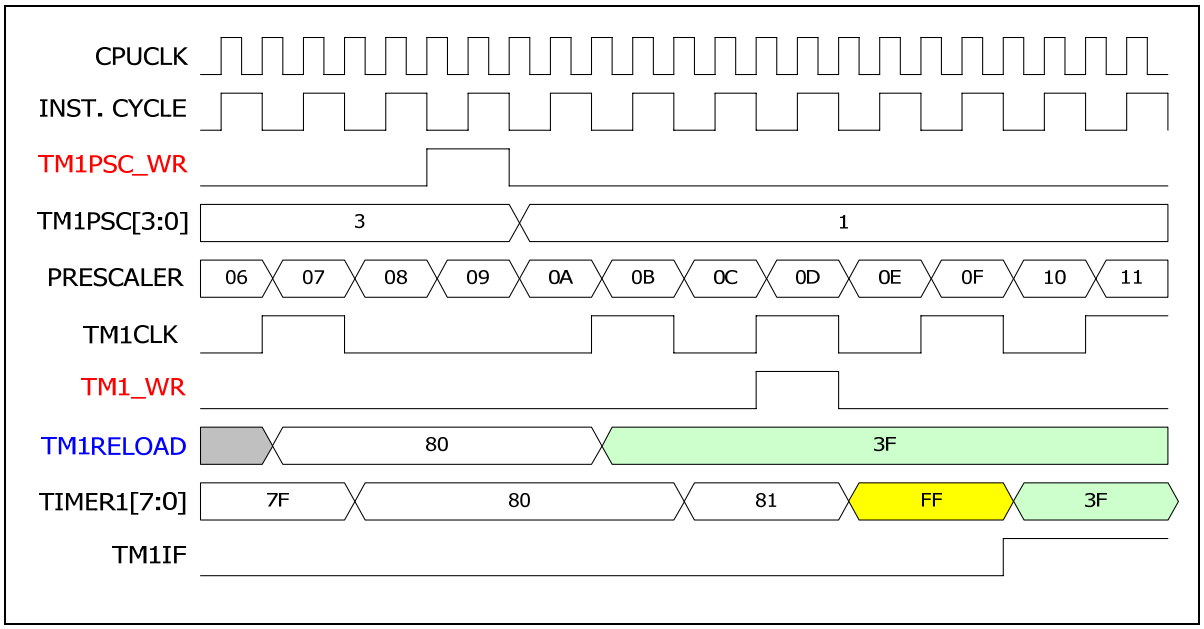




3.3 Timer1: 8-bit Timer with Pre-scale (PSC)

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 automatic reloads a new “offset value” (TM1RELD) while it rolls over base 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.

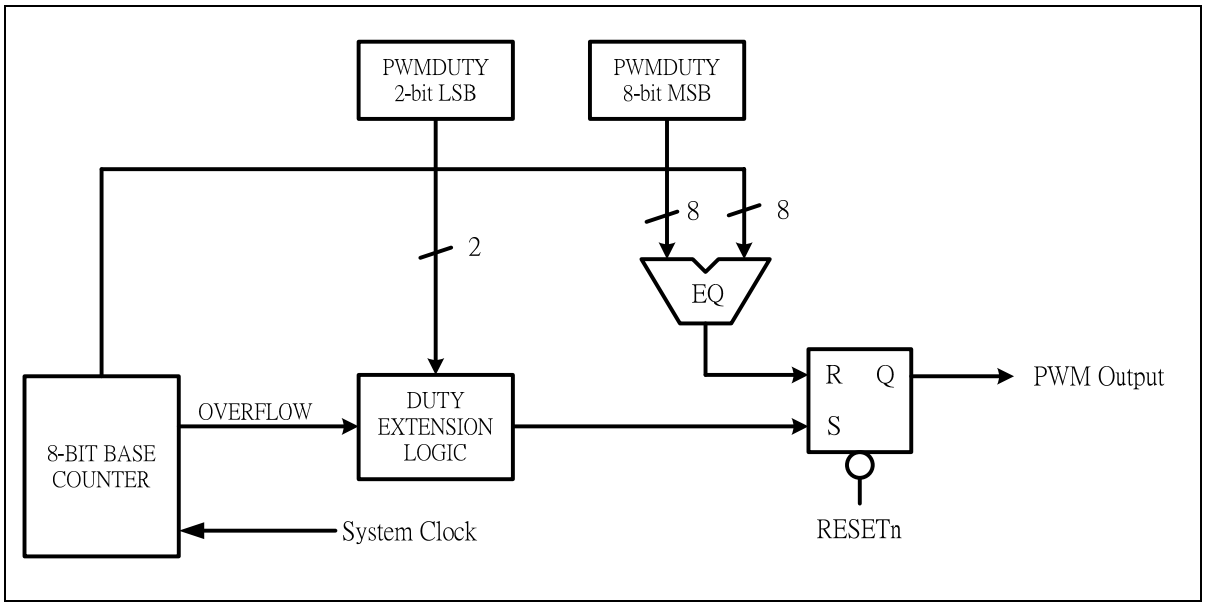


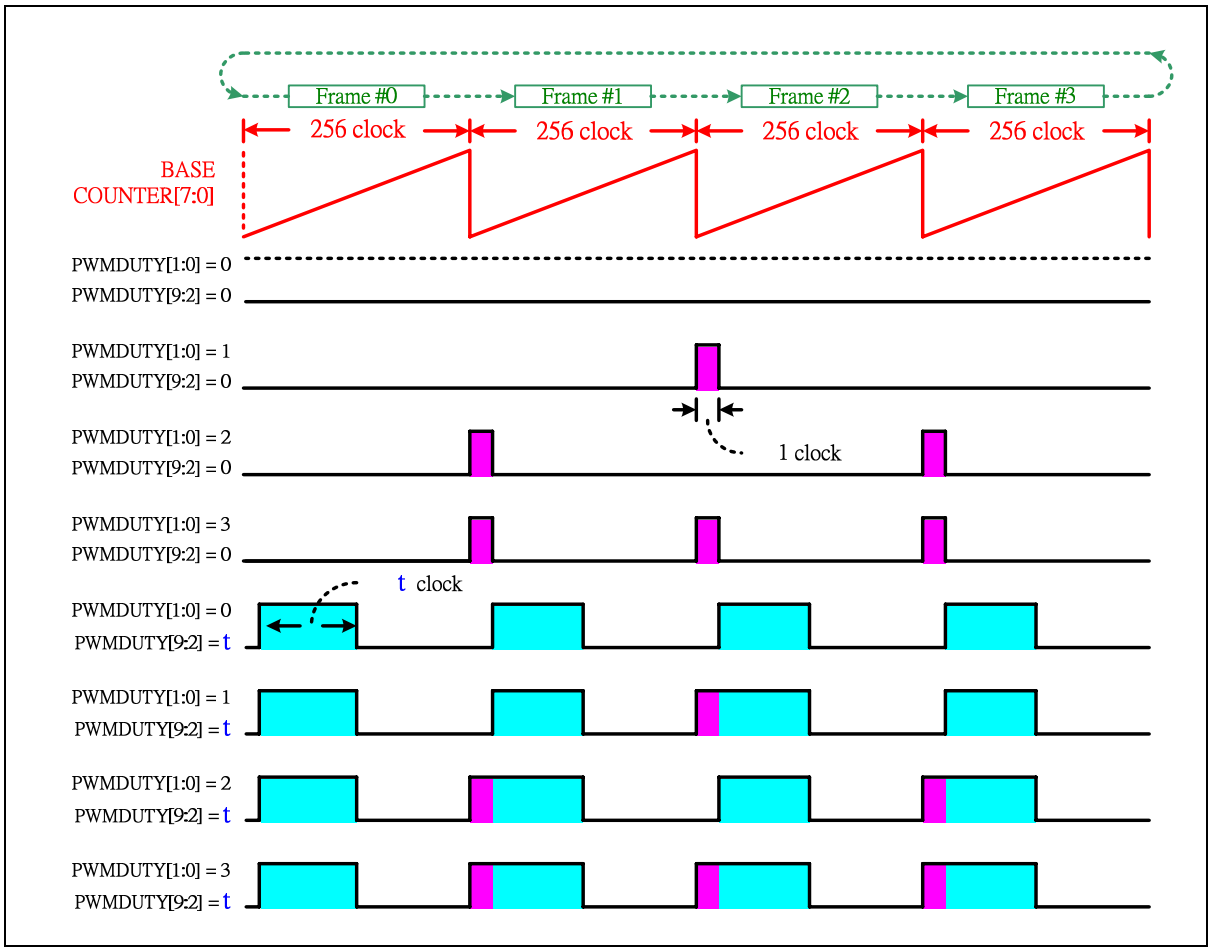


3.4 8+2 bit PWM

PWM0 and PWM1 have the same structure. The PWM can generate fix frequency waveform with 1024 duty resolution base on System Clock. A spread LSB technique allow 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 fast 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.

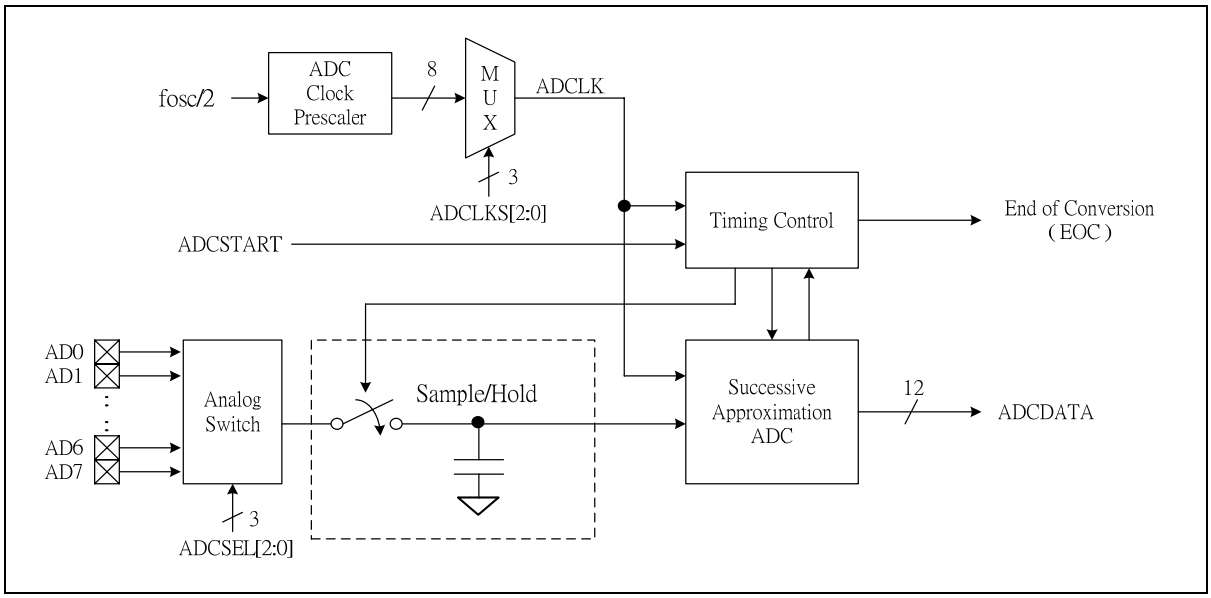




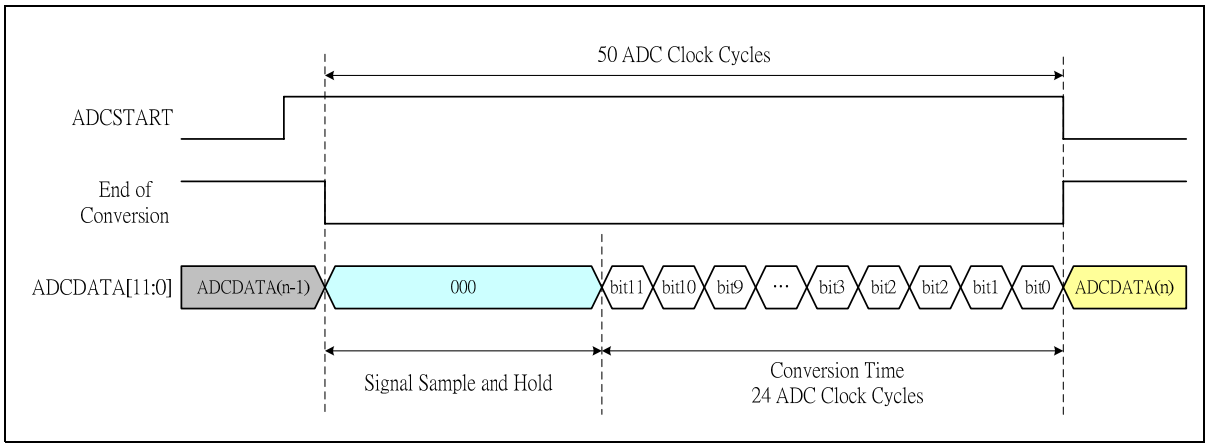
PWM example code:

```
        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 a 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 2MHz. User then launch the ADC conversion by set the ADCSTART control bit. After end of conversion, H/W automatic 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 a ADC input. The setting can disable the pin logical input path to save power consumption.



ADC example code:

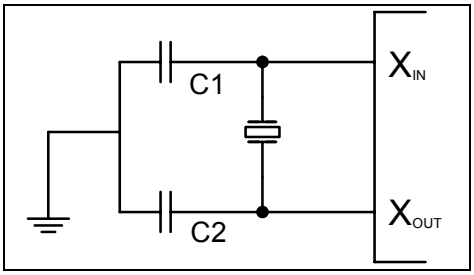
```
      movlw       00000110b
      movwf       11h         ;ADC channel select,ADC6(PA0) (ADCSEL)

      movlw       00000001h
      movwr       08h         ;disable PA0 pull up resistor (nPAPU)
```

```
      movlw       10111111b
      movwr       12h         ;set ADC6 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)
      :
```

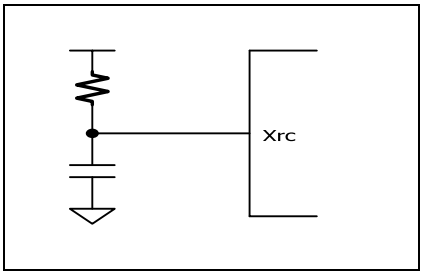
3.6 System Clock Oscillator

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 4MHz system clock.

.



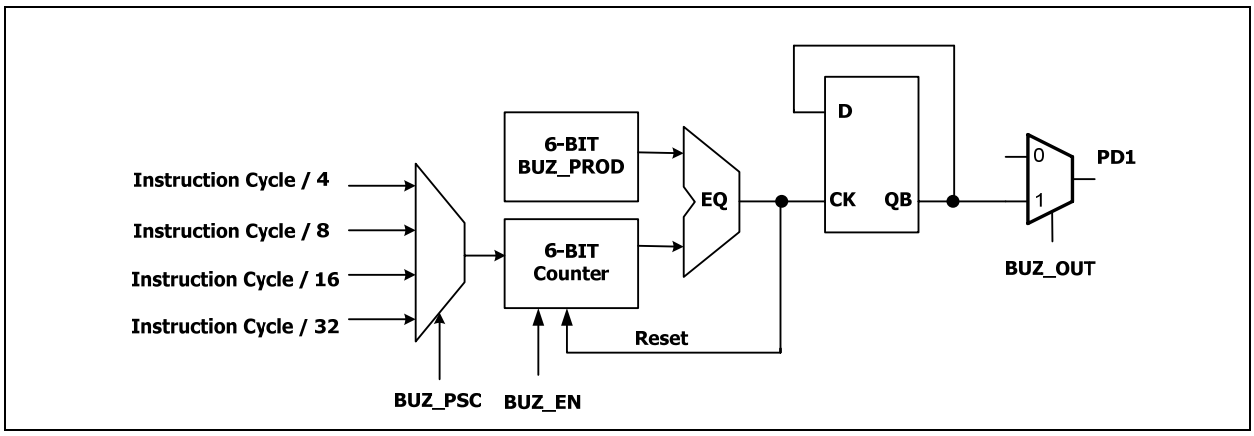
External Oscillator Circuit
(Crystal or Ceramic)

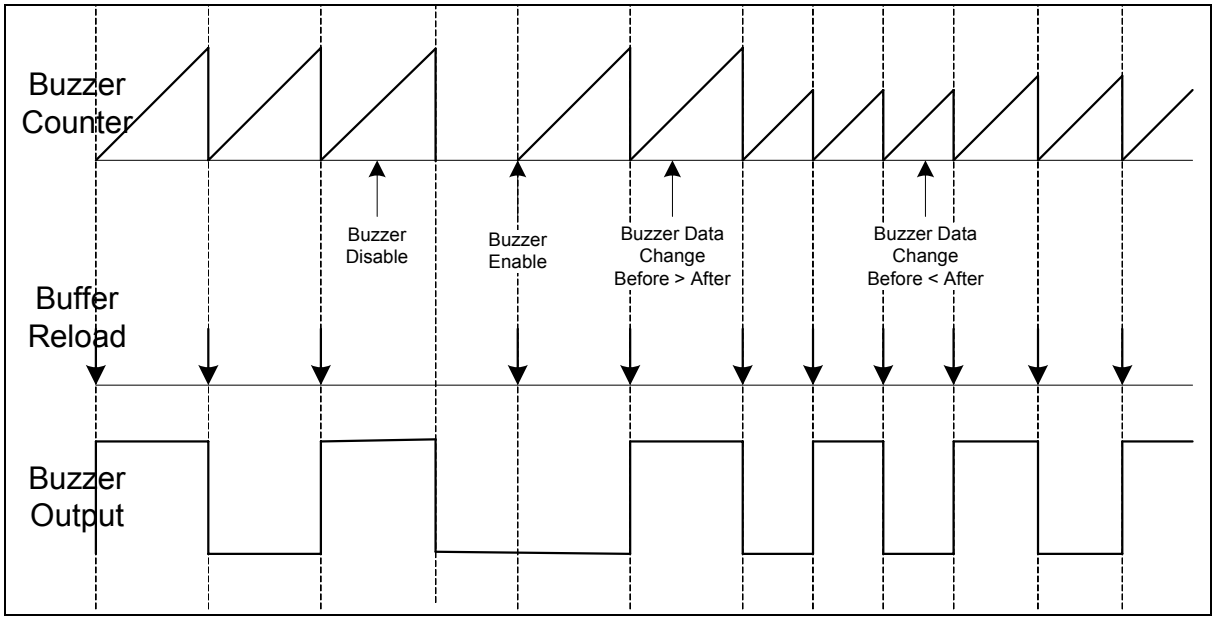


External RC Oscillator

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 need 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.

$$F_{BZ} = (f_{OSC}/2) / BUZ\_PSC / (BUZ\_PROD + 1)$$

Output frequency calculation

CPU Clock (fosc) = 8192KHz
Instruction Cycle = fosc/2 = 8192KHz/2 = 4096KHz
Prescaler Ratio (BUZ_PSC) = 11 (Instruction Cycle / 32)
Period Data (BUZ_PROD) = 9

F_{BZ} = (8192KHz/2) / 32 / (9+1) = 12.8 (KHz)

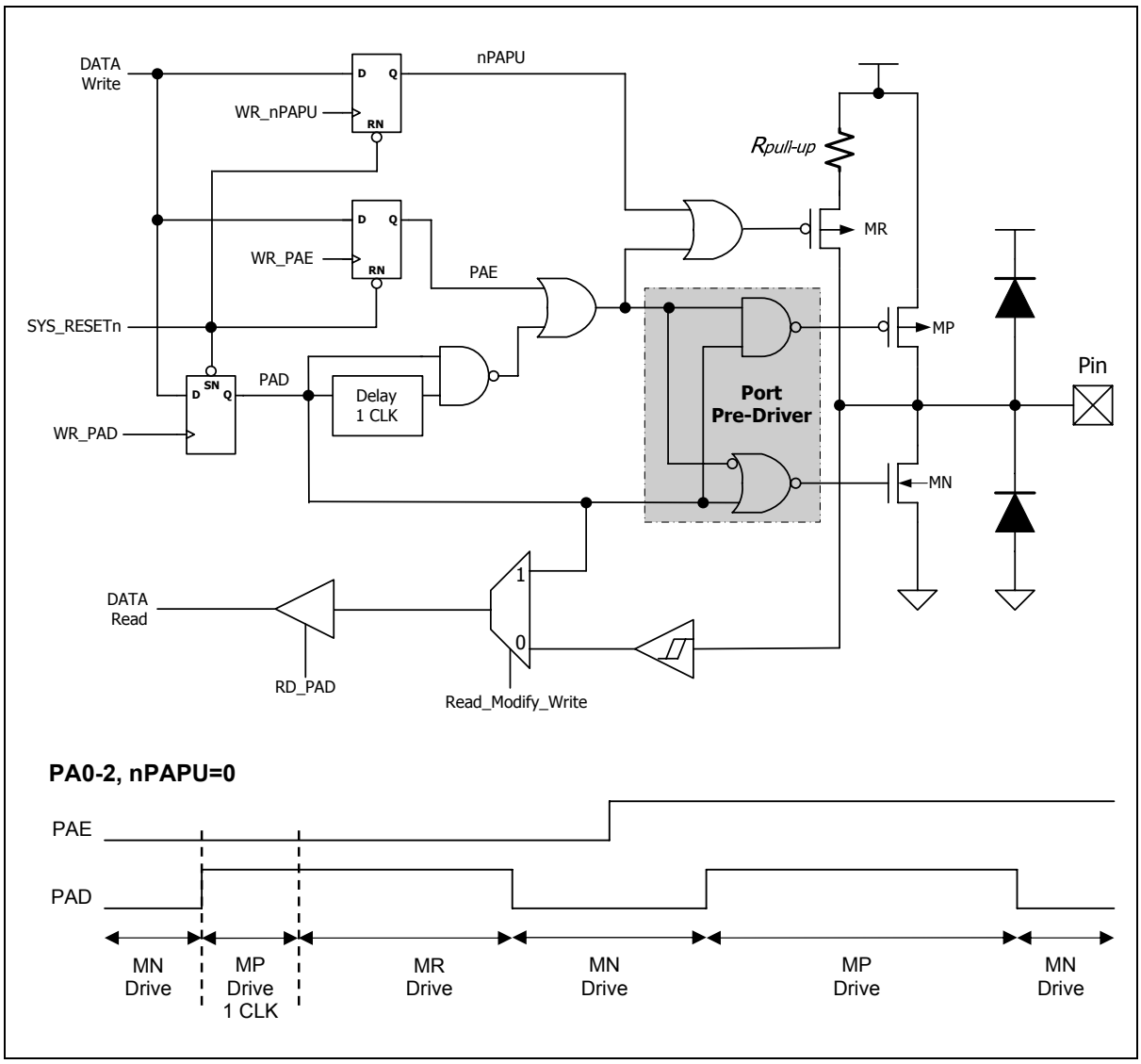
BUZZER example code:

```
      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

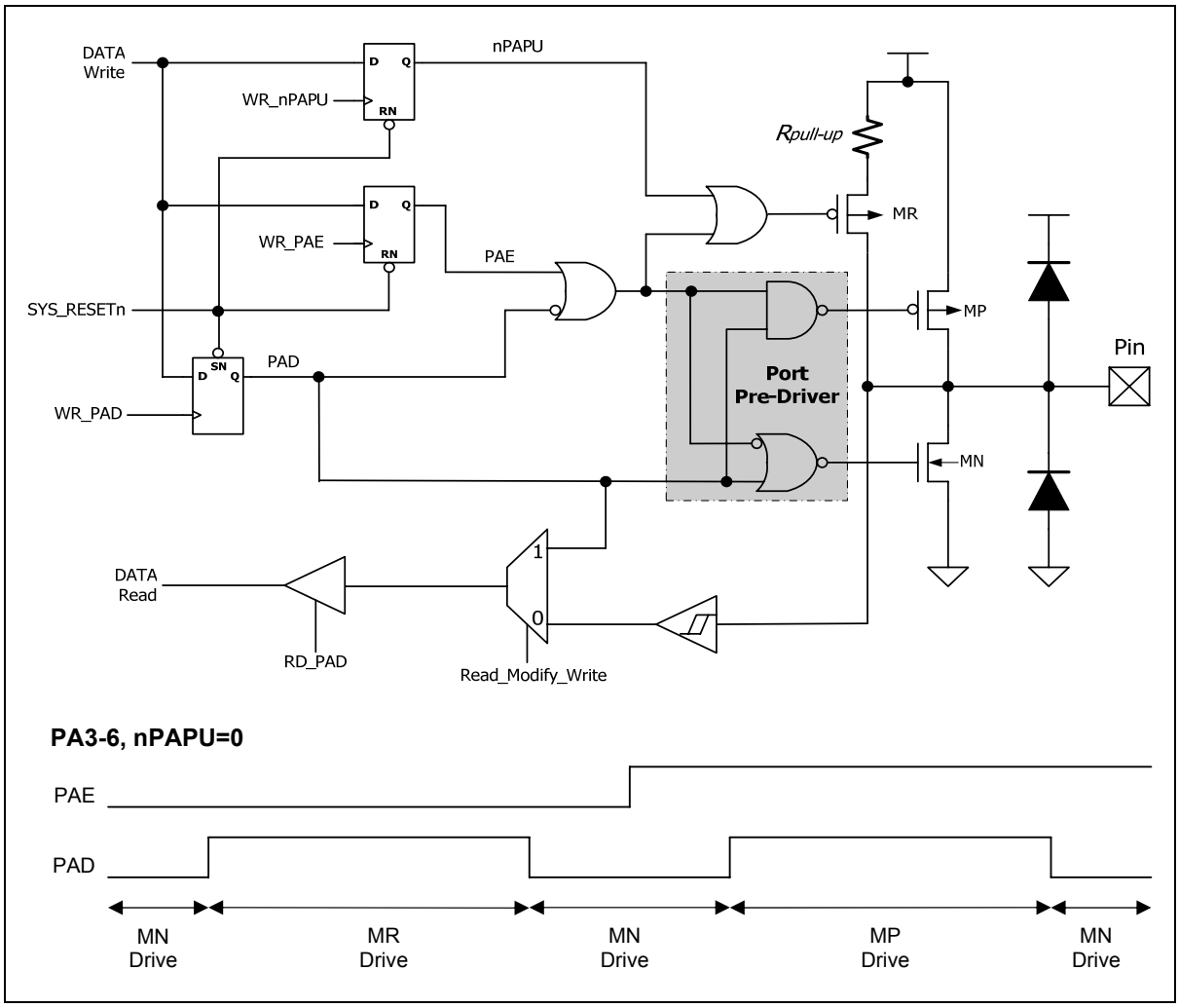
4.1 PA0-2

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 others instructions, CPU reads the pin state. The so-called “Read-Modify-Write” instruction includes BSF, BCF and all instructions using F-Plane as destination.



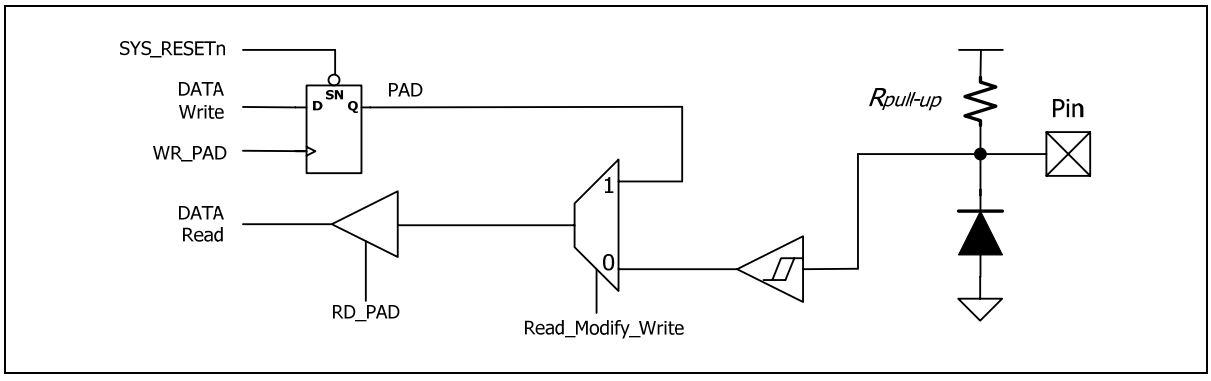
4.2 PA3-6, PB0-1, PD0-7

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, clear by 'SLEEP', 'CLRWDT' instruction
PD	03.3	R	0	Sleep mode flag, clear 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	04.6~0	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 overflow
		W	0	write 1: clear this flag; write 1: no action
TM0I	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
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 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 or rising 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
ADCDQ	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 clear this bit after ADC end of conversion

		W	0	S/W set 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 00: TCOUT output clock is “Instruction Cycle” divided by 1 01: TCOUT output clock is “Instruction Cycle” divided by 2 10: TCOUT output clock is “Instruction Cycle” divided by 4 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 1: T0I(PA2) falling edge to increase Timer0/PSC count
SELT0I	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
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 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
PBE	06.1~0	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
PDE	07.7~0	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
nPAPU	08.6~0	W	0	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 d. PA0 is working for PWM0 output 1: the pin pull up resistor is disable
nPBPU	09.1~0	W	0	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 (PBE=1) c. PB0 is working for PWM1 output 1: the pin pull up resistor is disable
nPDPU	0a.7~0	W	0	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 (PDD) is 0 b. the pin’s CMOS push-pull mode is chosen (PDE=1) c. the pin is working for T1OUT/BUZZER/TCOUT output 1: the pin pull up resistor is disable
BUZ_OUT	0b.7	W	0	0: disable BUZZER output to PD1 pin 1: enable BUZZER output to PD1 pin
PWM0_OUT	0b.6	W	0	0: disable PWM0 output to PA0 pin 1: enable PWM0 output to PA0 pin
PWM1_OUT	0b.5	W	0	0: disable PWM1 output to PB0 pin

				1: enable PWM1 output to PB0 pin
INT1EDGE	0b.4	W	0	0: INT1 pin (PA1) falling edge to trigger interrupt event 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 1: enable Instruction Clock divider output to PD6 pin
TM1_OUT	0b.2	W	0	0: disable Timer1 match out (T1OUT) to PD0 1: enable Timer1 match out (T1OUT) to PD0
WKTPSC	0b.1~0	W	11	WDT/WKT period (VDD=5V) 00: 13mS 01: 25mS 10: 50mS 11: 100mS
BUZ_EN	0c.7	W	0	0: disable BUZZER timer counting 1: enable BUZZER timer counting
ADCLKS	0c.6~4	W	0	000: ADC clock is “Instruction Cycle” divided by 128 001: ADC clock is “Instruction Cycle” divided by 64 ~ 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 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
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 do not write it
BUZ_PSC	10.7~6	W	0	BUZZER Clock Prescaler 00: BUZZER clock is “Instruction Cycle” divided by 4 01: BUZZER clock is “Instruction Cycle” divided by 8 10: BUZZER clock is “Instruction Cycle” divided by 16 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
ADC_TRIM	11.2~0	W	0	Test mode register, for manufacturer only, user do not write it
nADC_IE	12.7~0	W	ff	Each bit controls its corresponding ADC7~0 enable pin, if the bit is 0: the corresponding pin is ADC input 1: the corresponding pin is digital input

INSTRUCTION SET

Each instruction is a 14-bit word divided into an OPCODE, which specified 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 and enable Watch Dog 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 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		
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 ~ 7FFh	
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	WDTE ← 00h	
Status Affected	TO, PD	
OP-Code	00 0000 0000 0100	
Description	CLRWDT instruction enables and resets the Watchdog 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 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 ~ 7FFh	
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 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 ~ 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

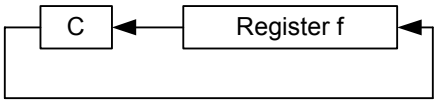
NOP	No Operation	
Syntax	NOP	
Operands	-	
Operation	No Operation	
Status Affected	Z	
OP-Code	00 0000 0000 0000	
Description	No Operation	
Cycle	1	
Example	NOP	-

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	RETFIE	A : PC = TOS, GIE = 1

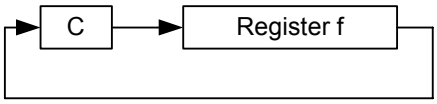
RETLW	Return with Literal in W	
Syntax	RETLW k	
Operands	k : 00h ~ FFh	
Operation	PC ← TOS, (W) ← k	
Status Affected	-	
OP-Code	01 1000 kkkk kkkk	
Description	The W register is loaded with the eightbit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.	
Cycle	2	
Example	CALL TABLE : TABLE ADDWF PCL,1 RETLW k1 RETLW k2 : RETLW kn	B : W = 0x07 A : W = value of k8

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 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 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 °C)

Parameter	Rating	Unit
Supply voltage	– 0.3 to + 6.5	V
Input voltage	– 0.3 to V_{DD} + 0.3	
Output voltage	– 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	– 25 to + 85	°C
Storage temperature	– 65 to + 150	

2. DC Characteristics (T_A = – 25 °C to + 85 °C, V_{DD} = 2.0 V to 5.5 V)

Parameter	Sym	Conditions		Min	Typ	Max	Unit
Input High Voltage	V_{IH}	All Input, except PA7	V_{DD} = 5 V	0.44V_{DD}			V
			V_{DD} = 3 V	0.32V_{DD}			V
		PA7	V_{DD} = 5 V	0.58V_{DD}			V
			V_{DD} = 3 V	0.63V_{DD}			V
Input Low Voltage	V_{IL}	All Input, except PA7	V_{DD} = 5 V			0.28V_{DD}	V
			V_{DD} = 3 V			0.27V_{DD}	V
		PA7	V_{DD} = 5 V			0.32V_{DD}	V
			V_{DD} = 3 V			0.3V_{DD}	V
Output High Voltage	V_{OH}	All Output	V_{DD} = 5 V,I_{OH}=7mA	4.5			V
			V_{DD} = 3 V,I_{OH}=4mA	2.7			V
Output Low Voltage	V_{OL}	All Output	V_{DD} = 5 V,I_{OL}=20mA			0.5	V
			V_{DD} = 3 V,I_{OL}=10mA			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
Power Supply Current	I_{DD}	Run 10 MHz, No Load	V_{DD} = 4.5 to 5.5 V	–	5		mA
		Run 4 MHz, No Load	V_{DD} = 2.0 V		1		
		Stop mode, No Load	V_{DD} = 4.5 to 5.5 V (LVR disable)	–		0.5	uA
			V_{DD} = 4.5 to 5.5 V (LVR enable)			1	
			V_{DD} = 3.0 V (LVR enable)			1	
System Clock Frequency	f_{OSC}	VDD > LVR_{th}	V_{DD} = 5 V	–		24	MHz
			V_{DD} = 3 V			18	
			V_{DD} = 2.2 V			10	
Pull-Up Resistor	R_P	V_{IN} = 0 V Ports A/B/D	V_{DD} = 5 V		90		k
			V_{DD} = 3 V		200		
		V_{IN} = 0 V PA7	V_{DD} = 5 V		15		k
			V_{DD} = 3 V		20		

3. Clock Timing (T_A = – 25°C to + 85°C)

Parameter	Condition			Min	Typ	Max	Unit
External RC Frequency	V_{DD} = 3V	R = 5K	C = 33pF	–	2.3	–	MHz
		R = 10K	C = 100pF	–	0.85	–	
		R = 100K	C = 300pF	–	0.046	–	
	V_{DD} = 5V	R = 5K	C = 33pF	–	2.8	–	
		R = 10K	C = 100pF	–	0.74	–	
		R = 100K	C = 300pF	–	0.032	–	
Internal RC Frequency	V_{DD} = 4.75 to 5.25 V			3.85	4	4.15	
	V_{DD} = 2.8 to 3.2 V			3.8	3.9	4.1	

4. Reset Timing Characteristics (T_A = – 25°C to + 85°C, V_{DD} = 2.0 V to 5.5 V)

Parameter	Conditions	Min	Typ	Max	Unit
RESET Input Low width	Input V_{DD} = 5 V ± 10 %	3	–	–	µs
WDT wakeup time	V_{DD} = 5 V, WKTPSC = 11	90	100	110	mS
	V_{DD} = 3 V, WKTPSC = 11	115	128	140	
CPU start up time	V_{DD} = 5 V	–	3.5	-	mS

5. LVR Circuit Characteristics (T_A = – 25°C to + 85°C, V_{DD} = 2.0 V to 5.5 V)

Parameter	Symbol	Min	Typ	Max	Unit
LVR reference Voltage	V_{LVR}	1.8	2.1	2.3	V
		2.9	3.2	3.5	
LVR Hysteresis Voltage	V_{HYST}	–	±0.1	–	V
Low Voltage Detection time	t_{LVR}	10	–	–	µs

6. ADC Electrical Characteristics (T_A = – 25°C to + 85°C, V_{DD} = 2.0 V to 5.5 V, V_{SS} = 0 V)

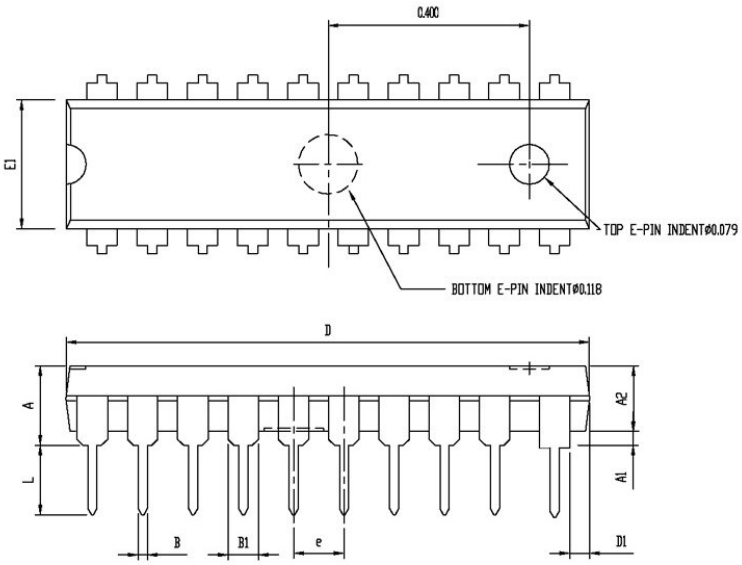
Parameter	Conditions	Min	Typ	Max	Units
Total Accuracy	V_{DD} = 5.12 V, V_{SS} = 0 V	–	± 2.5	± 4	LSB
Integral Non-Linearity		–	± 3.2	± 5	
Max Input Clock (f_{ADC})	–	–	–	2	MHz
Conversion Time	f_{ADC} = 2 MHz	–	25	–	µs
Input Voltage	–	V_{SS}	–	V_{DD}	V

PACKAGING INFORMATION

The TM57PA40 order information: “IC Type” “XX” “YY” “C” “Z”.

1. “IC TYPE”: TM57PA40
2. “XX”: Package Type
   - DIP Code: D
   - SOP Code: S
   - SSOP Code: SS
3. “YY”: IC Pin Number
   - Pin Number: 16 Code: 16
   - Pin Number: 20 Code: 20
4. “C”: Reserve (Must write be “C”)
5. “Z”: Package material
   - Package material: Pb-free Code: W
   - Package material: Green Package Code: G

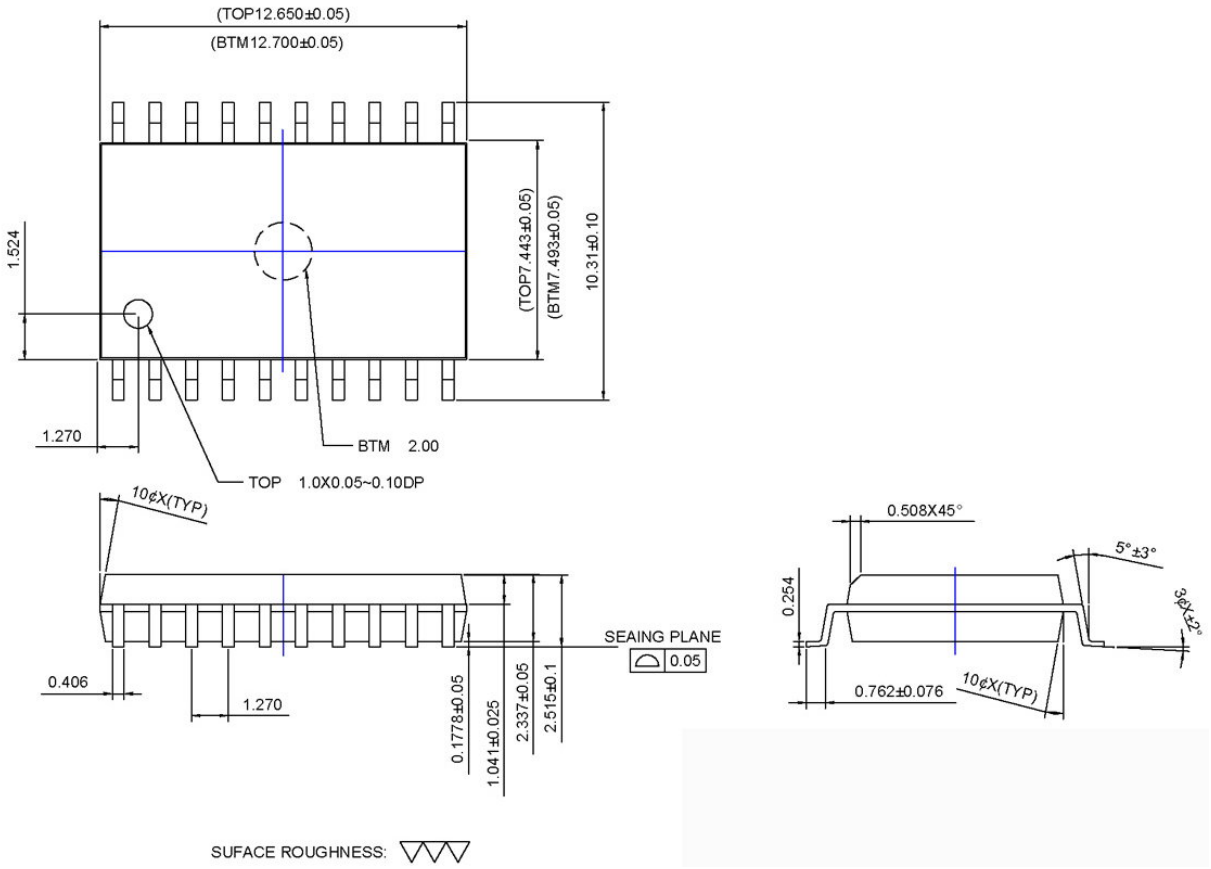
● 20-DIP Package Dimension



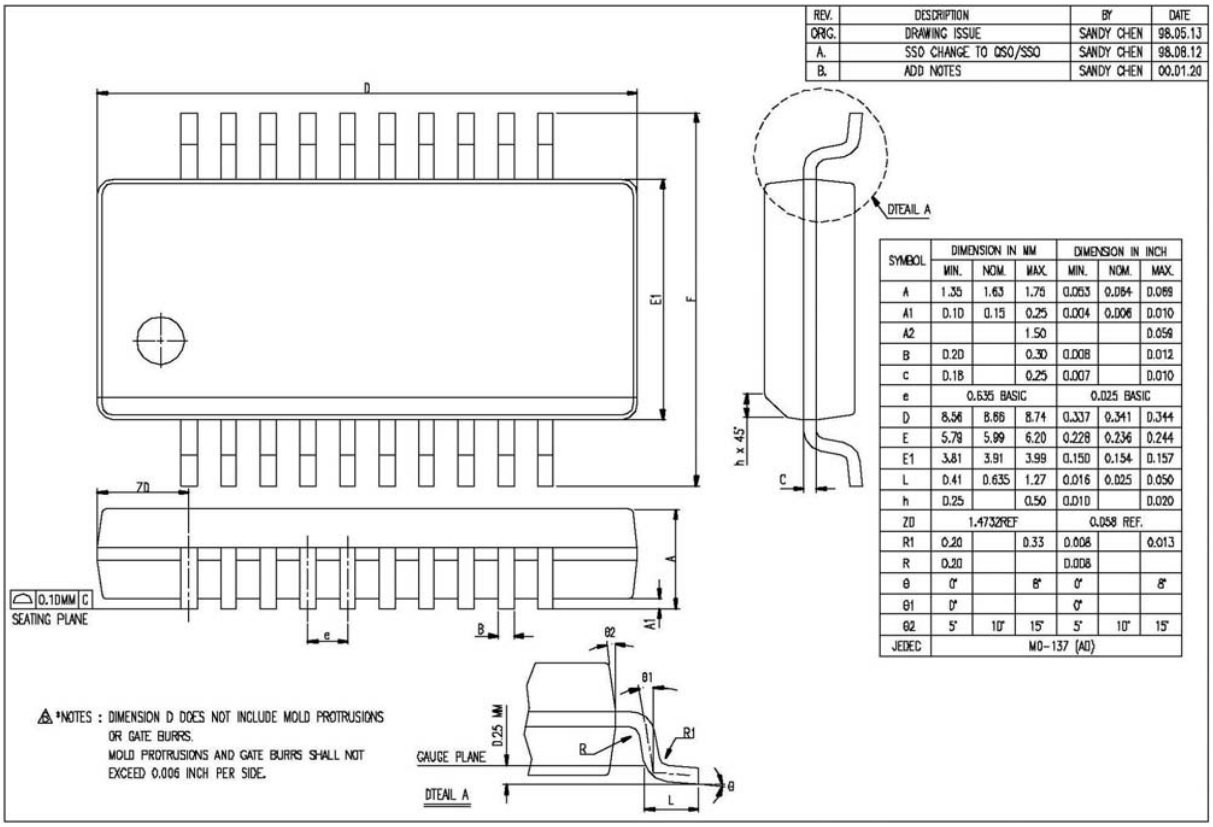
NOTE:
1. CONTROLLING DIMENSION :INCH
2. LEAD FRAME MATERIAL : A194
3. PACKAGE DIMENSION EXCLUDE MOLD FLASH OR PROTRUSION.
4. ALLOWABLE MOLD FLASH OR PROTRUSION SHALL NOT EXCEED 0.010"
5. TOLERANCE : 0.010"UNLESS OTHERWISE SPECIFIED.
6. AFTER SOLDER DIPPING LEAD THICKNESS WILL BE 0.020"MAX.

SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	—	—	4.57	—	—	0.180
A1	0.38	—	—	0.015	—	—
A2	—	3.30	3.56	—	0.130	0.140
B	0.36	0.46	0.56	0.014	0.018	0.022
B1	1.27	1.52	1.78	0.050	0.060	0.070
C	0.20	0.25	0.33	0.008	0.010	0.013
D	26.32	26.416	26.52	1.036	1.040	1.044
D1	0.43	0.56	0.69	0.017	0.022	0.027
E	7.62	—	8.26	0.300	—	0.325
E1	6.40	6.50	6.65	0.252	0.256	0.262
e	—	2.54	—	—	0.100	—
L	3.18	—	—	0.125	—	—
eB	8.38	—	9.65	0.330	—	0.380

- **20-SOP Package Dimension**



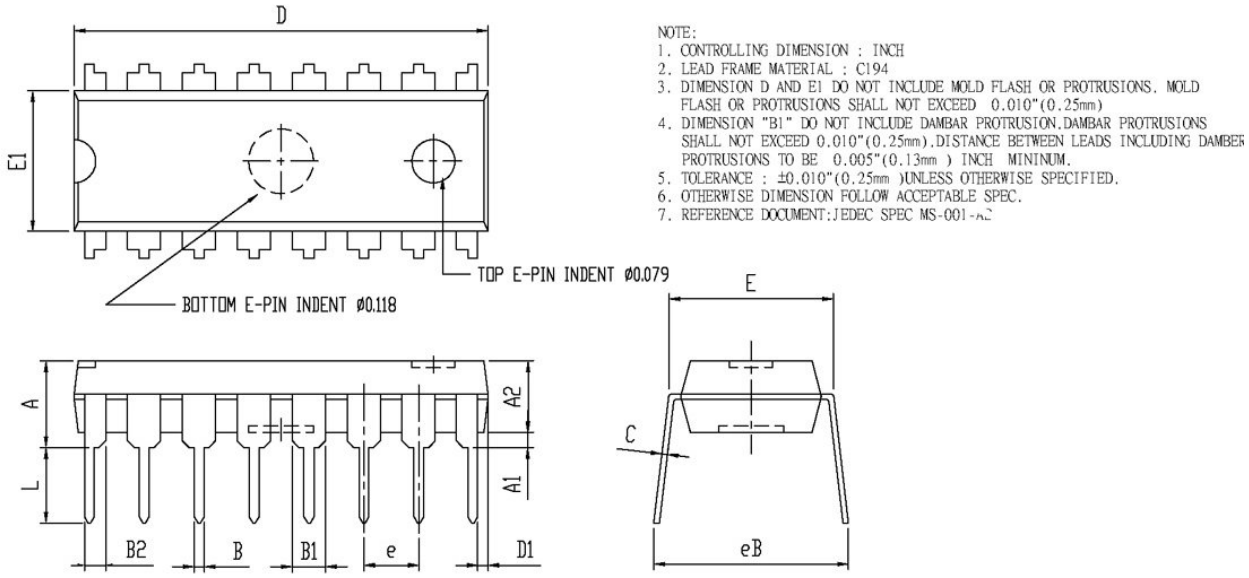
- **20-SSOP Package Dimension**



REV.	DESCRIPTION	BY	DATE
ORG.	DRAWING ISSUE	SANDY CHEN	98.05.13
A.	SSO CHANGE TO QSO/SSO	SANDY CHEN	98.08.12
B.	ADD NOTES	SANDY CHEN	00.01.20

SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	1.35	1.63	1.75	0.053	0.064	0.069
A1	0.10	0.15	0.25	0.004	0.006	0.010
A2			1.50			0.059
B	0.20		0.30	0.008		0.012
c	0.18		0.25	0.007		0.010
e	0.635 BASIC			0.025 BASIC		
D	8.56	8.66	8.74	0.337	0.341	0.344
E	5.79	5.99	6.20	0.228	0.236	0.244
E1	3.81	3.91	3.99	0.150	0.154	0.157
L	0.41	0.635	1.27	0.016	0.025	0.050
h	0.25		0.50	0.010		0.020
ZD	1.4732REF			0.058 REF.		
R1	0.20		0.33	0.008		0.013
R	0.20			0.008		
θ	0°		8°	0°		8°
θ1	0°			0°		
θ2	5°	10°	15°	5°	10°	15°
JEDEC	MO-137 (AD)					

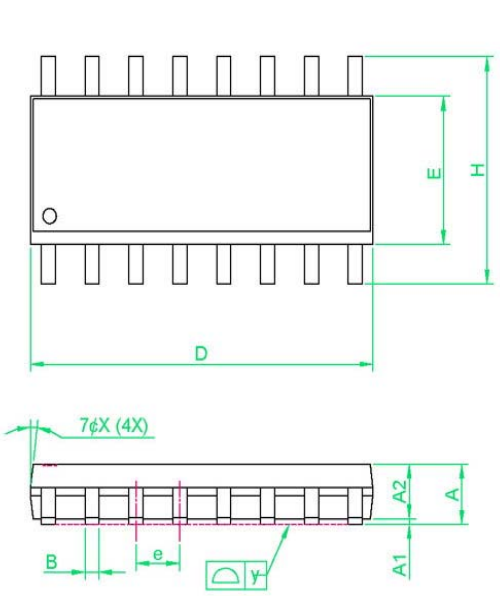
- **16-DIP Package Dimension**



NOTE:
1. CONTROLLING DIMENSION : INCH
2. LEAD FRAME MATERIAL : C194
3. DIMENSION D AND E1 DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED 0.010"(0.25mm)
4. DIMENSION "B1" DO NOT INCLUDE DAMBAR PROTRUSION.DAMBAR PROTRUSIONS SHALL NOT EXCEED 0.010"(0.25mm).DISTANCE BETWEEN LEADS INCLUDING DAMBER PROTRUSIONS TO BE 0.005"(0.13mm) INCH MININUM.
5. TOLERANCE : ±0.010"(0.25mm)UNLESS OTHERWISE SPECIFIED.
6. OTHERWISE DIMENSION FOLLOW ACCEPTABLE SPEC.
7. REFERENCE DOCUMENT:JEDEC SPEC MS-001-AC

SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	—	—	4.57	—	—	0.180
A1	0.38	—	—	0.015	—	—
A2	3.25	3.30	3.45	0.128	0.130	0.136
B	0.36	0.46	0.56	0.014	0.018	0.022
B1	1.27	1.52	1.78	0.050	0.060	0.070
B2	0.81	0.99	1.17	0.032	0.039	0.046
C	0.20	0.25	0.33	0.008	0.010	0.013
D	18.90	19.15	19.30	0.744	0.754	0.760
D1	0.33	0.46	0.58	0.013	0.018	0.023
E	7.62	—	8.26	0.300	—	0.325
E1	6.35	6.50	6.65	0.250	0.256	0.262
e	—	2.54	—	—	0.100	—
L	3.18	—	—	0.125	—	—
eB	8.64	—	9.65	0.340	—	0.380

- **16-SOP Package Dimension**

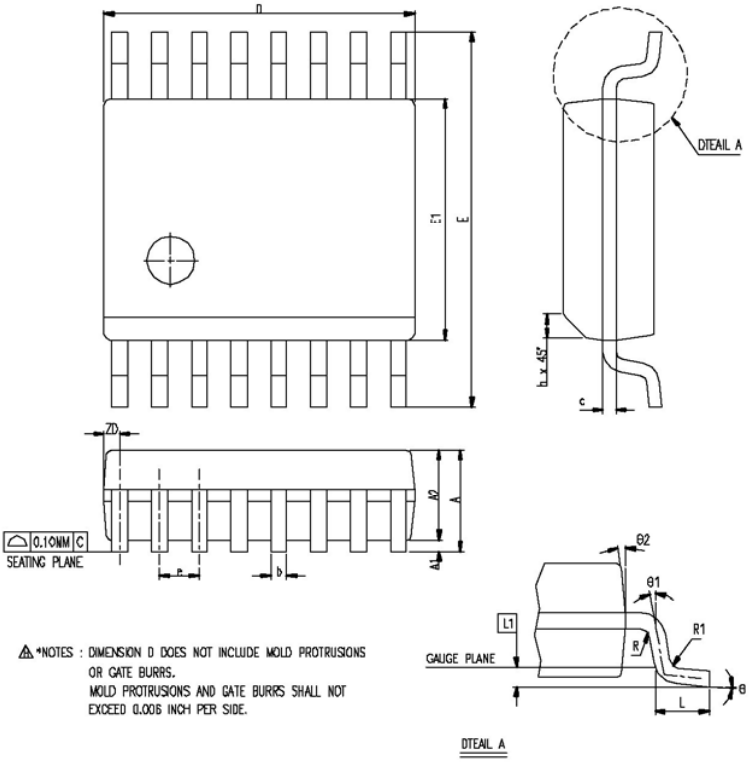


SYMBOLS	DIMENSIONS IN MILLIMETERS			DIMENSIONS IN INCHES		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.60	1.75	0.053	0.063	0.069
A1	0.10	----	0.25	0.004	----	0.010
A2	----	1.45	----	----	0.057	----
B	0.33	----	0.51	0.013	----	0.020
C	0.19	----	0.25	0.007	----	0.010
D	9.80	----	10.00	0.386	----	0.394
E	3.80	----	4.00	0.150	----	0.157
e	----	1.27	----	----	0.050	----
H	5.80	----	6.20	0.228	----	0.244
L	0.40	----	1.27	0.016	----	0.050
y	----	----	0.10	----	----	0.004
θ	0°	----	8°	0°	----	8°

NOTE
1. PACKAGE BODY SIZES EXCLUDE MOLD FLASH AND GATE BURRS
2. DIMENSION L IS MEASURED IN GAGE PLANE
3. TOLERANCE 0.10 mm UNLESS OTHERWISE SPECIFIED
4. CONTROLLING DIMENSION IS MILLIMETER. CONVERTED INCH DIMENSIONS ARE NOT NECESSARILY EXACT.
5. FOLLOWED FROM JEDEC MS-012

- **16-SSOP Package Dimension**

REV.	DESCRIPTION	BY	DATE
ORIG.	DRAWING ISSUE	SANDY CHEN	98.01.16
A.	SSO CHANGE TO QSO/SSO	SANDY CHEN	98.08.12
B.	ADD NOTES	SANDY CHEN	00.01.20
C.	DIMENSION IN MM CHANGE TO IN INCH	SANDY CHEN	01.01.15



⚠ *NOTES : DIMENSION D DOES NOT INCLUDE MOLD PROTRUSIONS OR GATE BURRS.
MOLD PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED 0.006 INCH PER SIDE.

SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	1.35	1.63	1.75	0.053	0.064	0.069
A1	0.10	0.15	0.25	0.004	0.006	0.010
A2			1.50			0.059
b	0.20		0.30	0.008		0.012
c	0.18		0.25	0.007		0.010
e	0.635 BASIC			0.025 BASIC		
D	4.80	4.90	5.00	0.189	0.193	0.197
E	5.79	5.99	6.20	0.228	0.236	0.244
E1	3.81	3.91	3.99	0.150	0.154	0.157
L	0.41	0.635	1.27	0.016	0.025	0.050
h	0.25		0.50	0.010		0.020
L1	0.254 BASIC			0.010 BASIC		
ZD	0.229 REF			0.009 REF		
R1	0.20		0.33	0.008		0.013
R	0.20			0.008		
θ	0°		8°	0°		8°
θ1	0°			0°		
θ2	5°	10°	15°	5°	10°	15°
JEDEC	MO-137 (AB)					