TM57PT20B

DATA SHEET

Rev 0.92

AMENDMENT HISTORY

Version	Date	Description
0.90	Nov, 2014	New release.
0.91	Nov, 2014	Added TM57PT20A/B comparison table in FEATURES (P.7)
0.92	Jun, 2018	Add Comparator Characteristics (P.84)

CONTENTS

FEATURES

1. **ROM: 2K x 14 bits OTP or 1K x 14 bits TTP™ (Two Time Programmable ROM)**
2. **RAM: 184 x 8 bits**
3. **STACK: 5 Levels**
4. **I/O Ports: Three bit-programmable I/O ports (Max. 18 pins)**
5. **Two Independent Timers**
   - Timer0
     - 8-bit Timer0 divided by 1 ~ 256 pre-scale option / counter / interrupt / stop function
   - T2
     - 15-bit T2 with 4 interrupt interval time options
     - IDLE mode wake-up timer or used as one simple 15-bit time base
     - Clock source: SXT or SIRC/2
6. **Three Independent PWMs**
   - One 8-bit PWM0 with pre-scale / period-adjustment / buffer-reload / clear and hold function
   - One 8-bit PWM1 with simple fixed frequency and duty cycle
   - One 8-bit PWM2 with simple fixed frequency and duty cycle
7. **One analog voltage comparator**
8. **14 channels Touch Key**
9. **Min. Operating Voltage (power on) and Speed: VDD can be lowest to 1.5V when the Fsys is 4 MHz**
10. **PA1 ~ PA6, PB1 ~ PB6 individual pin low level wake up**
11. **System Oscillation Sources**
    - Fast-clock
      - FXT (Fast Crystal): 1 MHz ~ 24 MHz
      - FIRC (Fast Internal RC): 8 MHz
    - Slow-clock
      - SXT (Slow Crystal): 32768 Hz
      - SIRC (Slow Internal RC)

        V_{DD} = 5V, SIRC = 110 KHz

        V_{DD} = 3V, SIRC = 88 KHz
12. **System Clock Prescaler: System Oscillation Sources can be divided by 16 / 4 / 2 / 1 as System Clock (Fsys)**

13. **Power Saving Operation Modes**
   - FAST Mode: Fast-clock keeps CPU running
   - SLOW Mode: Fast-clock stops, Slow-clock keeps CPU running
   - IDLE Mode: Fast-clock and CPU stop. T2 keeps running
   - STOP Mode: All Clocks stop, T2 stops
14. **Dual System Clock**
   - FIRC + SIRC
   - FIRC + SXT
   - FXT + SIRC
15. **Reset Sources**
   - Power On Reset
   - Watchdog Reset
   - Low Voltage Reset
   - External pin Reset
16. **3-Level Low Voltage Reset: 1.6V / 2.1V / 3.0V (can be disabled)**
17. **2-Level Low Voltage Detect: 2.2V / 3.1V (can be disabled)**
18. **Enhanced Power Noise Rejection**
19. **Built-in Power Management circuitry**
20. **Operation Voltage: Low Voltage Reset Level to 5.5V**
   - Fsys = 4 MHz, 1.5V ~ 5.5V
   - Fsys = 8 MHz, 1.8V ~ 5.5V
   - Fsys = 16 MHz, 2.3V ~ 5.5V
21. **Operating Temperature Range: -40℃ to +85℃**
22. **Interrupts**
   - Three External Interrupt Pins
     - Two pins are falling edge triggered
     - One pin is rising or falling edge triggered
   - Timer0 / T2 / Comparator Interrupts
23. **Watchdog Timer (WDT)**
   - Clocked by built-in RC oscillator with 4 adjustable Reset time options

     V_{DD} = 5V, WDT = 152 ms / 76 ms / 38 ms / 19 ms

     V_{DD} = 3V, WDT = 192 ms / 96 ms / 48 ms / 24 ms
   - Watchdog timer can be disabled/enabled in Power-down mode

24. I/O Port Modes

- Pseudo-Open-Drain Output (PA2 ~ PA0)
- Open-Drain Output
- CMOS Push-Pull Output
- Schmitt Trigger Input with pull-up resistor option

25. Table Read Instruction: 14-bit ROM data lookup table

26. Support 5-wire program

27. Instruction set: 39 Instructions

28. Package Types:

- 16-pin DIP (300 mil)
- 16-pin SOP (150 mil)
- 18-pin DIP (300 mil)
- 18-pin SOP (300 mil)
- 20-pin DIP (300 mil)
- 20-pin SOP (300 mil)

29. Supported EV board on ICE

EV board: EV2774

30. Comparison Table:

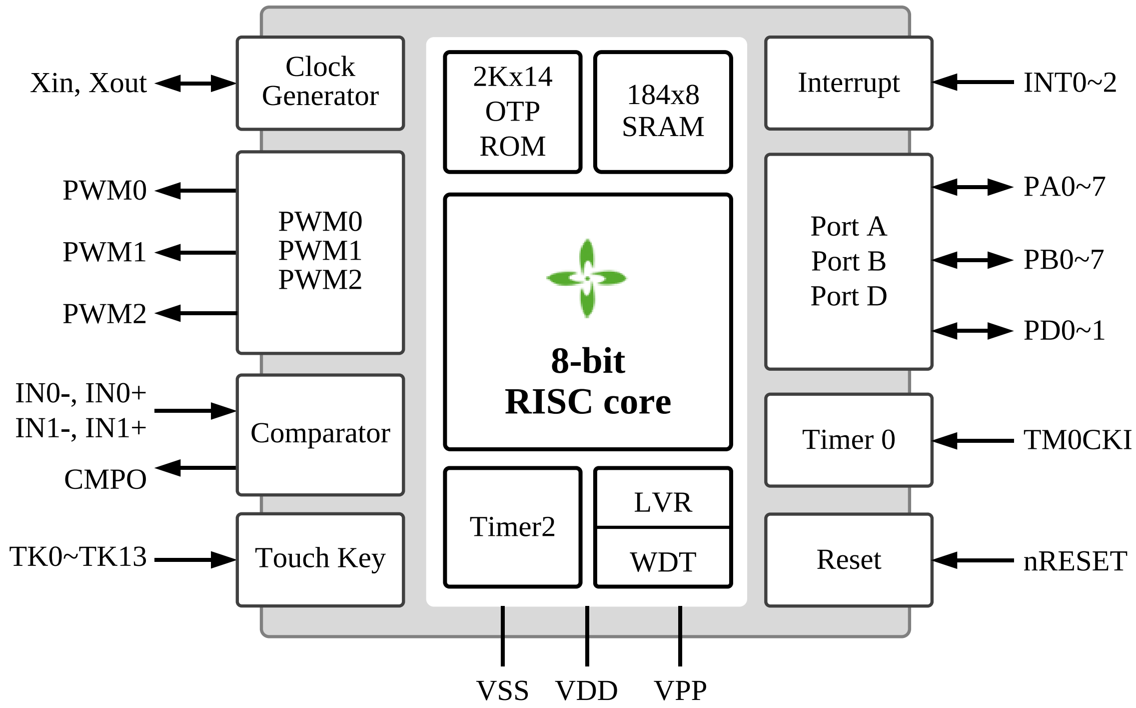
	EV2774	TM57PE20A	TM57PT20A	TM57PE20B	TM57PT20B
EV Board	–	EV2774	EV2774	EV2774	EV2774
Touch Key	14	X	14	X	14+Reference key(1)
PWM2	Always disable(2)	Always disable(2)	Always disable(2)	F16 = PWM2 duty R11.5 = PWM2OE	F16 = PWM2 duty R11.5 = PWM2OE
ESD Enhancement		+	+	++	++
EFT Immunity		+	+	++ (LVR=1.6V)	++ (LVR=1.6V)
Operating Voltage		1.6~5.5V@4MHz 2.1~5.5V@8MHz	1.6~5.5V@4MHz 2.1~5.5V@8MHz	1.5~5.5V@4MHz 1.8~5.5V@8MHz	1.5~5.5V@4MHz 1.8~5.5V@8MHz
Ioh typ. @5V	8mA	8mA	8mA	12mA	12mA
Iol typ. @5V	20mA	20mA	20mA	30mA	30mA
DC		Other unspecified items, refer to the DC characteristics for more details.			

Note:

(1): Reference key is available only for TM57PT20B (Set TKCHS=4'b1111)

(2): If PE20A/PT20A code replace with PE20B/PT20B, should take care and force R11.5=0

BLOCK DIAGRAM



TM57PT20B Block Diagram

PIN ASSIGNMENT

TM57PT20B — DIP-16 / SOP-16

TKCLD/PA5	1	16	PA6/PWM2/TK11
TK0/PWM1/PA1	2	15	PA0/TK10
TK1/INT0/TM0CKI/PA2	3	14	PA4/Xin
VPP/nRESET/INT2/PA7	4	13	PA3/Xout/TCOUT
VSS	5	12	VDD
TK2/IN0-/PB0	6	11	PB6/CMPO/TK8
TK3/IN1-/PB1	7	10	PB4/PWM0/TK6
TK4/IN0+/PB2	8	9	PB3/IN1+/TK5

TM57PT20B — DIP-18 / SOP-18

TKCLD/PA5	1	18	PA6/PWM2/TK11
TK0/PWM1/PA1	2	17	PA0/TK10
TK1/INT0/TM0CKI/PA2	3	16	PA4/Xin
VPP/nRESET/INT2/PA7	4	15	PA3/Xout/TCOUT
VSS	5	14	VDD
TK2/IN0-/PB0	6	13	PB7/TK9
TK3/IN1-/PB1	7	12	PB6/CMPO/TK8
TK4/IN0+/PB2	8	11	PB5/TK7
TK5/IN1+/PB3	9	10	PB4/PWM0/TK6

TM57PT20B — DIP-20 / SOP-20

TK13/INT1/PD0	1	20	PD1/TK12
TKCLD/PA5	2	19	PA6/PWM2/TK11
TK0/PWM1/PA1	3	18	PA0/TK10
TK1/INT0/TM0CKI/PA2	4	17	PA4/Xin
VPP/nRESET/INT2/PA7	5	16	PA3/Xout/TCOUT
VSS	6	15	VDD
TK2/IN0-/PB0	7	14	PB7/TK9
TK3/IN1-/PB1	8	13	PB6/CMPO/TK8
TK4/IN0+/PB2	9	12	PB5/TK7
TK5/IN1+/PB3	10	11	PB4/PWM0/TK6

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/O	Bit-programmable I/O port for Schmitt-trigger input or open-drain output. Pull-up resistor is assignable by software.
PB0–PB7	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.
PD0–PD1	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
TCOUT	O	Instruction cycle clock output. The instruction clock frequency is system clock frequency divided by two (Fsys/2)
VDD, VSS	P	Power Voltage input pin and ground
VPP	I	PROM programming high voltage input
INT0–INT2	I	External interrupt input
PWM0–PWM2	O	PWM output
TM0CKI	I	Timer0’s input in counter mode
IN0-, IN0+ IN1-, IN1+	I	Comparator voltage input
CMPO	O	Comparator output
TK0–TK13	I	Touch key input
TKCLD	I	Touch key capacitor input

PIN SUMMARY

Pin Number			Pin Name	Type	GPIO					Function After Reset	Alternate Function			
					Input		Output							
20-SOP/DIP	18-SOP/DIP	16-SOP/DIP			Weak Pull-up	Ext. Interrupt	O.D	P.O.D	P.P		PWM	Touch Key	ADC	MISC
1	-	-	TK13/INT1/PD0	I/O	○	○	○		○	PD0		○		
2	1	1	TKCLD/PA5	I/O	○		○		○	PA5		○		
3	2	2	TK0/PWM1/PA1	I/O	○			○	○	PA1	○	○		
4	3	3	TK1/INT0/ TM0CKI/PA2	I/O	○	○		○	○	PA2		○		TM0CKI
5	4	4	VPP/nRESET/ INT2/PA7	I/O	○	○	○			PA7				nRESET
6	5	5	VSS	P										
7	6	6	TK2/IN0-/PB0	I/O	○		○		○	PB0		○		IN0-
8	7	7	TK3/IN1-/PB1	I/O	○		○		○	PB1		○		IN1-
9	8	8	TK4/IN0+/PB2	I/O	○		○		○	PB2		○		IN0+
10	9	9	TK5/IN1+/PB3	I/O	○		○		○	PB3		○		IN1+
11	10	10	PB4/PWM0/TK6	I/O	○		○		○	PB4	○	○		
12	11	-	PB5/TK7	I/O	○		○		○	PB5		○		
13	12	11	PB6/CMPO/TK8	I/O	○		○		○	PB6		○		CMPO
14	13	-	PB7/TK9	I/O	○		○		○	PB7		○		
15	14	12	VDD	P										
16	15	13	PA3/Xout/TCOUT	I/O	○		○		○	PA3				Xout/TCOUT
17	16	14	PA4/Xin	I/O	○		○		○	PA4				Xin
18	17	15	PA0/TK10	I/O	○			○	○	PA0		○		
19	18	16	PA6/PWM2/TK11	I/O	○		○		○	PA6	○	○		
20	-	-	PD1/TK12	I/O	○		○		○	PD1		○		

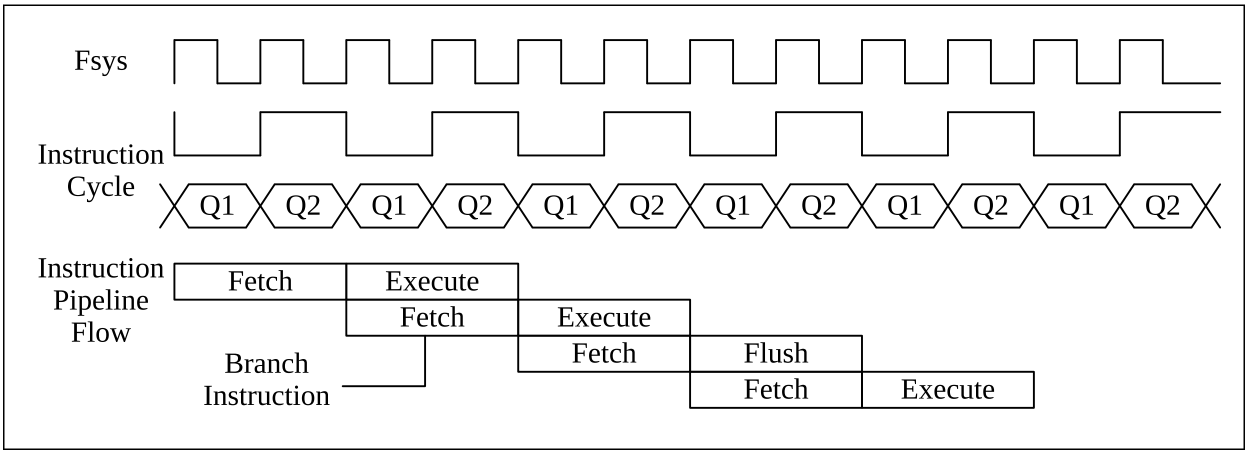
Symbol：P.P. = Push-Pull Output
P.O.D. = Pseudo Open Drain
O.D. = Open Drain

FUNCTIONAL DESCRIPTION

1. CPU Core

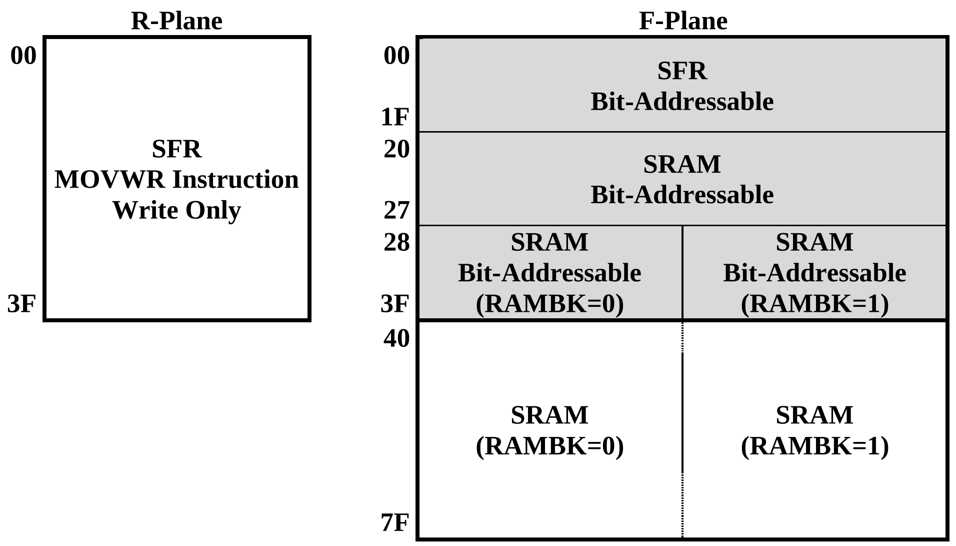
1.1 Clock Scheme and Instruction Cycle

The system clock (Fsys) 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 RAM 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 (F04.6~0) 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. And there are two RAM banks can be selected by RAMBK (F03.5).



◇Example: Write immediate data into R-Plane register

```
MOVLW    AAH              ; Move immediate AAH into W register
MOVWR    05H              ; Move W value into R-Plane location 05H
```

◇Example: Write immediate data into F-Plane register

```
MOVLW    55H              ; Move immediate 55H into W register
MOVWF    20H              ; Move W value into F-Plane location 20H
```

◇Example: Move F-Plane location 20H data into W register

```
MOVFW    20H              ; To get a content of F-Plane location 20H to W
```

◇Example: Clear SRAM Bank0 data by indirect addressing mode

```
         MOVLW    20H              ; W = 20H (SRAM start address)
         MOVWF    FSR              ; Set start address of user SRAM into FSR register
         BCF      STATUS, RAMBK    ; Set RAMBK = 0
LOOP:
         MOVLW    00H
         MOVFW    INDF             ; Clear user SRAM data
         INCF     FSR, 1           ; Increment the FSR for next address
         MOVLW    80H              ; W = 80H (SRAM end address)
         XORWF    FSR, 0           ; Check the FSR is end address of user SRAM?
         BTFSS    STATUS, Z        ; Check the Z flag
         GOTO     LOOP             ; If Z = 0, goto LOOP label
         …                         ; If Z = 1, exit LOOP
```

1.3 Programming Counter (PC) and Stack

The Programming Counter is 11-bit wide capable of addressing a 2K x 14 OTP 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 11 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[10:8] keeps unchanged. Therefore, the data of a lookup table must be located with the same PC[10:8]. The STACK is 11-bit wide and 5-level in depth. The CALL instruction and hardware interrupt will push STACK level in order. While the RET/RETI/RETLW instructions pop the STACK level in order.

For table lookup, the device offers the powerful table read instructions TABRL, TABRH to return the 14-bit ROM data into W register by setting the DPTR = {DPH, DPL} registers in F-Plane.

◇Example: To look up the PROM data located “TABLE”

```
            ORG       000H          ; Reset Vector
            GOTO      START         ; Goto user program address
START:
            MOVLW     00H
            MOVWF     INDEX         ; Set lookup table’s address (INDEX)
LOOP:
            MOVFW     INDEX         ; Move INDEX value to W register
            CALL      TABLE         ; To Lookup data (W = 55H when INDEX = 00H)
            …
            INCF      INDEX, 1      ; Increment the INDEX for next address
            …
            GOTO      LOOP          ; Goto LOOP label

            ORG       X00H          ; X = 1, 2, 3, …, 6, 7
TABLE:
            ADDWF     PCL, 1        ; (Addr = X00H) Add the W with PCL, the result
                                    ; back in PCL
            RETLW     55H           ; W = 55H when return
            RETLW     56H           ; W = 56H when return
            RETLW     58H           ; W = 58H when return
```

Note: TM57PT20B defines 256 ROM addresses as one page, so that TM57PT20B has eight pages, 000H~0FFH, 100H~1FFH, 200H~2FFH, …, and 700H~7FFH. On the other words, PC[10:8] can be defined as page. A lookup table must be located at the same page to avoid getting wrong data. Thus, the lookup table has maximum 255 data for above example with starting a lookup table at X00H (X=1, 2, 3, …, 6, 7). If a lookup table has fewer data, it needs not set the starting address at X00H, just only confirm all lookup table data are located at the same page.

◇Example: To look up the PROM data located in “TABLE” by TABRL and TABRH instructions

```
          ORG       000H               ; Reset Vector
          GOTO      START              ; Goto user program address
START:
          MOVLW     (TABLE>>8)&0xff    ; Get high byte address of TABLE label
          MOVWF     DPH                ; DPH (F17.1~0) = 02H
          MOVLW     (TABLE)&0xff       ; Get low byte address of TABLE label
          MOVWF     DPL                ; DPL (F04.7~0) = 80H
LOOP:
          TABRL                        ; W = 86H when DTPR = {DPH, DPL} = 0280H
          TABRH                        ; W = 19H when DTPR = {DPH, DPL} = 0280H
          …
          INCF      DPL, 1             ; Increment the DPL for next address
          …
          GOTO      LOOP               ; Goto LOOP label

          ORG       280H
TABLE:
          DT        0x1986             ; 14-bit ROM data
          DT        0x3719             ; 14-bit ROM data
```

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 (F-Plane 03H)

This register contains the arithmetic status of ALU, the reset status, and the voltage 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. The RAMBK bit is used to the SRAM Bank selection. The LVD bit is a voltage status flag. It is affected by the power supply voltage (V_{DD}). The LVD threshold voltage is chosen by SYSCFG[11:10].

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

Bit	Description	
7	**LVD**: Low Voltage Detect Flag LVD threshold is 2.2V/3.1V when LVR is 2.1V/3.0V 0: V_{DD} voltage is more than LVD threshold, LVR is disabled or VDDFLT (R0E.6) = 1 1: V_{DD} voltage is less than LVD threshold	
6	**GB0**: General Purpose Bit 0	
5	**RAMBK**: SRAM Bank Selection 0: SRAM Bank0 1: SRAM Bank1	
4	**TO**: Time Out Flag 0: after Power On Reset, LVR Reset, or CLRWDT/SLEEP instructions 1: WDT time out occurs	
3	**PD**: Power Down Flag 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
	0: no carry 1: a carry from the low nibble bits of the result occurs	0: a borrow from the low nibble bits of the result occurs 1: no borrow
0	**C**: Carry Flag or /Borrow Flag	
	ADD instruction	SUB instruction
	0: no carry 1: a carry occurs from the MSB	0: a borrow occurs from the MSB 1: no borrow

◇Example: Write immediate data into STATUS register

```
        MOVLW     00H
        MOVWF     STATUS              ; Clear STATUS register
```

◇Example: Bit addressing set and clear STATUS register

```
        BSF       STATUS, C           ; Set C = 1
        BCF       STATUS, C           ; Clear C = 0
```

◇Example: Determine the C flag by BTFSS instruction

```
        BTFSS     STATUS, C           ; Check the C flag
        GOTO      LABEL_1             ; If C = 0, goto LABEL_1 label
        GOTO      LABEL_2             ; If C = 1, goto LABEL_2 label
```

◇Example: Detect low supply voltage by the LVD flag

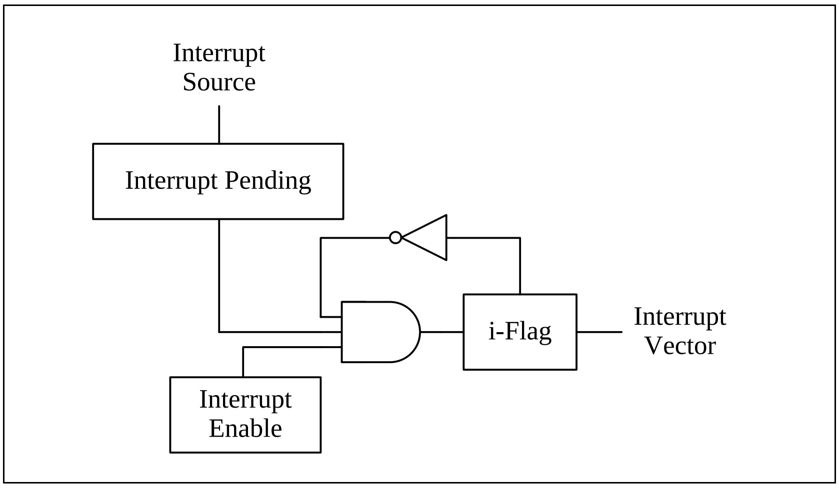
```
LOOP:
        BTFSC     STATUS, LVD         ; Check the LVD flag
        GOTO      LowBattery          ; If LVD = 1, goto LowBattery label
        GOTO      LOOP                ; If LVD = 0, goto LOOP label
```

1.6 Interrupt

The TM57PT20B has 1 level, 1 vector and 6 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 TM57PT20B 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 would trigger CPU to service the interrupt. CPU accepts interrupt in the end of current executed instruction cycle. In the mean while, a "CALL 001" instruction is inserted to CPU, and i-flag is set to prevent recursive interrupt nesting.

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



◇Example: Setup INT0 (PA2) interrupt request with rising edge trigger

```
        ORG     000H            ; Reset Vector
        GOTO    START           ; Goto user program address

        ORG     001H            ; All interrupt vector
        GOTO    INT             ; If INT0 (PA2) input occurred rising edge

        ORG     002H
START:
        MOVLW   xx00xxxxB
        MOVWR   PAMODL          ; Select INT0 (PA2) pin mode as
                                ; Open drain output low or input with Pull-up

        MOVLW   xxxxx1xxB
        MOVWF   PAD             ; Release INT0 (PA2), it becomes Schmitt-trigger
                                ; input mode with input pull-up resistor

        MOVLW   0xx1x0xxB
        MOVWR   R0B             ; Set INT0 interrupt trigger as rising edge
        MOVLW   11111110B
```

```
            MOVWF    INTIF            ; Clear INT0 interrupt request flag
            MOVLW    00000001B
            MOVWF    INTIE            ; Enable INT0 interrupt
MAIN:
            …
            GOTO     MAIN

INT:
            MOVWF    20H              ; Store W data to SRAM 20H
            MOVFW    STATUS           ; Get STATUS data
            MOVWF    21H              ; Store STATUS data to SRAM 21H

            BTFSS    INT0IF           ; Check INT0IF bit
            GOTO     EXIT_INT         ; INT0IF = 0, exit interrupt subroutine
            …                         ; INT0 interrupt service routine
            MOVLW    11111110B
            MOVWF    INTIF            ; Clear INT0 interrupt request flag
EXIT_INT:
            MOVFW    21H              ; Get SRAM 21H data
            MOVWF    STATUS           ; Restore STATUS data
            MOVFW    20H              ; Restore W data
            RETI                      ; Return from interrupt
```

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	–	T2IE	CMPIE	TM0IE	–	INT2IE	INT1IE	INT0IE
R/W	–	R/W	R/W	R/W	–	R/W	R/W	R/W
Reset	–	0	0	0	–	0	0	0

F08.6 **T2IE**: T2 interrupt enable
0: disable
1: enable

F08.5 **CMPIE**: Comparator interrupt enable
0: disable
1: enable

F08.4 **TM0IE**: Timer0 interrupt enable
0: disable
1: enable

F08.2 **INT2IE**: INT2 (PA7) pin interrupt enable
0: disable
1: enable

F08.1 **INT1IE**: INT1 (PD0) pin interrupt enable
0: disable
1: enable

F08.0 **INT0IE**: INT0 (PA2) pin interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	–	T2IF	CMPIF	TM0IF	–	INT2IF	INT1IF	INT0IF
R/W	–	R/W	R/W	R/W	–	R/W	R/W	R/W
Reset	–	0	0	0	–	0	0	0

F09.6 **T2IF**: T2 interrupt event pending flag
This bit is set by H/W while T2 overflows, write 0 to this bit will clear this flag

F09.5 **CMPIF**: Comparator interrupt event pending flag
This bit is set by H/W at Comparator output falling/rising edge, write 0 to this bit will clear this flag

F09.4 **TM0IF**: Timer0 interrupt event pending flag
This bit is set by H/W while Timer0 overflows, write 0 to this bit will clear this flag

F09.2 **INT2IF**: INT2 interrupt event pending flag
This bit is set by H/W at INT2 pin's falling edge, write 0 to this bit will clear this flag

F09.1 **INT1IF**: INT1 interrupt event pending flag
This bit is set by H/W at INT1 pin's falling edge, write 0 to this bit will clear this flag

F09.0 **INT0IF**: INT0 interrupt event pending flag
This bit is set by H/W at INT0 pin's falling/rising edge, write 0 to this bit will clear this flag

R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	–	T2PSC		INT0EDG	TCOE	–	WDTPSC	
R/W	–	W		W	W	–	W	
Reset	–	0	0	0	0	–	1	1

R0B.4 **INT0EDG:** INT0 pin (PA2) edge interrupt event
0: falling edge to trigger
1: rising edge to trigger

2. Chip Operation Mode

2.1 Reset

The TM57PT20B can be RESET in four ways.

- Power-On-Reset
- Low Voltage Reset (LVR)
- External Pin Reset (PA7)
- Watchdog Reset (WDT)

After Power-On-Reset, all system and peripheral control registers are then set to their default hardware Reset values. The clock source, LVR level and chip operation mode are selected by the SYSCFG register value. The Low Voltage Reset features static reset when supply voltage is below a threshold level. There are three threshold levels can be selected. The LVR's operation mode is defined by the SYSCFG register.

There are three voltage selections for the LVR threshold level, one is higher level which is suitable for application with V_{DD} is more than 3.6V, the second one is suitable for application with V_{DD} is more than 3.0V, while another one is suitable for application with V_{DD} is less than 3.0V. 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
LVR3.1	5.5V > V_{DD} > 3.6V
LVR2.3	5.5V > V_{DD} > 3.0V
LVR1.6	V_{DD} is wide voltage range

The External Pin Reset and Watchdog Reset can be disabled or enabled by the SYSCFG register. These two resets also set all the control registers to their default reset value.

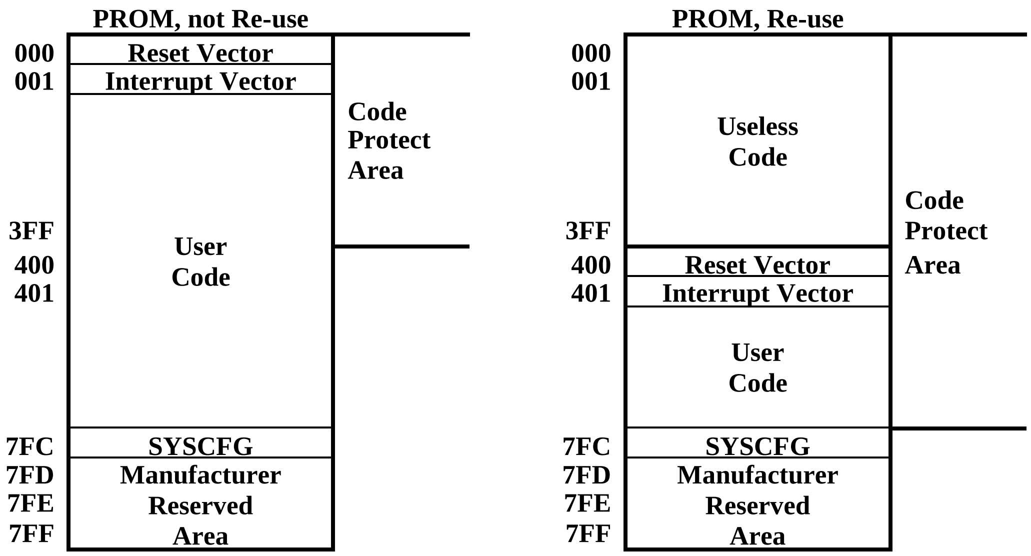
2.2 System Configuration Register (SYSCFG)

The System Configuration Register (SYSCFG) is located at ROM address 7FCh. The SYSCFG determines the option for initial condition of MCU. It is written by PROM Writer only. User can select 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, the data in PROM will be protected, when user reads PROM.

Bit	**13~0**	
Default Value	**11111111111111**	
Bit	Description	
13	**PROTECT**: Code protection selection	
	0	Enable
	1	Disable
12	**REUSE**: PROM Re-use control	
	0	Enable
	1	Disable
11-10	**LVR**: Low Voltage Reset Mode	
	00	LVR = 3.0V, LVD = 3.1V, always enable
	01	LVR = 2.1V, LVD = 2.2V, always enable
	10	LVR disable, LVD disable
	11	LVR = 1.6V; always enable. LVD disable
9-8	**Reserved**	
7	**XRSTE**: External Pin (PA7) Reset Enable	
	0	Disable, PA7 as IO pin
	1	Enable
6-5	**WDTE**: WDT Reset Enable	
	0x	WDT Reset Disable
	10	WDT Reset Enable in Fast/Slow Mode, Disable in Power-down Mode
	11	WDT Reset Always Enable
4-0	**Reserved**	

2.3 PROM Re-use ROM

The PROM of this device is 2K words. For some F/W program, the program size could be less than 1K 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 400h. In the SYSCFG, if protect mode is enabled and not Re-use, 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

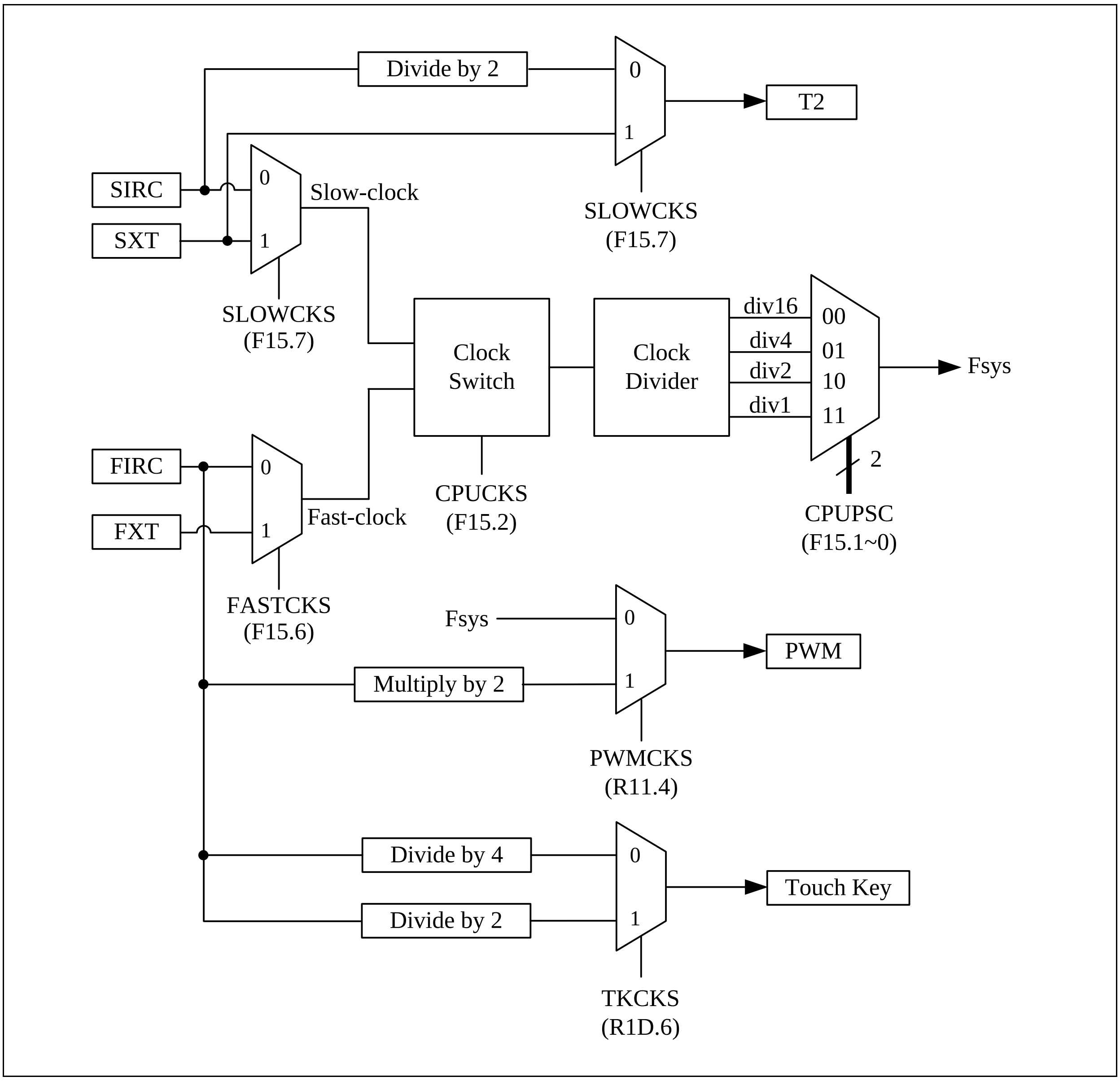
The Power-down mode includes IDLE Mode and STOP Mode. It is activated by SLEEP instruction. During the Power-down mode, the system clock and peripherals stop to minimize power consumption. The T2 Timer is working or not depends on F/W setting, and WDT is set by SYSCFG. The Power-down mode can be terminated by Reset, or enabled Interrupts (External pins and T2 interrupts) or PA1-6 and PB1-6 pins low level wake up.

R03	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWRDN	PWRDN							
R/W	W							
Reset	–	–	–	–	–	–	–	–

R03.7~0 **PWRDN:** Write this register to enter Power Down (STOP/IDLE) Mode

2.5 Dual System Clock

TM57PT20B 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. When in IDLE mode, only SXT or SIRC/2 can be configured to keep oscillating to provide clock source to T2 block. Refer to the Figure as below.



FAST Mode:

TM57PT20B enters FAST mode by setting the CPUCKS (F15.2). In FAST mode, TM57PT20B can select FXT or FIRC as its system clock source by setting FASTCKS (F15.6). However, change Fast-clock type under FAST mode is not allowed. User should let TM57PT20B 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 (R11.4).

SLOW Mode:

After power on or reset, TM57PT20B enters SLOW mode, the default Slow-clock is SIRC. User can select SXT or SIRC as its System clock by setting SLOWCKS (F15.7). However, change Slow-clock type under SLOW mode is not allowed. User should let TM57PT20B enter FAST mode first, change SLOWCKS, then back to SLOW mode.

IDLE Mode:

When SLOWSTP (F15.4) is cleared, the TM57PT20B will enter the “IDLE Mode” after executing the SLEEP instruction. In this mode, the Slow-clock will continue running to provide clock to T2 block. CPU stops fetching code and all blocks are stop except T2 related circuits.

T2 is independent and has its own control registers. It is possible to keep T2 working and wake-up in the IDLE mode.

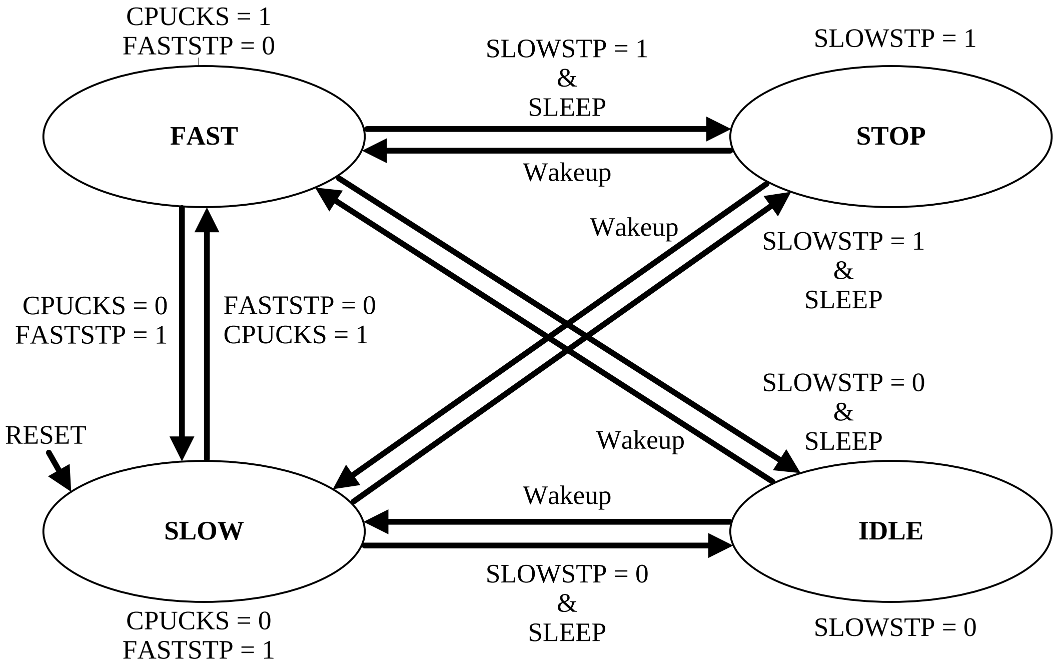
STOP Mode:

When SLOWSTP (F15.4) is set, all blocks will be turned off and the TM57PT20B will enter the “STOP Mode” after executing the SLEEP instruction. STOP mode is similar to IDLE mode. The difference is all clock oscillators either Fast-clock or Slow-clock are stopped and no clocks are generated.

2.6 Dual System Clock Modes Transition

TM57PT20B is operated in one of four modes: FAST Mode, SLOW Mode, IDLE Mode, and STOP Mode.

Modes Transition Diagram:



Note:
- SLEEP denotes SLEEP instruction
- Wakeup denotes wake-up events, such as External pin, WKT, T2 interrupts or pin wake-up
- CPUCKS (F15.2), FASTSTP (F15.3), SLOWSTP (F15.4)

CPU Mode & Clock Functions Table:

Mode	Oscillator	Fsys	Fast-clock	Slow-clock	TM0	T2	PWM0-2	Wakeup event
FAST	FIRC, FXT	Fast-clock	Run	Run	Run	Run	Run	X
SLOW	SIRC, SXT	Slow-clock	Stop	Run	Run	Run	Run	X
IDLE	SIRC, SXT	Stop	Stop	Run	Stop	Run	Stop	T2/IO
STOP	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 (F15.7). 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 (F15.6). 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
```

IDLE Mode Setting:

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

(1) Enable Slow-clock (SLOWSTP = 0)

(2) Execute SLEEP instruction

IDLE mode can be woken up by interrupts (XINT or T2) or PA1-6 and PB1-6 pins low level wake up.

◇Example: Switch operating mode to IDLE mode

```
BCF     SLOWSTP     ; Enable Slow-clock
SLEEP               ; Enter IDLE mode
```

STOP Mode Setting:

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

(1) Stop Slow-clock (SLOWSTP = 1)

(2) Execute SLEEP instruction

STOP mode can be woken up by interrupt (XINT) or PA1-6 and PB1-3 pins low level wake up.

◇Example: Switch operating mode to STOP mode

```
BSF     SLOWSTP     ; Stop Slow-clock
SLEEP               ; Enter STOP mode
```

IO setting notes in STOP/IDLE mode:

Note: In STOP/IDLE mode, PA3 and PA4 must be set as input mode with internal pull-up enable to avoid floating state when select FXT or SXT mode. The PA3 and PA4 IO setting list as below.

	Fast-clock	Slow-clock	PAMODL[7]	PAMODL[6]	PAD3	PAMODH[1]	PAMODH[0]	PAD4
1	FIRC	SIRC	※	※	※	※	※	※
2	FIRC	SXT	0	0	1	0	0	1
3	FXT	SIRC	0	0	1	0	0	1

※：Don't care

F15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCTL	SLOWCKS	FASTCKS	–	SLOWSTP	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

F15.7 **SLOWCKS**: Slow-clock type select or T2 clock source select
For Slow-clock type
0: SIRC
1: SXT
For T2 clock source
0: SIRC/2
1: SXT

F15.6 **FASTCKS**: Fast-clock type select
0: FIRC
1: FXT

F15.4 **SLOWSTP**: Slow-clock Enable / Disable
0: enable
1: disable in Power-down mode

F15.3 **FASTSTP**: Fast-clock Enable / Disable
0: enable
1: disable

F15.2 **CPUCKS**: System clock source select
0: Slow-clock
1: Fast-clock

F15.1~0 **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 (F15) can't be set directly for CPU modes transition. It may cause the transition fail. Please refer the mentioned steps for transition in this chapter.

2.7 Internal Power Management

The TM57PT20B has built-in Power Management circuitry and scheme to adapt user's system operation voltage and clock speed. The Power Management related control bits are listed below.

NOPUMP: (R0E.3, Default = 0)

If this bit is "1", the TM57PT20B's internal Voltage Pump circuitry has stopped working. Otherwise, the TM57PT20B works in the auto-pump-mode. It turns on Voltage Pump when V_{DD}<2.7V, turns off Voltage Pump when V_{DD}>2.7V.

MODE3V: (R0E.2, Default = 0)

This bit enables the TM57PT20B to work in the extremely high clock speed and/or low voltage (V_{DD}=1.1V) environment. When MODE3V is set, the TM57PT20B continuously turns on the Voltage Pump circuitry no matter V_{DD}>2.7V or V_{DD}<2.7V. So that it is suggested enable this mode when the operating voltage range covers 2.7V.

Warning: User must set MODE3V = 0 when V_{DD} > 3.2V

VDDFLT: (R0E.6, Default = 0)

If this bit is "1", the TM57PT20B turns on the power noise filter circuitry to enhance the chip's power noise immunity. The LVD flag is disabled in such setting.

The following table shows the relationship of operation voltage and system clock.

Fsys Type	Frequency or Option	NOPUMP = 0		NOPUMP = 1
		MODE3V=1	MODE3V=0	MODE3V = 0 or 1
		PUMP always ON	auto-pump-mode	PUMP always OFF
FXT	4 MHz	1.5V ~ 3.2V	1.5V ~ 5.5V	1.7V ~ 5.5V
	8 MHz	1.8V ~ 3.2V	1.8V ~ 5.5V	2.0V ~ 5.5V
	12 MHz	2.1V ~ 3.2V	2.1V ~ 5.5V	2.4V ~ 5.5V
	16 MHz	2.3V ~ 3.2V	2.3V ~ 5.5V	2.7V ~ 5.5V
	20 MHz	2.6V ~ 3.2V	2.6V ~ 5.5V	3.1V ~ 5.5V
	24 MHz	2.9V ~ 3.2V	3.5V ~ 5.5V	3.5V ~ 5.5V
FIRC*	0.5 MHz (CPUPSC=00)	1.2V ~ 3.2V	1.2V ~ 5.5V	1.5V ~ 5.5V
	2 MHz (CPUPSC=01)	1.3V ~ 3.2V	1.3V ~ 5.5V	1.5V ~ 5.5V
	4 MHz (CPUPSC=10)	1.4V ~ 3.2V	1.4V ~ 5.5V	1.7V ~ 5.5V
	8 MHz (CPUPSC=11)	1.8V ~ 3.2V	1.8V ~ 5.5V	2.0V ~ 5.5V
SXT	32768 Hz	1.4V ~ 3.2V	1.4V ~ 5.5V	1.5V ~ 5.5V
SIRC*	6785 Hz (CPUPSC=00)	1.1V ~ 3.2V	1.1V ~ 5.5V	1.5V ~ 5.5V
	27.5 KHz (CPUPSC=01)	1.1V ~ 3.2V	1.1V ~ 5.5V	1.5V ~ 5.5V
	55 KHz (CPUPSC=10)	1.1V ~ 3.2V	1.1V ~ 5.5V	1.5V ~ 5.5V
	110 KHz (CPUPSC=11)	1.1V ~ 3.2V	1.1V ~ 5.5V	1.5V ~ 5.5V

Note: FIRC and SIRC are very low accuracy when operating at low voltage.

The TM57PT20B starts at the Slow-clock mode after power on or reset. It can be switched to Fast-clock mode as long as the supply voltage is within related operating voltage range.

R0E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0E	–	VDDFLT	–	–	NOPUMP	MODE3V	–	–
R/W	–	W	–	–	W	W	–	–
Reset	–	0	–	–	0	0	–	–

R0E.6 **VDDFLT:** Power noise filter
0: disable
1: enable

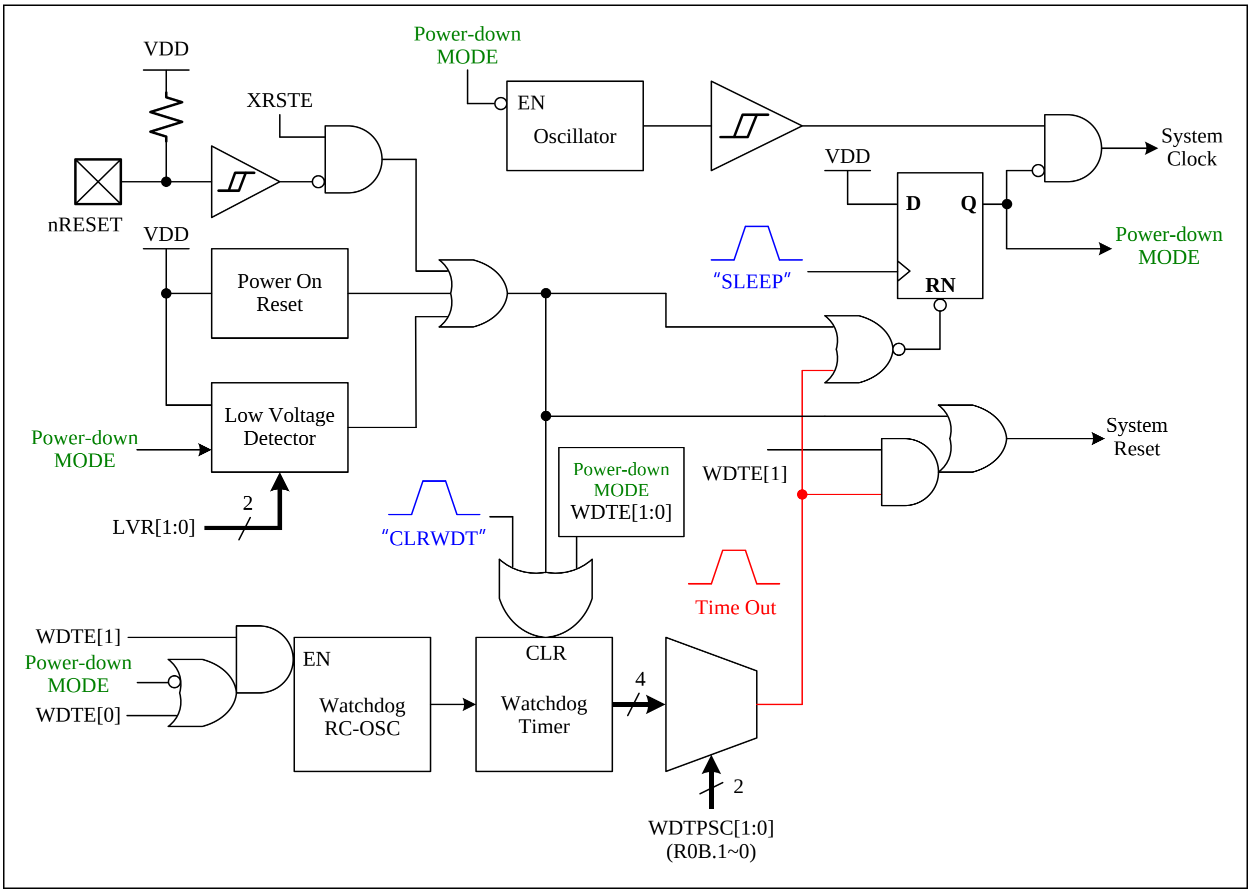
R0E.3 **NOPUMP:** Voltage PUMP control
0: enable auto-pump-mode or PUMP always ON
1: disable voltage pump

R0E.2 **MODE3V**: MODE 3V control
0: disable
1: enable

3. Peripheral Functional Block

3.1 Watchdog (WDT) Timer

The WDT clock source is internal RC Timer. It is enabled by setting the WDTE[1:0] (SYSCFG[6:5]). The overflow period of WDT can be selected from 19 ms to 192 ms. The WDT timer is cleared by the CLRWDT instruction. The WDT works in both normal (SLOW and FAST mode) mode and IDLE mode. In normal mode, the WDT is enabled by setting WDTE[1], no matter WDTE[0] is set or cleared. In other words, the internal RC Timer stops for power saving when WDTE[1] is cleared. In IDLE mode, the WDT is only enabled when WDTE[1] and WDTE[0] are both set. Otherwise it will be disabled and stopped for power saving. Refer to the following table and figure.



WDT Block Diagram

The WDT and WKT's behavior in different Mode are shown as below table.

Mode	WDTE[1:0]		Watchdog RC Oscillator
Normal Mode	0	0	Stop
	0	1	
	1	0	Run
	1	1	
Power-down Mode	0	0	Stop
	0	1	
	1	0	
	1	1	Run

F03	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	LVD	GB0	RAMBK	TO	PD	Z	DC	C
R/W	R	R/W	R/W	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F03.4 **TO:** WDT time out flag, read-only

0: after Power On Reset, LVR Reset, or CLRWDT/SLEEP instructions

1: WDT time out occurs

R04	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTCLR	WDTCLR							
R/W	W							
Reset	–	–	–	–	–	–	–	–

R04.7~0 **WDTCLR:** Write this register to clear WDT

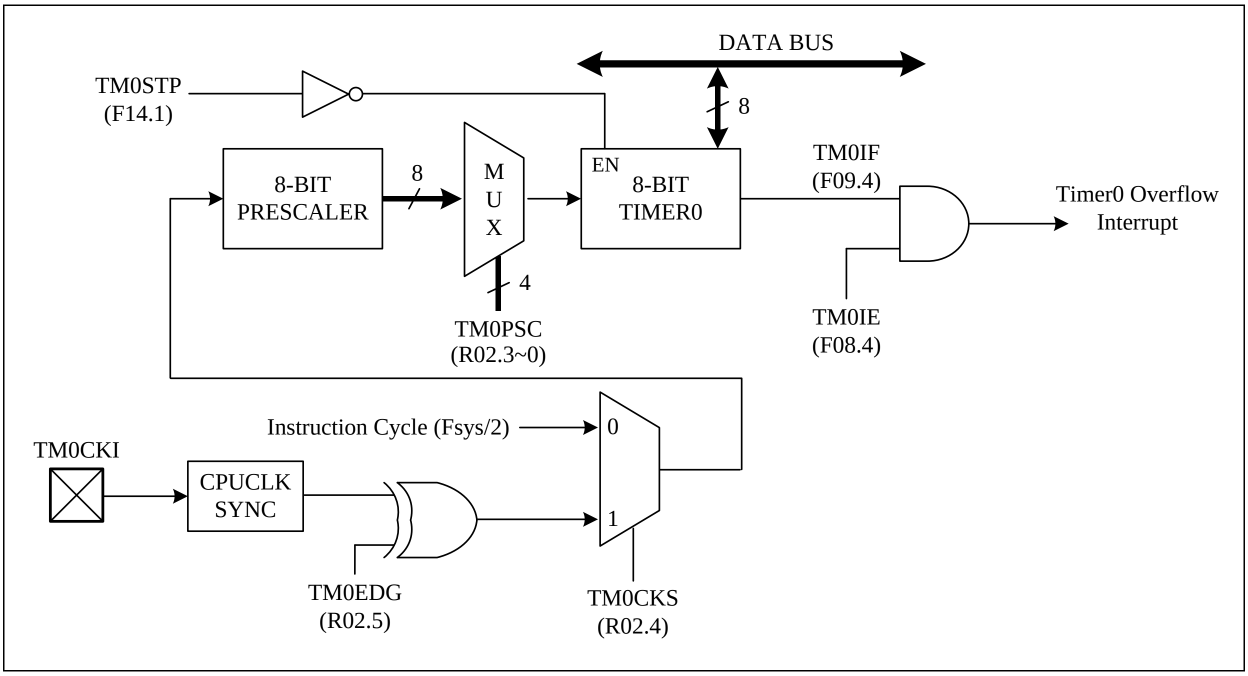
R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	–	T2PSC		INT0EDG	TCOE	–	WDTPSC	
R/W	–	W		W	W	–	W	
Reset	–	0	0	0	0	–	1	1

R0B.1~0 **WDTPSC:** WDT pre-scale select:

Bit 1	Bit 0	5V	3V
0	0	19 ms	24 ms
0	1	38 ms	48 ms
1	0	76 ms	96 ms
1	1	152 ms	192 ms

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

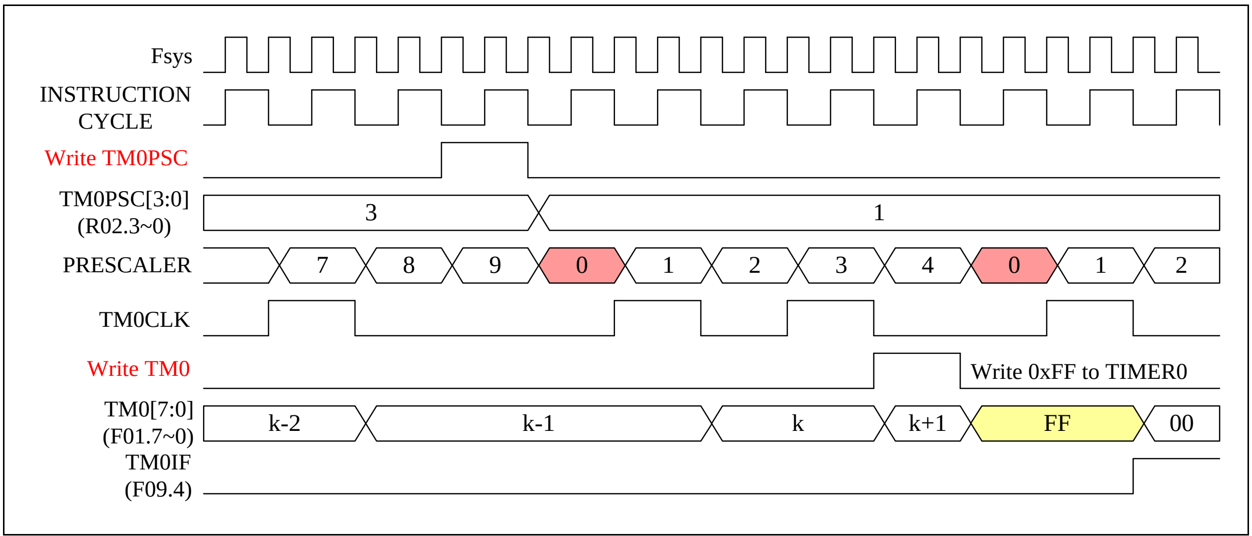
The Timer0 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatically rolls over based on the pre-scaled clock source, which can be the instruction cycle or TM0CKI (PA2) rising/falling input. The Timer0's increasing rate is determined by the TM0PSC[3:0] (R02.3~0). The Timer0 can generate interrupt flag TM0IF (F09.4) when it rolls over. It generates Timer0 interrupt if the TM0IE (F08.4) bit is set. Timer0 can be stopped counting if the TM0STP (F14.1) bit is set.



Timer0 Block Diagram

Timer Mode:

When the Timer0 prescaler (TM0PSC) is written, the internal 8-bit prescaler will be cleared to 0 to make the counting period correct at the first Timer0 count. TM0CLK is the internal signal that causes the Timer0 to increase by 1 at the end of TM0CLK. TM0WR is also the internal signal that indicates the Timer0 is directly written by instruction; meanwhile, the internal 8-bit prescaler will be cleared. When Timer0 counts from FFh to 00h, TM0IF (Timer0 Interrupt Flag) will be set to 1 and generate interrupt if TM0IE (Timer0 Interrupt Enable) is set. The following timing diagram describes the Timer0 works in pure Timer mode.



Timer0 works in Timer mode (TM0CKS = 0)

The equation of Timer0 interrupt timer value is as following:

Timer0 interrupt interval cycle time = Instruction cycle time / TM0PSC / 256

◇Example: Setup Timer0 work in Timer mode, Fsys = Fast-clock / CPUPSC = FXT 4MHz / 1 = 4MHz

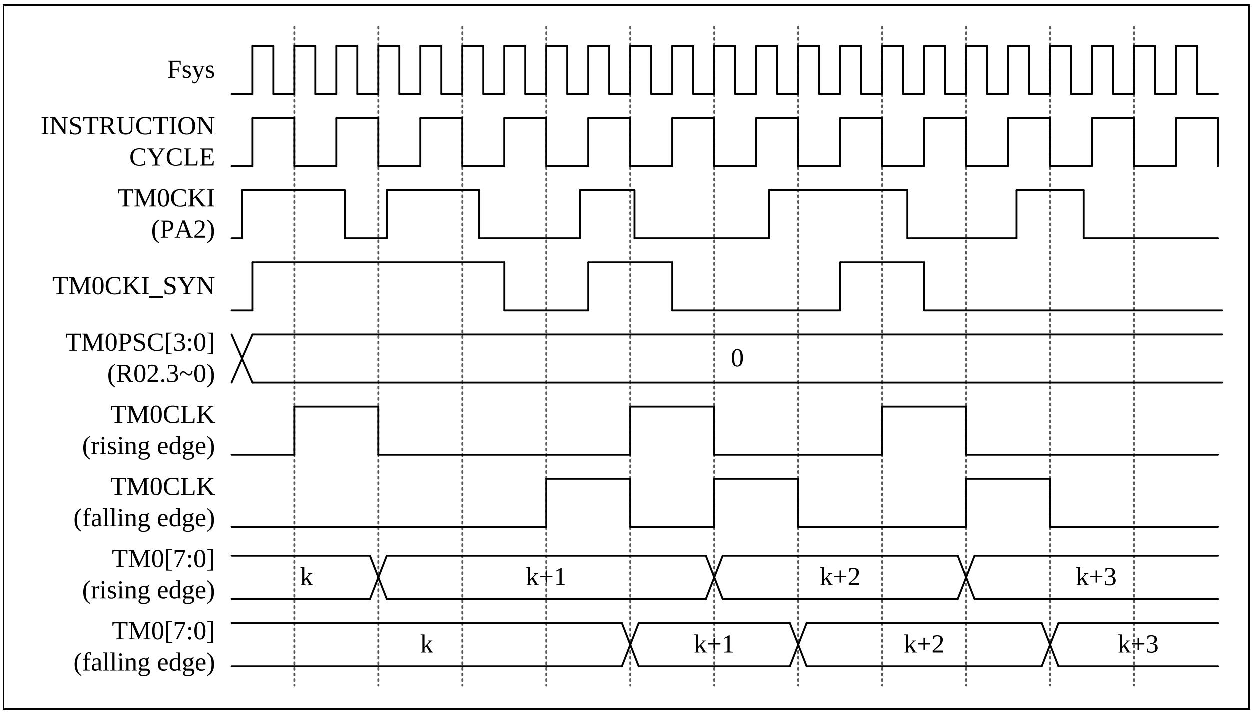
```
; Setup Timer0 clock source and divider
        MOVLW   00x00101B       ; TM0CKS = 0, Timer0 clock is instruction cycle
        MOVWR   TM0CTL          ; TM0PSC = 0101b, divided by 32

; Setup Timer0
        BSF     TM0STP          ; Timer0 stops counting
        CLRF    TM0             ; Clear Timer0 content

; Enable Timer0 and interrupt function
        MOVLW   11101111B
        MOVWF   INTIF           ; Clear Timer0 request interrupt flag
        BSF     TM0IE           ; Enable Timer0 interrupt function
        BCF     TM0STP          ; Enable Timer0 counting
```

Timer0 clock source is Fsys/2 = 4 MHz / 2 = 2 MHz, Timer0 divided by 32

Timer0 interrupt frequency = 2 MHz / 32 / 256 = 244.14 Hz

Counter Mode:

If TM0CKS = 1, then Timer0 counter source clock is from TM0CKI (PA2) pin. TM0CKI signal is synchronized by instruction cycle that means the high/low time durations of TM0CKI must be longer than one instruction cycle time to guarantee each TM0CKI's change will be detected correctly by the synchronizer. The following timing diagram describes the Timer0 works in Counter mode.



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

◇Example: Setup Timer0 works in Counter mode

```
; Setup Timer0 clock source and divider
          MOVLW     00110000B          ; TM0EDG = 1, counting edge is falling edge
          MOVWR     TM0CTL             ; TM0CKS = 1, Timer0 clock is TM0CKI (PA2)
                                       ; TM0PSC = 0000b, divided by 1

; Setup Timer0
          BSF       TM0STP             ; Timer0 stops counting
          CLRF      TM0                ; Clear Timer0 content

; Enable Timer0 and read Timer0 counter
          BCF       TM0STP             ; Enable Timer0 counting
          …
          BSF       TM0STP             ; Timer0 stops counting
          MOVFW     TM0                ; Read Timer0 content
```

F01	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0	TM0							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F01.7~0 **TM0:** Timer0 content

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	–	T2IE	CMPIE	TM0IE	–	INT2IE	INT1IE	INT0IE
R/W	–	R/W	R/W	R/W	–	R/W	R/W	R/W
Reset	–	0	0	0	–	0	0	0

F08.4 **TM0IE**: Timer0 interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	–	T2IF	CMPIF	TM0IF	–	INT2IF	INT1IF	INT0IF
R/W	–	R/W	R/W	R/W	–	R/W	R/W	R/W
Reset	–	0	0	0	–	0	0	0

F09.4 **TM0IF**: Timer0 interrupt event pending flag
This bit is set by H/W while Timer0 overflows, write 0 to this bit will clear this flag

F14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF14	–	–	–	–	CMPST	T2CLR	TM0STP	PWM0CLR
R/W	–	–	–	–	R	R/W	R/W	R/W
Reset	–	–	–	–	0	0	0	1

F14.1 **TM0STP**: Timer0 counter stop
0: Timer0 is counting
1: Timer0 stops counting

R02	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0CTL	–	–	TM0EDG	TM0CKS	TM0PSC			
R/W	–	–	W	W	W			
Reset	–	–	0	0	0	0	0	0

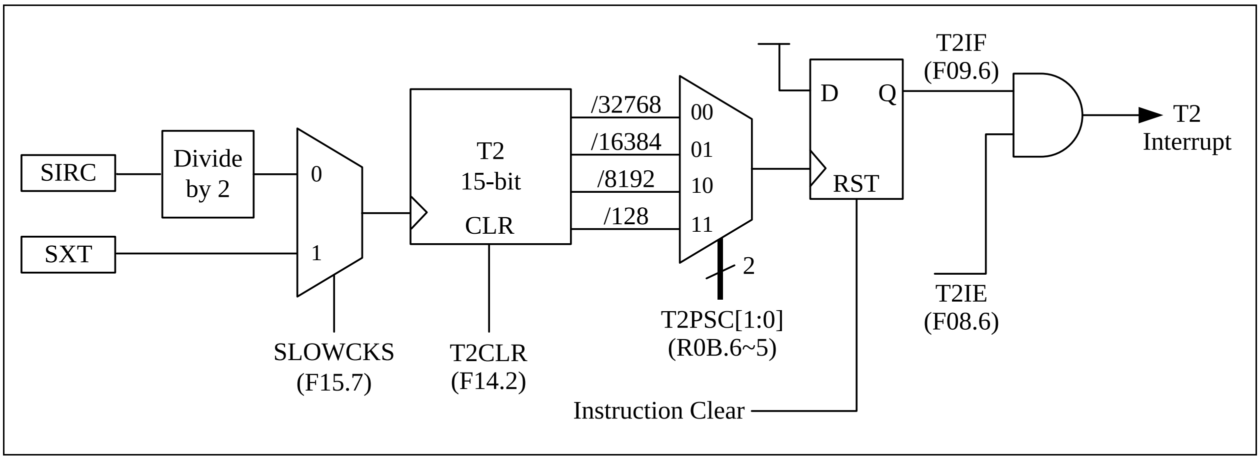
R02.5 **TM0EDG:** TM0CKI (PA2) edge selection for Timer0 prescaler count
0: TM0CKI (PA2) rising edge for Timer0 prescaler count
1: TM0CKI (PA2) falling edge for Timer0 prescaler count

R02.4 **TM0CKS:** Timer0 clock source select
0: Instruction Cycle (Fsys/2) as Timer0 prescaler clock
1: TM0CKI (PA2) as Timer0 prescaler clock

R02.3~0 **TM0PSC:** Timer0 prescaler. Timer0 clock source
0000: divided by 1
0001: divided by 2
0010: divided by 4
0011: divided by 8
0100: divided by 16
0101: divided by 32
0110: divided by 64
0111: divided by 128
1xxx: divided by 256

3.3 T2: 15-bit Timer

The T2 is a 15-bit counter and the clock sources are from either SIRC/2 or SXT. The clock source is used to generate time base interrupt and T2 counter block clock. It is selected by SLOWCKS (F15.7). The T2's 15-bit content cannot be read by instructions. It generates interrupt flag T2IF (F09.6) with the clock divided by 32768, 16384, 8192, or 128 depends on the T2PSC[1:0] (R0B.6~5) bits. The following figure shows the block diagram of T2.



T2 Block Diagram

◇Example: T2 clock source is SXT and divided by 32768

```
; Setup T2 clock source and divider
          BSF       SLOWCKS          ; SLOWCKS=1, T2 clock source is SXT
          MOVLW     000xx0xxB        ; T2PSC = 00b, divided by 32768
          MOVWR     R0B              ;
          BSF       T2CLR            ; T2CLR = 1, clear T2 counter

; Enable T2 interrupt function
          MOVLW     10111111B
          MOVWF     INTIF            ; Clear T2 request interrupt flag
          BSF       T2IE             ; Enable T2 interrupt function
```

T2 clock source is Slow-clock = 32768 Hz, T2 divided by 32768

T2 interrupt frequency = 32768 Hz / 32768 = 1 Hz

T2 interrupt period = 1 / 1 Hz = 1s

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	–	T2IE	CMPIE	TM0IE	–	INT2IE	INT1IE	INT0IE
R/W	–	R/W	R/W	R/W	–	R/W	R/W	R/W
Reset	–	0	0	0	–	0	0	0

F08.6 **T2IE**: T2 interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	–	T2IF	CMPIF	TM0IF	–	INT2IF	INT1IF	INT0IF
R/W	–	R/W	R/W	R/W	–	R/W	R/W	R/W
Reset	–	0	0	0	–	0	0	0

F09.6 **T2IF**: T2 interrupt event pending flag
This bit is set by H/W while T2 overflows, write 0 to this bit will clear this flag

F14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF14	–	–	–	–	CMPST	T2CLR	TM0STP	PWM0CLR
R/W	–	–	–	–	R	R/W	R/W	R/W
Reset	–	–	–	–	0	0	0	1

F14.2 **T2CLR**: T2 counter clear
0: T2 is counting
1: T2 is cleared immediately, this bit is auto cleared by H/W

F15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCTL	SLOWCKS	FASTCKS	GB1	SLOWSTP	FASTSTP	CPUCKS	CPUPSC	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	1	1

F15.7 **SLOWCKS**: Slow-clock type select or T2 clock source select
For Slow-clock type
0: SIRC
1: SXT
For T2 clock source
0: SIRC/2
1: SXT

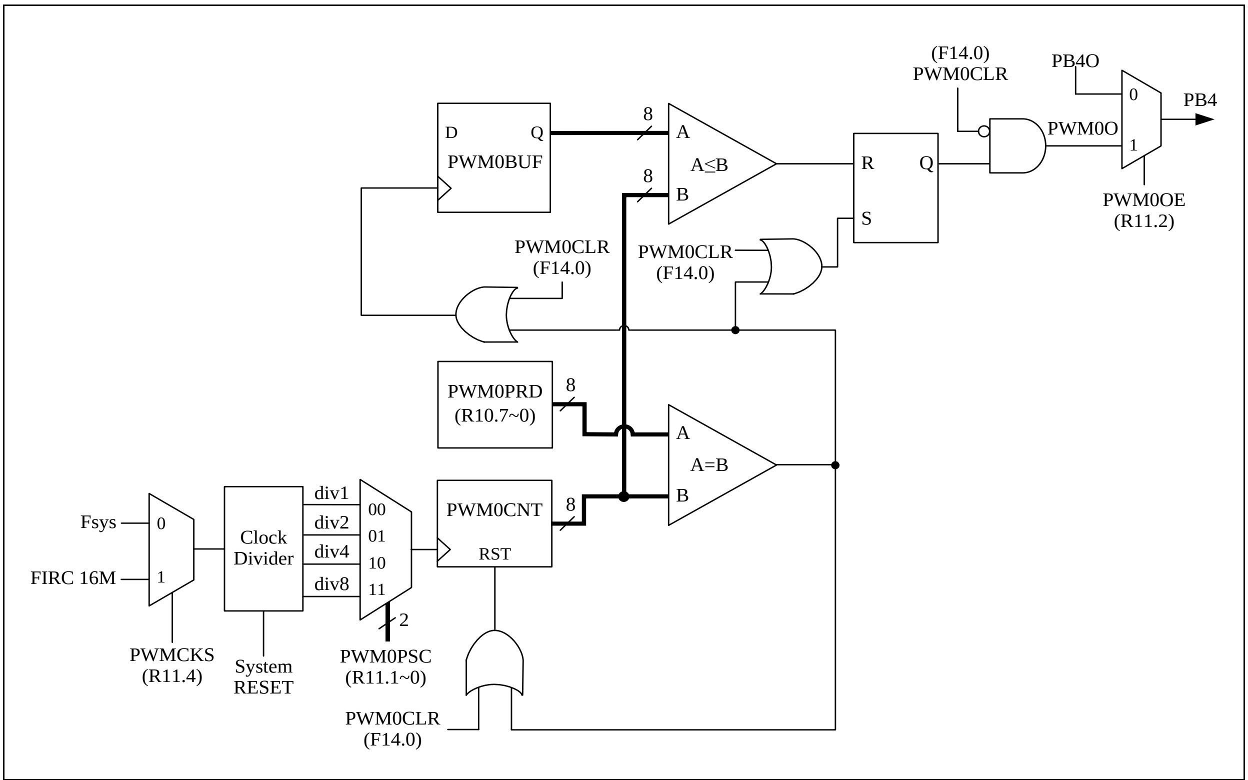
R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	–	T2PSC		INT0EDG	TCOE	–	WDTPSC	
R/W	–	W		W	W	–	W	
Reset	–	0	0	0	0	–	1	1

R0B.6~5 **T2PSC:** T2 prescaler. T2 clock source
00: divided by 32768
01: divided by 16384
10: divided by 8192
11: divided by 128

3.4 PWM0: 8-bit PWM

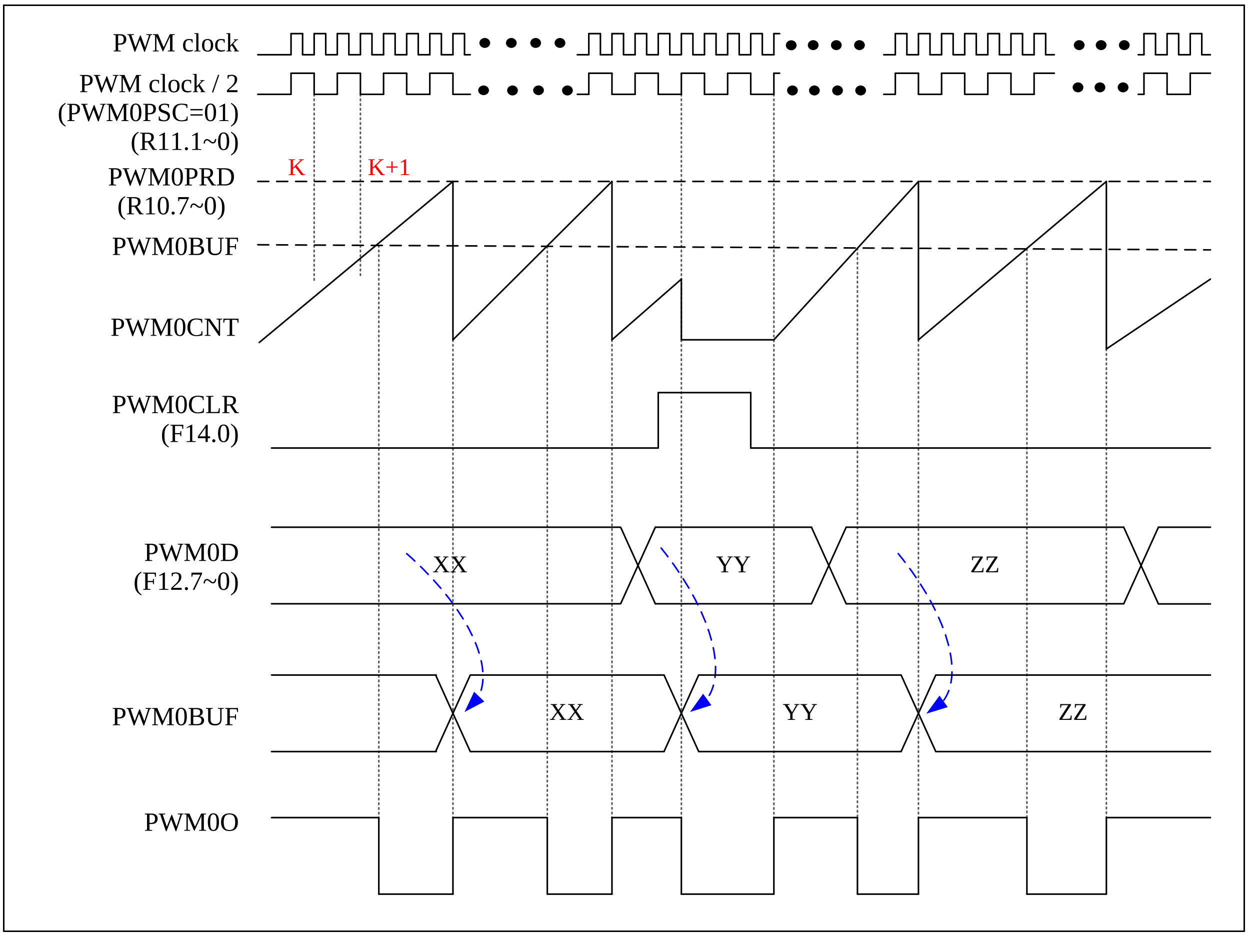
TM57PT20B has three built-in 8-bit PWM generators. There are PWM0, PWM1 and PWM2. All of them use the same clock source. The PWM clock source can be chosen by PWMCKS (R11.4) bit. If PWMCKS bit is set, the PWM clock source is FIRC 16 MHz, otherwise is system clock (Fsys). And it also can be divided by 1, 2, 4, and 8 according to PWM0PSC (R11.1~0). The PWM0 duty cycle can be changed with writing to PWM0D (F12.7~0). Writing to PWM0D will not change the current PWM0 duty until the current PWM0 period completes. When current PWM0 period is finish, the new value of PWM0D will be updated to the PWM0BUF.

The PWM0 will output to PB4 if PWM0OE (R11.2) is set. With I/O mode setting, the PWM0 output can be set as CMOS push-pull output mode or open-drain output mode. When PBMODH[1] (R07.1) is set and PBMODH[0] (R07.0) is cleared, the PB4 output is CMOS push-pull output mode. When PBMODH[1] is cleared, the PB4 output is open-drain output mode. Setting the PWM0CLR (F14.0) bit will clear the PWM0 counter and load the PWM0D to PWM0BUF, PWM0CLR bit must be cleared so that the PWM0 counter can count. Figure shows the block diagram of PWM0.



PWM0 Block Diagram

Figure shows the PWM0 waveforms. When PWM0CLR (F14.0) bit is set or PWM0BUF equals to zero, the PWM0 output is cleared to ‘0’ no matter what its current status is. Once the PWM0CLR bit is cleared and PWM0BUF is not zero, the PWM0 output is set to ‘1’ to begin a new PWM cycle. PWM0 output will be ‘0’ when PWM0CNT is greater than or equals to PWM0BUF. PWM0CNT keeps counting up when equals to PWM0PRD (R10.7~0), the PWM0 output is set to ‘1’ again.



PWM0 Timing Diagram

◇Example: CPU is running at FAST mode, Fsys = Fast-clock / CPUPSC = FXT 4 MHz / 1 = 4 MHz

```
; Setup PWM0 prescaler, period, and duty
        BSF      PWM0CLR        ; PWM0CLR = 1, PWM0 clear and hold
        MOVLW    00000101B      ; PWMCKS = 0, PWM-clock source is Fsys
        MOVWR    PWMCTL         ; PWM0OE = 1, PWM0 output to PB4 pin
                                ; PWM0PSC = 01b, divided by 2
        MOVLW    FFH
        MOVWR    PWM0PRD        ; Set PWM0 period = FFH + 1 = 256

        MOVLW    80H
        MOVWF    PWM0D          ; Set PWM0 duty = 80H = 128
        BCF      PWM0CLR        ; PWM0CLR = 0, PWM0 is running
```

PWM0 output duty = PWM0D / (PWM0PRD + 1) = 128 / (255 + 1) = 1 / 2

PWM clock = Fsys = 4 MHz, PWM clock divided by 2

PWM0 output frequency = 4 MHz / 2 / (255 + 1) = 7812.5 Hz

F12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0D	PWM0D							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F12.7~0 **PWM0D**: PWM0 duty

F14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF14	–	–	–	–	CMPST	T2CLR	TM0STP	PWM0CLR
R/W	–	–	–	–	R	R/W	R/W	R/W
Reset	–	–	–	–	0	0	0	1

F14.0 **PWM0CLR**: PWM0 clear and hold
0: PWM0 is running
1: PWM0 is clear and hold

R10	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0PRD	PWM0PRD							
R/W	W							
Reset	1	1	1	1	1	1	1	1

R10.7~0 **PWM0PRD**: PWM0 period data

R11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMWCTL	–	–	PWM2OE	PWMCKS	PWM01E	PWM0OE	PWM0PSC	
R/W	–	–	W	W	W	W	W	
Reset	–	–	0	0	0	0	0	0

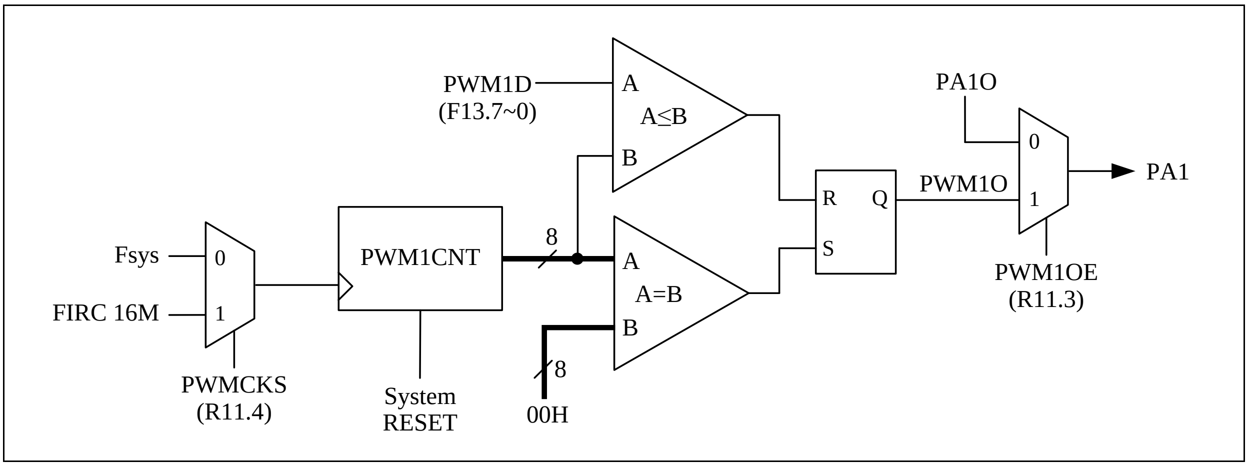
R11.4 **PWMCKS**: PWM Clock source select
0: System clock (Fsys)
1: FIRC 16 MHz

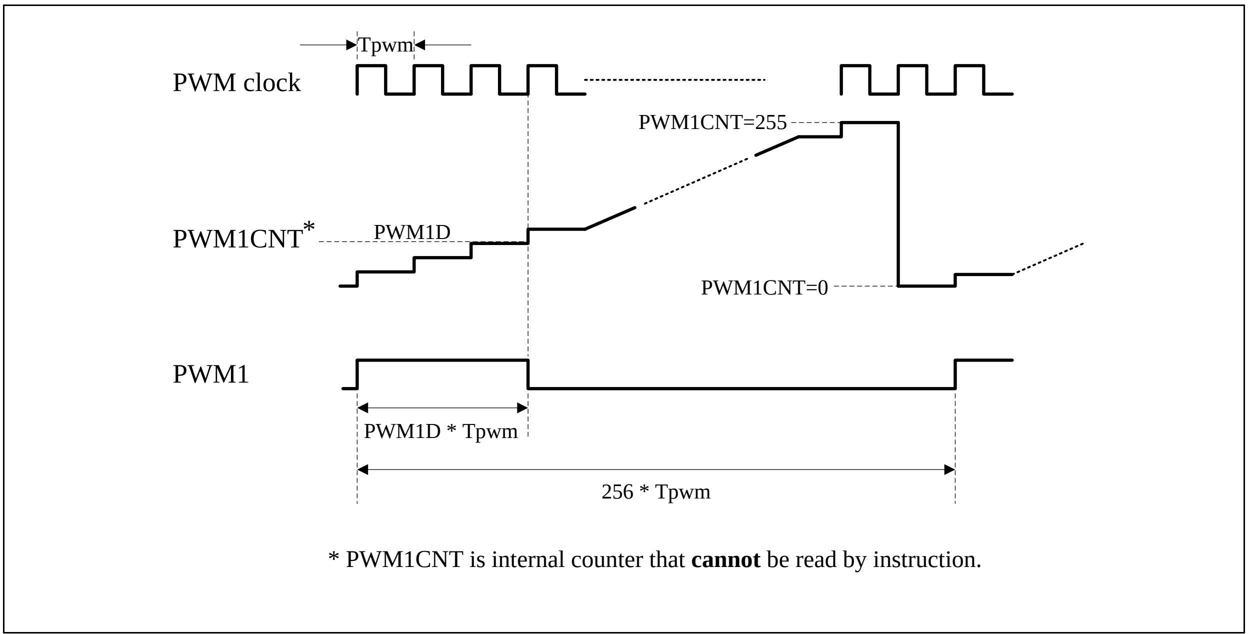
R11.2 **PWM0OE**: PWM0 positive output to PB4 pin
0: disable
1: enable

R11.1~0 **PWM0PSC**: PWM0 prescaler, PWM0 clock source
00: divided by 1
01: divided by 2
10: divided by 4
11: divided by 8

3.5 PWM1: 8-bit PWM

PWM1 is a simple fixed frequency and duty cycle variable PWM generator. System clock (Fsys) and FIRC Clock (16 MHz) can be selected as the PWM clock by PWMCKS (R11.4) bit. The PWM frequency is fixed, the period is PWM clock counts from 0 to 255. The duty can be set via PWM1D (F13.7~0). The output of PWM1 shares the pin PA1 that can be selected by PWM1OE (R11.3) control bit. Figure is the block diagram of PWM1.





PWM1 output duty = [PWM1D / 256]

When PWM1D = 80H, PWM1 output duty will be 1/2

PWM1 output frequency = PWM clock / 256

When PWM clock = FIRC 16 MHz, PWM1 output frequency = 16 MHz / 256 = 62.5 KHz

F13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1D	PWM1D							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F13.7~0 **PWM1D**: PWM1 duty

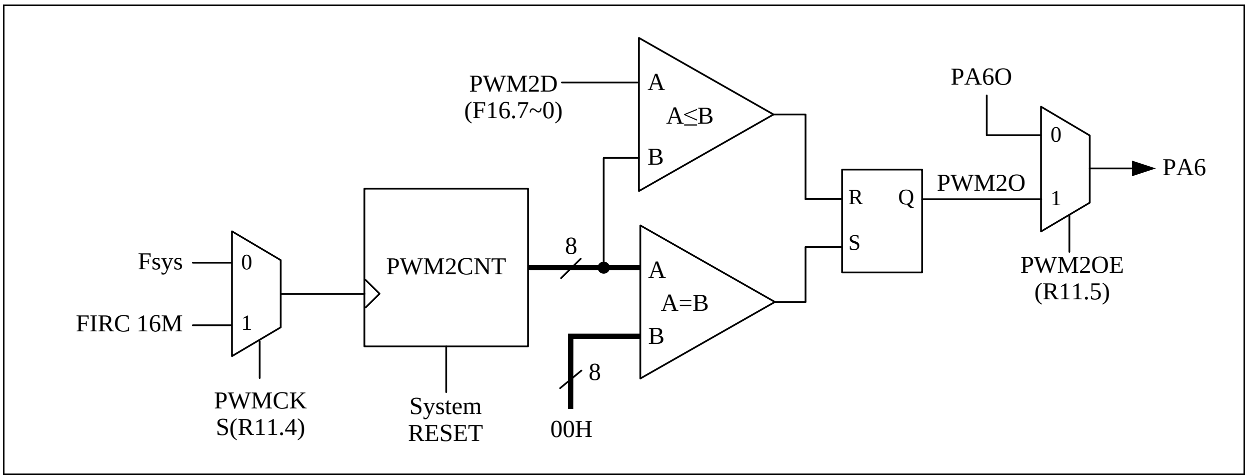
R11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMWCTL	–	–	PWM2OE	PWMCKS	PWM1OE	PWM0OE	PWM0PSC	
R/W	–	–	W	W	W	W	W	
Reset	–	–	0	0	0	0	0	0

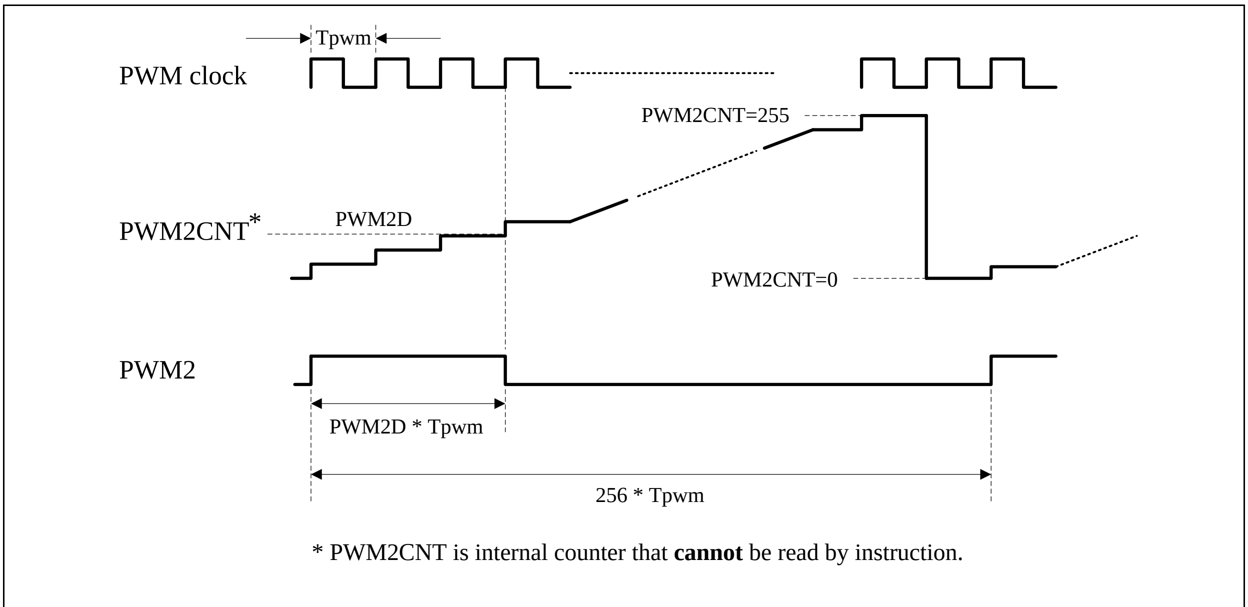
R11.4 **PWMCKS**: PWM Clock source select
0: System clock (Fsys)
1: FIRC 16 MHz

R11.3 **PWM1OE**: PWM1 positive output to PA1 pin
0: disable
1: enable

3.6 PWM2: 8-bit PWM

PWM2 is a simple fixed frequency and duty cycle variable PWM generator. The structure is the same as PWM1. Please refer to chapter 3.5 to see more details for PWM2 setting.





PWM2 output duty = [PWM2D / 256]

When PWM2D = 80H, PWM2 output duty will be 1/2

PWM2 output frequency = PWM clock / 256

When PWM clock = FIRC 16 MHz, PWM2 output frequency = 16 MHz / 256 = 62.5 KHz

F16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2D	PWM2D							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F16.7~0 **PWM2D**: PWM2 duty

R11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMWCTL	–	–	PWM2OE	PWMCKS	PWM1OE	PWM0OE	PWM0PSC	
R/W	–	–	W	W	W	W	W	
Reset	–	–	0	0	0	0	0	0

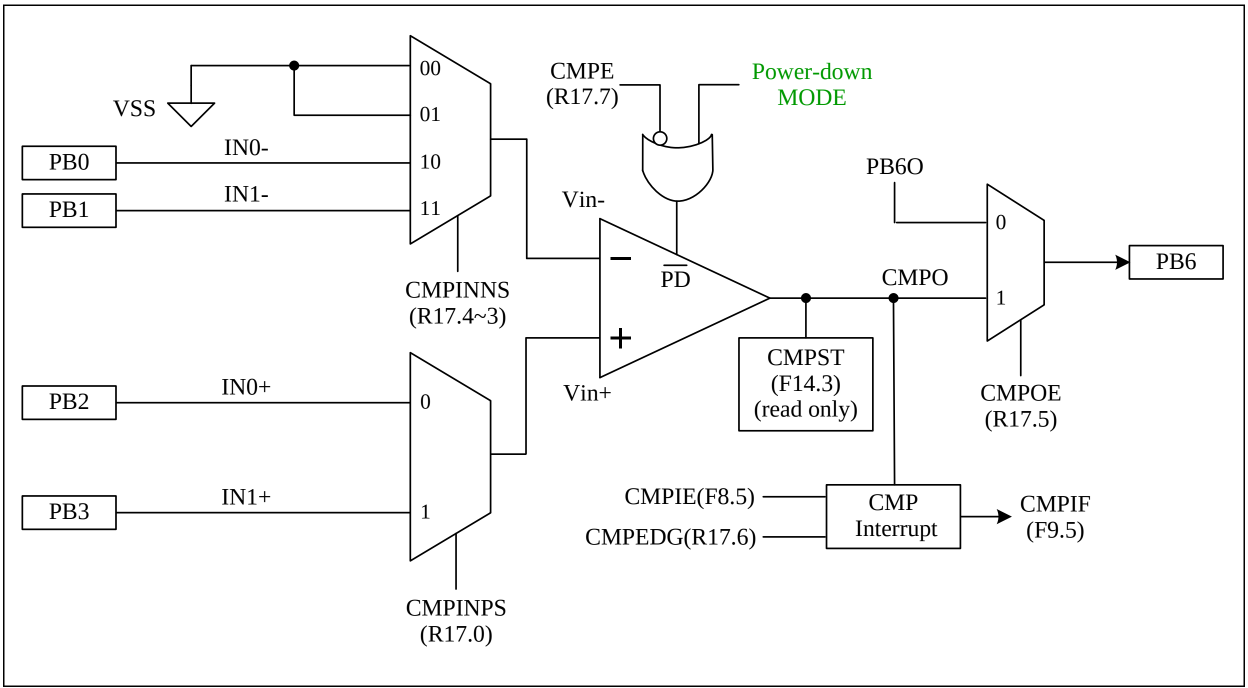
R11.5 **PWM2OE**: PWM2 positive output to PA6 pin
0: disable
1: enable

R11.4 **PWMCKS**: PWM Clock source select
0: System clock (Fsys)
1: FIRC 16 MHz

3.7 Analog Comparator

TM57PT20B includes an analog comparator. It can be enabled by CMPE (R17.7) in normal mode (SLOW and FAST mode). The analog comparator has four analog inputs (IN0-, IN1-, IN0+ and IN1+) and one digital output (CMPO). The input source of negative pin can be selected from VSS, IN0- or IN1- by CMPINNS (R17.4~3), and the input source of positive pin can be selected from IN0+ or IN1+ by CMPINPS (R17.0) bit. The analog comparator compares the input values on the positive pin Vin+ and negative pin Vin-. When the voltage on positive pin is higher than the voltage on negative pin, the analog comparator output (CMPO) is set. The output status can not only be read from CMPST (F14.3) bit, but also output to PB6 pin by setting CMPOE (R17.5) bit. The comparator output can be set as CMOS push-pull output mode or open-drain output mode. When PBMODH[5] (R07.5) is set and PBMODH[4] (R07.4) is cleared, the PB6 output is CMOS push-pull output mode. When PBMODH[5] is cleared, the PB6 output is open-drain output mode.

The analog comparator can generate interrupt flag CMPIF (F9.5) when the output status rising or falling. The comparator interrupt can be enabled by CMPIE (F8.5) bit, and the interrupt trigger edge can be selected by CMPEDG (R17.6) bit. A block diagram of the analog comparator is shown below.



◇Example: Compare channel IN0- (input: 2V) and channel IN0+ (input: 4V)

MOVLW	xx**<u>10</u>**xxxxB	; PBMODL[5:4] = 10B
MOVWR	PBMODH	; Set PB6 for comparator output
MOVLW	xx**<u>11</u>**xx**<u>11</u>**B	; PBMODL[5:4] = 11B, PBMODL[1:0] = 11B
MOVWR	PBMODL	; Set PB0 as IN0- for comparator analog input
		; Set PB2 as IN0+ for comparator analog input
MOVLW	**<u>1</u>**0**<u>1</u>** **<u>10</u>**00**<u>0</u>**B	; Channel select: IN0- vs. IN0+
MOVWR	CMPCTL	; comparator enable, comparator output enable

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	–	T2IE	CMPIE	TM0IE	–	INT2IE	INT1IE	INT0IE
R/W	–	R/W	R/W	R/W	–	R/W	R/W	R/W
Reset	–	0	0	0	–	0	0	0

F08.5 **CMPIE**: Comparator interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	–	T2IF	CMPIF	TM0IF	–	INT2IF	INT1IF	INT0IF
R/W	–	R/W	R/W	R/W	–	R/W	R/W	R/W
Reset	–	0	0	0	–	0	0	0

F09.5 **CMPIF**: Comparator interrupt event pending flag
Set by H/W at Comparator output falling/rising edge, write 0 to this bit will clear this flag

F14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF14	–	–	–	–	CMPST	T2CLR	TM0STP	PWM0CLR
R/W	–	–	–	–	R	R/W	R/W	R/W
Reset	–	–	–	–	0	0	0	1

F14.3 **CMPST**: Comparator output state

R17	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMPCTL	CMPE	CMPEDG	CMPOE	CMPINNS		–	–	CMPINPS
R/W	W	W	W	W		–	–	W
Reset	0	0	0	0	0	–	–	0

R17.7 **CMPE**: Comparator enable
0: disable
1: enable

R17.6 **CMPEDG**: Comparator interrupt edge
0: falling edge
1: rising edge

R17.5 **CMPOE**: Comparator output to pin enable
0: disable
1: enable

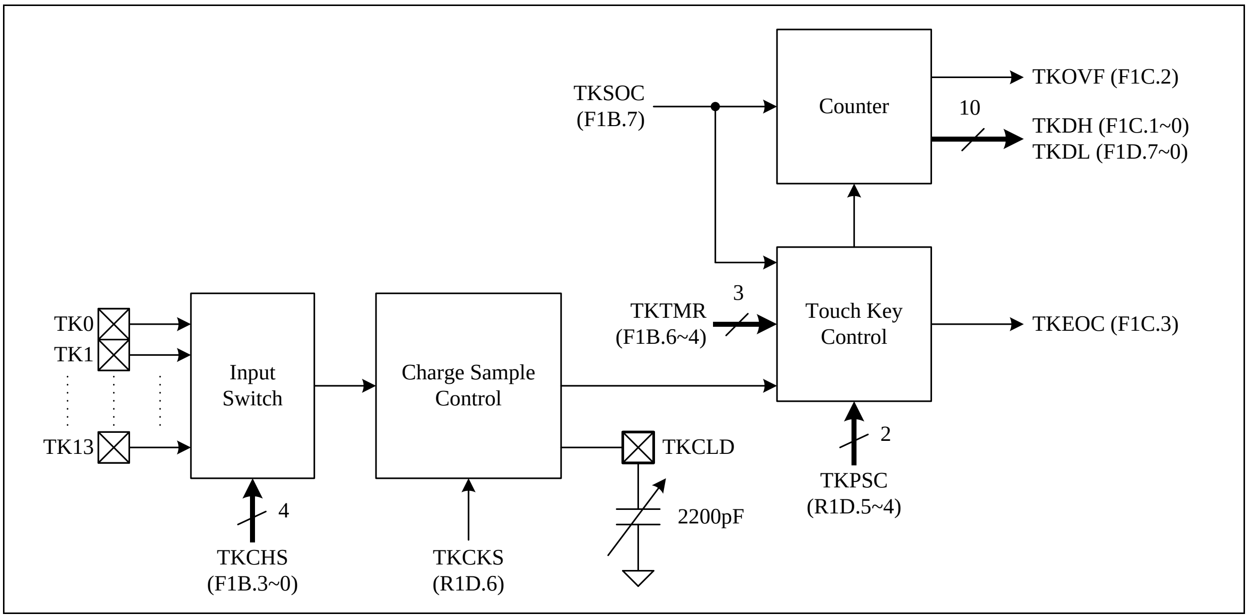
R17.4~3 **CMPINNS**: Comparator negative input source select
0x: VSS
10: IN0-
11: IN1-

R17.0 **CMPINPS**: Comparator positive input source select
0: IN0+
1: IN1+

3.8 Touch Key

The Touch Key offers an easy, simple and reliable method to implement finger touch applications. For most applications, only requires an external capacitor component on TKCLD pin. The TKCKS default is 4 MHz is sufficient for general touch plane.

Setting the TKSOC (F1B.7) bit to start touch key conversion, the TKSOC bit has to be cleared manually. "TKEOC=0" means conversion is in process, while "TKEOC=1" means the conversion is finish. After TKEOC's (F1C.3) edge rising, user must wait at least 10 us for next conversion. The touch key counting value is stored into TKDATA[9:0] (TKDH, TKDL). If TKOVF=1, it means the conversion has exceeded in period time, reduce TKTMR (F1B.6~4) or increase TKPSC (R1D.5~4) to fit the range of TKDATA[9:0]. On the other hand, if TKOVF=0, but TKDATA[9:0] is too small, increase TKTMR or reduce TKPSC to adapting the system board circumstances. The more detailed information, refer to touch key application note.



Touch Key Block Diagram

◇Example: Touch key channel = TK5 (PB3).

```
            MOVLW    11xxxxxxB         ; PBMODL[7:6] = 11B
            MOVWR    PBMODL            ; Set PB3 for touch key input

            MOVLW    xxxx11xxB         ; PAMODH[3:2] = 11B
            MOVWR    PAMODH            ; Set PA5 as TKCLD for connecting capacitor

            MOVLW    0100 0101B
            MOVWF    TKCTL1            ; TKTMR=4, TKCHS=5 (TK5)

            MOVLW    0 1 000000B       ; TKPD=0
            MOVWR    TKCTL2            ; TKCKS=1 (4 MHz), TKPSC=div1 (4 MHz)
            :
            :
            BSF      TKSOC             ; touch key start conversion
            NOP
            NOP
            NOP
            BCF      TKSOC

 WAIT_TK:
            BTFSS    TKEOC             ; wait touch key conversion finish
            GOTO     WAIT_TK

            MOVFW    TKDH              ; read TKDATA[9:8]
            MOVFW    TKDL              ; read TKDATA[7:0]
```

F1B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCTL1	TKSOC	TKTMR			TKCHS			
R/W	R/W	R/W			R/W			
Reset	0	1	0	0	0	0	0	0

F1B.7 **TKSOC**: Touch key start of conversion, rising edge to start
H/W auto cleared while end of conversion

F1B.6~4 **TKTMR**: Touch key conversion time
000: shortest
…
111: longest

F1B.3~0 **TKCHS**: Touch key channel select
0000: TK0 (PA1)
0001: TK1 (PA2)
0010: TK2 (PB0)
0011: TK3 (PB1)
0100: TK4 (PB2)
0101: TK5 (PB3)
0110: TK6 (PB4)
0111: TK7 (PB5)
1000: TK8 (PB6)
1001: TK9 (PB7)
1010: TK10 (PA0)
1011: TK11 (PA6)
1100: TK12 (PD1)
1101: TK13 (PD0)

F1C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKDH	–	–	–	–	TKEOC	TKOVF	TKDH	
R/W	–	–	–	–	R	R	R	
Reset	–	–	–	–	1	0	0	0

F1C.3 **TKEOC**: Touch key end of conversion
0: conversion is in process
1: end of conversion

F1C.2 **TKOVF**: Touch key counter overflow flag
0: not overflow
1: overflow

F1C.1~0 **TKDH**: Touch key data MSB [9~8]

F1D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKDL	TKDL							
R/W	R							
Reset	0	0	0	0	0	0	0	0

F1C.7~0 **TKDL**: Touch key data LSB [7~0]

R1D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCTL2	TKPD	TKCKS	TKPSC		–	–	–	–
R/W	W	W	W		–	–	–	–
Reset	0	0	0	0	–	–	–	–

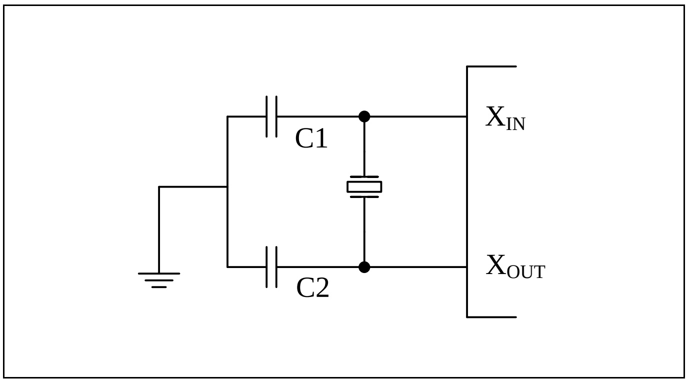
R1D.7 **TKPD**: Touch key power down
0: Touch key running
1: Touch key power down

R1D.6 **TKCKS**: Touch key clock select
0: 2 MHz
1: 4 MHz

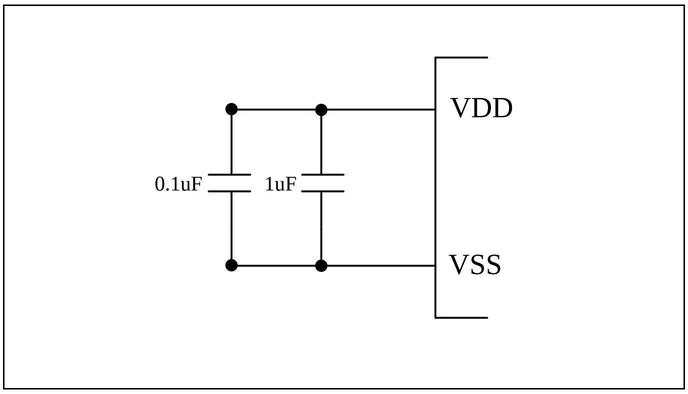
R1D.5~4 **TKPSC**: Touch key data prescaler, touch key data
00: divided by 1
01: divided by 2
10: divided by 4
11: divided by 8

3.9 System Clock Oscillator

System clock can be operated in four different oscillation modes. Four oscillation modes are FIRC, FXT, SIRC and SXT, respectively. In Fast/Slow Crystal mode (FXT/SXT), a crystal or ceramic resonator is connected to the Xin and Xout pins to establish oscillation. In the Fast Internal RC mode (FIRC), the on-chip oscillator generates 8 MHz system clock. Since power noise degrades the performance of Fast Internal Clock Oscillator, placing power supply bypass capacitors 1 uF and 0.1 uF very close to VDD/VSS pins to improve the stability of clock and the overall system. In the Slow Internal RC mode (SIRC), it provides a lower speed and accuracy of the oscillator for power saving purpose.



External Oscillator Circuit
(Crystal or Ceramic)

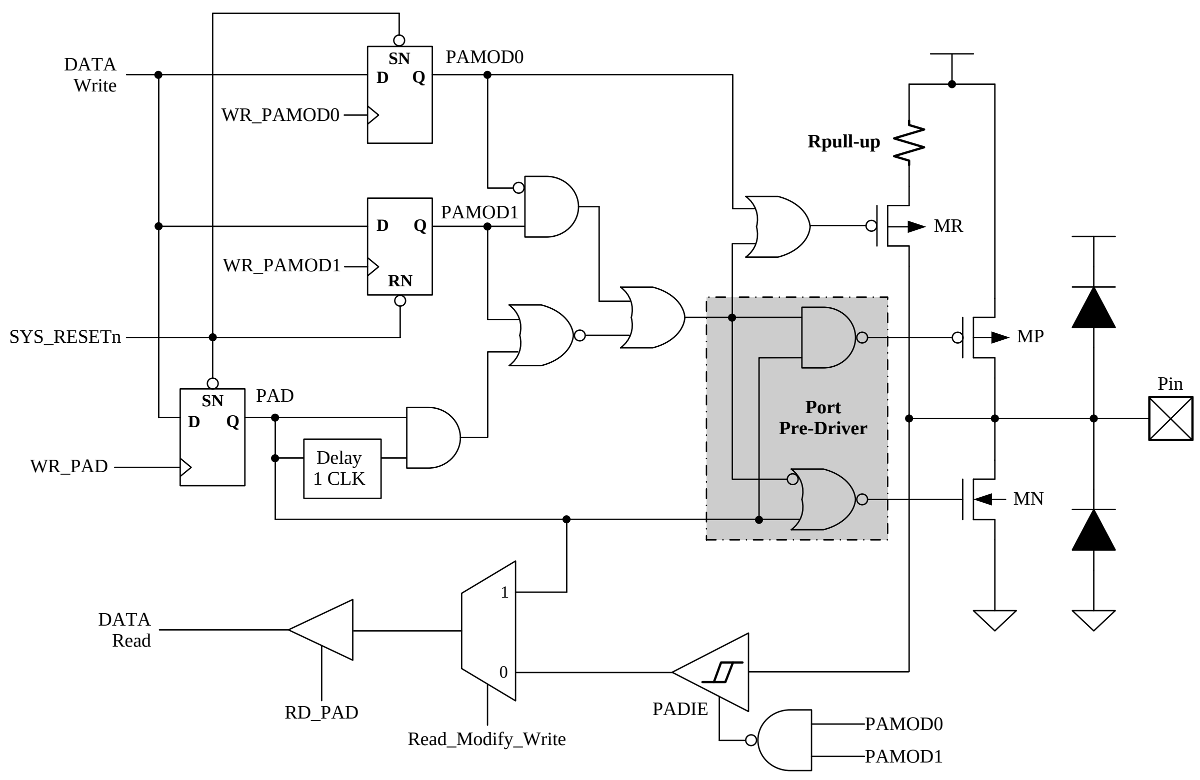


Fast Internal RC Mode

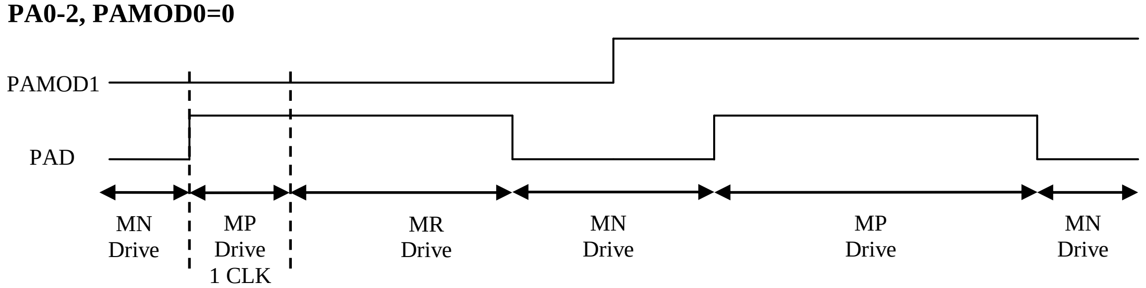
4. I/O Port

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 PAMOD1=0 and PAD=1. To use the pin in pseudo-open-drain mode, S/W set the PAMOD1=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 PAMOD1=1 and PAMOD0=0 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.



PA0-2, PAMOD0=0

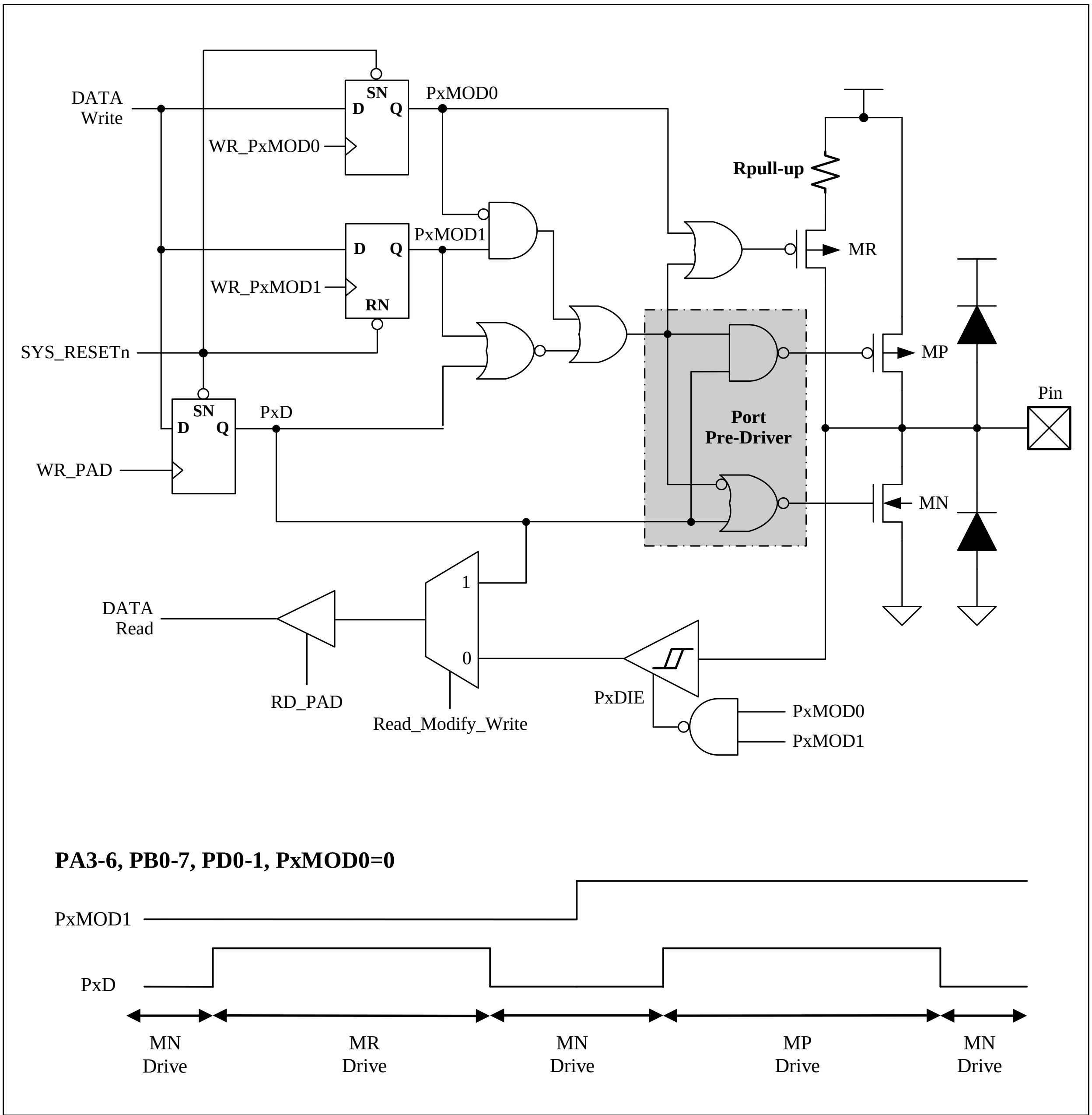


How to control PA0-2 status can be concluded as following list.

Register Setting			PIN STATE	Pull-up	Mode
PAMODE1	PAMODE0	PAD0-2			
0	0	0	Low	No	Pseudo-open-drain output
		1	High	Yes	Pseudo-open-drain output or Input with pull-high
0	1	0	Low	No	Pseudo-open-drain output
		1	Hi-Z	No	Pseudo-open-drain output or Input without pull-high
1	0	0	Low	No	CMOS push-pull output
		1	High	No	

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

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.

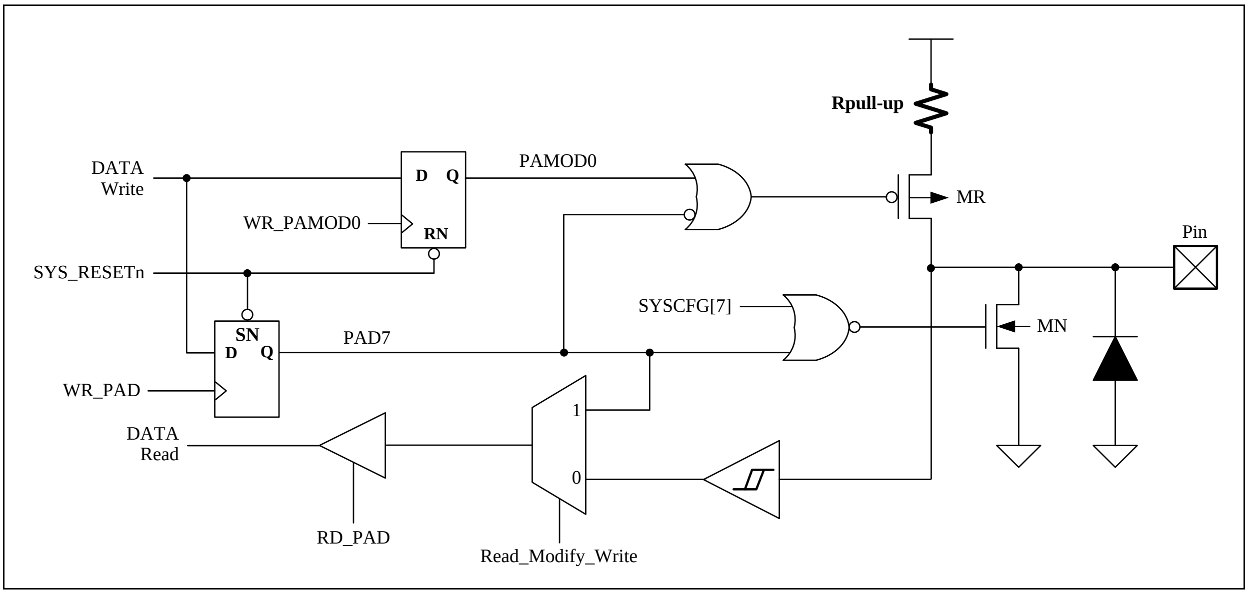


How to control PA3-6, PB0-7 and PD0-1 status can be concluded as following list.

Register Setting			PIN STATE	Pull-up	Mode
PxMODE1	**PxMODE0**	**PxD**			
0	0	0	Low	No	Open-drain output
		1	High	Yes	Open-drain output or Input with pull-high
0	1	0	Low	No	Open-drain output
		1	Hi-Z	No	Open-drain output or Input without pull-high
1	0	0	Low	No	CMOS push-pull output
		1	High	No	
1	1	x	Hi-Z	No	Touch key / Comparator input

4.3 PA7

PA7 can be used in Schmitt-trigger input or open-drain output which is setting by the PAD[7] (F05.7) bit. When the PAD[7] bit is set, PA7 is assigned as Schmitt-trigger input mode, otherwise is assigned as open-drain output mode and output low. The pull-up resistor connected to this pin default, and can be disabled by S/W. In open-drain output mode, the pull-up resistor will be disabled automatically for power saving. When SYSCFG[7] is set, PA7 is only used in Schmitt-trigger input for external active low reset.



How to control PA7 status can be concluded as following list.

SYSCFG[7]	Register Setting			PIN STATE	Pull-up	Mode
	PAMODE1	PAMODE0	PAD7			
0	x	0	0	Low	No	Open-drain output
			1	High	Yes	Input with pull-high
		1	0	Low	No	Open-drain output
			1	Hi-Z	No	Input without pull-high
1	x	0	0	Hi-Z	No	Reset input without pull-high
			1	High	Yes	Reset input with pull-high
		1	x	Hi-Z	No	Reset input without pull-high

F05	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD	PAD7	PAD						
R/W	R/W	R/W						
Reset	1	1	1	1	1	1	1	1

F05.7 **PAD7:** PA7 data or pin mode control
0: PA7 is open-drain output mode and output low
1: PA7 is Schmitt-trigger input mode

F05.6~0 **PAD:** PA6~PA0 data
0: output low
1: output high or Schmitt-trigger input mode

F06	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBD	PBD							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

F06.7~0 **PBD:** PB7~PB0 data
0: output low
1: output high or Schmitt-trigger input mode

F07	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDD	–	–	–	–	–	–	PDD	
R/W	–	–	–	–	–	–	R/W	
Reset	–	–	–	–	–	–	1	1

F07.1~0 **PDD:** PD1~PD0 data
0: output low
1: output high or Schmitt-trigger input mode

R05	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAMODH	PAMODH							
R/W	W		W		W		W	
Reset	0	0	0	1	0	1	0	1

R05.7~0 **PAMODH**: PA7~PA4 Pin Mode Control
00: Open Drain output low, or input with pull-up
The PA4's pull-up resistor is disabled automatically for external oscillation in this mode
01: Open Drain output low, or input without pull-up
10: CMOS output low, or CMOS output high
11: Touch key input

R06	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAMODL	PAMODL							
R/W	W		W		W		W	
Reset	0	1	0	1	0	1	0	1

R06.7~0 **PAMODL**: PA3~PA0 Pin Mode Control
00: Open Drain output low, or input with pull-up
the PA3's pull-up resistor is disabled automatically for external oscillation in this mode
01: Open Drain output low, or input without pull-up
10: CMOS output low, or CMOS output high
11: Touch key input

R07	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBMODH	PBMODH							
R/W	W		W		W		W	
Reset	0	1	0	1	0	1	0	1

R07.7~0 **PBMODH**: PB7~PB4 Pin Mode Control
00: Open Drain output low, or input with pull-up
01: Open Drain output low, or input without pull-up
10: CMOS output low, or CMOS output high
11: Touch key input

R08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBMODL	PBMODL							
R/W	W		W		W		W	
Reset	0	1	0	1	0	1	0	1

R08.7~0 **PBMODL**: PB3~PB0 Pin Mode Control
00: Open Drain output low, or input with pull-up
01: Open Drain output low, or input without pull-up
10: CMOS output low, or CMOS output high
11: Touch key / Comparator input

R0A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDMOD	–	–	–	–	PDMOD			
R/W	–	–	–	–	W		W	
Reset	–	–	–	–	0	1	0	1

R0A.3~0 **PDMOD**: PD1~PD0 Pin Mode Control
00: Open Drain output low, or input with pull-up
01: Open Drain output low, or input without pull-up
10: CMOS output low, or CMOS output high
11: Touch key input

R13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAWKEN	–	PAWKEN						–
R/W	–	W						–
Reset	–	0	0	0	0	0	0	–

R13.6~1 **PAWKEN:** PA6~PA1 individual pin low level wake up control
0: disable
1: enable

R18	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBWKEN	–	PBWKEN						–
R/W	–	W						–
Reset	–	0	0	0	0	0	0	–

R18.6~1 **PBWKEN:** PB6~PB1 individual pin low level wake up control
0: disable
1: enable

MEMORY MAP

F-Plane

Name	Address	R/W	Rst	Description
(F00) INDF				**Function related to : RAM W/R**
INDF	00.7~0	R/W	-	Not a physical register, addressing INDF actually point to the register whose address is contained in the FSR register
(F01) TM0				**Function related to : Timer0**
TM0	01.7~0	R/W	0	Timer0 content
(F02) PCL				**Function related to : PROGRAM COUNT**
PCL	02.7~0	R/W	0	Programming Counter LSB[7~0]
(F03) STATUS				**Function related to : STATUS**
LVD	03.7	R	0	Low voltage detector flag
GB0	03.6	R/W	0	General purpose bit 0
RAMBK	03.5	R/W	0	SRAM Bank selection, 0: Bank0, 1: Bank1
TO	03.4	R	0	WDT timeout flag
PD	03.3	R	0	Power-down mode flag
Z	03.2	R/W	0	Zero flag
DC	03.1	R/W	0	Decimal Carry flag or Decimal /Borrow flag
C	03.0	R/W	0	Carry flag or /Borrow flag
(F04) FSR				**Function related to : RAM W/R / Table Read**
DPL	04.7~0	R/W	-	Table read low address, data ROM pointer (DPTR) low byte
FSR	04.6~0	R/W	-	File Select Register, indirect address mode pointer
(F05) PAD				**Function related to : Port A**
PAD7	05.7	R	-	PA7 pin or “data register” state
		W	1	0: PA7 is open-drain output mode 1: PA7 is Schmitt-trigger input mode
PAD	05.6~0	R	-	Port A pin or “data register” state
		W	7F	Port A output data register
(F06) PBD				**Function related to : Port B**
PBD	06.7~0	R	-	Port B pin or “data register” state
		W	FF	Port B output data register
(F07) PDD				**Function related to : Port D**
PDD	07.1~0	R	-	Port D pin or “data register” state
		W	3	Port D output data register

Name	Address	R/W	Rst	Description
(F08) INTIE				**Function related to : Interrupt Enable**
-	08.7	-	-	Reserved
T2IE	08.6	R/W	0	T2 interrupt enable 0: disable 1: enable
CMPIE	08.5	R/W	0	Comparator interrupt enable 0: disable 1: enable
TM0IE	08.4	R/W	0	Timer0 interrupt enable 0: disable 1: enable
-	08.3	-	-	Reserved
INT2IE	08.2	R/W	0	INT2 (PA7) pin interrupt enable 0: disable 1: enable
INT1IE	08.1	R/W	0	INT1 (PD0) pin interrupt enable 0: disable 1: enable
INT0IE	08.0	R/W	0	INT0 (PA2) pin interrupt enable 0: disable 1: enable
(F09) INTIF				**Function related to : Interrupt Flag**
-	09.7	-	-	Reserved
T2IF	09.6	R	-	T2 interrupt event pending flag, set by H/W while T2 overflows
		W	0	0: clear this flag 1: no action
CMPIF	09.5	R	-	Comparator interrupt event pending flag, set by H/W at Comparator output falling/rising edge
		W	0	0: clear this flag 1: no action
TM0IF	09.4	R	-	Timer0 interrupt event pending flag, set by H/W while Timer0 overflows
		W	0	0: clear this flag 1: no action
-	09.3	-	-	Reserved
INT2IF	09.2	R	-	INT2 interrupt event pending flag, set by H/W at INT2 pin's falling edge
		W	0	0: clear this flag 1: no action
INT1IF	09.1	R	-	INT1 interrupt event pending flag, set by H/W at INT1 pin's falling edge
		W	0	0: clear this flag 1: no action
INT0IF	09.0	R	-	INT0 interrupt event pending flag, set by H/W at INT0 pin's falling/rising edge
		W	0	0: clear this flag 1: no action

Name	Address	R/W	Rst	Description
(F12) PWM0D				**Function related to : PWM0**
PWM0D	12.7~0	R/W	0	PWM0 duty
(F13) PWM1D				**Function related to : PWM1**
PWM1D	13.7~0	R/W	0	PWM1 duty
(F14) MF14				**Function related to : Comparator / T2 / TM0 / PWM0**
CMPST	14.3	R	0	CMPO state
T2CLR	14.2	R/W	0	T2 counter clear 0: T2 is counting 1: T2 is clear immediately, this bit is auto cleared by H/W
TM0STP	14.1	R/W	0	Timer0 counter stop 0: Timer0 is counting 1: Timer0 stops counting
PWM0CLR	14.0	R/W	1	PWM0 clear and hold 0: PWM0 is running 1: PWM0 is clear and hold
(F15) CLKCTL				**Function related to : CPUCLK / T2**
SLOWCKS	15.7	R/W	0	Slow-clock type select or T2 clock source select For Slow-clock type 0: SIRC 1: SXT For T2 clock source 0: SIRC/2 1: SXT
FASTCKS	15.6	R/W	0	Fast-clock type select 0:FIRC 1:FXT
GB1	15.5	R/W	0	General purpose bit 1
SLOWSTP	15.4	R/W	0	Slow-clock Enable / Disable 0: enable 1: disable in Power-down mode
FASTSTP	15.3	R/W	0	Fast-clock Enable / Disable 0: enable 1: disable
CPUCKS	15.2	R/W	0	System clock source select 0: Slow-clock 1: Fast-clock
CPUPSC	15.1~0	R/W	11	System clock source prescaler. System clock source 00: divided by 16 01: divided by 4 10: divided by 2 11: divided by 1
(F16) PWM2D				**Function related to : PWM2**
PWM2D	16.7~0	R/W	0	PWM2 duty
(F17) DPH				**Function related to : Table Read**
DPH	17.2~0	R/W	0	Table read high address, data rom pointer (DPTR) high byte

Name	Address	R/W	Rst	Description
(F1B) TKCTL1				**Function related to : Touch Key**
TKSOC	1b.7	R/W	0	Touch key start of conversion, rising edge to start H/W auto cleared while end of conversion
TKTMR	1b.6~4	R/W	4	Touch key conversion time 0: shortest, 7: longest
TKCHS	1b.3~0	R/W	0	Touch key channel select 0000: TK0 (PA1) 0001: TK1 (PA2) 0010: TK2 (PB0) 0011: TK3 (PB1) 0100: TK4 (PB2) 0101: TK5 (PB3) 0110: TK6 (PB4) 0111: TK7 (PB5) 1000: TK8 (PB6) 1001: TK9 (PB7) 1010: TK10 (PA0) 1011: TK11 (PA6) 1100: TK12 (PD1) 1101: TK13 (PD0)
(F1C) TKDH				**Function related to : Touch Key**
TKEOC	1c.3	R	1	Touch key end of conversion 0: convert in process 1: end of conversion
TKOVF	1c.2	R	0	Touch key counter overflow flag 0: not overflow 1: overflow
TKDH	1c.1~0	R	0	Touch key counter MSB [9~8]
(F1D) TKDL				**Function related to : Touch Key**
TKDL	1d.7~0	R	0	Touch key counter LSB [7~0]
User Data Memory				
SRAM	20~27	R/W	-	SRAM common area (8 bytes)
	28~7f	R/W	-	SRAM Bank0 area (RAMBK=0, 88 bytes)
	28~7f	R/W	-	SRAM Bank1 area (RAMBK=1, 88 bytes)

R-Plane

Name	Address	R/W	Rst	Description
(R02) TM0CTL				**Function related to : Timer0**
TM0EDG	02.5	W	0	TM0CKI (PA2) edge selection for Timer0 prescaler count 0: TM0CKI (PA2) rising edge for Timer0 prescaler count 1: TM0CKI (PA2) falling edge for Timer0 prescaler count
TM0CKS	02.4	W	0	Timer0 clock source select 0: Instruction Cycle (Fsys/2) as Timer0 prescaler clock 1: TM0CKI (PA2) as Timer0 prescaler clock
TM0PSC	02.3~0	W	0	Timer0 prescaler. Timer0 clock source 0000: divided by 1 0001: divided by 2 0010: divided by 4 0011: divided by 8 0100: divided by 16 0101: divided by 32 0110: divided by 64 0111: divided by 128 1xxx: divided by 256
(R03) PWRDN				**Function related to : POWER DOWN**
PWRDN	03	W	-	Write this register to enter Power-down (STOP/IDLE) Mode
(R04) WDTCLR				**Function related to : WDT**
WDTCLR	04	W	-	Write this register to clear WDT timer
(R05) PAMODH				**Function related to : Port A**
PAMODH	05.7~0	W	15	PA7~PA4 I/O mode control 00: Open Drain output low, or input with pull-up The PA4's pull-up resistor is disabled automatically for external oscillation 01: Open Drain output low, or input without pull-up 10: CMOS output low, or CMOS output high 11: Touch key input
(R06) PAMODL				**Function related to : Port A**
PAMODL	06.7~0	W	55	PA3~PA0 I/O mode control 00: Open Drain output low, or input with pull-up The PA3's pull-up resistor is disabled automatically for external oscillation 01: Open Drain output low, or input without pull-up 10: CMOS output low, or CMOS output high 11: Touch key input
(R07) PBMODH				**Function related to : Port B**
PBMODH	07.7~0	W	55	PB7~PB4 I/O mode control 00: Open Drain output low, or input with pull-up 01: Open Drain output low, or input without pull-up 10: CMOS output low, or CMOS output high 11: Touch key input
(R08) PBMODL				**Function related to : Port B**
PBMODL	08.7~0	W	55	PB3~PB0 I/O mode control 00: Open Drain output low, or input with pull-up 01: Open Drain output low, or input without pull-up 10: CMOS output low, or CMOS output high 11: Touch key / Comparator input

<table>
<tr><th>Name</th><th>Address</th><th>R/W</th><th>Rst</th><th>Description</th></tr>
<tr><td colspan="4">(R0A) PDMOD</td><td>Function related to : Port D</td></tr>
<tr><td>PDMOD</td><td>0a.3~0</td><td>W</td><td>5</td><td>PD1~PD0 I/O mode control
00: Open Drain output low, or input with pull-up
01: Open Drain output low, or input without pull-up
10: CMOS output low, or CMOS output high
11: Touch key input</td></tr>
<tr><td colspan="4">(R0B) MR0B</td><td>Function related to : T2 / INT0 / TCOUT / WDT</td></tr>
<tr><td>T2PSC</td><td>0b.6~5</td><td>W</td><td>0</td><td>T2 prescaler. T2 clock source
00: divided by 32768
01: divided by 16384
10: divided by 8192
11: divided by 128</td></tr>
<tr><td>INT0EDG</td><td>0b.4</td><td>W</td><td>0</td><td>INT0 pin (PA2) edge interrupt event
0: falling edge to trigger
1: rising edge to trigger</td></tr>
<tr><td>TCOE</td><td>0b.3</td><td>W</td><td>0</td><td>Enable Instruction Cycle (Fsys/2) output to PA3 pin (TCOUT)
0: disable
1: enable</td></tr>
<tr><td>-</td><td>0b.2</td><td>-</td><td>-</td><td>Reserved</td></tr>
<tr><td>WDTPSC</td><td>0b.1~0</td><td>W</td><td>11</td><td>WDT pre-scale selections:
<table><tr><th>Bit 1</th><th>Bit 0</th><th>5V</th><th>3V</th></tr><tr><td>0</td><td>0</td><td>19 ms</td><td>24 ms</td></tr><tr><td>0</td><td>1</td><td>38 ms</td><td>48 ms</td></tr><tr><td>1</td><td>0</td><td>76 ms</td><td>96 ms</td></tr><tr><td>1</td><td>1</td><td>152 ms</td><td>192 ms</td></tr></table></td></tr>
<tr><td colspan="4">(R0E) MR0E</td><td>Function related to : Power Filter / Voltage Pump / Operating Voltage</td></tr>
<tr><td>VDDFLT</td><td>0e.6</td><td>W</td><td>0</td><td>Power noise filter
0: disable
1: enable</td></tr>
<tr><td>NOPUMP</td><td>0e.3</td><td>W</td><td>0</td><td>Voltage PUMP control
0: enable auto-pump-mode
1: disable voltage pump</td></tr>
<tr><td>MODE3V</td><td>0e.2</td><td>W</td><td>0</td><td>MODE 3V control
0: disable
1: enable</td></tr>
<tr><td colspan="4">(R10) PMW0PRD</td><td>Function related to : PWM0</td></tr>
<tr><td>PWM0PRD</td><td>10.7~0</td><td>W</td><td>FF</td><td>PWM0 period data</td></tr>
</table>

Name	Address	R/W	Rst	Description
(R11) PWMCTL				**Function related to : PWM0 / PWM1 / PWM2**
PWM2OE	11.5	W	0	PWM2 positive output to PA6 pin 0: disable 1: enable
PWMCKS	11.4	W	0	PWM clock source select 0: System clock (Fsys) 1: FIRC 16 MHz
PWM1OE	11.3	W	0	PWM1 positive output to PA1 pin 0: disable 1: enable
PWM0OE	11.2	W	0	PWM0 positive output to PB4 pin 0: disable 1: enable
PWM0PSC	11.1~0	W	0	PWM0 prescaler, PWM0 clock source 00: divided by 1 01: divided by 2 10: divided by 4 11: divided by 8
(R13) PAWKEN				**Function related to : Port A / WAKE UP**
PAWKEN	13.6~1	W	0	PA6~PA1 individual pin low level wake up control Each bit controls its corresponding pin, if the bit is 0: disable 1: enable
(R17) CMPCTL				**Function related to : Comparator**
CMPE	17.7	W	0	Comparator enable 0: disable 1: enable
CMPEDG	17.6	W	0	Comparator interrupt edge 0: falling edge to trigger 1: rising edge to trigger
CMPOE	17.5	W	0	Comparator output to pin enable 0: disable 1: enable
CMPINNS	17.4~3	W	0	Comparator negative input source select 0x: VSS 10: IN0- (PB0) 11: IN1- (PB1)
-	17.2~1	-	-	Reserved
CMPINPS	17.0	W	0	Comparator positive input source select 0: IN0+ (PB2) 1: IN1+ (PB3)
(R18) PBWKEN				**Function related to : Port B / WAKE UP**
PBWKEN	18.6~1	W	0	PB6~PB1 individual pin low level wake up control Each bit controls its corresponding pin, if the bit is 0: disable 1: enable

Name	Address	R/W	Rst	Description
(R1D) TKCTL2				**Function related to : Touch Key**
TKPD	1d.7	W	1	Touch key power down 0: Touch key running 1: Touch key power down
TKCKS	1d.6	W	1	Touch key clock select 0: 2 MHz 1: 4 MHz
TKPSC	1d.5~4	W	0	Touch key data prescaler, Touch Key data 00: divided by 1 01: divided by 2 10: divided by 4 11: divided by 8

INSTRUCTION SET

Each instruction is a 14-bit word divided into an Op Code, which specifies the instruction type, and one or more operands, which further specify the operation of the instruction. The instructions can be categorized as byte-oriented, bit-oriented and literal operations list in the following table.

For byte-oriented instructions, "f" 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 or /Borrow Flag
DC	Decimal Carry Flag or Decimal /Borrow Flag
PC	Program Counter
TOS	Top Of Stack
GIE	Global Interrupt Enable Flag (i-Flag)
[]	Option Field
( )	Contents
.	Bit Field
B	Before
A	After
←	Assign direction

Mnemonic		Op Code	Cycle	Flag Affect	Description
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
SUBLW	k	01 1011 kkkk kkkk	1	C, DC, Z	Subtract W from Literal "k"
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 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 Power-down mode, Clock oscillation stops
TABRH		00 0000 0101 1000	2	-	Lookup ROM high data to W
TABRL		00 0000 0101 0000	2	-	Lookup ROM low data to W
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 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 ~ 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	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 : 000h ~ FFFh	
Operation	Operation: TOS ← (PC) + 1, PC.11~0 ← k	
Status Affected	-	
OP-Code	10 kkkk 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 Z bit is set.	
Cycle	1	
Example	CLRW	B : W = 0x5A A : W = 0x00, Z = 1

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

COMF — Complement "f"

Syntax	COMF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← ($\overline{f}$)	
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 ~ 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 : 000h ~ FFFh	
Operation	PC.11~0 ← k	
Status Affected	-	
OP-Code	11 kkkk kkkk kkkk	
Description	GOTO is an unconditional branch. The 12-bit immediate value is loaded into PC bits <11:0>. GOTO is a two-cycle instruction.	
Cycle	2	
Example	LABEL1 GOTO SUB1	B : PC = LABEL1 A : PC = SUB1

INCF — Increment "f"

Syntax	INCF f [,d]
Operands	f : 00h ~ 7Fh
Operation	(destination) ← (f) + 1
Status Affected	Z
OP-Code	00 1010 dfff ffff
Description	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.
Cycle	1
Example	INCF CNT, 1 — B : CNT = 0xFF, Z = 0 A : CNT = 0x00, Z = 1

INCFSZ — Increment "f", Skip if 0

Syntax	INCFSZ f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	(destination) ← (f) + 1, skip next instruction if result is 0
Status Affected	-
OP-Code	00 1111 dfff ffff
Description	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead, making it a 2 cycle instruction.
Cycle	1 or 2
Example	LABEL1 INCFSZ CNT, 1 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 are OR'ed with the eight-bit literal 'k'. The result is placed in the W register.
Cycle	1
Example	IORLW 0x35 — B : W = 0x9A A : W = 0xBF, Z = 0

IORWF — Inclusive OR W with "f"

Syntax	IORWF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	(destination) ← (W) OR k
Status Affected	Z
OP-Code	00 0100 dfff ffff
Description	Inclusive OR the W register with register 'f'. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.
Cycle	1
Example	IORWF RESULT, 0 — B : RESULT = 0x13, W = 0x91

A : RESULT = 0x13, W = 0x93, Z = 0

MOVFW	**Move "f" to W**	
Syntax	MOVFW f	
Operands	f : 00h ~ 7Fh	
Operation	(W) ← (f)	
Status Affected	-	
OP-Code	00 1000 0fff ffff	
Description	The contents of register 'f' are moved to W register.	
Cycle	1	
Example	MOVFW FSR	B : FSR = 0xC2, W = ? A : FSR = 0xC2, W = 0xC2

MOVLW	**Move Literal to W**	
Syntax	MOVLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← k	
Status Affected	-	
OP-Code	01 1001 kkkk kkkk	
Description	The eight-bit literal 'k' is loaded into W register. The don't cares will assemble as 0's.	
Cycle	1	
Example	MOVLW 0x5A	B : W = ? A : W = 0x5A

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

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

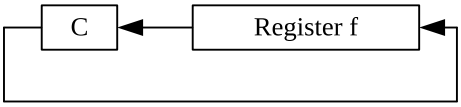
NOP	No Operation	
Syntax	NOP	
Operands	-	
Operation	No Operation	
Status Affected	-	
OP-Code	00 0000 0000 0000	
Description	No Operation	
Cycle	1	
Example	NOP	-

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

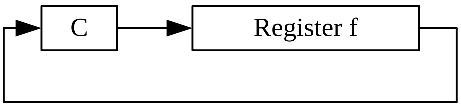
RETI	Return from Interrupt	
Syntax	RETI	
Operands	-	
Operation	PC ← TOS, GIE ← 1	
Status Affected	-	
OP-Code	00 0000 0110 0000	
Description	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in to the PC. Interrupts are enabled. This is a two-cycle instruction.	
Cycle	2	
Example	RETI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

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 Power-down mode, Clock oscillation stops

Syntax	SLEEP
Operands	-
Operation	-
Status Affected	TO, PD
OP-Code	00 0000 0000 0011
Description	Go into Power-down mode with the oscillator stops.
Cycle	1
Example	SLEEP — -

SUBLW — Subtract W from Literal "k"

Syntax	SUBLW k
Operands	k : 00h ~ FFh
Operation	(W) ← (k) – (W)
Status Affected	C, DC, Z
OP-Code	01 1101 kkkk kkkk
Description	The contents of the W register are subtracted (2's complement method) from the eight-bit literal 'k' and the result is placed in the W register.
Cycle	1
Example	SUBLW 0x15 — B : W = 0x10, C = ?, Z = ? A : W = 0x05, C = 1, Z = 0
	SUBLW 0x10 — B : W = 0x10, C = ?, Z = ? A : W = 0x00, C = 1, Z = 1
	SUBLW 0x05 — B : W = 0x10, C = ?, Z = ? A : W = 0xF5, C = 0, Z = 0

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 = 0x03, W = 0x02, C = ?, Z = ? A : REG1 = 0x01, W = 0x02, C = 1, Z = 0
	SUBWF REG1, 1 — B : REG1 = 0x02, W = 0x02, C = ?, Z = ? A : REG1 = 0x00, W = 0x02, C = 1, Z = 1
	SUBWF REG1, 1 — B : REG1 = 0x01, W = 0x02, C = ?, Z = ? A : REG1 = 0xFF, W = 0x02, 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

TABRH — Return DPTR high byte to W

Syntax	TABRH
Operands	-
Operation	(W) ← ROM[DPTR] high byte content, Where DPTR = {DPH[max:8], FSR[7:0]}
Status Affected	-
OP-Code	00 0000 0101 1000
Description	The W register is loaded with high byte of ROM[DPTR]. This is a two-cycle instruction.
Cycle	2
Example	

```
MOVLW   (TAB1&0xFF)
MOVWF   FSR                 ;Where FSR is F-Plane register
MOVLW   (TAB1>>8)&0xFF
MOVWF   DPH                 ;Were DPH is F-Plane register

TABRL                       ;W = 0x89
TABRH                       ;W = 0x37

        ORG 0234H
TAB1:
DT      0x3789, 0x2277      ;ROM data 14bits
```

TABRL — Return DPTR low byte to W

Syntax	TABRL
Operands	-
Operation	(W) ← ROM[DPTR] low byte content, Where DPTR = {DPH[max:8], FSR[7:0]}
Status Affected	-
OP-Code	00 0000 0101 0000
Description	The W register is loaded with low byte of ROM[DPTR]. This is a two-cycle instruction.
Cycle	2
Example	

```
MOVLW   (TAB1&0xFF)
MOVWF   FSR                 ;Where FSR is F-Plane register
MOVLW   (TAB1>>8)&0xFF
MOVWF   DPH                 ;Where DPH is F-Plane register

TABRL                       ;W = 0x89
TABRH                       ;W = 0x37

        ORG 0234H
TAB1:
DT      0x3789, 0x2277      ;ROM data 14bits
```

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°C, V_{DD} = 1.1V to 5.5V)

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Operating Voltage	V_{DD}	FAST mode, 25°C, Fsys = 24 MHz		2.9	–	5.5	V
		FAST mode, 25°C, Fsys = 16 MHz		2.3	–	5.5	
		FAST mode, 25°C, Fsys = 8 MHz		1.8	–	5.5	
		FAST mode, 25°C, Fsys = 4 MHz		1.5	–	5.5	
		SLOW mode, 25°C, SIRC		1.1	–	5.5	
Input High Voltage	V_{IH}	All Input, except PA7	V_{DD} = 5V	$0.6V_{DD}$	–	–	V
			V_{DD} = 3V	$0.6V_{DD}$	–	–	V
		PA7	V_{DD} = 5V	$0.7V_{DD}$	–	–	V
			V_{DD} = 3V	$0.7V_{DD}$	–	–	V
Input Low Voltage	V_{IL}	All Input	V_{DD} = 5V	–	–	$0.2V_{DD}$	V
			V_{DD} = 3V	–	–	$0.2V_{DD}$	V
I/O Port Source Current	I_{OH}	All Output	V_{DD} = 5V, V_{OH} = $0.9V_{DD}$	6	12	–	mA
			V_{DD} = 3V, V_{OH} = $0.9V_{DD}$	3	6	–	
I/O Port Sink Current	I_{OL}	All Output, except PA7	V_{DD} = 5V, V_{OL} = $0.1V_{DD}$	15	30	–	mA
			V_{DD} = 3V, V_{OL} = $0.1V_{DD}$	7.5	15	–	
		PA7	V_{DD} = 5V, V_{OL} = $0.1V_{DD}$	22.5	45	–	
			V_{DD} = 3V, V_{OL} = $0.1V_{DD}$	9	18	–	
Input Leakage Current (pin high)	I_{ILH}	All Input	V_{IN} = V_{DD}	–	–	1	μA
Input Leakage Current (pin low)	I_{ILL}	All Input	V_{IN} = 0V	–	–	-1	

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Supply Current	I_{DD}	FAST mode, LVR enable, WDT enable	V_{DD} = 5V, FXT =12 MHz	–	3.0	–	mA
			V_{DD} = 3V, FXT = 12MHz	–	1.9	–	
			V_{DD} = 5V, FXT = 8 MHz	–	2.3	–	
			V_{DD} = 3V, FXT = 8 MHz	–	1.4	–	
			V_{DD} = 5V, FXT = 4 MHz	–	1.4	–	
			V_{DD} = 3V, FXT = 4 MHz	–	0.8	–	
			V_{DD} = 5V, FIRC = 8 MHz	–	2.1	–	
			V_{DD} = 3V, FIRC = 8 MHz	–	1.5	–	
		SLOW mode, LVR enable	V_{DD} = 5V, SXT = 32 KHz	–	33	–	μA
			V_{DD} = 3V, SXT = 32 KHz	–	11	–	
			V_{DD} = 5 V, SIRC, CPUPSC = 11	–	51	–	
			V_{DD} = 3 V, SIRC, CPUPSC = 11	–	22	–	
		IDLE mode, LVR enable	V_{DD} = 5V, SXT = 32 KHz	–	14	–	
			V_{DD} = 3V, SXT = 32 KHz	–	3.5	–	
			V_{DD} = 5 V, SIRC, CPUPSC = 11	–	15.5	–	
			V_{DD} = 3 V, SIRC, CPUPSC = 11	–	4.5	–	
		IDLE mode, LVR disable	V_{DD} = 5V, SXT = 32 KHz	–	9	–	
			V_{DD} = 3V, SXT = 32 KHz	–	2	–	
			V_{DD} = 5 V, SIRC, CPUPSC = 11	–	10	–	
			V_{DD} = 3 V, SIRC, CPUPSC = 11	–	2.5	–	
		STOP mode, LVR enable	V_{DD} = 5V	–	5.0	–	
			V_{DD} = 3V	–	1.5	–	
		STOP mode, LVR disable	V_{DD} = 5V	–	–	0.1	
			V_{DD} = 3V	–	–	0.1	
System Clock Frequency	Fsys	V_{DD} > LVR_{th}	V_{DD} = 3.0V	–	–	20	MHz
			V_{DD} = 2.1V	–	–	12	
			V_{DD} = 1.6V	–	–	4	
LVR Reference Voltage	V_{LVR}	T_A = 25°C		–	3	–	V
				–	2.1	–	V
				–	1.6	–	V
LVR Hysteresis Voltage	V_{HYST}	T_A = 25°C		–	±0.1	–	V
LVD Reference Voltage	V_{LVD}	T_A = 25°C		–	3.1	–	V
				–	2.2	–	V
Low Voltage Detection time	t_{LVR}	T_A = 25°C		100	–	–	μs
Pull-Up Resistor	R_P	V_{IN} = 0 V Port A, B, D	V_{DD} = 5V	–	65	–	KΩ
			V_{DD} = 3V		120		
		V_{IN} = 0 V PA7	V_{DD} = 5V	–	60	–	
			V_{DD} = 3V		140		

3. Clock Timing (T_A = -40°C to +85°C)

Parameter	Condition	Min	Typ	Max	Unit
Internal RC Frequency	25°C, V_{DD} = 3 ~ 5.5V	7.75	8	8.25	MHz
	25°C, V_{DD} = 2.6 ~ 3V	7.6	8	8.4	
	-40°C ~ 85°C, V_{DD} = 2.6 ~ 5.5V	7.5	8	8.5	

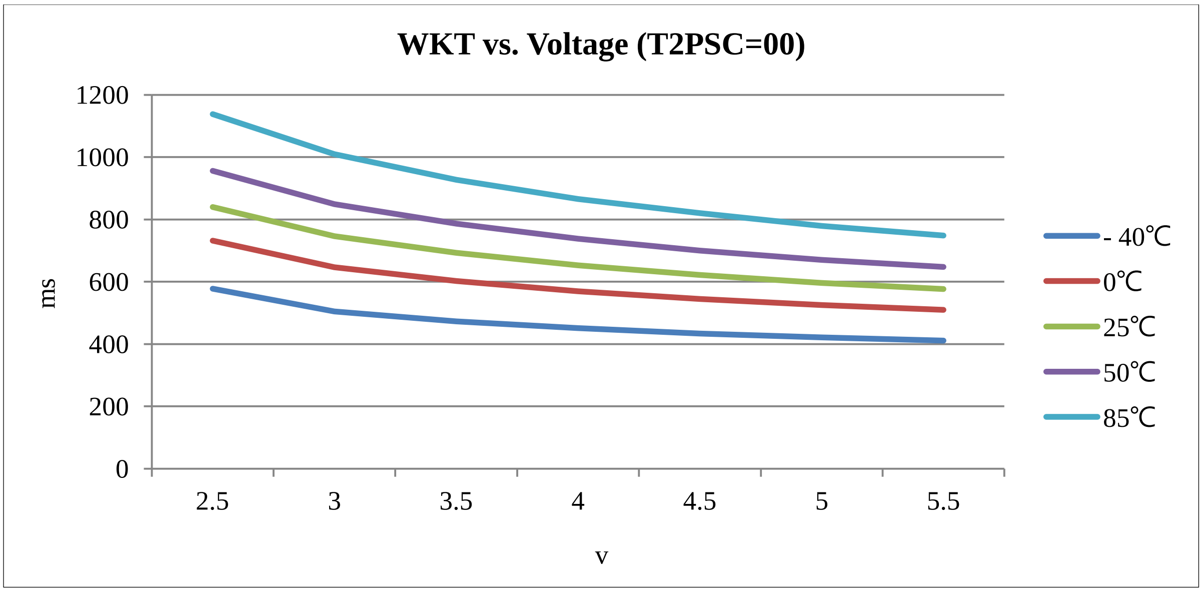
4. Reset Timing Characteristics (T_A = -40°C to +85°C, V_{DD} = 3V to 5V)

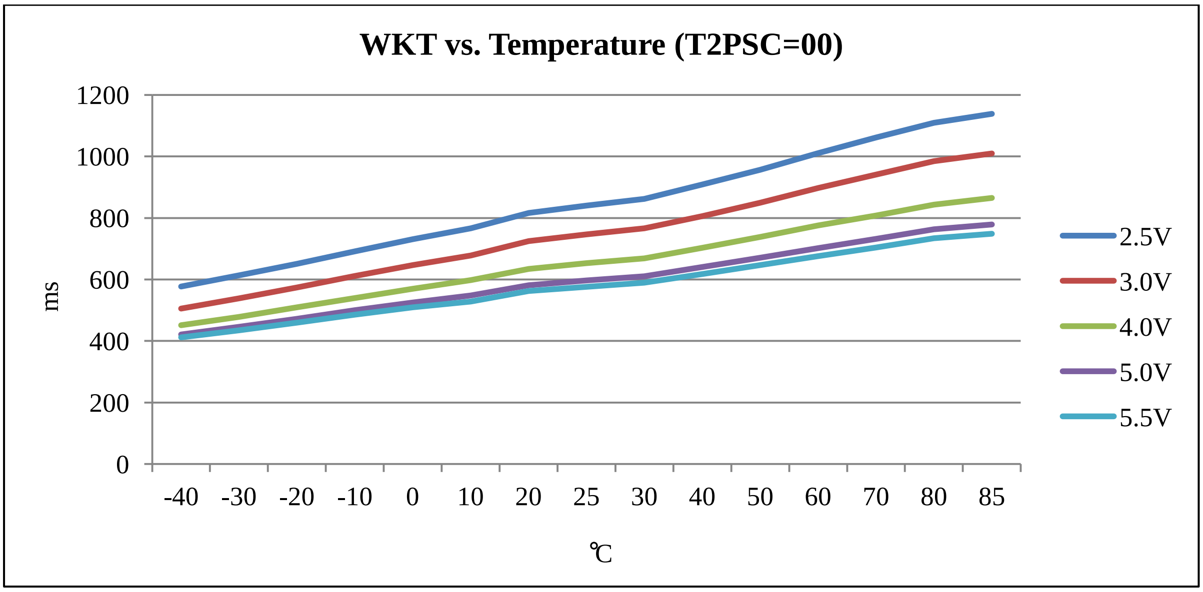
Parameter	Conditions	Min	Typ	Max	Unit
RESET Input Low width	Input V_{DD} = 5 V ± 10 %	3	–	–	µs
WDT wakeup time	V_{DD} = 5V, WDTPSC = 00	–	19	–	ms
	V_{DD} = 3V, WDTPSC = 00	–	24	–	
CPU start up time	V_{DD} = 5V	–	19	–	ms
	V_{DD} = 3V	–	24	–	

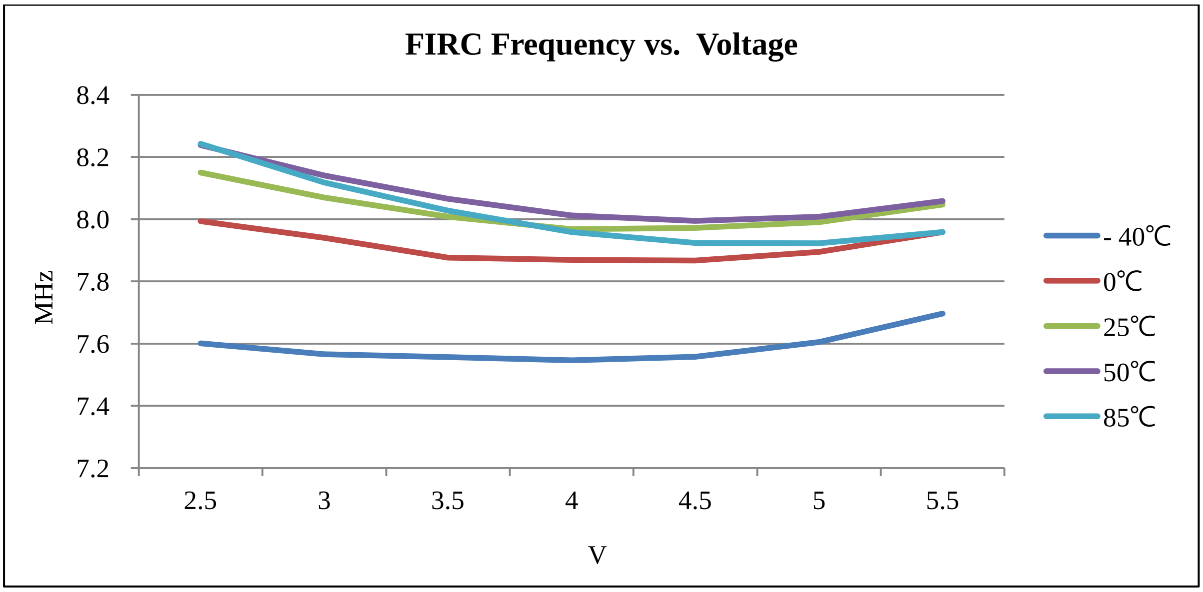
5. Comparator Characteristics (T_A = 25°C, V_{DD} = 5V)

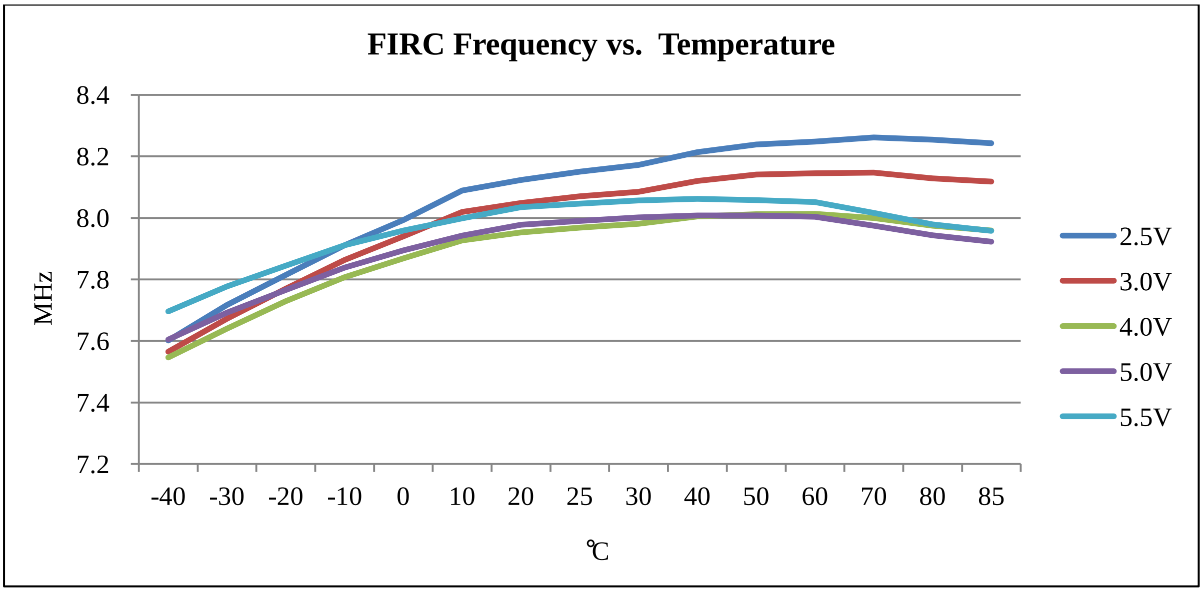
Parameter	Symbol	Conditions	Min	Typ	Max	Unit
CMP Operating Current	I_{CMP}	V_{DD} = 5V	–	230	–	µA
CMP Quiescent Current			–	180	–	
CMP Common Mode Voltage Range	V_{CM}	–	0	-	V_{DD} -1	V
CMP Hysteresis Width	V_{HYS}	V_{DD} = 5V	–	10	–	mV

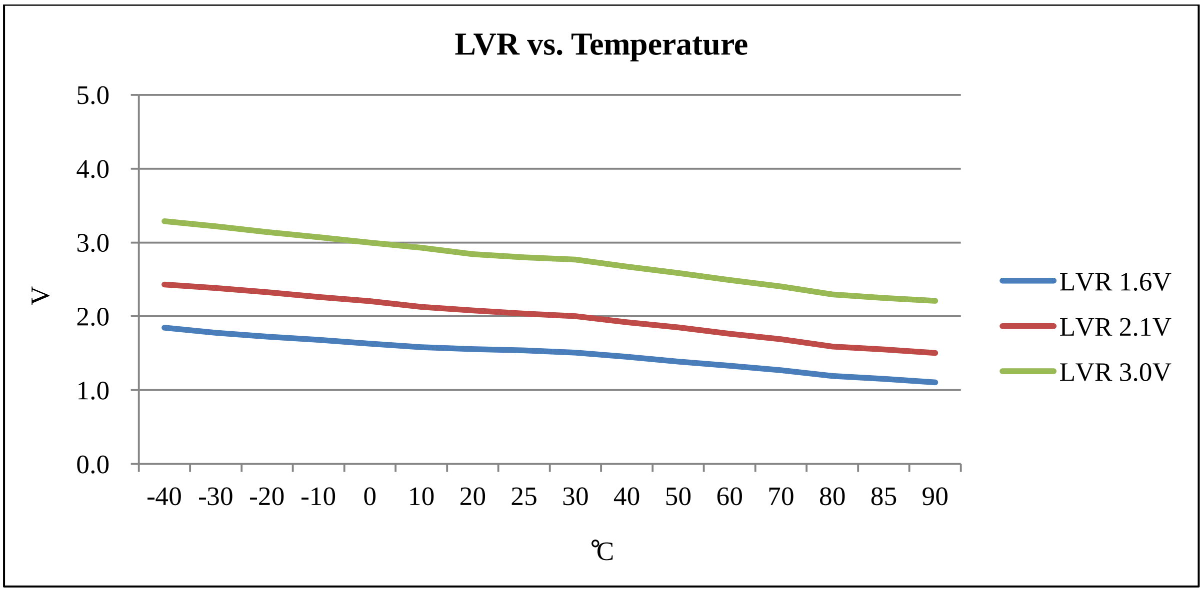
6. Characteristic Graphs

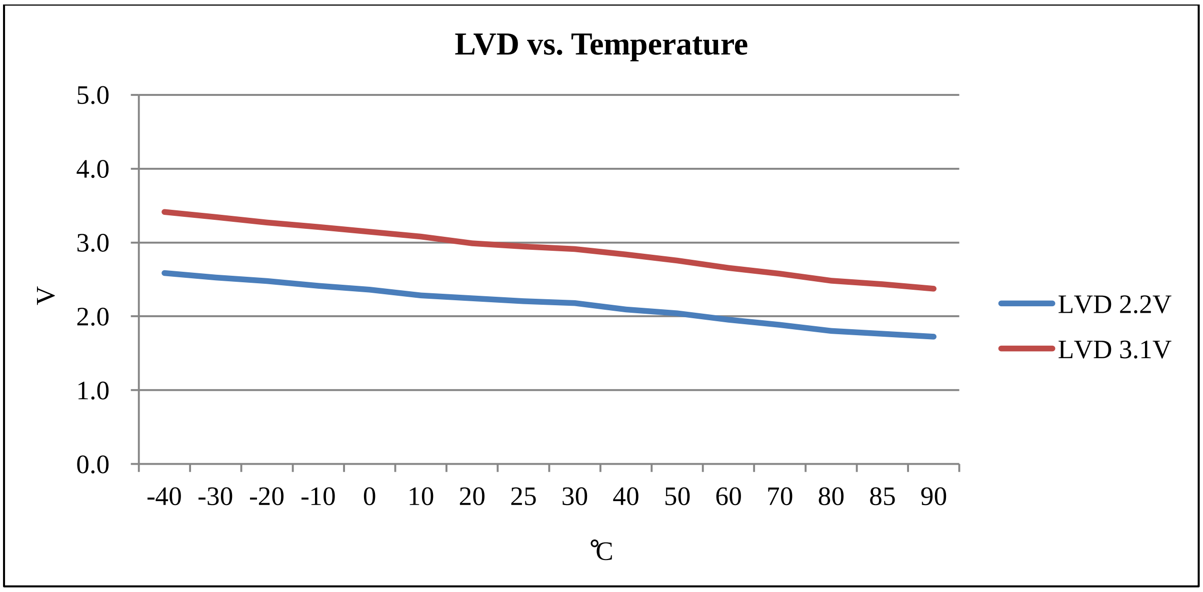


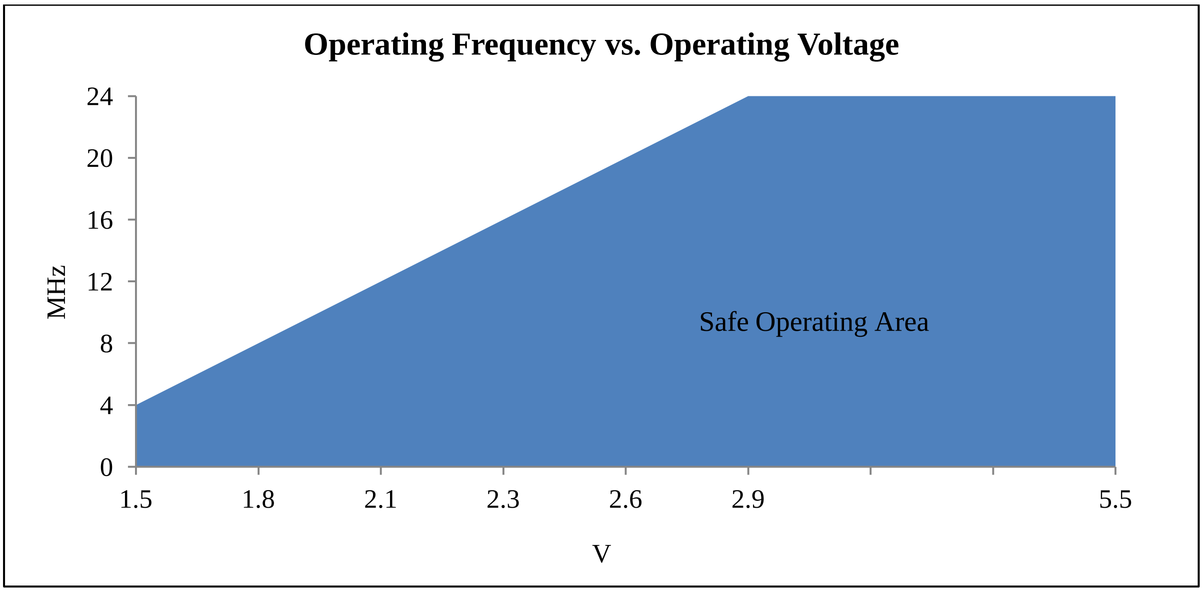


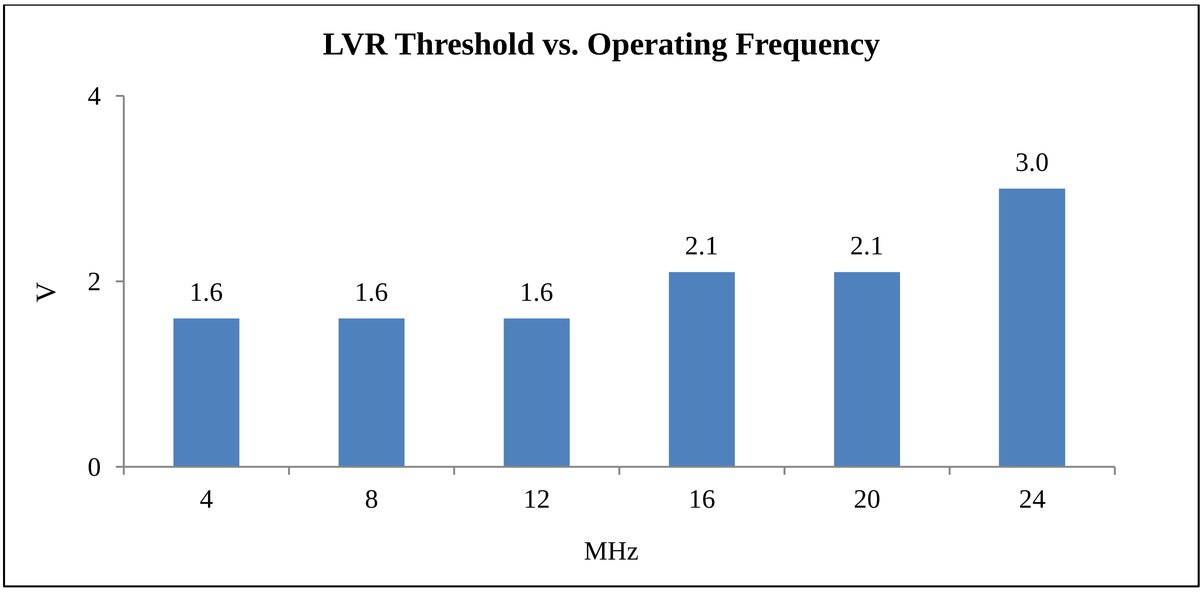

FIRC Frequency vs. Voltage
8.4
8.2
8.0
7.8
7.6
7.4
7.2
MHz
2.5
3
3.5
4
4.5
5
5.5
V
- 40℃
0℃
25℃
50℃
85℃


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
℃
2.5V
3.0V
4.0V
5.0V
5.5V


LVR vs. Temperature
5.0
4.0
3.0
2.0
1.0
0.0
V
-40 -30 -20 -10 0 10 20 25 30 40 50 60 70 80 85 90
℃
LVR 1.6V
LVR 2.1V
LVR 3.0V


LVD vs. Temperature
5.0
4.0
3.0
2.0
1.0
0.0
V
-40 -30 -20 -10 0 10 20 25 30 40 50 60 70 80 85 90
℃
LVD 2.2V
LVD 3.1V


Operating Frequency vs. Operating Voltage
24
20
16
12
8
4
0
MHz
Safe Operating Area
1.5
1.8
2.1
2.3
2.6
2.9
5.5
V

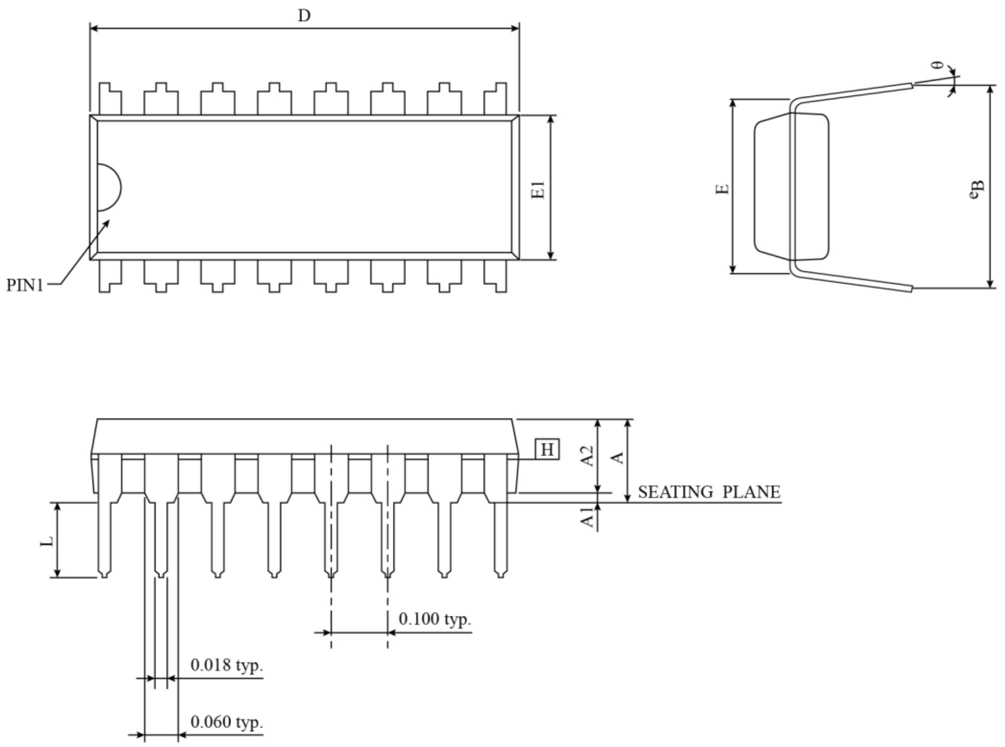

LVR Threshold vs. Operating Frequency
4
2
0
V
1.6
1.6
1.6
2.1
2.1
3.0
4
8
12
16
20
24
MHz

PACKAGING INFORMATION

The ordering information:

Ordering number	Package
TM57PT20B-OTP	Wafer / Dice blank chip
TM57PT20B-COD	Wafer / Dice with code
TM57PT20B-OTP-05	DIP 20-pin (300 mil)
TM57PT20B-OTP-21	SOP 20-pin (300 mil)
TM57PT20B-OTP-04	DIP 18-pin (300 mil)
TM57PT20B-OTP-20	SOP 18-pin (300 mil)
TM57PT20B-OTP-03	DIP 16-pin (300 mil)
TM57PT20B-OTP-16	SOP 16-pin (150 mil)

16-DIP Package Dimension

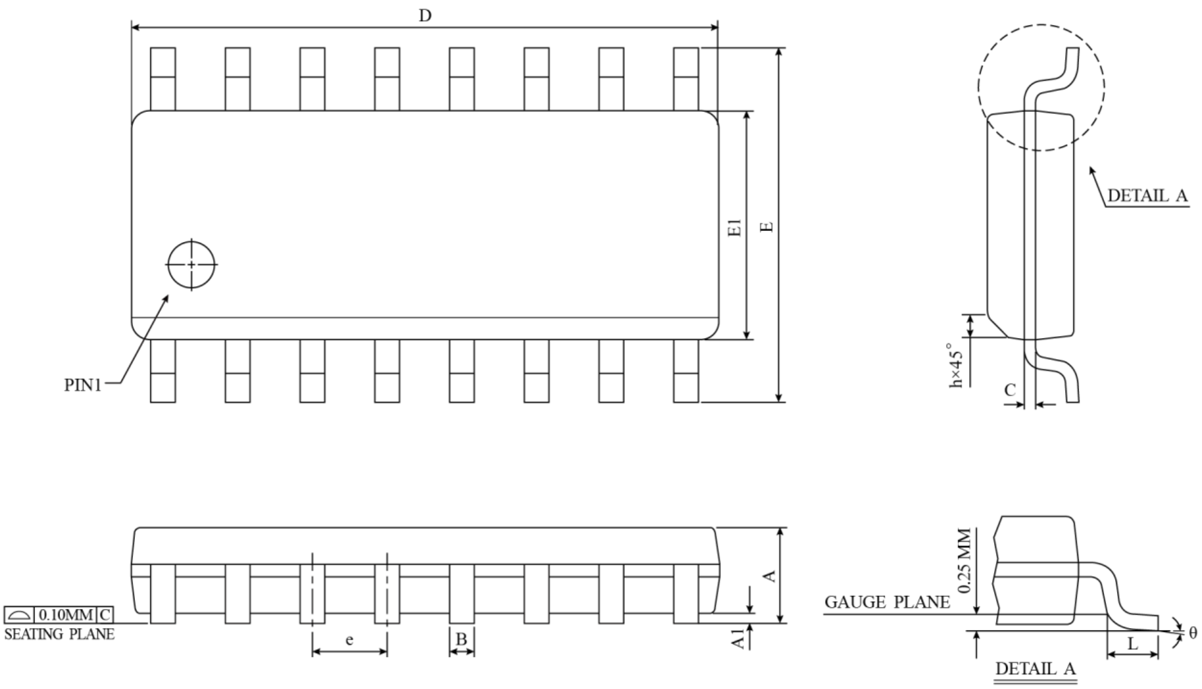


SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN	MAX	MIN	MAX
A	-	4.369	-	0.172
A1	0.381	0.965	0.015	0.038
A2	3.175	3.429	0.125	0.135
D	18.669	19.685	0.735	0.775
E	7.620 BSC		0.300 BSC	
E1	6.223	6.477	0.245	0.255
L	2.921	3.810	0.115	0.150
eB	8.509	9.525	0.335	0.375
θ	0°	15°	0°	15°
JEDEC	MS-001 (BB)			

NOTES：

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

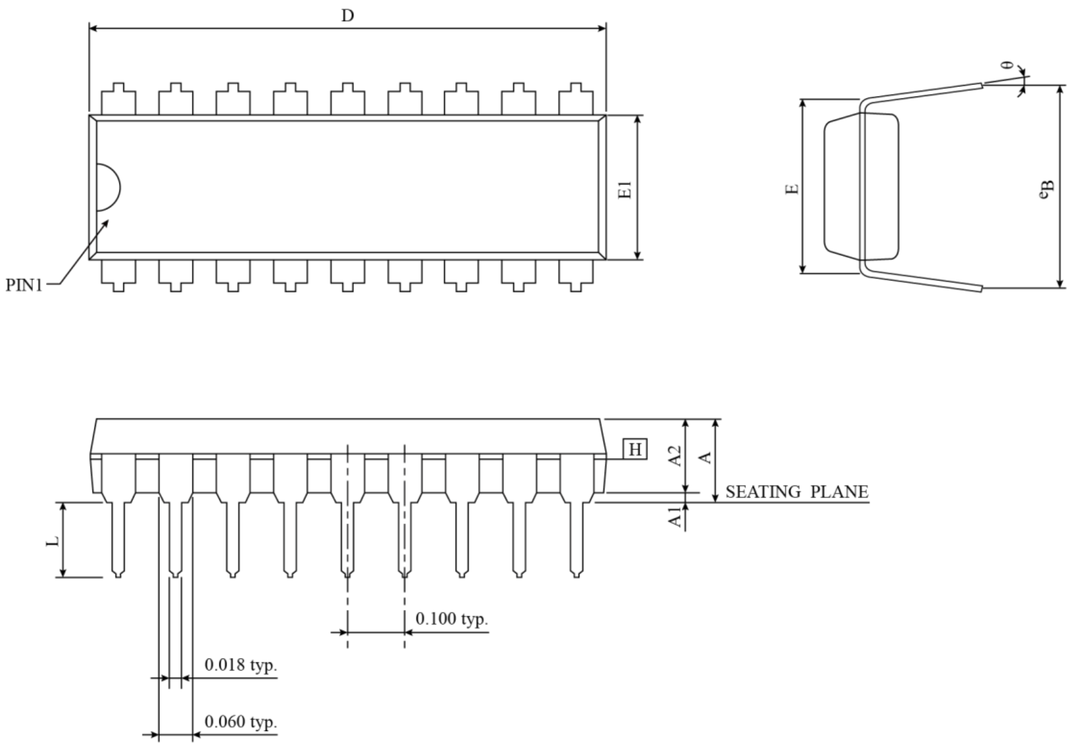
16-SOP Package Dimension



SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN	MAX	MIN	MAX
A	1.35	1.75	0.0532	0.0688
A1	0.10	0.25	0.0040	0.0098
B	0.33	0.51	0.013	0.020
C	0.19	0.25	0.0075	0.0098
D	9.80	10.00	0.3859	0.3937
E	5.80	6.20	0.2284	0.2440
E1	3.80	4.00	0.1497	0.1574
e	1.27 BSC		0.050 BSC	
h	0.25	0.50	0.0099	0.0196
L	0.40	1.27	0.016	0.050
θ	0°	8°	0°	8°
JEDEC	MS-012 (AC)			

△A * 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.

18-DIP Package Dimension

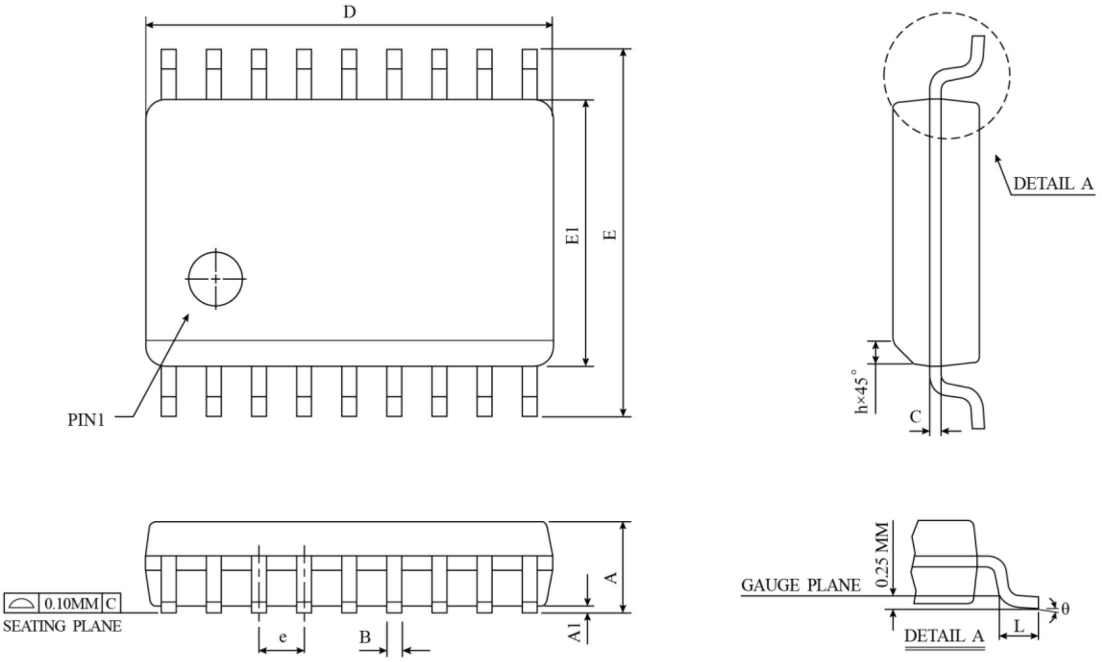


SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN	MAX	MIN	MAX
A	-	5.334	-	0.210
A1	0.381	-	0.015	-
A2	3.175	3.429	0.125	0.135
D	22.352	23.368	0.880	0.920
E	7.620 BSC		0.300 BSC	
E1	6.223	6.477	0.245	0.255
L	2.921	3.810	0.115	0.150
e_B	8.509	9.525	0.335	0.375
θ	0°	15°	0°	15°
JEDEC	MS-001 (AC)			

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.

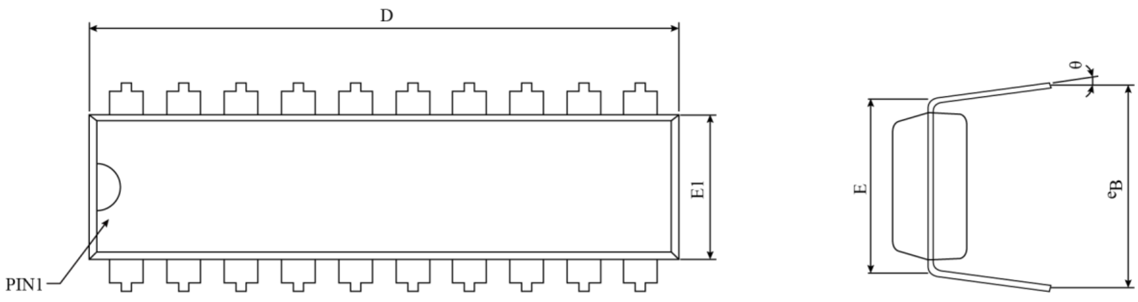
18-SOP Package Dimension

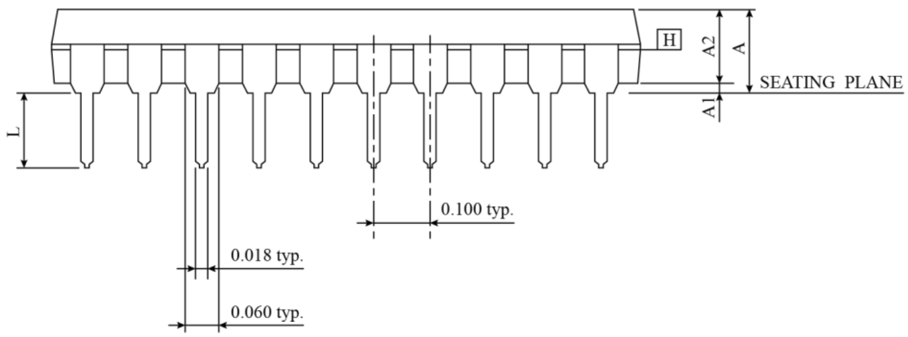


SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN	MAX	MIN	MAX
A	2.362	2.642	0.093	0.104
A1	0.102	0.305	0.004	0.012
B	0.406 typ		0.016 typ	
C	0.254 typ		0.010 typ	
D	11.354	11.760	0.447	0.463
E	10.008	10.643	0.394	0.419
E1	7.391	7.595	0.291	0.299
e	1.27 typ		0.050 typ	
h	0.508 typ		0.020 typ	
L	0.406	1.270	0.016	0.050
θ	0°	8°	0°	8°
JEDEC	MS-012 (AB)			

A * NOTES : 1. 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.
2. DIMENSION "E1" DOES NOT INCLUDE INTER-LEAD FLASH, OR PROTRUSIONS. INTER-LEAD FLASH AND PROTRUSIONS SHALL NOT EXCEED 0.25 MM (0.010 INCH) PER SIDE.

20-DIP Package Dimension



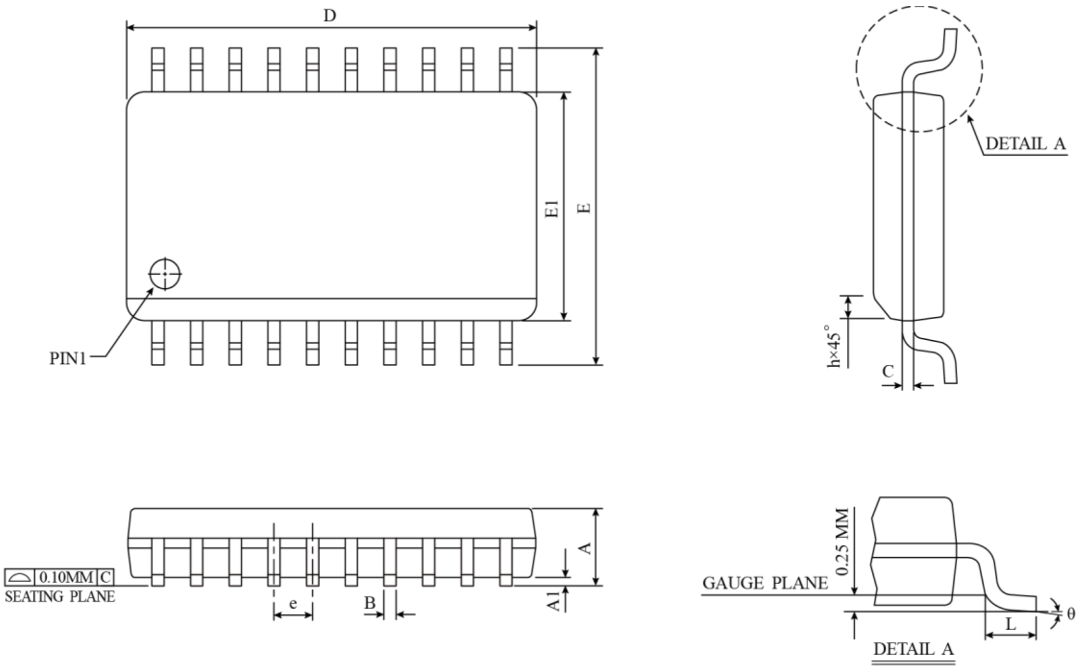


SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN	MAX	MIN	MAX
A	-	4.445	-	0.175
A1	0.381	-	0.015	-
A2	3.175	3.429	0.125	0.135
D	25.705	26.416	1.012	1.040
E	7.620	7.874	0.300	0.310
E1	6.223	6.477	0.245	0.255
L	3.048	3.556	0.120	0.140
eB	8.509	9.525	0.335	0.375
θ	0°	15°	0°	15°
JEDEC	MS-001 (AD)			

NOTES：

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

20-SOP Package Dimension



SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN	MAX	MIN	MAX
A	2.35	2.65	0.0926	0.1043
A1	0.10	0.30	0.0040	0.0118
B	0.33	0.51	0.013	0.020
C	0.23	0.32	0.0091	0.0125
D	12.60	13.00	0.4961	0.5118
E	10.00	10.65	0.394	0.491
E1	7.40	7.60	0.2914	0.2992
e	1.27 BSC		0.050 BSC	
h	0.25	0.75	0.010	0.029
L	0.40	1.27	0.016	0.050
θ	0°	8°	0°	8°
JEDEC	MS-013 (AC)			

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