十速

TM57PT45/PA45/ PT45C/PA45C

DATA SHEET

Rev 1.2

AMENDMENT HISTORY

Version	Date	Description
D0.93	Dec, 2015	1. Modify Operating Voltage (p7) 2. Add LVR selection table (p24) 3. Modify DC (p95) 4. Modify LVR vs. temperature (p99)
1.0	Aug, 2016	**(原 Doc No: TM57PT45_PA45_PT45C_PA45C _EDV093 因不再標示 Detailed 版，所以檔名修改為 DS- TM57PT45_PA45_PT45C_PA45C _EV10)** 1. p46: add PWM0PSC issue and PWM0 block diagram modified 2. p48: PWM0 example code modified 3. p51: PWM0PSC issue 4. p52: add PWM1PSC issue and PWM1 block diagram modified 5. p54: PWM1 example code modified 6. p57: PWM1PSC issue 7. p80: PWM0PSC, PWM1PSC descriptions modified 8. p84 : instruction table modified 9. p93 : remove SUBLW instruction 10. p95 : add table read instructions
1.1	Feb, 2017	1. p5:remove dual system clock statement 2. p7:remove 24-pin DIP package type 3. p9:remove 24-pin DIP pin assignment 4. p11-12:remove SKINNY DIP 24-pin (300mil) 5. p104:remove 24-Skinny DIP Package Dimension 6. p28:remove "Dual" from section name and related statement below 7. p30:remove "Dual System" from section name 8. p3:modified 2.5/2.6 section name
1.2	Mar, 2018	1. p7,p100 modify packages type 2. p6 modify operating voltage & freq.

CONTENTS

FEATURES

1. **ROM: 4K x 14 bits OTP, or 2K x 14 bits TTP**
2. **RAM: 184 x 8 bits**
3. **STACK: 6 Levels**
4. **System Oscillation Sources (Fsys)**
   - Fast-clock
     - FXT (Fast Crystal): 1M~24 MHz
     - FIRC (Fast Internal RC): 2/4/8/16 MHz
   - Slow-clock
     - SXT (Slow Crystal): 32768 Hz
     - SIRC (Slow Internal RC): 150K/37.5K/9.4K/2.3 KHz @5V; 116K/29K/7.25K/1.8 KHz @3V
5. **Power Saving Operation Mode**
   - FAST mode: Slow-clock can be disabled or enabled
   - SLOW mode: Fast-clock stops, CPU is running
   - Fast Mode and Slow Mode can be chosen by CPUCKS control bit.
   - STOP mode: All Clocks stop, Wake-up Timer is disabled or enabled
6. **Operation Voltage and Speed: VDD=2.2V @4 MHz**
7. **2 Independent Timers**
   - Timer0
     - 8-bit timer divided by 1~256 pre-scaler option, Counter/Interrupt/Stop function
     - Capture – high duty or low duty (pulse width measurement)
     - Overflow and Toggle out
   - Timer1
     - 16-bit timer with two pre-scalers, Counter/Interrupt/Stop/Clear&Hold/Set/Reload function
     - Capture – period time
     - Overflow and Toggle out
8. **Interrupt**
   - Three External Interrupt pins
     - 2 pins are falling edge wake-up triggered
     - 1 pin is rising or falling edge wake-up triggered
   - Timer0/Timer1/WKT (wake-up) Interrupts
   - PWM0/PWM1 Interrupt

9. **Port B individual pin low level wake up**
10. **Wake-up (WKT) Timer**
    - Clocked by built-in RC oscillator with 4 adjustable Interrupt times
      1.1 ms/2.2 ms/36 ms/144 ms @5V, 1.4 ms/2.8 ms/46 ms/184 ms @3V
11. **Watchdog Timer**
    - Clocked by built-in RC oscillator with 4 adjustable Reset Times
      144 ms/289 ms/1155 ms/2312 ms @5V, 184 ms/367 ms/1469 ms/2939 ms @3V
      Watchdog timer can be disabled/enabled in STOP mode (WDTSTP, (R0D.5))
12. **2 Independent PWMs**
    - PWM0:
      - 8+2 bits, period-adjustable/duty-adjustable/Clear&Hold
      - Clock source: 16 MHz (double of FIRC 8MHz) or system clock (Fsys)
      - With differential output pair
      - Non-overlap durations adjustable
    - PWM1:
      - 8+2 bits, period-adjustable/duty-adjustable/Clear&Hold
      - Clock sources: 16 MHz (double of FIRC 8 MHz) or system clock (Fsys)
      - With differential output pair
      - Non-overlap durations adjustable
13. **12-bit ADC converter with 12 input channels**
14. **16 channel Touch Key (TM57PT45 only)**
15. **2 Operational Amplifiers with series/parallel applications**
16. **Reset Sources**
    - Power On Reset / Watchdog Reset / Low Voltage Reset / External Pin Reset
17. **Low Voltage Reset Option: LVR2.0V (Battery application don't suggest to select this level), LVR2.0V disable in STOP mode, LVR2.9V**
18. **Operating Voltage: LVR Level to 5.5V** **(Ref. Characteristic Graph of Fsys Minimum Operating Voltage & LVR)**
    - Fsys=4 MHz @2.2V~5.5V & LVR=2.9V & -40℃ < Ta< 85℃
    - Fsys=8 MHz @2.4V~5.5V & LVR=2.9V & -40℃ < Ta< 70℃
    - Fsys=12 MHz @2.8V~5.5V & LVR=2.9V
    - Fsys=16 MHz @3.3V~5.5V & LVR=2.9V

19. Enhanced Power Noise Rejection.

20. Instruction set: 36 Instructions

21. Instruction Execution Time

- 2 oscillation clocks per instruction except branch

22. I/O ports: Maximum 22 programmable I/O pins

- Pseudo-Open-Drain Output
- Open-Drain Output
- CMOS Push-Pull Output
- Schmitt Trigger Input with pull-up resistor option
- 6 high sink current output pins

23. Package Types:

- 16-pin SSOP (150mil)
- 20-pin DIP (300 mil), SOP (300 mil)
- 24-pin SOP (300 mil), SSOP (150 mil)

24. Supported EV board on ICE

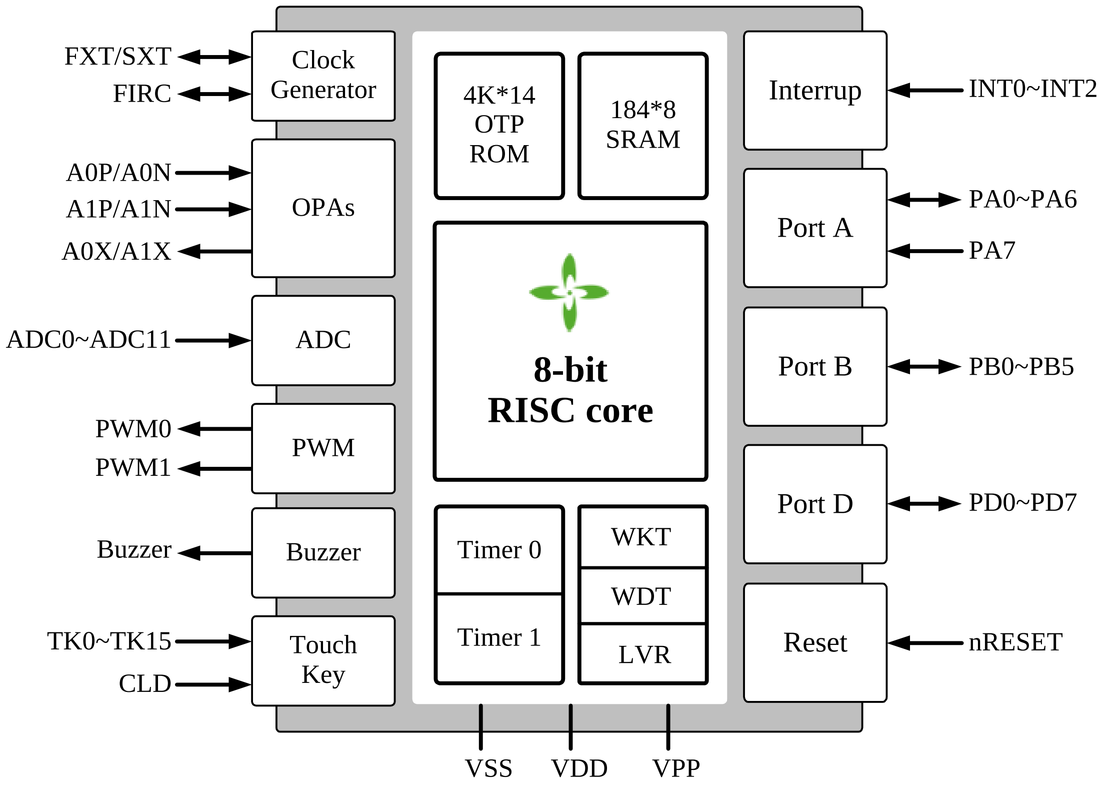
EV board: EV2768

Difference between TM57PT45/PA45 and TM57PT45C/PA45C

	PWM0P, PWM0N PWM1P, PWM1N	
	Drive Current	Sink Current
TM57PT45/PA45	8 mA @VDD=5V 3 mA @VDD=3V	20 mA @VDD=5V 10 mA @VDD=3V except PB0(PWM1P)
TM57PT45C/PA45C	27 mA @VDD=5V 10 mA @VDD=3V	45 mA @VDD=5V 20 mA @VDD=3V

TM57PT45C/PA45C enlarge the drive and sink currents of the PWM0P, PWM0N, PWM1P, and PWM1N.

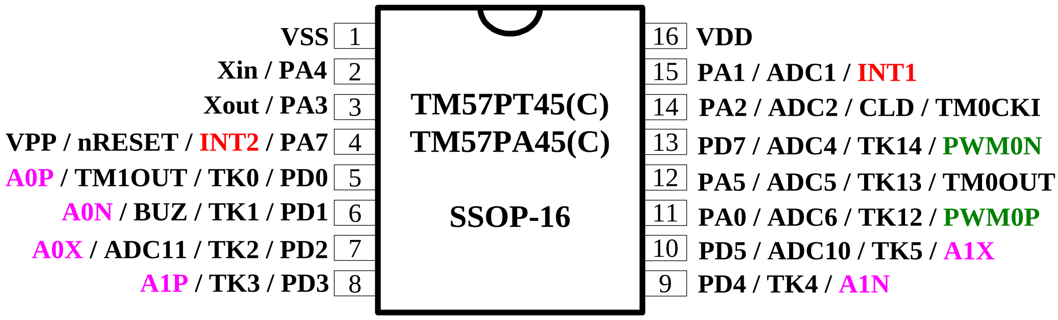
BLOCK DIAGRAM

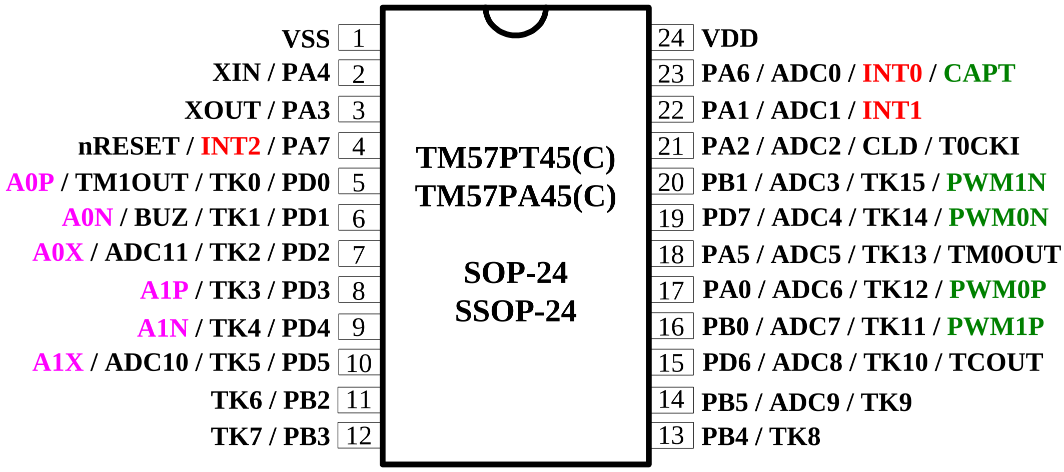


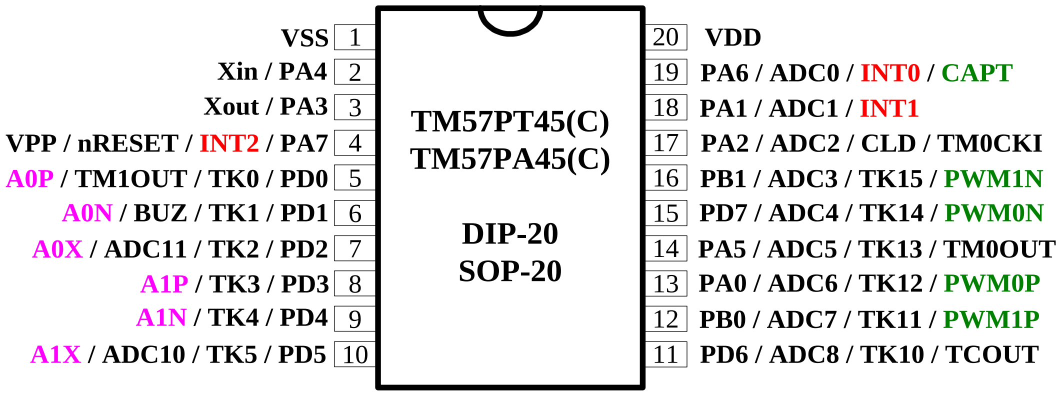
TM57PT45 Block Diagram

Note that Touch Key block is only for TM57PT45/PT45C

PIN ASSIGNMENT







*** Note that TM57PA45/PA45C do not has TK0~TK15 and CLD pins.**

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 PB0–PB5 PD0–PD7	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.
PB0, PB2 PB3, PB4 PB5, PD6	I/O	High Sink current pins
VPP/nRESET/ PA7	I	Schmitt-trigger input with pull-high configurable, External active low reset, normal stay to "high".
Xin, Xout	–	Crystal/Resonator oscillator connection for system clock.
VDD, VSS	P	Power Voltage input pin and ground
VPP	I	PROM programming high voltage input
INT0–INT2	I	External interrupt input
PWM0N PWM0P PWM1P PWM1N	O	PWM outputs
TCOUT	O	Instruction cycle clock divided by N output. Where N is 1,2,4,8. The instruction clock frequency is system clock frequency divided by two (Fsys/2).
TM0CKI	I	Timer0's input in counter mode
CAPT	I	Timer0/Timer1 Capture input
BUZ	O	Buzzer output
TM0OUT	O	Timer0 overflow toggle output
TM1OUT	O	Timer1 overflow toggle output
ADC0~ADC11	I	A/D converter input
TK0~TK15	I	Touch Key input (for TM57PT45 only)
CLD	I	Touch Key capacitor input (for TM57PT45 only)
A0P, A1P	I	Positive inputs of OPA0 and OPA1
A0N, A1N	I	Negative inputs of OPA0 and OPA1
A0X, A1X	O	Outputs of OPA0 and OPA1

PROGRAMMING PINS:

VDD/VSS/PA0/PA1/PA3/PA4/PA7 (VPP)

PIN SUMMARY

Pin Number		Pin Name	Type	GPIO					Function After Reset	Alternate Function				
				Input		Output								
24-SOP/ SSOP	20-SOP/DIP			Weak Pull-up	Ext. Interrupt	O.D	P.O.D	P.P		PWM	Touch Key	ADC	OPA	MISC
1	1	VSS	P											
2	2	Xin/PA4	I/O	○		○		○	PA4					Xin
3	3	Xout/PA3	I/O	○		○		○	PA3					Xout
4	4	VPP/nRESET/ INT2/PA7	I/O	○	○	○			PA7					VPP nRESET
5	5	A0P/TM1OUT/ TK0/PD0	I/O	○		○		○	PD0		○		○	TM1OUT
6	6	A0N/BUZ/TK1/PD1	I/O	○		○		○	PD1		○		○	BUZ
7	7	A0X/ADC11/TK2/ PD2	I/O	○		○		○	PD2		○	○	○	
8	8	A1P/TK3/PD3	I/O	○		○		○	PD3		○		○	
9	9	A1N/TK4/PD4	I/O	○		○		○	PD4		○		○	
10	10	A1X/ADC10/TK5/ PD5	I/O	○		○		○	PD5		○	○	○	
11	-	TK6/PB2	I/O	○		○		○	PB2		○			H.S.
12	-	TK7/PB3	I/O	○		○		○	PB3		○			H.S.
13	-	TK8/PB4	I/O	○				○	PB4		○			H.S.
14	-	TK9/ADC9 PB5	I/O	○		○		○	PB5		○	○		H.S.
15	11	TK10/TCOUT/ ADC8/PD6	I/O	○		○		○	PD6		○	○		TCOUT H.S.
16	12	TK11/PWM1P/ ADC7/PB0	I/O	○		○		○	PB0	○	○	○		H.S.
17	13	TK12/PWM0P /ADC6/PA0	I/O	○			○	○	PA0	○	○	○		
18	14	TK13/TM0OUT/ ADC5/PA5	I/O	○	○	○		○	PA5		○	○		TM0OUT
19	15	TK14/PWM0N/ ADC4/PD7	I/O	○	○	○		○	PD7	○	○	○		
20	16	TK15/PWM1N/ ADC3/PB1	I/O	○		○		○	PB1	○	○	○		

Pin Number		Pin Name	Type	GPIO					Function After Reset	Alternate Function				
				Input		Output								
24-SOP / SSOP	20-SOP/DIP			Weak Pull-up	Ext. Interrupt	O.D	P.O.D	P.P		PWM	Touch Key	ADC	OPA	MISC
21	17	CLD/TM0CKI/ ADC2/PA2	I/O	○			○	○	PA2		○	○		TM0CKI
22	18	INT1/ADC1/ PA1	I/O	○			○	○	PA1			○		
23	19	CAPT/INT0/ ADC0/PA6	I/O	○		○		○	PA6			○		CAPT
24	20	VDD	P											

Symbol: P.P. = Push-Pull Output
P.O.D. = Pseudo Open Drain
O.D. = Open Drain
H.S. = High Sink Pins

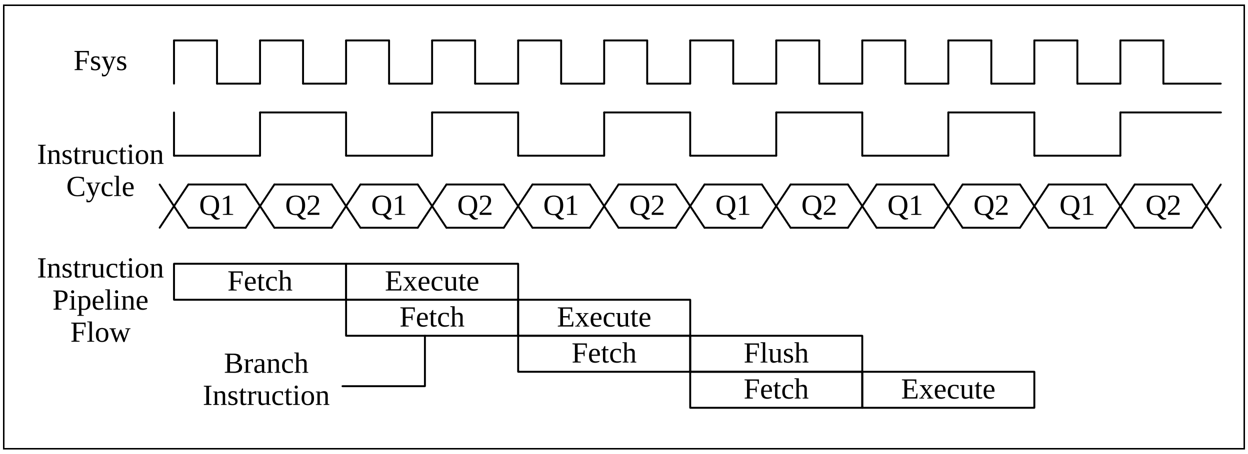
*** Note that TM57PA45 do not has TK0~TK15 and CLD pins.**

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.



Terminology definitions:

Fsys: System clock. The main clock that drives the core logic and all peripherals. The clock source can be either Fast-clock or Slow-clock which can be set by registers.

Fast-clock: The clock source that contains Fast crystal (FXT), Fast Internal RC oscillator (FIRC), and External RC oscillator (XRC). *

Slow-clock: The clock source that contains Slow crystal (SXT), Slow Internal RC oscillator (SIRC), and External RC oscillator (XRC).

Instruction Cycle=Fsys/2

FXT: Fast Crystal

FIRC: Fast Internal RC oscillator

*XRC: Fast or Slow External RC oscillator

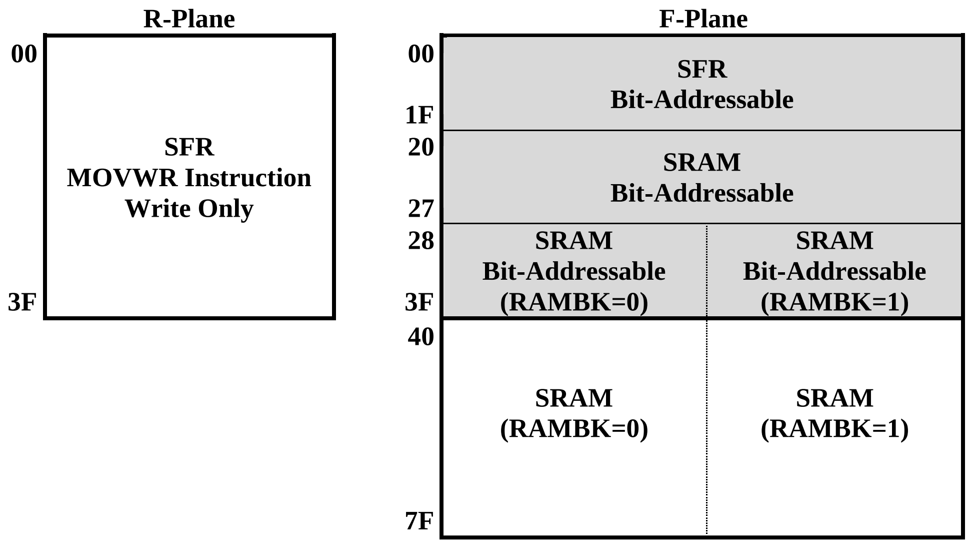
SXT: Slow Crystal (32 KHz)

SIRC: Slow Internal RC oscillator

* TM57PT45/PA45/PT45C/PA45C doesn't support XRC mode.

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. 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, 5    ; Set RAMBK=0
LOOP:
        MOVLW    00H
        MOVWF    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, 2    ; 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 12-bit wide capable of addressing a 4K 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 12 bits address from instruction word. For RET/RETI/RETLW instructions, PC retrieves its content from the top level STACK. For the other instructions updating PC[7:0], the PC[11:8] keeps unchanged. Therefore, the data of a lookup table must be located with the same PC[11:8]. The STACK is 12-bit wide and 6-level in depth. The CALL instruction and hardware interrupt will push STACK level in order. While the RET/RETI/RETLW instructions pop the STACK level in order.

◇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
                                        ; is stored back in PCL
          RETLW     55H                 ; W=55H when return
          RETLW     56H                 ; W=56H when return
          RETLW     58H                 ; W=58H when return
```

Note: TM57PT45 defines 256 ROM addresses as one page, so that TM57PT45 has 16 pages, 000H~0FFH, 100H~1FFH, 200H~2FFH, …, and F00H~FFFH. On the other words, PC[11:8] 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 does not need to set the starting address at X00H, just only confirm all lookup table data are located at the same page.

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.

<table>
<tr><th>STATUS</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td>Reset Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R</td><td>R</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Bit</td><td colspan="8">Description</td></tr>
<tr><td>7</td><td colspan="8">GB0: General Purpose Bit 0</td></tr>
<tr><td>6</td><td colspan="8">GB1: General Purpose Bit 1</td></tr>
<tr><td>5</td><td colspan="8">RAMBK: SRAM Bank Selection
0: SRAM Bank0
1: SRAM Bank1</td></tr>
<tr><td>4</td><td colspan="8">TO: Time Out Flag
0: after Power On Reset, LVR Reset, or CLRWDT/SLEEP instructions
1: WDT time out occurs</td></tr>
<tr><td>3</td><td colspan="8">PD: Power Down Flag
0: after Power On Reset, LVR Reset, or CLRWDT instruction
1: after SLEEP instruction</td></tr>
<tr><td>2</td><td colspan="8">Z: Zero Flag
0: the result of a logic operation is not zero
1: the result of a logic operation is zero</td></tr>
<tr><td rowspan="3">1</td><td colspan="8">DC: Decimal Carry Flag or Decimal/Borrow Flag</td></tr>
<tr><td colspan="4">ADD instruction</td><td colspan="4">SUB instruction</td></tr>
<tr><td colspan="4">0: no carry
1: a carry from the low nibble bits of the result occurs</td><td colspan="4">0: a borrow from the low nibble bits of the result occurs
1: no borrow</td></tr>
<tr><td rowspan="3">0</td><td colspan="8">C: Carry Flag or/Borrow Flag</td></tr>
<tr><td colspan="4">ADD instruction</td><td colspan="4">SUB instruction</td></tr>
<tr><td colspan="4">0: no carry
1: a carry occurs from the MSB</td><td colspan="4">0: a borrow occurs from the MSB
1: no borrow</td></tr>
</table>

◇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 WDT time out event occurs

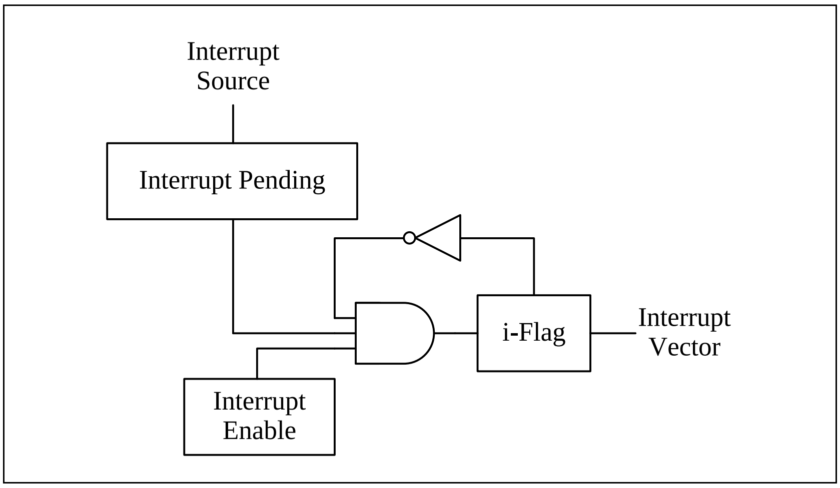
```
  LOOP:
                BTFSC     STATUS, TO          ; Check the LVD flag
                GOTO      WDT_Timeout_Proc    ; If TO=1, goto WDT_Timeout_Proc
  MAIN:         …….
```

1.6 Interrupt

The TM57PT45/PA45/PT45C/PA45C has 1 level, 1 vector and 8 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 TM57PT45/PA45/PT45C/PA45C has only 1 vector, there is not an interrupt priority register. The interrupt priority is determined by F/W.

If the corresponding interrupt enable bit has been set (INTE), it would trigger CPU to service the interrupt. CPU accepts interrupt in the end of current executed instruction cycle. In the mean while, a "CALL 001" instruction is inserted to CPU, and i-flag is set to prevent recursive interrupt nesting.

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



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

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

            ORG      001H          ; All interrupt vector
            GOTO     INT           ; If INT1 (PA1) input occurred rising edge

            ORG      002H
START:
            MOVLW    xxxxxx0xB
            MOVWR    PAPUN         ; Select INT1 (PA1) pin mode pull-up enable

            MOVLW    xxxxxx1xB
            MOVWF    PAD           ; Release INT1 (PA1), it becomes Schmitt-trigger
                                   ; input mode with input pull-up resistor

            MOVLW    xxxxxx0xB
            MOVWR    PAE           ;Disable INT1(PA1) push-pull output

            MOVLW    1xxxxxxxB
            MOVWR    R0D           ; Set INT1 interrupt trigger as rising edge
            MOVLW    11111101B
            MOVWF    INTF          ; Clear INT1 interrupt request flag
            MOVLW    00000010B
            MOVWF    INTE          ; Enable INT1 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    XINT1F        ; Check XINT1F bit
            GOTO     EXIT_INT      ; XINT1F=0, exit interrupt subroutine
            …                      ; INT1 interrupt service routine
            MOVLW    11111101B
            MOVWF    INTF          ; Clear INT1 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	PWM0IE	PWM1IE	TM1IE	TM0IE	WKTIE	XINT2E	XINT1E	XINT0E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F08.7 **PWM0IE**: PWM0 interrupt enable
0: disable
1: enable

F08.6 **PWM1IE**: PWM1 interrupt enable
0: disable
1: enable

F08.5 **TM1IE**: Timer1 interrupt enable
0: disable
1: enable

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

F08.3 **WKTIE**: WKT interrupt enable
0: disable
1: enable

F08.2 **XINT2E**: External pin XINT2 (PA7) interrupt enable
0: disable
1: enable

F08.1 **XINT1E**: External pin XINT1 (PA1) interrupt enable
0: disable
1: enable

F08.0 **XINT0E**: External pin XINT0 (PA6) interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	PWM0IF	PWM1IF	TM1IF	TM0IF	WKTIF	XINT2F	XINT1F	XINT0F
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F09.7 **PWM0I**: PWM0 interrupt event pending flag
This bit is set by H/W while PWM0 period is completed, write 0 to this bit will clear this flag

F09.6 **PWM1IF**: PWM1 interrupt event pending flag
This bit is set by H/W while PWM1 period is completed, write 0 to this bit will clear this flag

F09.5 **TM1IF**: Timer1 interrupt event pending flag
This bit is set by H/W while Timer1 overflows, 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

F09.3 **WKTIF**: WKT interrupt event pending flag
This bit is set by H/W while WKT overflows, write 0 to this bit will clear this flag

F09.2 **XINT2F**: 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 **XINT1F**: INT1 interrupt event pending flag
This bit is set by H/W at INT1 pin’s falling/rising edge, write 0 to this bit will clear this flag

F09.0 **XINT0F**: INT0 interrupt event pending flag
This bit is set by H/W at INT0 pin’s falling edge, write 0 to this bit will clear this flag

R0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0D	INT1EDG	TM1CM	WDTSTP	FIRCKS		ADCKS		
R/W	–	W		W		–		
Reset	–	0	0	0		–		

R0D.7 **INT1EDG:** INT1 pin (PA1) edge interrupt event
0: falling edge to trigger
1: rising edge to trigger

2. Chip Operation Mode

2.1 Reset

The TM57PT45/PA45/PT45C/PA45C can be RESET in four ways.

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

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

There are two voltage selections for the LVR threshold level, one is higher level which is suitable for application with V_{DD} is more than 3V, the other one is suitable for application with V_{DD} is more than 2.0V. 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
LVR2.0	5.5V > V_{DD} > 2.0V
LVR2.9	5.5V > V_{DD} > 3.3V or V_{DD}=5.0V

Different Fsys have different system minimum operating voltage, reference to Operating Voltage of DC characteristics, if current system voltage is lower than minimum operating voltage and lower LVR is selected, then the system maybe enter dead-band and error occur.

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

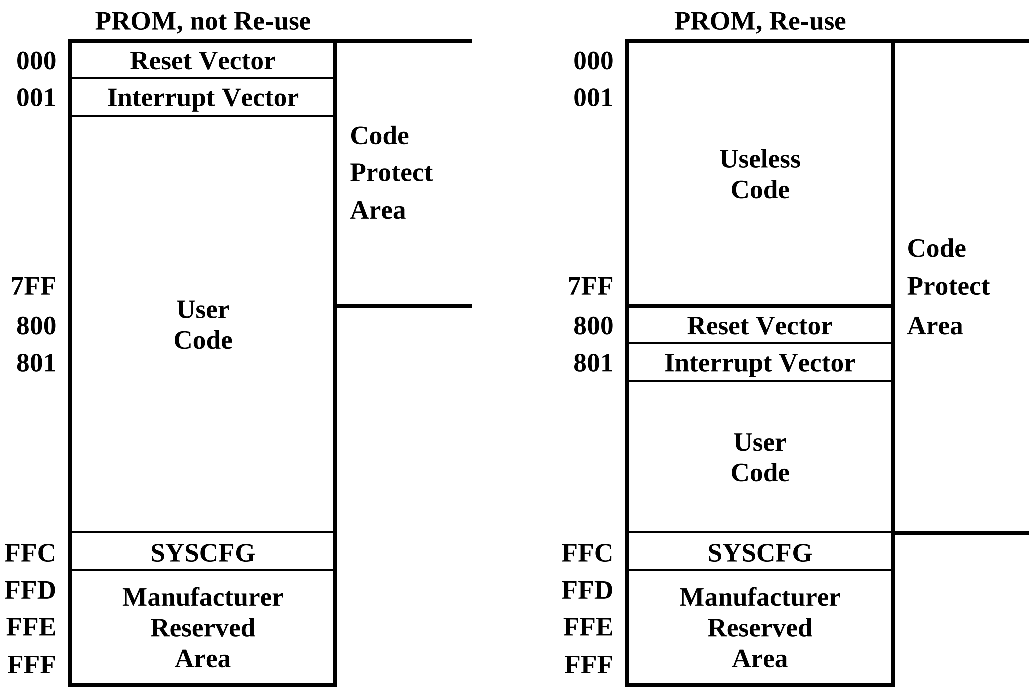
2.2 System Configuration Register (SYSCFG)

The System Configuration Register (SYSCFG) is located at ROM address FFCh. The SYSCFG determines the option for initial condition of MCU. It is written by PROM Writer only. User can select LVR threshold voltage and chip operation mode by SYSCFG register. The default value of SYSCFG is 3FFFh. The 14th 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 disable
	01	LVR=2.9V, always enable
	10	LVR=2.0V, disable at STOP mode
	11	LVR=2.0V; always enable
9-8	**CLKT**: Clock type selection	
	01	FIRC
	10	Slow Crystal
	11	Fast Crystal
7	**XRSTE**: External Pin (PA7) Reset Enable	
	0	Disable, PA7 as IO pin
	1	Enable
6	**WDTE**: WDT Reset Enable	
	0	WDT Reset Disable
	1	WDT Reset Always Enable
5-0	**Reserved**	

2.3 PROM Re-use ROM

The PROM of this device is 4K words. For some F/W program, the program size could be less than 2K words. To fully utilize the PROM, the device allows users to reuse the PROM. This feature is named as Two Time Programmable (TTP) ROM. While the first half of PROM is occupied by a useless program code and the second half of the PROM remains blank, users can re-write the PROM with the updated program code into the second half of the PROM. In the Re-use mode, the Reset Vector and Interrupt Vector are re-allocated at the beginning of the PROM’s second half by the Assembly Compiler. Users simply choose the “REUSE” option in the ICE tool interface, 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 800h. 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

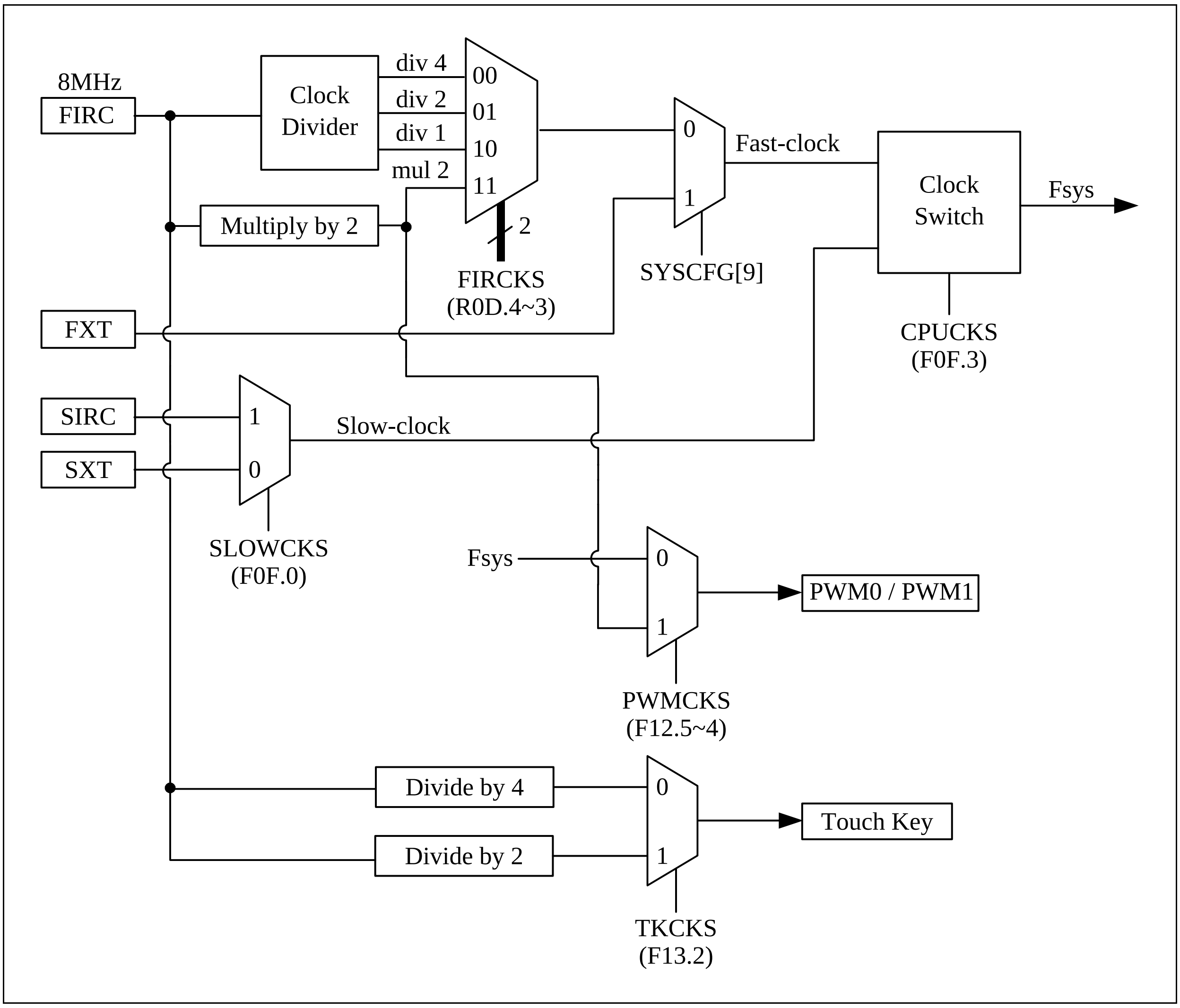
The Power-down mode of TM57PT45/PA45/PT45C/PA45C has only STOP Mode. It is activated by SLEEP instruction. During the Power-down mode, the system clock and peripherals stop to minimize power consumption. The WDT is working or not depends on SYSCFG. The WKT is working or not depends on WKTIE (MF08.3). The Power-down mode can be terminated by Reset, or enabled Interrupts (External pins and WKT) 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) Mode

2.5 System Clock

TM57PT45/PA45/PT45C/PA45C four kinds of clock source, FXT (Fast Crystal) Clock, SXT (Slow Crystal) Clock, SIRC (Slow Internal RC) Clock and FIRC (Fast Internal RC) Clock. Each clock source can be applied to CPU kernel as system clock source. Refer to the Figure as below.



FAST Mode:

After power-on or reset, TM57PT45/PA45/PT45C/PA45C enters FAST or SLOW mode depends on SYSCFG[9:8]. In FAST mode, TM57PT45/PA45/PT45C/PA45C can select FXT or FIRC as its CPU clock. TM57PT45/PA45/PT45C/PA45C enters FAST mode by setting the CPUCKS (F0F.3) when it is in SLOW mode. If user wants to change to SLOW mode, Slow-clock should be enabled first (F0F.2=1), then switch to Slow-clock as CPU clock (F0F.3=1), turn off Fast-clock (F0F.4=1) in the end.

In this mode, the program is executed using Fast-clock as system clock source. The Timer0 and Timer1 blocks are driven by Fast-clock. PWMs can be driven by Fast-clock or FIRC 16 MHz by setting PWMCKS (F12.5~4).

SLOW Mode:

After power on or reset, TM57PT45/PA45/PT45C/PA45C enters SLOW mode if SYSCFG[9:8]=10. User can select SXT or SIRC as its System clock by setting SLOWCKS (F0F.0). However, changing Slow-clock type under SLOW mode is not allowed. User should let TM57PT45/PA45/PT45C/PA45C enter FAST mode first, change SLOWCKS, then back to SLOW mode.

IDLE Mode:

The TM57PT45/PA45/PT45C/PA45C does not support IDLE mode because there is no T2 exist in this model.

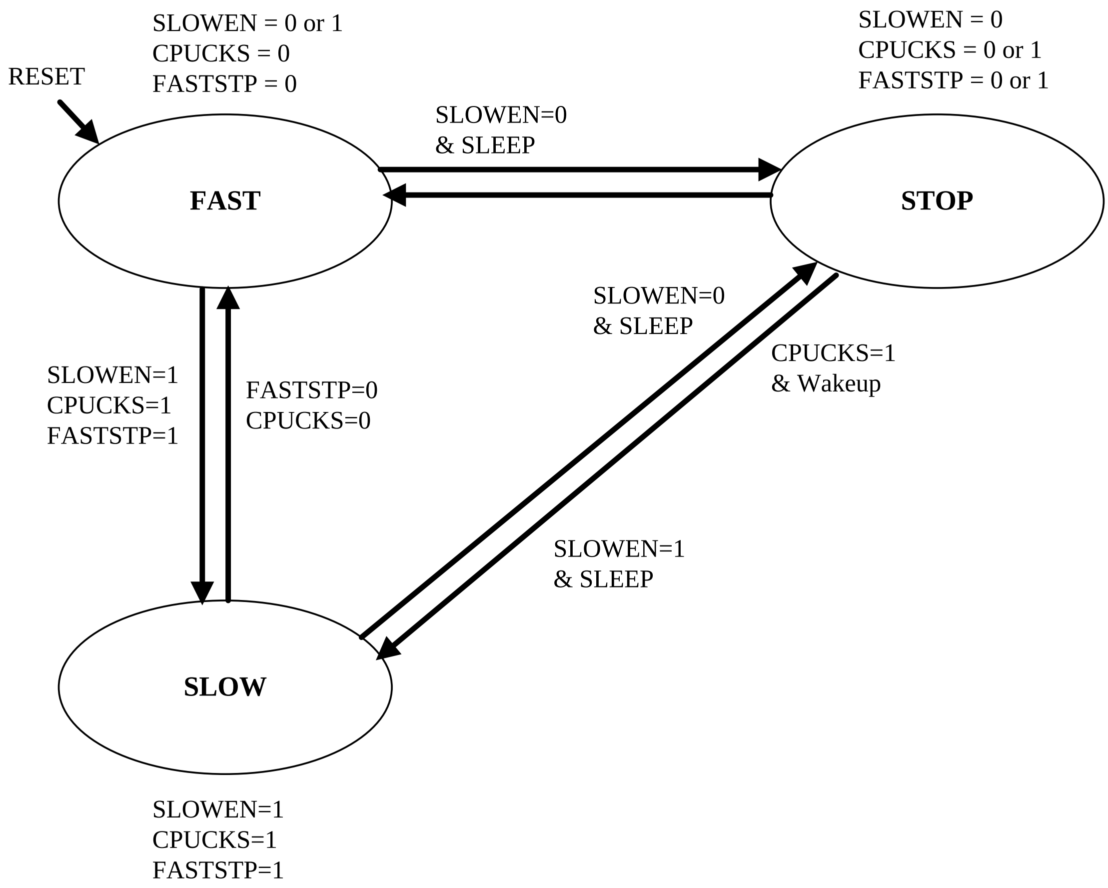
STOP Mode:

If Slow-clock is disabled, all blocks will be turned off and the TM57PT45/PA45/PT45C/PA45C 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 Clock Modes Transition

TM57PT45/PA45/PT45C/PA45C is operated in one of three modes: FAST Mode, SLOW Mode, and STOP Mode.

Modes Transition Diagram:



Note:

- SLEEP denotes SLEEP instruction
- Wakeup denotes wake-up events, such as External pin, WKT, or Port B pin wake-up
- Slow-clock will be turn-on (SLOWEN=1) if one of the following conditions is true:
  1. WDT is enabled (SYSCFG.6=1)
  2. Wakeup Timer (WKT) interrupt is enabled
  3. SLOWCKS=1 and (CPUCKS=1 or SLOWEN=1)

  Make sure the above 3 conditions is false to enter STOP mode.
- CPUCKS (F0F.3), FASTSTP (F0F.4), SLOWEN (F0F.2)

CPU Mode & Clock Functions Table:

Mode	Oscillator	Fsys	Fast-clock	Slow-clock	TM0	TM1	PWM0/1	Wakeup event
FAST	FIRC, FXT	Fast-clock	Run	Run	Run	Run	Run	X
SLOW	SIRC, SXT	Slow-clock	Run	Run	Run	Run	Run	X
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 (F0F.0). 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=0, SIRC: SLOWCKS=1)

(2) Switch system clock source to Slow-clock (CPUCKS = 1)

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

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

```
BSF     SLOWCKS     ; Select SIRC as Slow-clock source
BSF     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 SYSCFG[9]. If SYSCFG[9] 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) Enable Fast-clock (FASTSTP=0)

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

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

```
BCF     FASTSTP     ; Enable Fast-clock
BCF     CPUCKS      ; Switch system clock source to Fast-clock
```

STOP Mode Setting:

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

(1) Stop Slow-clock (SLOWEN=0)

(2) Execute SLEEP instruction

Besides SLOWEN=0, user must make sure all possibilities to make Slow Internal RC running are disabled. First, make sure WDT is not enabled. Second, WKT interrupt is not enabled. Third, SLOWCKS is not set to SIRC and CPUCKS is not set to Slow-clock.

STOP mode can be woken up by interrupt (INT0, INT1, INT2), WKT, or PB0-5 pins low level wake up.

◇Example: Switch operating mode to STOP mode

```
BCF     SLOWEN      ; Stop Slow-clock
SLEEP               ; Enter STOP mode
```

IO setting notes in STOP 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 is as below.

	Fast-clock	Slow-clock	PAE3	PAPUN3	PAD3	PAE4	PAPUN4	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

F0F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCTL	–	–	–	FASTSTP	CPUCKS	SLOWEN	–	SLOWCKS
R/W	–	–	–	R/W	R/W	R/W	–	R/W
Reset	–	–	–	0	0	0	–	1

F0F.4 **FASTSTP**: Fast-clock Enable / Disable
0: enable
1: disable

F0F.3 **CPUCKS**: System clock source select
0: Fast-clock
1: Slow-clock

F0F.2 **SLOWEN**: Slow-clock Enable / Disable
0: disable in Power-down mode
1: enable

F0F.0 **SLOWCKS**: Slow-clock type select
0: SXT
1: SIRC

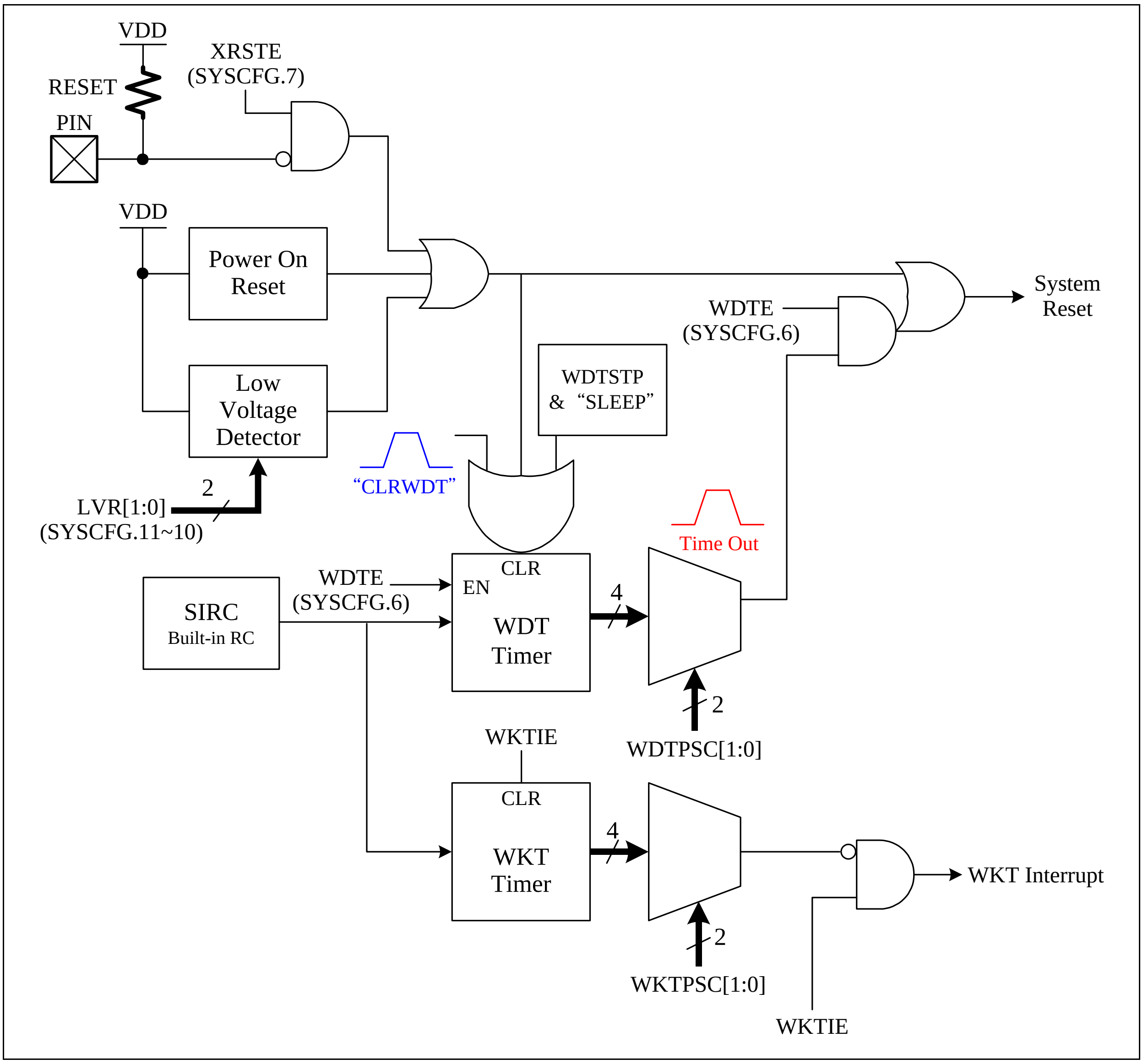
Warning: The CLKCTL (F0F) can't be set directly for CPU modes transition. It may cause the transition fail. Please refer the mentioned steps for transition in this chapter.

3. Peripheral Functional Block

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

The WDT and WKT share the same internal RC oscillator (SIRC). The overflow period of WDT, WKT can be selected by WDTPSC[1:0] and WKTPSC[1:0]. The WDT timer is cleared by the CLRWDT instruction. If the Watchdog is enabled (WDTE=1), the WDT generates the chip reset signal when WDT overflows. Set WDTSTP (R0D.5) to '1' can let WDT timer stop counting after executing SLEEP instruction, i.e. WDTSTP=0 WDT timer always keeps counting even if the SLEEP instruction is executed.

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



WDT/WKT Block Diagram

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

Mode	WDTE	WKTIE	WDTSTP	Internal SIRC Oscillator
Normal Mode	0	0	0/1	Stop
	0	1		Run
	1	0		
	1	1		
Power Down Mode	0	0	0	Stop
	0	1	0	Run
	1	0	0	Run
	1	1	0	Run
	0	0	1	Stop
	0	1	1	Run
	1	0	1	Stop
	1	1	1	Run

F03	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	GBIT1	GBIT0	RAMBK	TO	PD	Z	DC	C
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

R0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0C	WKTPSC		WDTPSC		TM1CKS	TM0OE	TCOE	TM1OE
R/W	–	W		W	W	W	W	W
Reset	–	0	0	0	0	0	0	0

R0C.7~6 **WKTPSC:** WKT pre-scale select: *(the time IS NOT precise enough for accurate timing applications)*

Bit 1	Bit 0	5V	3V
0	0	1.1 ms	1.4 ms
0	1	2.2 ms	2.7 ms
1	0	36 ms	44 ms
1	1	143 ms	177 ms

R0C.5~4 **WDTPSC:** WDT pre-scale select: *(the time IS NOT precise enough for accurate timing applications)*

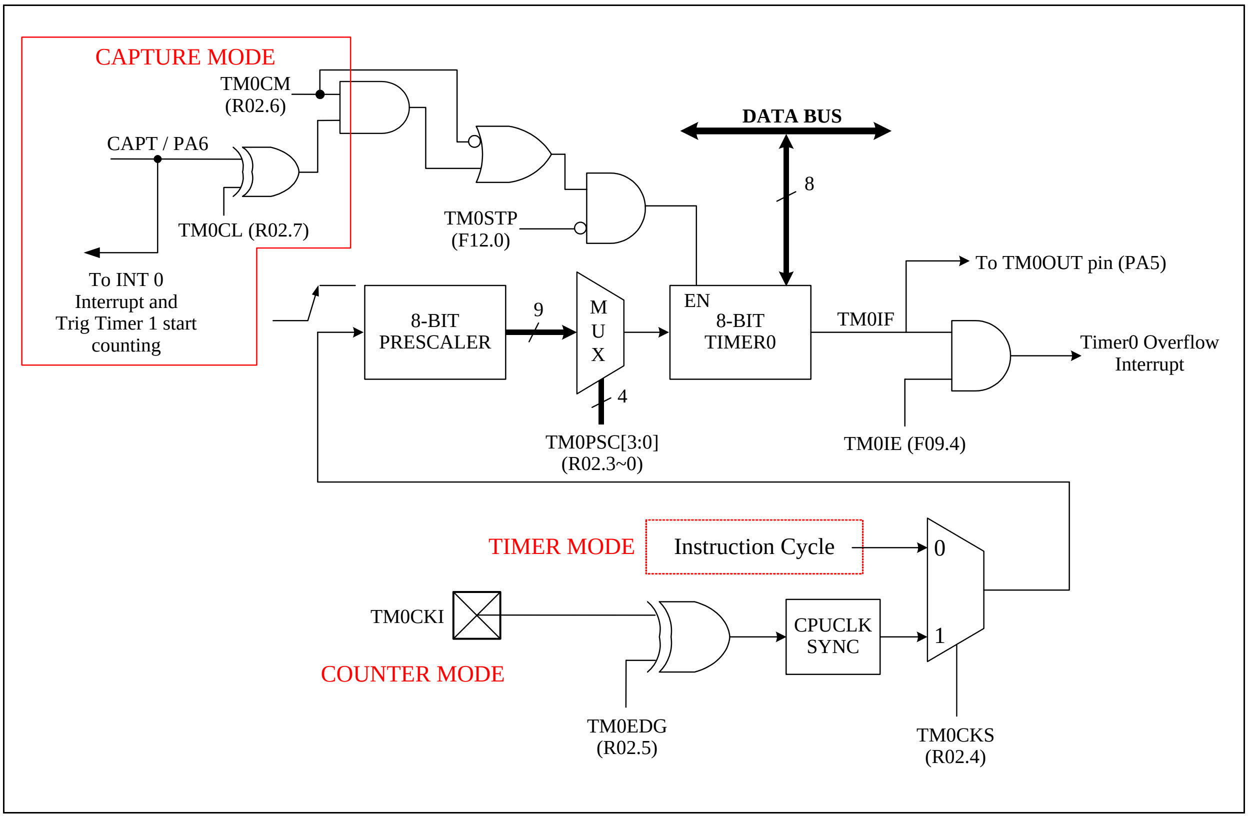
Bit 1	Bit 0	5V	3V
0	0	140 ms	175 ms
0	1	280 ms	355 ms
1	0	1140 ms	1440 ms
1	1	2280 ms	2880 ms

R0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0D	INT1EDG	TM1CM	WDTSTP	FIRCKS		ADCKS		
R/W	–	W		W		–		
Reset	–	0	0	0		–		

R0D.5 **WDTSTP:** WDT stops counting when in STOP mode
0: WDT keeps counting when in STOP mode
1: WDT stops counting when in STOP mode

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

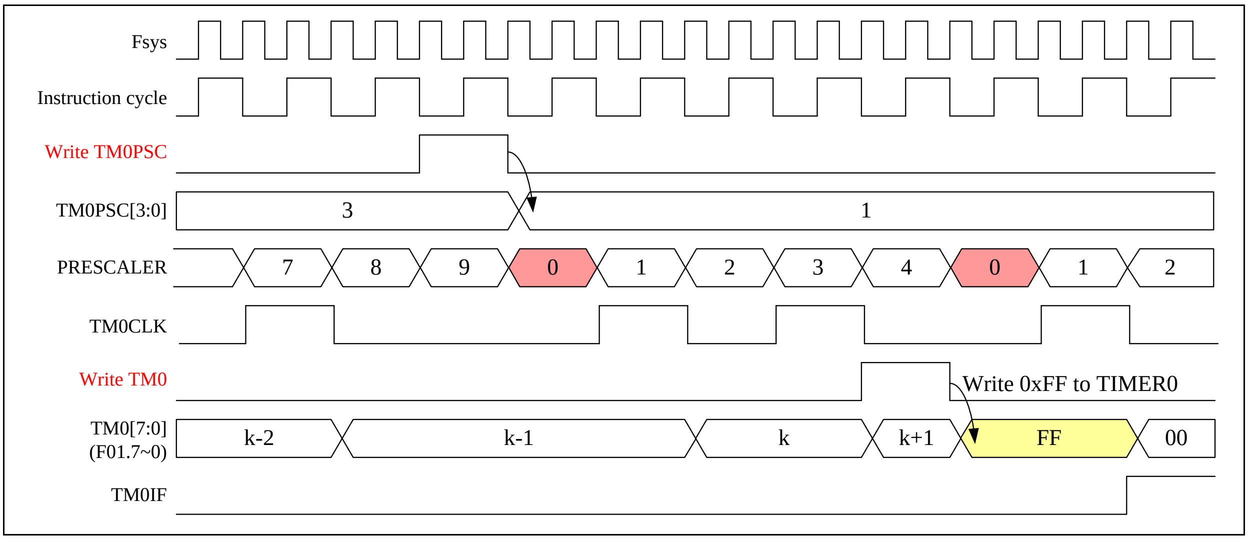
The Timer0 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatically rolls over based on the pre-scaled clock source, which can be the instruction cycle or TM0CKI (PA2) rising/falling input. The Timer0's 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 (F12.0) bit is set.



Timer0 Block Diagram

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

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



Timer0 works in Timer mode

The equation of TM0OUT initial value is as following.

TM0OUT output frequency=Instruction cycle/TM0PSC/ (256-TM0)

TM0OUT output time period=1/TM0OUT output frequency.

◇Example:

Setup TM0 Work in Timer mode and counting overflow toggle output to TM0OUT (PA5) pin configuration.

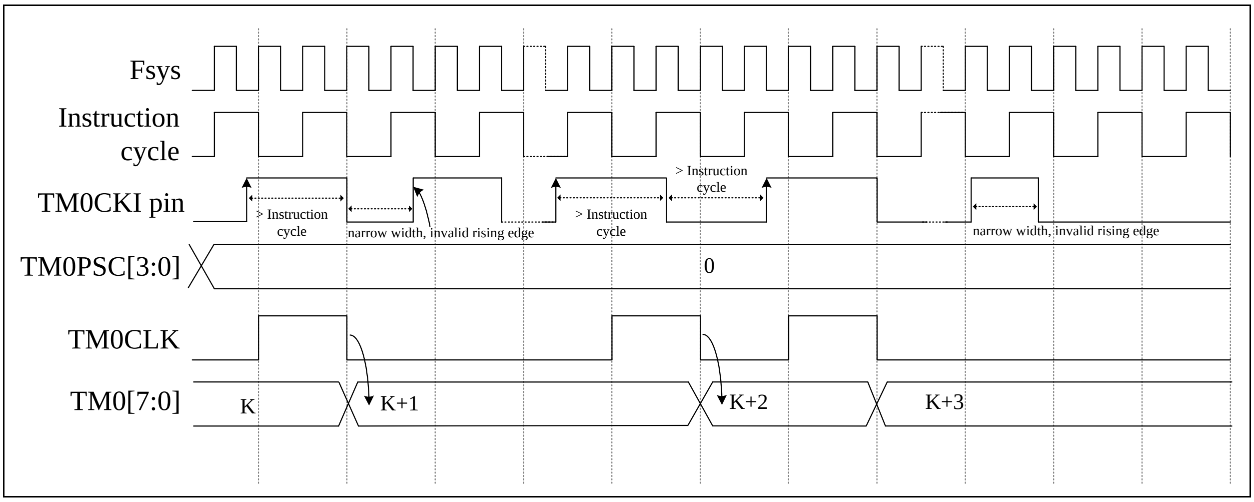
```
; Setup TM0 clock source and divider.
        MOVLW  00000101B
        MOVWR  R02            ; Setup  TM0=Timer mode.
                              ; TM0 clock source=Instruction cycle.
                              ; Divided by 32

; Set TM0 timer.
        BSF    TM0STP         ; Disable TM0 counting (Default "0").
        MOVLW  156
        MOVWF  TM0            ; Write 156 into TM0 register of F-Plane.

; Set TM0OUT pin function.
        MOVLW  11010100B
        MOVWR  R0C            ; Enable TM0 match toggle output to TM0OUT (PA5).

; Enable TM0 timer and interrupt function.
        MOVLW  11101111B      ; Clear TM0 request interrupt flag
        MOVWF  INTIF
        BSF    TM0IE          ; Enable TM0 interrupt function.
        BCF    TM0STP         ; Enable TM0 counting (Default "0").
```

Example:

TM0 clock source is Fsys=4 MHz, Instruction cycle=2 MHz, TM0PSC=/32, TM0=156,
TM0OUT output frequency=2 MHz/32/ (256-156) =2 MHz/32/100=312.5 Hz
TM0OUT output time period=1/312.5 Hz=3.2 ms.

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

TM0CKS=1 if Timer0 counter source clock is from TM0CKI pin. TM0CKI signal is synchronized by instruction cycle, which 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.



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

◇Example:

Setup TM0 Work in counter mode and clock source from TM0CKI pin (PA2) configuration.

```
; Setup TM0 clock source from TM0CKI pin (PA2) and divider.
        MOVLW  00010000B
        MOVWR  R02              ; Setup TM0=Counter mode.
                                ; Select TM0 prescaler counting edge=rising edge.
                                ; TM0 clock source=TM0CKI pin (PA2)
                                ; Divided by 1

; Set TM0 timer and stop TM0 counting.
        BSF    TM0STP           ; Disable TM0 counting (Default "0").
        MOVLW  00H
        MOVWF  TM0              ; Write 0 into TM0 register of F-Plane.

; Start TM0 count and read TM0 count.
        BCF    TM0STP           ; Enable TM0 counting.
        NOP
        NOP
        NOP
        BSF    TM0STP           ; Disable TM0 counting (Default "0")
        MOVFW  TM0
```

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	PWM0IE	PWM1IE	TM1IE	TM0IE	WKTIE	XINT2E	XINT1E	XINT0E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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	PWM0IF	PWM1IF	TM1IF	TM0IF	WKTIF	XINT2F	XINT1F	XINT0F
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

F12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF14	PWM0CLR	PWM1CLR	PWM0CKS	PWM1CKS	TM1SET	TM1CLR	TM1STP	TM0STP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	0	0	0	0	0

F12.0 **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	TM0CL	TM0CM	TM0EDG	TM0CKS	TM0PSC			
R/W	W	W	W	W	W			
Reset	0	0	0	0	0	0	0	0

R02.7 **TM0CL:** Timer0 Capture Mode Level
0: CAPT pin high level capture
1: CAPT pin low level capture

R02.6 **TM0CM:** Timer0 Mode Selection
0: Timer/Counter Mode , clock source from Instruction Cycle (Fsys/2) or TM0CKI
1: Capture Mode, counts CAPT pin level duration.

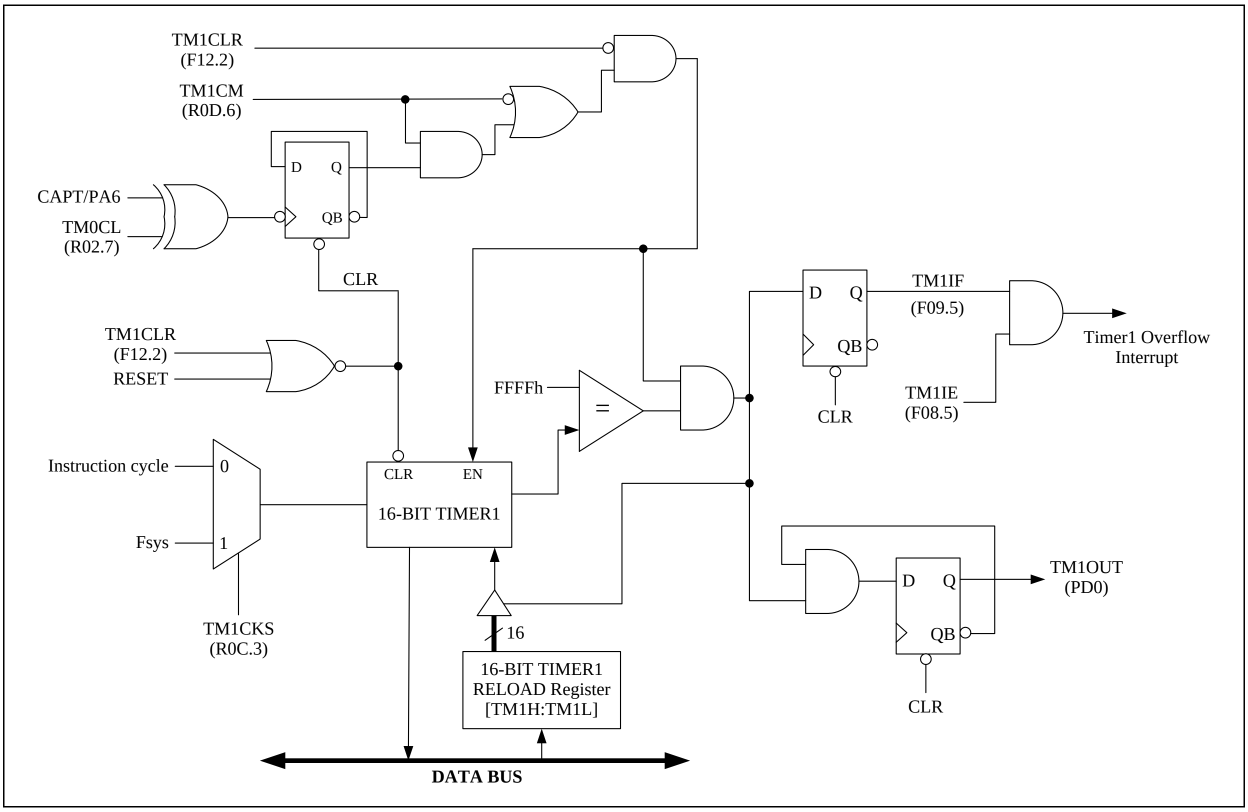
R02.5 **TM0EDG:** TM0CKI (PA2) edge selection for Timer0 prescaler count
0: TM0CKI (PA2) rising edge for Timer0 prescaler count
1: TM0CKI (PA2) falling edge for Timer0 prescaler count

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 Timer1

Timer1 is a 16-bit counter used as Capture/Timer mode with 16-bit auto-reload register. Timer1 can only be accessed by reading F-Plane TM1H and TM1L. Writing TM1H and TM1L is actually writing to Timer1 reload registers. The clock sources of Timer1 are Fsys and Instruction cycle, selected by TM1PSC. Setting the bit TM1CLR will clear Timer1 and hold Timer1 on 0000h. Setting the TM1STP bit will stop Timer1 counting. TM1OUT is an output signal that toggles when Timer1 overflow.



Timer1 Block Diagram

Note that writing to TM1H and TM1L is actually writing to Timer1 reload register, while reading TM1H and TM1L is actually reading the Timer1 counter itself. That is, Timer1 counter and Timer1 reload register share two addresses (0ah, 0bh) of F-Plane.

◇Example:

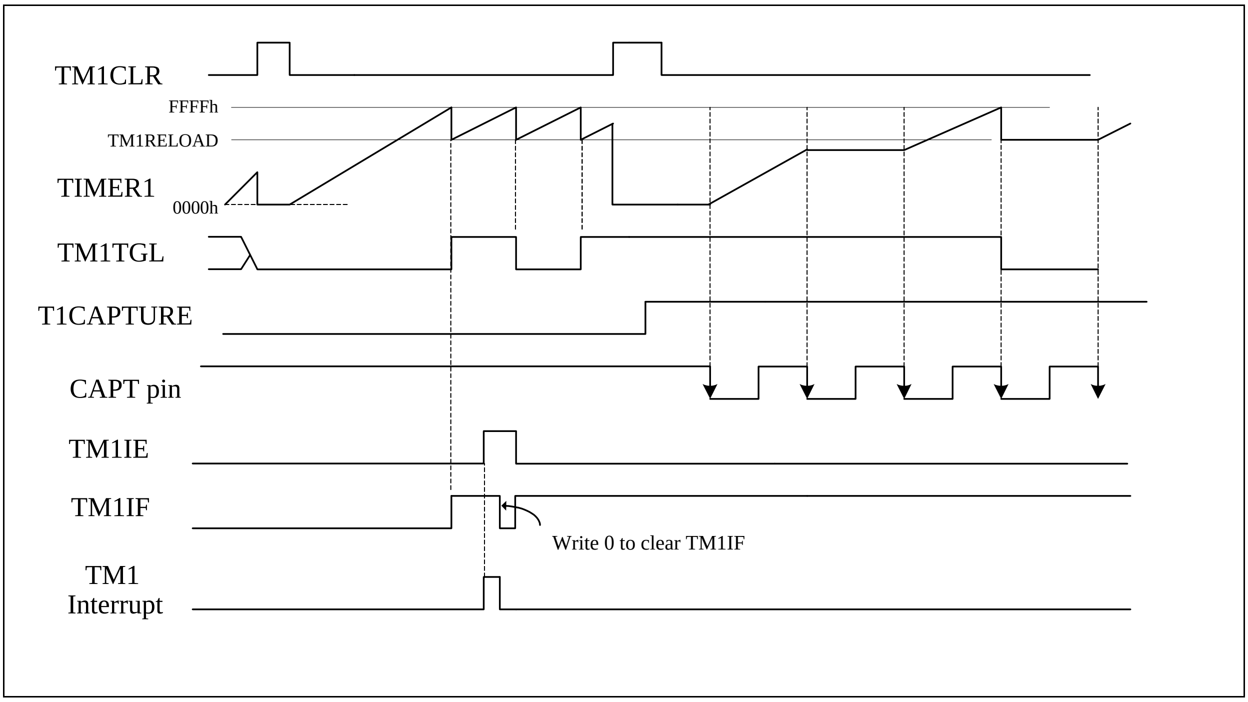
Setup TM1 Work in Timer mode and counting overflow toggle output to TM1OUT (PD0) pin configuration.

```
; Setup TM1 clock source and divider.
        MOVLW  00001001B     ; TM1OE=1 (Enable TM1OUT)
        MOVWR  R0C           ; TM1CKS=1 (Fsys as Timer1 clock source)
        MOVLW  0001000B      ; TM1CM=0 (Timer1 as timer mode)
        MOVWR  R0D

; Set TM1 timer.
        BSF    TM1STP        ; Stop TM1 counting (Default "0").
        BCF    TM1SET
        BSF    TM1CLR        ; Clear TM1 counter (Default "0").

        MOVLW  FFH
        MOVWF  TM1H          ; Write FFH into TM1 counting high byte.
        MOVLW  00H
        MOVWF  TM1L          ; Write 00H into TM1 counting low byte.

; Enable TM0 timer and interrupt function.
        MOVLW  11011111B     ; Clear TM1 request interrupt flag
        MOVWF  INTIF
        BSF    TM1IE         ; Enable TM1 interrupt function.

        BCF    TM1SET
        BCF    TM1CLR
        BCF    TM1STP        ; Enable TM1 counting (Default “0”).
```

Example:

TM1 clock source prescaler is Fsys=4 MHz, TM1 LSB=FFH, TM1 LSB=01H
TM1OUT output frequency=2 MHz/ (FFFF–FF00) =2 MHz/256=7.8 KHz
TM1OUT output time period=1/7.8 KHz=128 u

Timer1 can also works with Capture mode. When works in Capture mode, Timer1 will start counting when the TM1CLR bit is cleared and the first falling edge of CAPT pin (if TM0CL=0) is coming. When the 2nd falling edge of CAPT pin is coming, Timer1 stops counting and hold the value. When the 3rd falling edge of CAPT pin is coming, the Timer1 continues counting. The following figure shows the detail timing diagram.



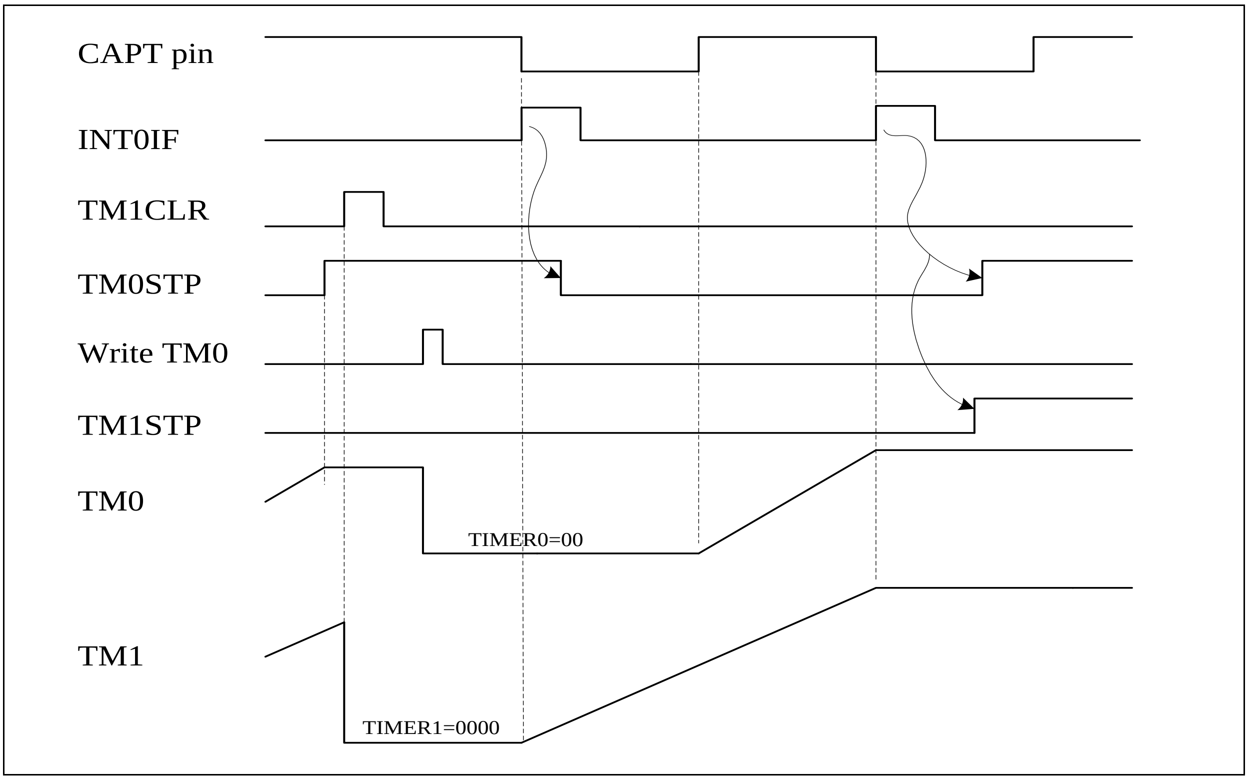
Timer1 works in Capture mode (TM0CL=0, implies CAPT falling edge)

Timer0 and Timer1 are used for Pulse Width and Period Capture

Timer0 and Timer1 can cooperate to measure the signal period and duty cycle time. The key is multi-function of PA6 (CAPT, INT0). Suppose that:

- TM0CKS=0, Timer0 prescaler increases per instruction cycle.
- TM0CM=1, TM1CM=1. Timer0 and Timer1 work in Capture mode.
- PA6 pin (CAPT pin) interrupts every falling edge. TM0CL=0, **Timer1** starts/holds in turn when PA6 pin (CAPT pin) falling edge is coming. **Timer0** starts counting when PA6 pin (CAPT pin) is in logic ‘1’ level, and holds the Timer0 value when PA6 pin (CAPT pin) is in logic ‘0’ level.
- Timer1 is used to measure the signal period, Timer0 is used to measure the PA6 (CAPT pin) in logic ‘1’ time (i.e. the duty cycle of the signal).

The following figure shows how to use Timer0 and Timer1 to measure the PA6 (CAPT pin) signal's period and duty cycle (TM0CL=0).

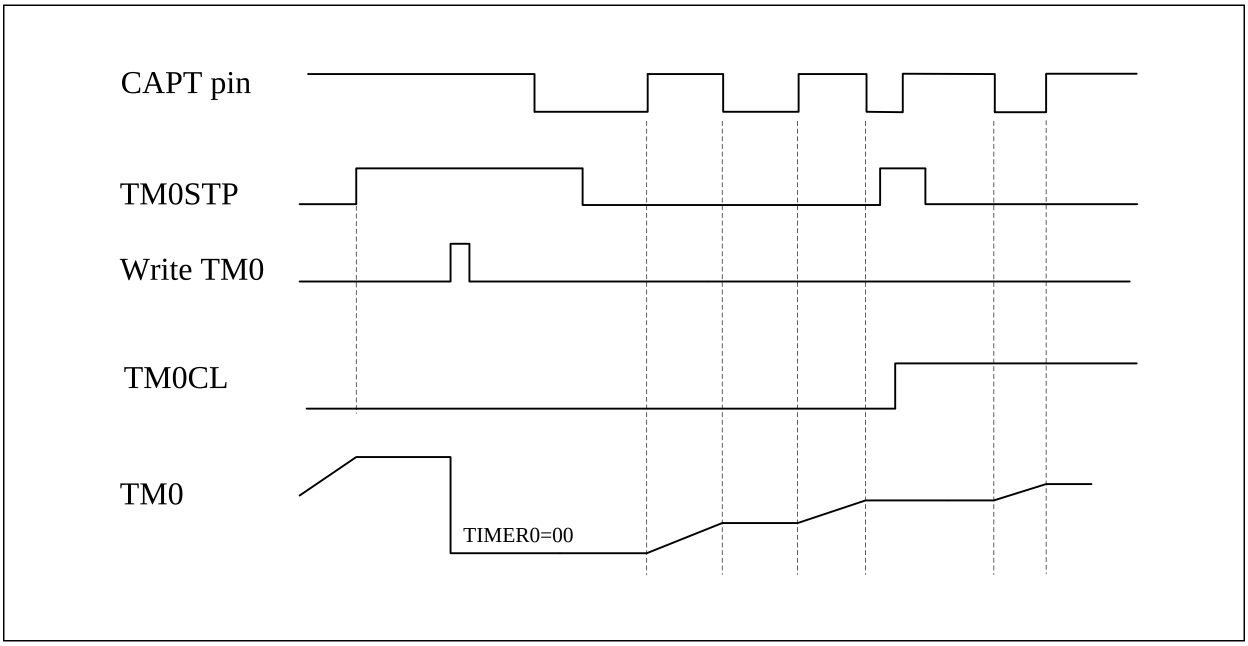


Timer0 and Timer1 are used to measure the signal on CAPT pin.

Follow the steps below to start measuring the CAPT pin's period and duty cycle.

1. Stop Timer0 by firmware (TM0STP=1, Timer0 will be stopped and hold)
2. Clear Timer1 by firmware (TM1CLR=1)
3. Clear Timer0 by directly write 00h to Timer0 (Timer0 is still hold). Once CAPT pin falling edge is coming, the Timer1 starts counting; meanwhile the PA6 interrupt is generated and clears the TM0STP by firmware. Now the Timer0 is ready to count when CAPT pin goes high.
4. CAPT pin rising edge is coming, Timer0 starts counting until the CAPT pin returns to 0 and holds the counting value. Timer1 also stops counting and holds the value.
5. PA6 interrupt is generated again, firmware stops Timer1 and Timer0 to read the period and duty cycle.

It is not necessary to use both Timer0 and Timer1. If only the duty cycle (CAPT high time) needs to be measured, there is no need to use Timer1 to measure the period. In such case, user can set the TM0CM=1 and TM1CM=0. Timer0 is counting up only when CAPT pin is ‘1’. Note that the internal prescaler will be kept to next Timer0 count, so it will not lose the counting accuracy.



Timer0 is used to measure the high (or low) time on CAPT pin

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	PWM0IE	PWM1IE	TM1IE	TM0IE	WKTIE	XINT2E	XINT1E	XINT0E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F08.6 **TM1IE**: Timer1 interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	PWM0IF	PWM1IF	TM1IF	TM0IF	WKTIF	XINT2F	XINT1F	XINT0F
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

F0A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1L	TM1L							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F0A.7~0 **TM1L**: Timer1 counter low byte
Read TM1L will get the Timer1 counter low byte. Write TM1L will write the Timer1 reload register low byte.

F0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1H	TM1H							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F0B.7~0 **TM1H**: Timer1 counter high byte
Read TM1H will get the Timer1 counter high byte. Write TM1H will write the Timer1 reload register high byte.

F12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF14	PWM0CLR	PWM1CLR	PWM0CKS	PWM1CKS	TM1SET	TM1CLR	TM1STP	TM0STP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	0	0	0	0	0

F12.1 **TM1STP**: Timer1 counter stop
0: Timer1 is counting
1: Timer1 stops counting

F12.2 **TM1CLR:** Timer1 counter clear
0: Release Timer1 clear
1: Clear Timer1 to ‘0000’h and hold

F12.3 **TM1SET:** Timer1 counter set to ‘FFFF’h
0: Release Timer1 set
1: Set Timer1 to ‘FFFF'

R0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0C	WKTPSC		WDTPSC		TM1CKS	TM0OE	TCOE	TM1OE
R/W	–	W		W	W	W	W	W
Reset	–	0	0	0	0	0	0	0

R0C.0 **TM1OE**: Timer1 overflow toggle output to PD0
0: disable output TM1OUT
1: enable output TM1OUT

R0C.3 **TM1CKS:** Timer1 clock source selection
0: Instruction cycle (Fsys/2)
1: System clock (Fsys)

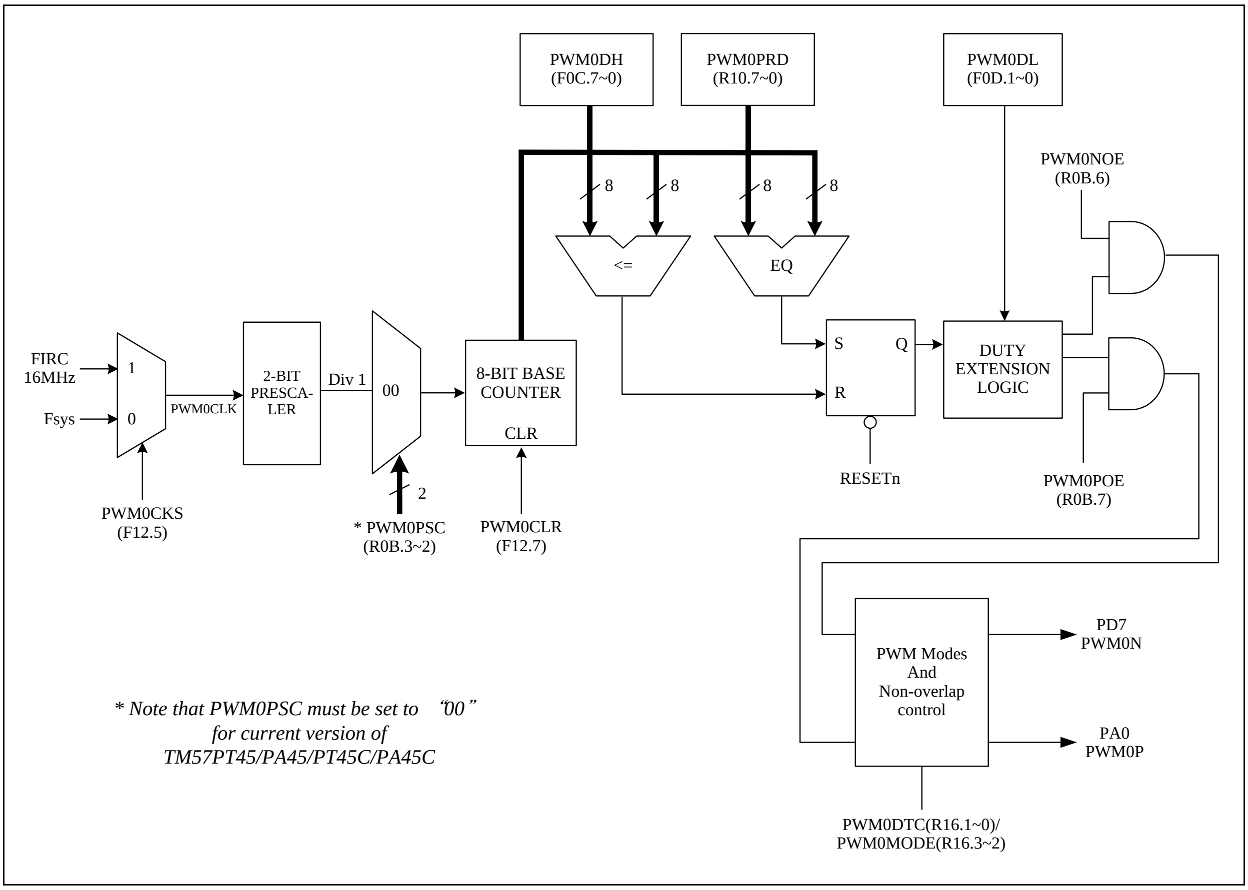
R0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0D	INT1EDG	TM1CM	WDTSTP	FIRCKS		ADCKS		
R/W	–	W		W		–		
Reset	–	0	0	0		–		

R0D.6 **TM1CM:** Timer1 Mode Selection
0: Timer1 in Timer Mode
1: Timer1 in Capture Mode to measure CAPT pin period time between successive rising or falling edges.

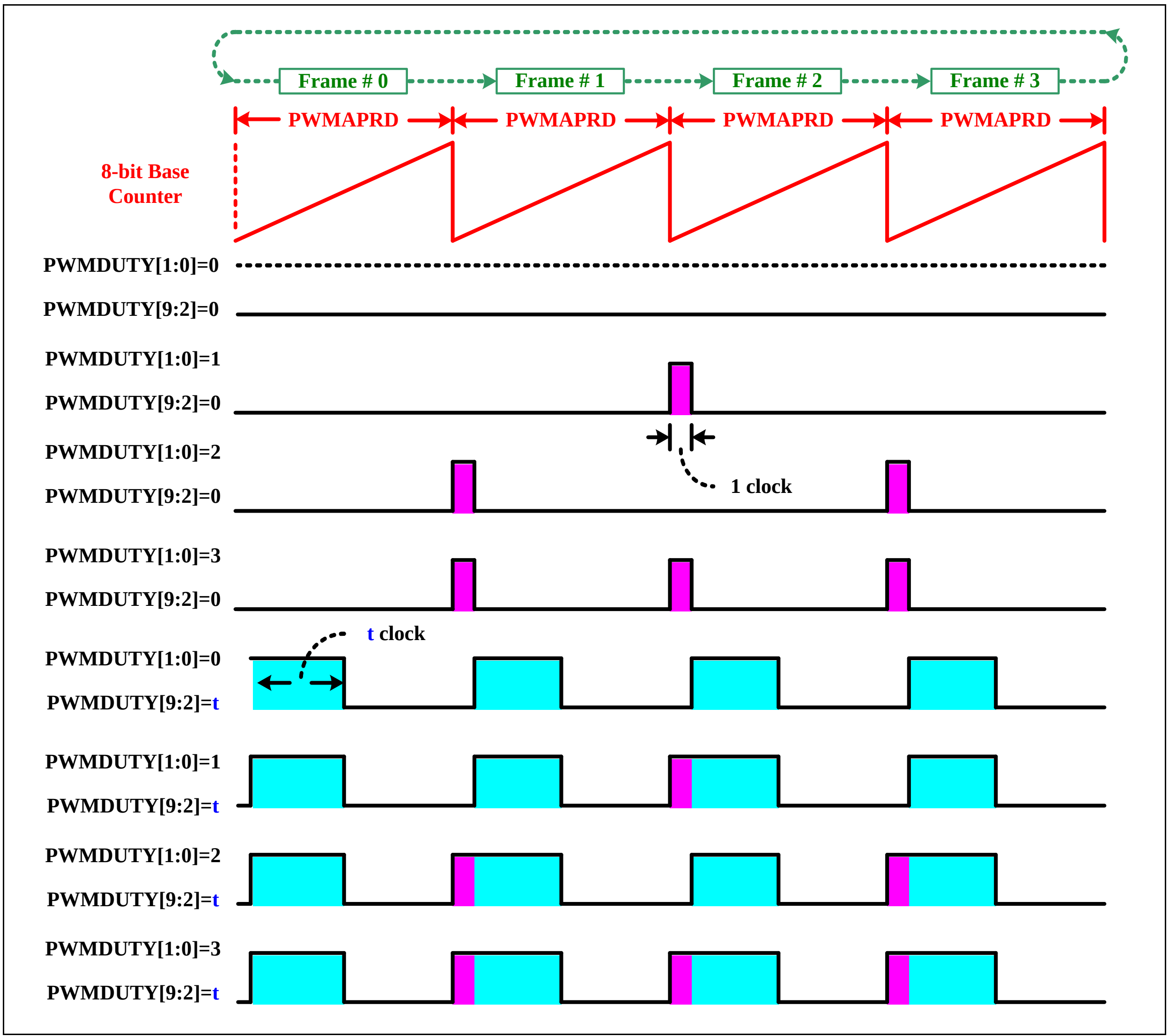
3.4 PWM0: (8+2) bits PWM

The PWM can generate fix frequency waveform with 1024 duty resolution based on System Clock (Fsys) or FIRC 16MHz. A spread LSB technique allows PWM to run its frequency at "System Clock divided by 256" instead of "System Clock divided by 1024", which means the PWM is 4 times faster than normal. The advantage of higher PWM frequency is that the post RC filter can transform the PWM signal to more stable DC voltage level. The PWM output signal reset to low level whenever the 8-bit base counter matches the 8-bit MSB of PWM duty register PWM0DH (F0C.7~0). When the base counter rolls over, the 2-bit LSB of PWM duty register PWM0DL (F0D.1~0) decides whether to set the PWM output signal high immediately or set it high after one clock cycle delay.

PWM0PSC is not be implemented in this version, user must set PWM0PSC to "00" to prevent malfunction.



PWM0 Block Diagram



PWM0 Timing Diagram

Example:

[CPU running at Fast mode, Fsys=FIRC 8 MHz]

◇Example:

```
; Setup PWM0 clock prescaler.
        BCF     PWM0CKS         ; PWM0 clock source=Fsys
        MOVLW   11000000B       ; Fsys=8 MHz, PWM0POE=1, PWM0NOE=1
        MOVWR   R0B             ;

        MVOLW   00000000B       ; PWM0 Mode=00
        MOVWR   R16             ; PWM0DTC=00

        MOVLW   80H
        MOVWR   PWM0PRD         ; Set PWM0 period=80H.

        MOVLW   00000000B
        MOVWF   F0D             ; Set PWM0DL duty=00H

        MOVLW   20H
        MOVWF   PWM0DH          ; Set PWM0DH duty=20H

        BCF     PWM0CLR         ; Enable PWM0 counting
```

Example:

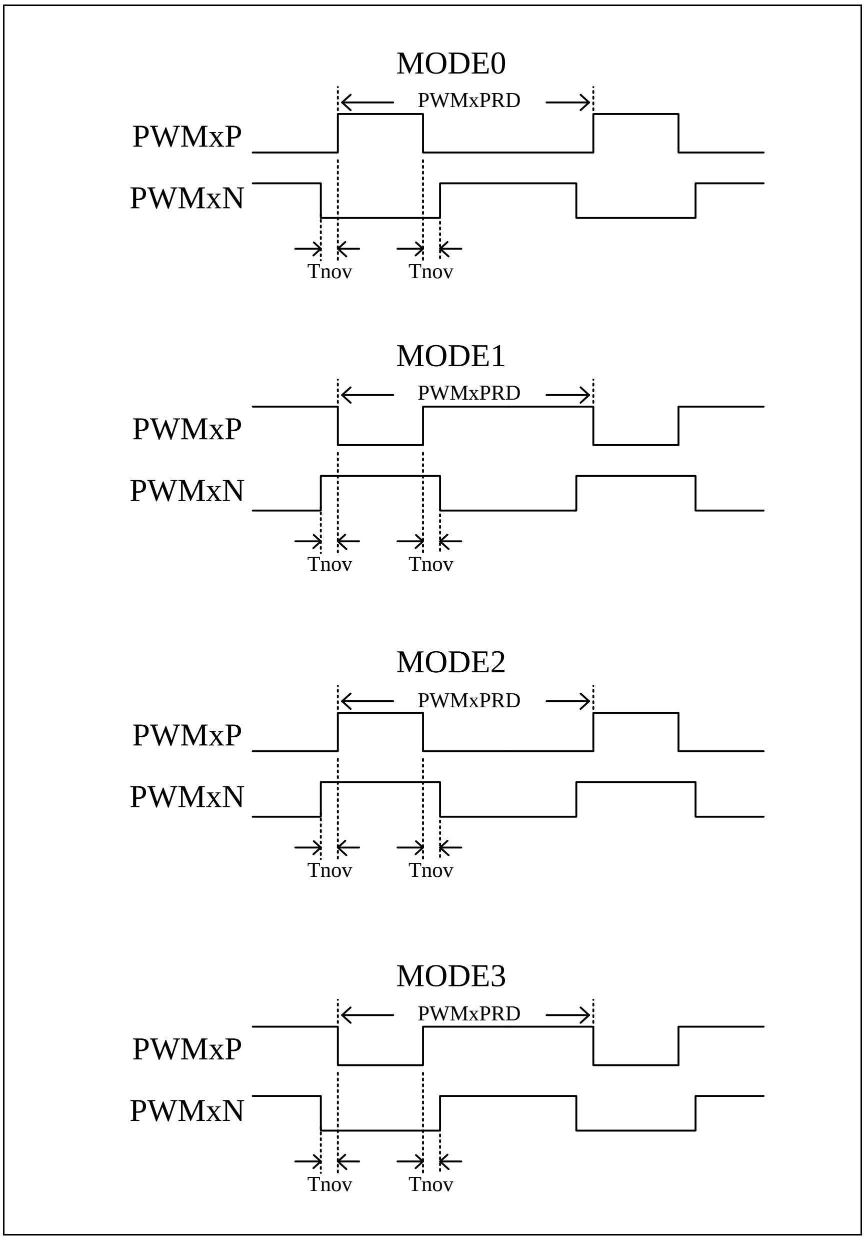
Fsys=8 MHz, PWM0PRD=80H,

PWM0DL=00H, PWM0DH=20H

PWM0 output frequency=8 MHz/ (PWM0PRD+1) =8 MHz/129=62 KHz.

PWM0P output duty=32:129=24.8%.

PWM0 can be output via PWM0P and PWM0N with four different modes. The edges of the PWM pulse can be separated with 4 different time non-overlap clocks intervals, 0s, 1 Fsys clock, 2 Fsys clocks, and 4 Fsys clocks which are selected by PWM0DTC (R16.1~0). The default output form is MODE0. The waveforms of the four output modes are shown below



PWM0/PWM1 output modes

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	PWM0IE	PWM1IE	TM1IE	TM0IE	WKTIE	XINT2E	XINT1E	XINT0E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F08.7 **PWM0IE**: PWM0 interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	PWM0IF	PWM1IF	TM1IF	TM0IF	WKTIF	XINT2F	XINT1F	XINT0F
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F09.7 **PWM0IF**: PWM0 interrupt event pending flag
This bit is set by H/W while PWM0 finish period, write 0 to this bit will clear this flag

F0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DH	PWM0DH							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F0C.7~0 **PWM0DH**: PWM0 duty 8-bit MSB

F0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0D	PCH				PWM1DL		PWM0DL	
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F0D.1~0 **PWM0DL**: PWM0 duty 2-bit LSB

F12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF14	PWM0CLR	PWM1CLR	PWM0CKS	PWM1CKS	TM1SET	TM1CLR	TM1STP	TM0STP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	0	0	0	0	0

F12.7 **PWM0CLR**: PWM0 clear and hold
0: PWM0 is running
1: PWM0 is clear and hold

F12.5 **PWM0CKS**: PWM0 clock selection
0: Fsys as PWM0 clock source
1: FIRC 16MHz as PWM0 clock source

R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	PWM0POE	PWM0NOE	PWM1POE	PWM1NOE	PWM0PSC		PWM1PSC	
R/W	W	W	W	W	W	W	W	W
Reset	1	1	0	0	0	0	0	0

R0B.7 PWM0POE: PWM0P output enable
0: disable PWM0P output
1: enable PWM0P output

R0B.6 **PWM0NOE**: PWM0N output enable
0: disable PWM0N output
1: enable PWM0N output

R0B.3~2 **PWM0PSC**: PWM0 clock source is divided by
Users must set these 2 bits to “00”.

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

R16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR16	PWM1MODE		PWM1DTC		PWM0MODE		PWM0DTC	
R/W	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

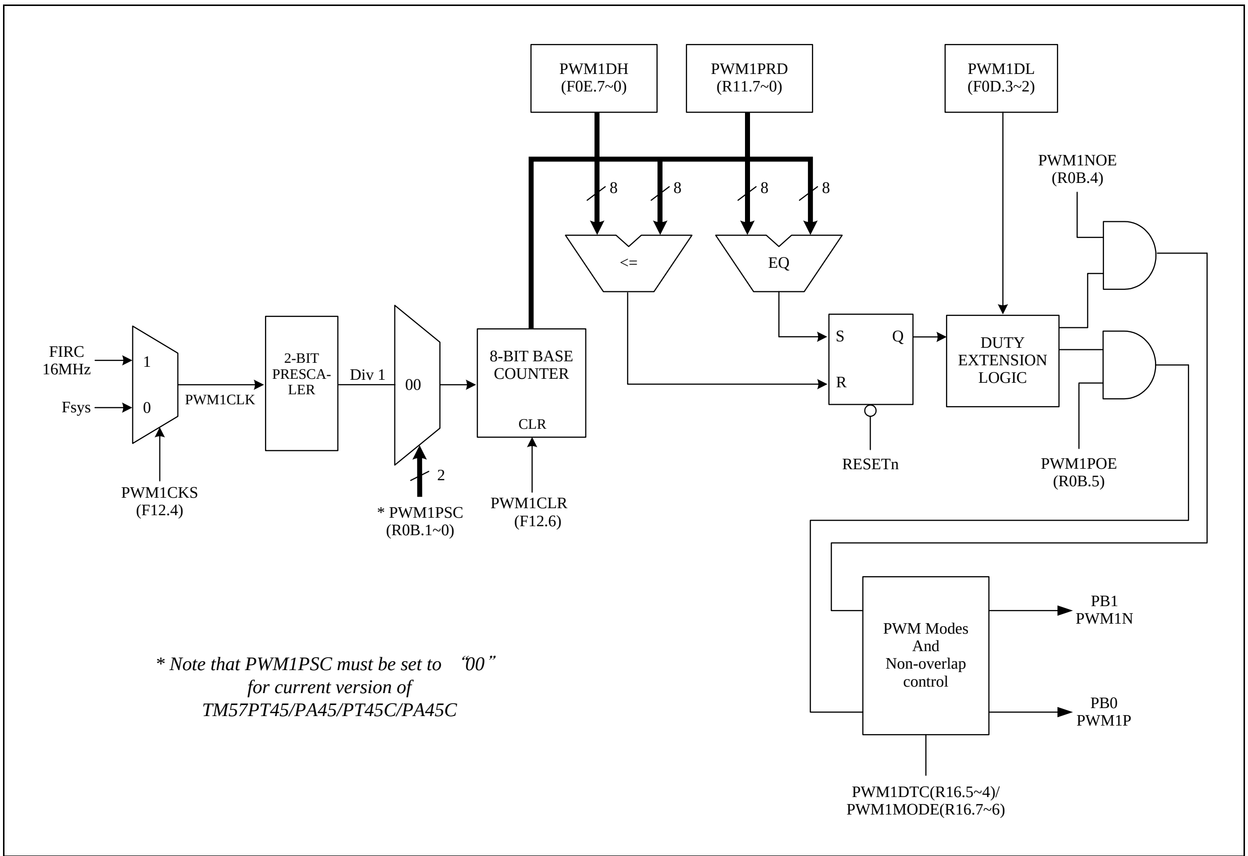
R16.1~0 **PWM0DTC**: PWM0 dead time control
00: 0 Fsys clock (original PWM0)
01: 1 Fsys clock
10: 2 Fsys clocks
11: 4 Fsys clocks

R16.3~2 **PWM0MODE**: PWM0P and PWM0N output mode
00: Mode 0
01: Mode 1
10: Mode 2
11: Mode 3

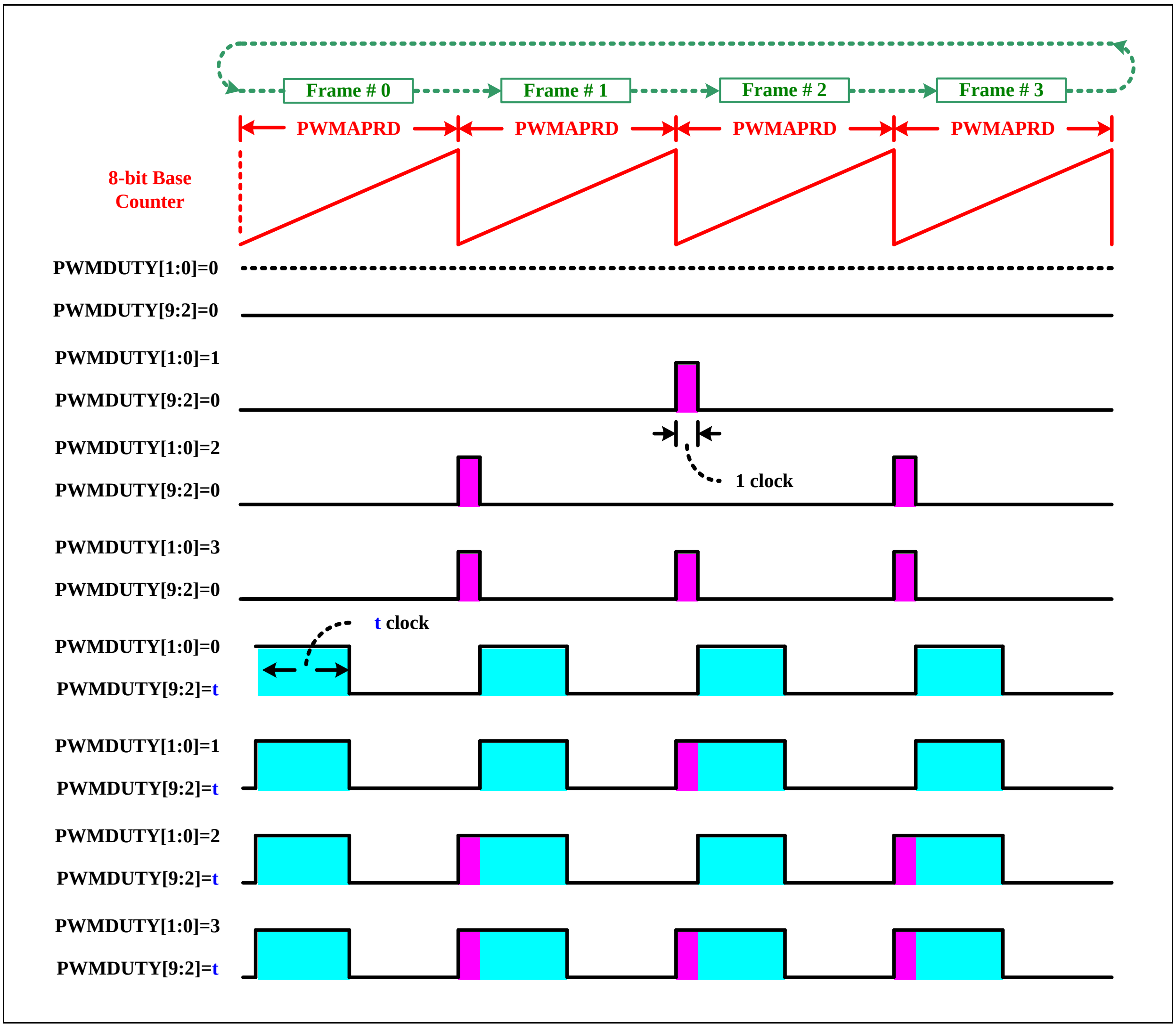
3.5 PWM1: (8+2) bits PWM

The PWM can generate fix frequency waveform with 1024 duty resolution based on System Clock (Fsys) or FIRC 16MHz. A spread LSB technique allows PWM to run its frequency at "System Clock divided by 256" instead of "System Clock divided by 1024", which means the PWM is 4 times faster than normal. The advantage of higher PWM frequency is that the post RC filter can transform the PWM signal to more stable DC voltage level. The PWM output signal reset to low level whenever the 8-bit base counter matches the 8-bit MSB of PWM duty register PWM1DH (F0E.7~0). When the base counter rolls over, the 2-bit LSB of PWM duty register PWM1DL (F0D.3~2) decides whether to set the PWM output signal high immediately or set it high after one clock cycle delay.

PWM1PSC is not be implemented in this version, user must set PWM1PSC to "00" to prevent malfunction.



PWM1 Bock Diagram



PWM1 Timing Diagram

Example:

[CPU running at Fast mode, Fsys=FIRC 8 MHz]

◇Example:

```
; Setup PWM0 clock prescaler.
        BCF         PWM1CKS         ;PWM1 clock source=Fsys
        MOVLW       00110000B       ; Fsys=8 MHz, PWM1POE=1, PWM1NOE=1
        MOVWR       R0B             ;

        MOVLW       00000000B       ; PWM1 MODE=00
        MOVWR       R16             ; PWM1DTC=00

        MOVLW       80H
        MOVWR       PRM1PRD         ; Set PWM1 period=80H

        MOVLW       00000000B
        MOVWF       F0D             ; Set PWM1DL duty=00H

        MOVLW       20H
        MOVWF       PWM1DH          ; Set PWM1DH=20H

        BCF         PWM1CLR         ; Enable PWM1 counting
```

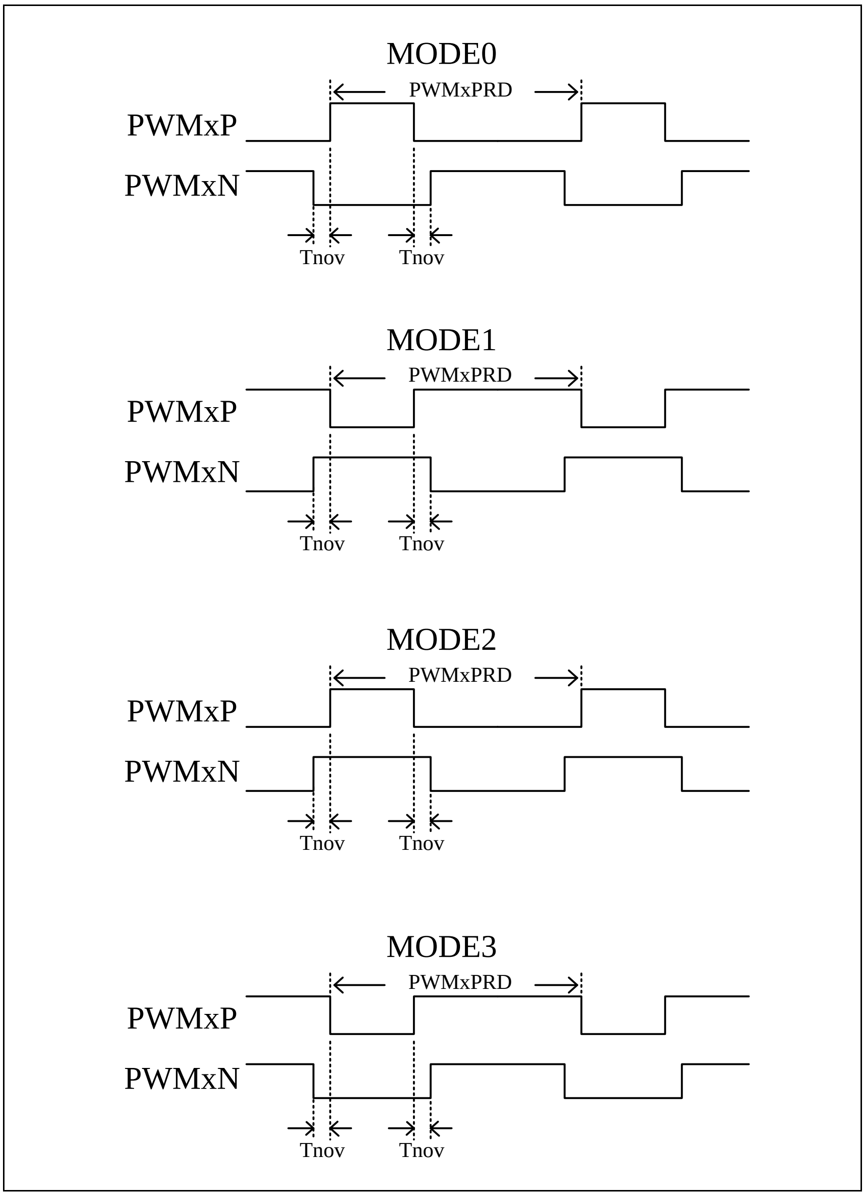
Fsys=8 MHz, PWM0PRD=80H,

PWM1DL=00H, PWM1DH=20H

PWM1 output frequency=8 MHz/ (PWM1PRD+1) =8 MHz/129=62 KHz.

PWM1P output duty=32:129=24.8%

PWM1 can be output via PWM1P and PWM1N with four different modes. The edges of the PWM pulse can be separated with 4 different time non-overlap clocks intervals, 0s, 1 Fsys clock, 2 Fsys clocks, and 4 Fsys clocks which are selected by PWM1DTC (R16.5~4). The default output form is MODE0. The waveforms of the four output modes are shown below.



PWM0/PWM1 output modes

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	PWM0IE	PWM1IE	TM1IE	TM0IE	WKTIE	XINT2E	XINT1E	XINT0E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F08.6 **PWM1IE**: PWM1 interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	PWM0IF	PWM1IF	TM1IF	TM0IF	WKTIF	XINT2F	XINT1F	XINT0F
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F09.6 **PWM1IF**: PWM1 interrupt event pending flag
This bit is set by H/W while PWM1 finish period, write 0 to this bit will clear this flag

F0E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1DH	PWM1DH							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F0E.7~0 **PWM1DH**: PWM1 duty 8-bit MSB

F0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0D	PCH				PWM1DL		PWM0DL	
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F0D.3~2 **PWM1DL**: PWM1 duty 2-bit LSB

F12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF14	PWM0CLR	PWM1CLR	PWM0CKS	PWM1CKS	TM1SET	TM1CLR	TM1STP	TM0STP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	0	0	0	0	0

F12.6 **PWM1CLR**: PWM1 clear and hold
0: PWM1 is running
1: PWM1 is clear and hold

F12.4 **PWM1CKS**: PWM1 clock selection
0: Fsys as PWM1 clock source
1: FIRC 16 MHz as PWM1 clock source

R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	PWM0POE	PWM0NOE	PWM1POE	PWM1NOE	PWM0PSC		PWM1PSC	
R/W	W	W	W	W	W	W	W	W
Reset	1	1	0	0	0	0	0	0

R0B.5 PWM1POE: PWM1P output enable
0: disable PWM1P output
1: enable PWM1P output

R0B.5 **PWM1NOE**: PWM1N output enable
0: disable PWM1N output
1: enable PWM1N output

R0B.1~0 **PWM1PSC**: PWM1 clock source
Users must set these 2 bits to "00".

R11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1PRD	PWM1PRD							
R/W	W							
Reset	1	1	1	1	1	1	1	1

R11.7~0 **PWM1PRD**: PWM1 period data

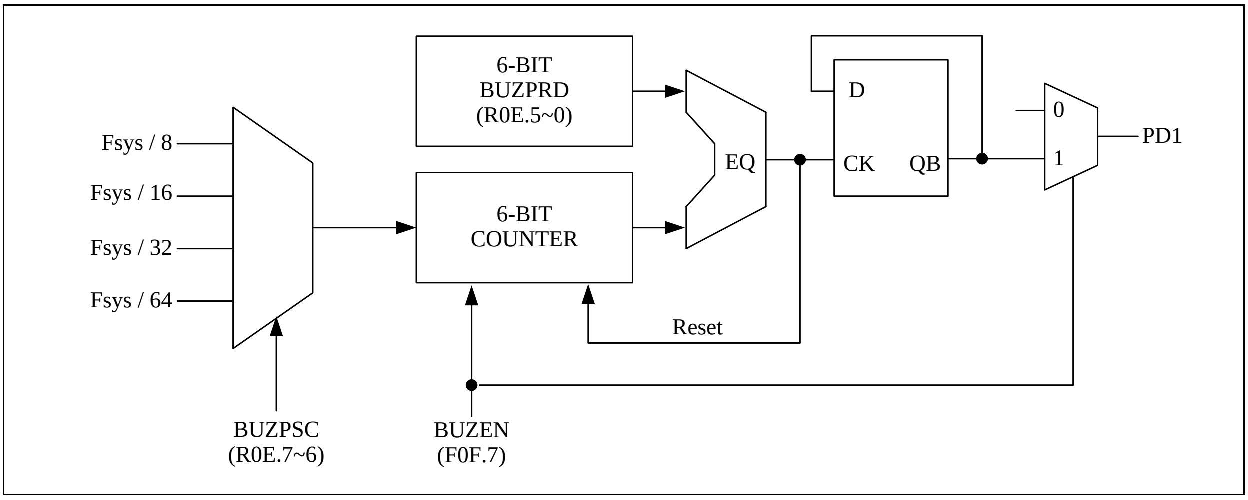
R16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR16	PWM1MODE		PWM1DTC		PWM0MODE		PWM0DTC	
R/W	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

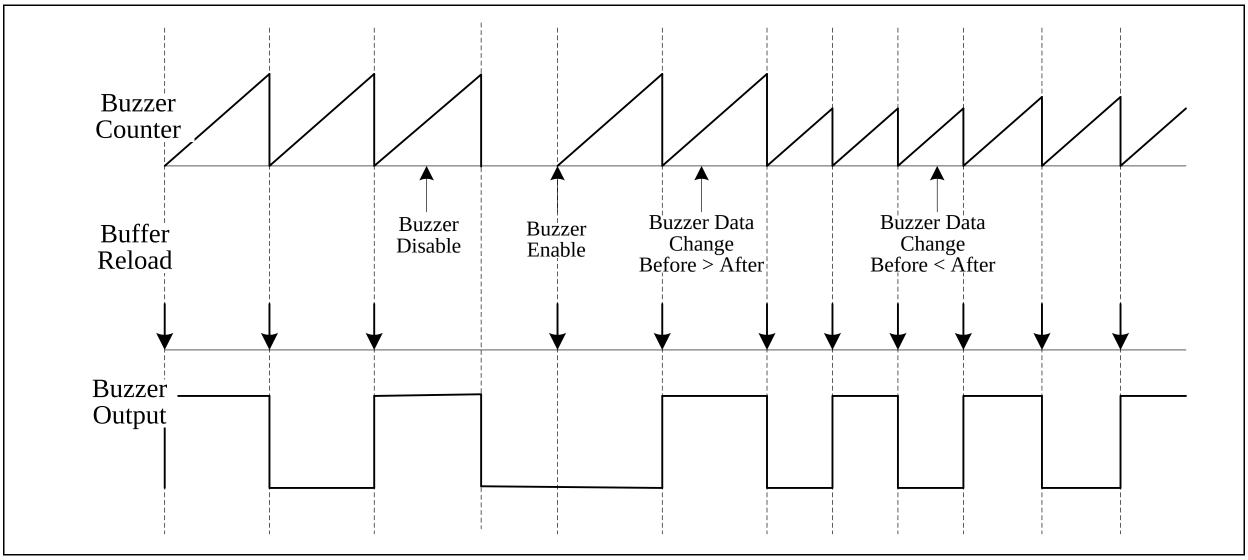
R16.5~4 **PWM1DTC**: PWM1 dead time control
00: 0 Fsys clock (original PWM1)
01: 1 Fsys clock
10: 2 Fsys clocks
11: 4 Fsys clocks

R16.7~6 **PWM1MODE**: PWM1P and PWM1N output mode
00: Mode 0
01: Mode 1
10: Mode 2
11: Mode 3

3.6 Buzzer Output

The Buzzer driver consists of 6-bit counter and a clock divider. It generates 50% duty square waveform with wide frequency range. To use the Buzzer function, user needs to set both the Buzzer enable control bit (BUZEN F0F.7)





Frequency calculation is as follows. F_{BZ}= (Fsys/BUZPSC) / (BUZPRD+1) /2

F_{BZ}= (4 MHz/32) / (9+1) /2=6.25 KHz

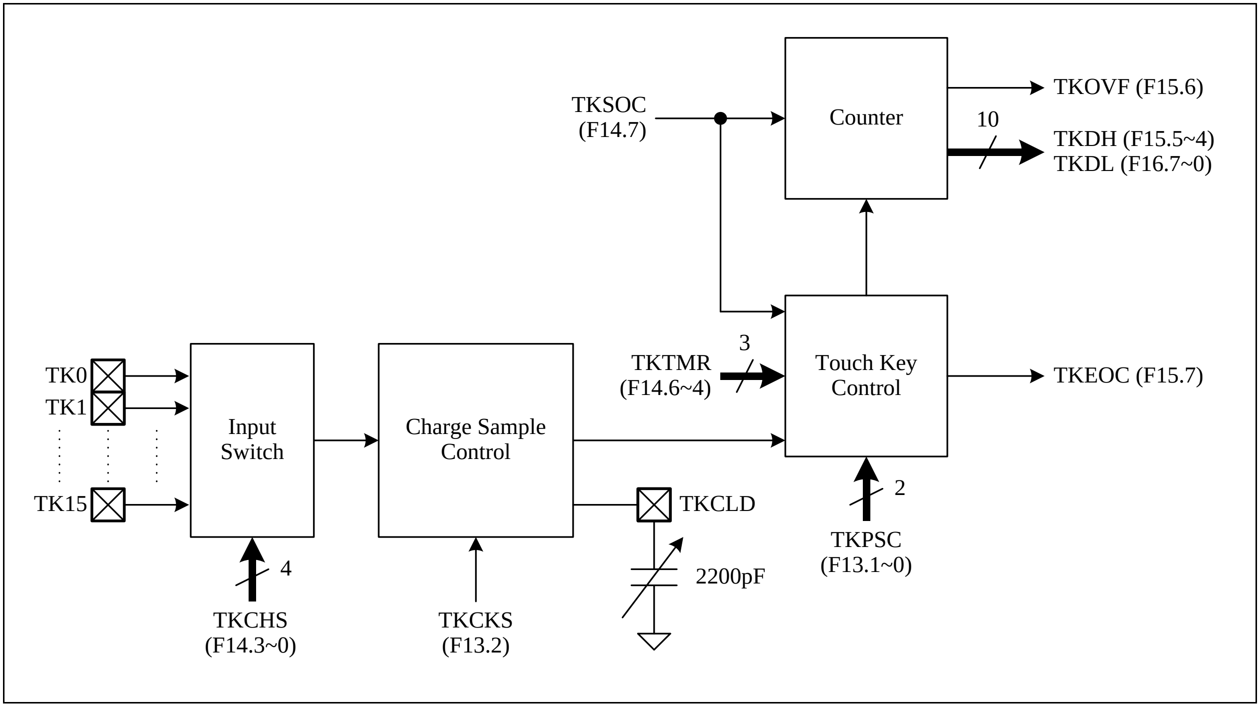
Example: [CPU running in FAST mode, Fsys=4 MHz]

```
MOVLW  10000001B
MOVWF  F0F          ; F0F.7 (BUZEN)=1, enable Buzzer counting and output to PD1
MOVLW  10 001001B   ; R0E.7~6 (BUZPSC)=Fsys/32
MOVWR  R0E          ; R0E.5~0 (BUZPRD)=9
```

3.7 Touch Key (The function only available for TM57PT45)

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 (F14.7) bit to start touch key conversion, the TKSOC bit will be cleared by H/W while end of conversion. "TKEOC=0" means conversion is in process, while "TKEOC=1" means the conversion is finish. After TKEOC's (F15.7) 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 (F14.6~4) or increase TKPSC (F13.1~0) 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=TK7 (PB3).

```
        MOVLW   xxxx0xxxB
        MOVWR   PBE             ; disable PB3 push-pull output
        MOVWR   PBM             ; disable PB3 digital input

        MOVLW   xxxx1xxxB       ; disable PB3 pull high
        MOVWR   PBPUN

        MOVLW   xxxxx0xxB       ; Set PA2 as TKCLD for connecting capacitor
        MOVWR   PAE
        MOVWR   PAM             ; disable PA2(CLD) digital input

        MOVLW   xxxxx1xxB       ; disable PA2 pull high
        MOVWR   PAPUN

        MOVLW   0100 0111B
        MOVWF   F14             ; TKTMR=4, TKCHS=7 (TK7)

        MOVLW   0000 0 1 00B    ; TKPD=0
        MOVWF   F13             ; 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]
```

F13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF13	OPA0EN	OPA1EN	OP1IPSEL	–	TKPD	TKCKS	TKPSC	
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W	
Reset	0	0	0	0	1	1	0	

F13.3 **TKPD**: Touch key power down
- 0: Touch key running
- 1: Touch key power down

R13.2 **TKCKS**: Touch key clock select
- 0: 2 MHz
- 1: 4 MHz

R13.1~0 **TKPSC**: Touch key data prescaler, touch key data
- 00: divided by 1
- 01: divided by 2
- 10: divided by 4
- 11: divided by 8

F14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF14	TKSOC	TKTMR			TKCHS			
R/W	R/W	R/W			R/W			
Reset	0	1	0	0	0			

F14.7 **TKSOC**: Touch key start of conversion, rising edge to start
H/W auto cleared while end of conversion

F14.6~4 **TKTMR**: Touch key conversion time
000: shortest
…
111: longest

F14.3~0 **TKCHS**: Touch key channel select
0000: TK0 (PD0)
0001: TK1 (PD1)
0010: TK2 (PD2)
0011: TK3 (PD3)
0100: TK4 (PD4)
0101: TK5 (PD5)
0110: TK6 (PB2)
0111: TK7 (PB3)
1000: TK8 (PB4)
1001: TK9 (PB5)
1010: TK10 (PD6)
1011: TK11 (PB0)
1100: TK12 (PA0)
1101: TK13 (PA5)
1110: TK14 (PD7)
1111: TK15 (PB1)

F15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCTL2	TKEOC	TKOVF	TKDH		–	–	–	–
R/W	R	R	R		–	–	–	–
Reset	1	0	0		–	–	–	–

F15.7 **TKEOC**: Touch key end of conversion
0: conversion is in process
1: end of conversion

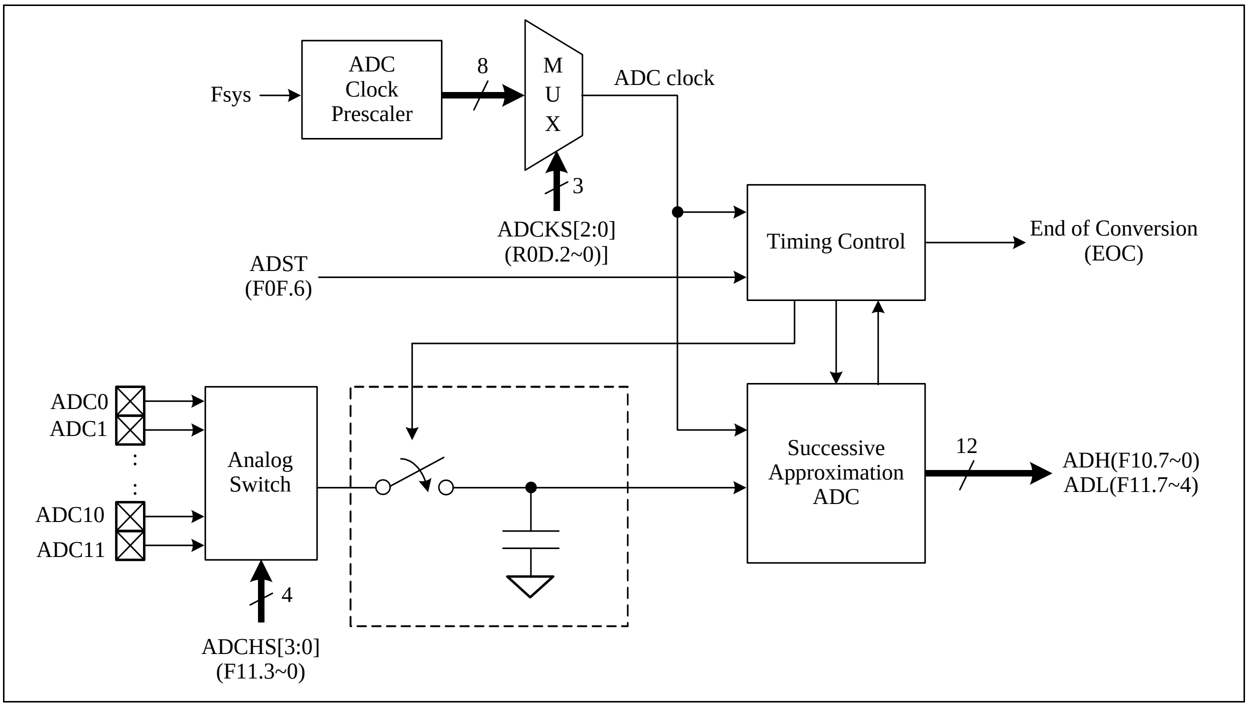
F15.6 **TKOVF**: Touch key counter overflow flag
0: not overflow
1: overflow

F15.5~4 **TKDH**: Touch key data MSB [9~8]

F16	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

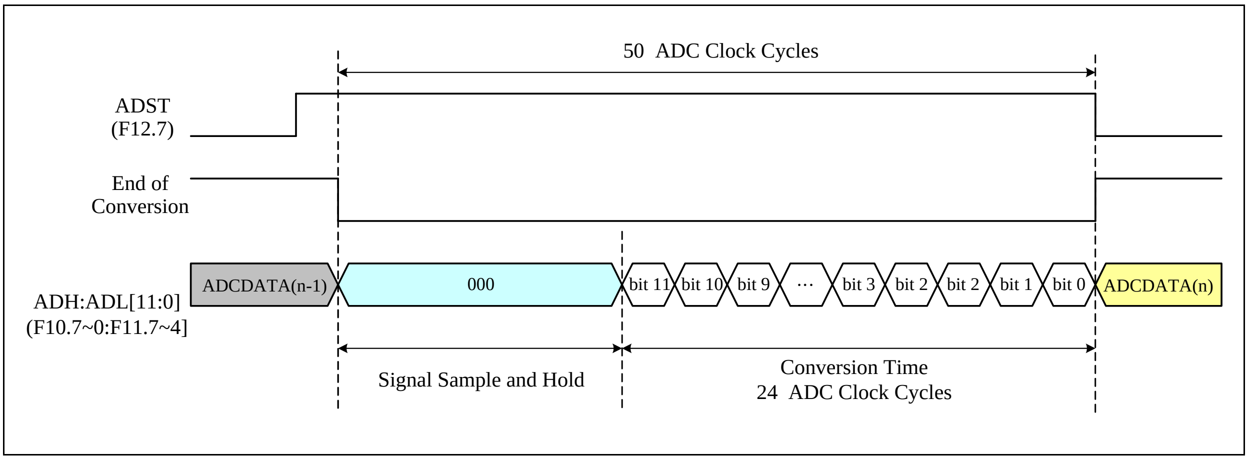
F16.7~0 **TKDL**: Touch key data LSB [7~0]

3.8 ADC: 12-bit Analog-to-Digital Converter



The 12-bit ADC (Analog to Digital Converter) consists of a 12-channel analog input multiplexer, control register, clock generator, 12-bit successive approximation register, and output data register. To use the ADC, user needs to set ADCKS(R0D.2~0) to choose a proper ADC clock frequency, which must be less than 1 MHz. User then launches the ADC conversion by setting the ADST (F0F.6) control bit. After end of conversion, H/W automatic clears the ADST (F0F.6) bit. User can poll this bit to know the conversion status. The PAM (R12.7~0), PBM (R13.5~0), PDM (R14.7~0) control registers are used for ADC pin type setting, user can write the corresponding bit to "0" when the pin is used as an ADC input. The setting can disable the pin logical input path to save power consumption.

The A/D conversion timing diagram



Example:

[CPU running at Fast mode , Fsys=FIRC 8 MHz]

ADC clock frequency=1 MHz, ADC channel=ADC2 (PA2).

◇Example:

```
            MOVLW   xxx10101B       ; Fsys=8 MHz
            MOVWR   R0D             ; ADC clock prescaler/8

            MOVLW   11111011B
            MOVWR   PAM             ; Enable PA2 pin (ADC2) analog input

            MOVLW   00000010B
            MOVWF   F11             ; ADC channel select ADC2 (PA2 pin)

            BSF     ADST            ; ADC start conversion

    WAIT_ADC:
            BTFSC   ADST            ; Wait ADC conversion
            GOTO    WAIT_ADC

            MOVFW   ADH             ; Read ADC value [11:4]
            MOVWF   ADC_MSB
            MOVFW   F11             ; Read ADC value[3:0]
            ANDLW   F0H
            MOVWF   ADC_LSB
      …
```

F10	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADH	ADH							
R/W	R							
Reset	0	0	0	0	0	0	0	0

F10.7~0 **ADCDH**: ADC Output MSB[11:4]

F11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF11	ADL				ADCHS			
R/W	R				R/W			
Reset	0	0	0	0	0	0	0	0

F11.7~4 **ADL**: ADC output LSB[3:0]

F11.3~0 **ADCHS**: ADC channel select

0000 : ADC0 (PA6)
0001 : ADC1 (PA1)
0010 : ADC2 (PA2)
0011 : ADC3 (PB1)
0100 : ADC4 (PD7)
0101 : ADC5 (PA5)
0110 : ADC6 (PA0)
0111 : ADC7 (PB0)
1000 : ADC8 (PD6)
1001 : ADC9 (PB5)
1010 : ADC10 (PD5)
1011 : ADC11 (PD2)

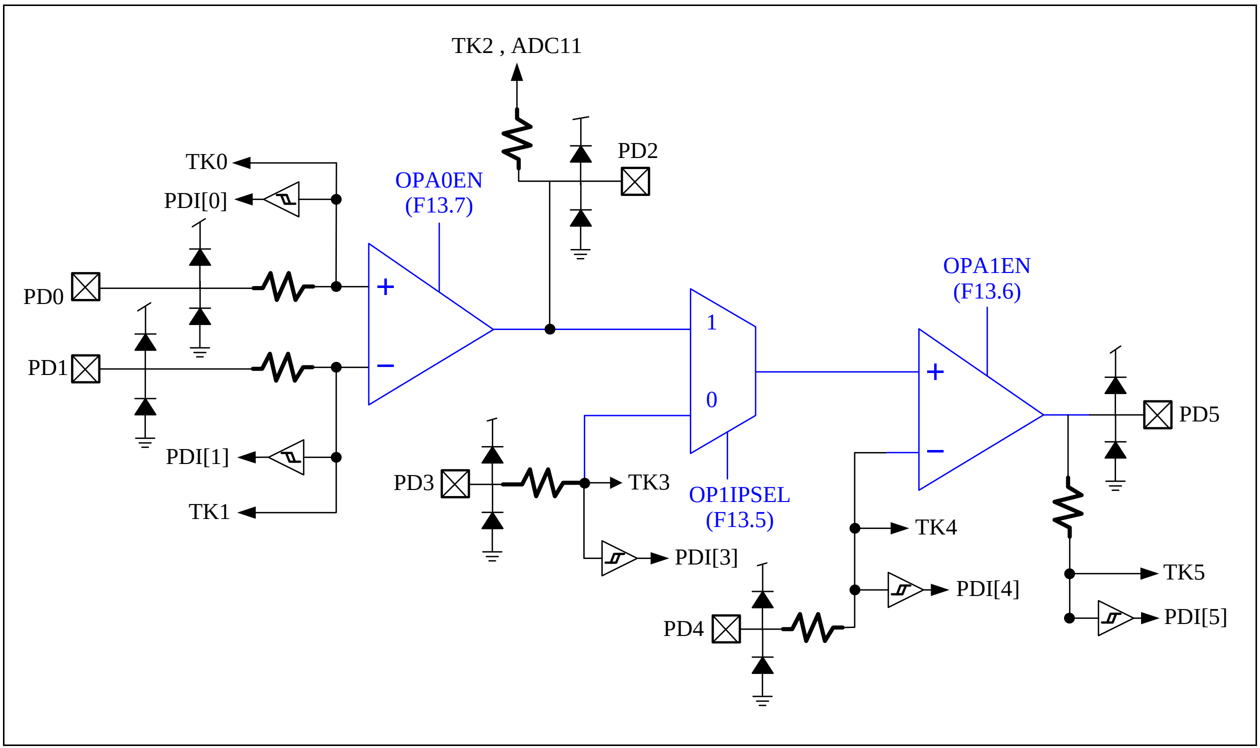
R0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0D	INT1EDG	TM1CM	WDTSTP	FIRCKS		ADCKS		
R/W	W	W	W	W	W	W		
Reset	0	0	0	0	1	1	1	1

R0D.2~0 **ADCKS:** ADC clock selection
000 : Fsys/256
001 : Fsys/128
010 : Fsys/64
011 : Fsys/32
100 : Fsys/16
101 : Fsys/8
110 : Fsys/4
111 : Fsys/2

3.9 OPA: 2 Sets Operational Amplifiers

There are two sets operational amplifiers (OPA0 and OPA1) in the TM57PT45. The default setting of OPA0 and OPA1 are in the power down mode. To use those OPAs, the OPA0EN (F13.7) and OPA1EN (F13.6) bit have to be cleared. The two OPAs can be used with independent or cascade application depends on OP1IPSEL (F13.5) bit. When OP1IPSEL is set, the OPA0 output and OPA1 positive input are connected together. When OP1IPSEL is cleared, the OPA0 and OPA1 are independent. In this way, the TM57PT45 can get more flexible for OPA applications. The OPA Block diagram is shown as below.

With I/O mode setting, the corresponding pins have to set as analog mode (please refer the chapter 4.3). The setting can disable the pin logical input path for save power consumption.



OPA Block Diagram

◇Example: OPA series application (connects 2 OPAs in series)

```
MOVLW    00000000B
MOVWF    PDE            ; Disable PD5-0 CMOS output

MOVLW    11111111B
MOVWR    PDPUN          ; Disable PD5-0 pull up resistor

MOVLW    11111111B
MOVWF    PDD            ; Do not output '0's to OPA pins

BSF      OP1IPSEL       ; Set OPA1 positive input from output of OPA0

BSF      OPA0EN         ; Enable OPA0
BSF      OPA1EN         ; Enable OPA1
```

F13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF13	OPA0EN	OPA1EN	OP1IPSEL	–	TKPD	TKCKS	TKPSC	
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	0	0

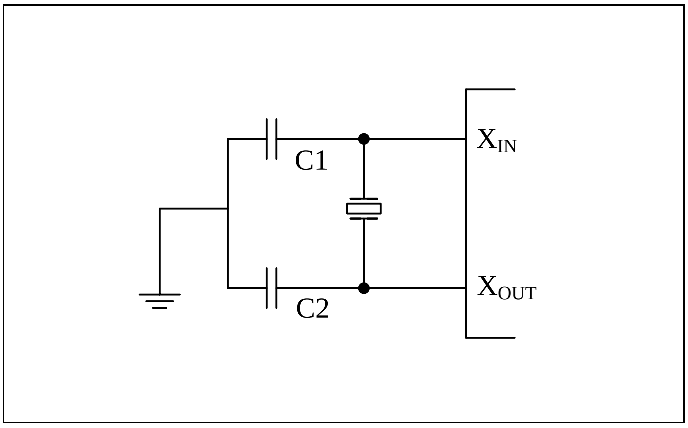
F13.7 **OPA0EN**: OPA0 Enable
0: OPA0 is power down
1: OPA0 is enabled

F13.6 **OPA1EN**: OPA1 Enable
0: OPA1 is power down
1: OPA1 is enabled

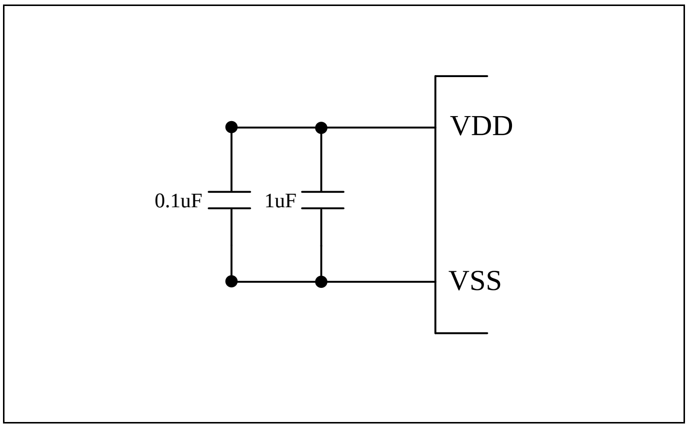
F13.5 **OP1IPSEL**: OPA1 positive input terminal selection
0: OPA1 positive terminal from PD3
1: OPA1 positive terminal from output of OPA0, a.k.a. PD2

3.10 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)

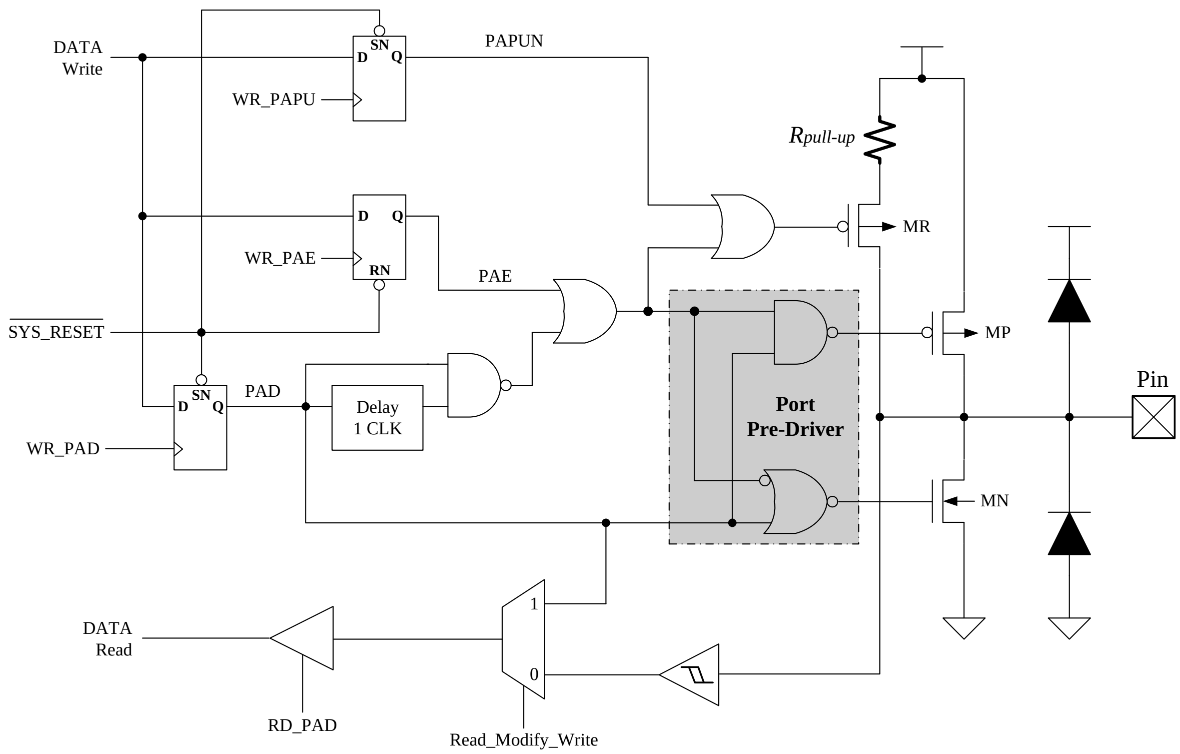


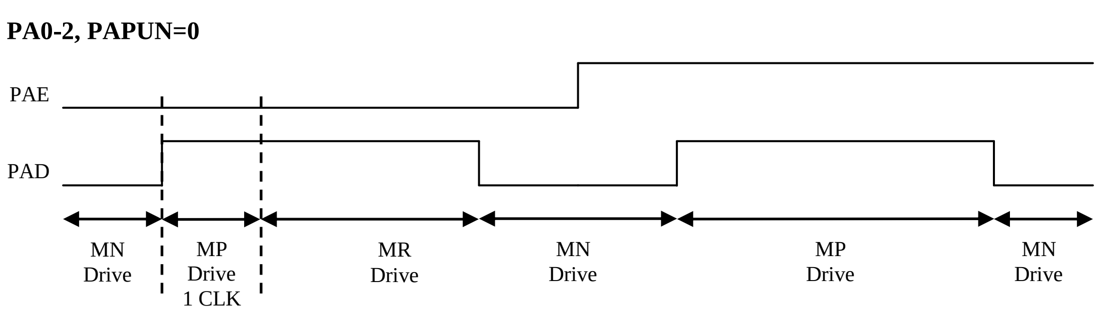
Fast Internal RC Mode

4. I/O Port

4.1 PA0-2

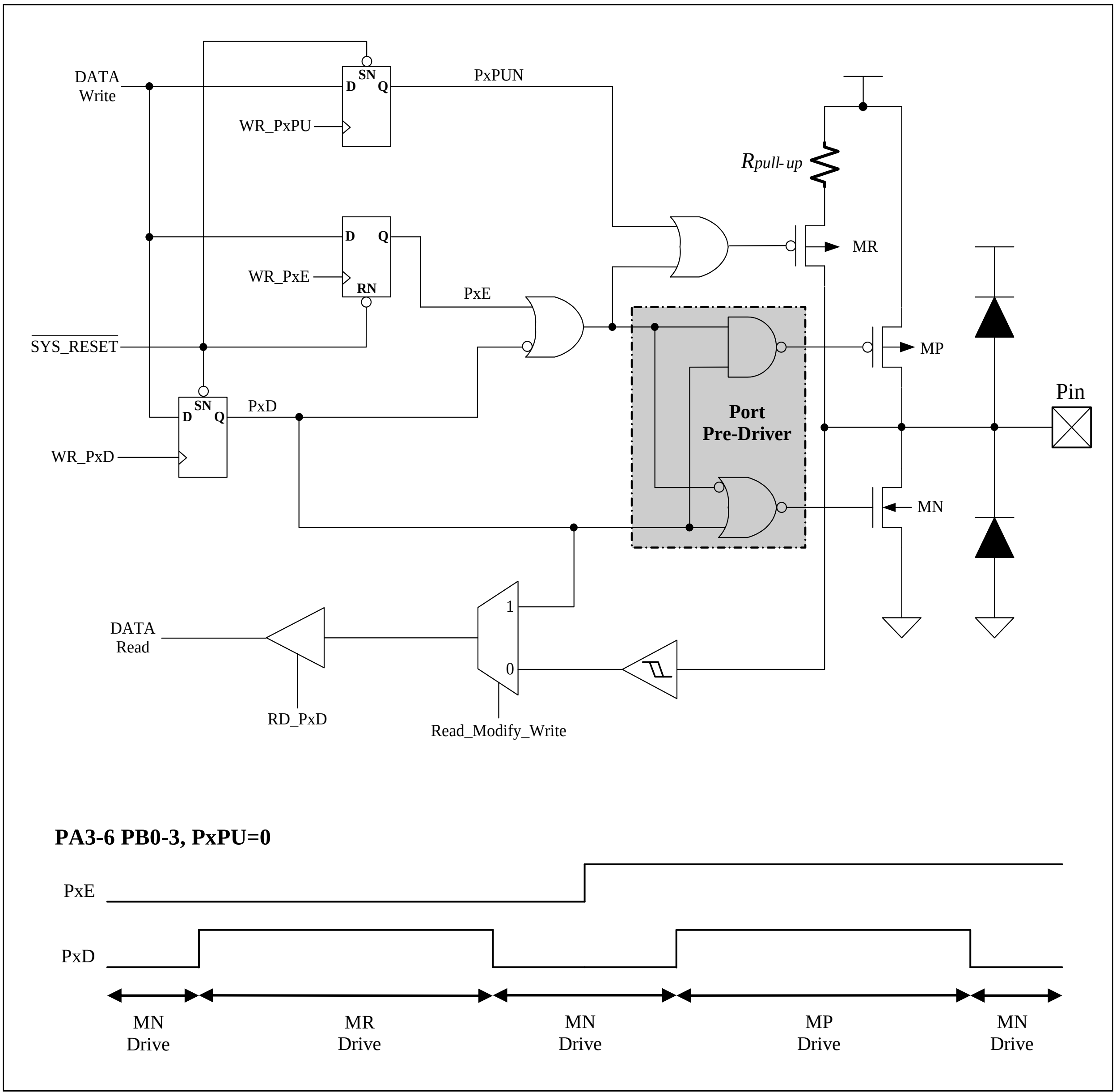
These pins can be used as Schmitt-trigger input, CMOS push-pull output or "pseudo-open-drain" output. The pull-up resistor is assignable to each pin by S/W setting. To use the pin in Schmitt-trigger input mode, S/W needs to set the PAE=0 and PAD=1. To use the pin in pseudo-open-drain mode, S/W sets the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





4.2 PA3-6, PB0-5, PD0-7

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



◇Example: I/O mode selecting

```
MOVLW  FFH
MOVWF  PAD
MOVWF  PBD
MOVWF  PDD
MOVLW  00H
MOVWR  PAE
MOVWR  PBE
MOVWR  PDE              ; Set all ports to be Schmitt-trigger input
```

◇Example: Set PA0-2 as pseudo-open-drain mode

```
MOVLW  xxxxx000B
MOVWR  PAE              ; Set PA2-PA0 as pseudo-open-drain mode

MOVLW  xxxxx000B
MOVWF  PAD              ; PA2~PA0 output low level
```

◇Example: Set PA0-2 is CMOS push-pull output mode.

```
MOVLW  xxxxx111B
MOVWR  PAE              ; Set PA2-PA0 as CMOS push-pull output mode
```

◇Example: Read data from input port.

```
MOVFW  PAD              ; Read data from Port A
MOVFW  PBD              ; Read data from Port B
MOVFW  PDD              ; Read data from Port D
```

◇Example: Write data to output port.

```
MOVLW  55H
MOVWF  PAD              ; Write data 55H to Port A
MOVWF  PBD              ; Write data 55H to Port B
```

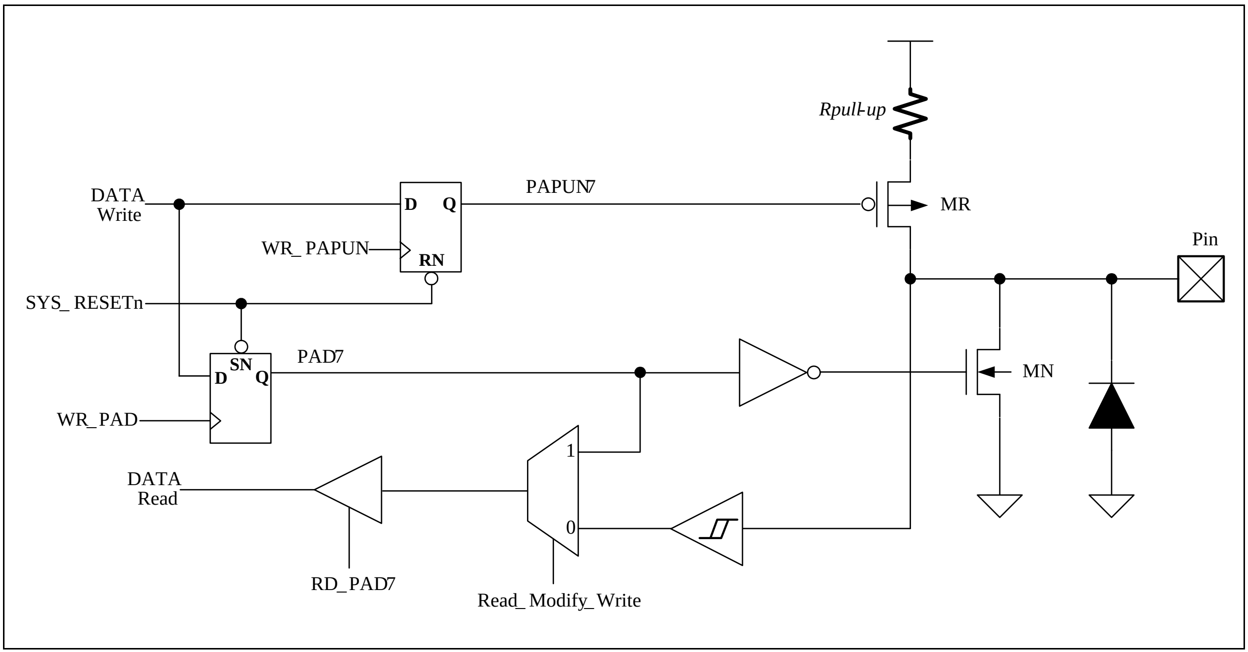
◇Example: Write one bit data to output port.

```
BCF    PAD,0
BCF    PBD,1
BCF    PDD,2            ; Set PA0, PB1 and PD2 to be "0"

BSF    PAD,3
BSF    PBD,4
BSF    PDD,7            ; Set PA3, PB4 and PD7 to be "1"
```

4.3 PA7

PA7 can be used in Schmitt-trigger input mode or open-drain output which is set 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 is connected to this pin by default and can be disabled by S/W. In open-drain output mode, the pull-up resistor will not be disabled automatically. The pull-up resistor can be disabled by S/W in open-drain output mode for power saving.

CAUTION: Before turning off the PA7 pull-up resistor (PAPUN.7=1), make sure the SYSCFG[7]: XRSTE bit is "0" that disable the external reset pin function. If XRSTE=1 and PAPUN.7=1, and the PA7 pin is in floating state, the chip will not work correctly.



◇Example: Read state from PA7.

Condition: SYSCFG[7] is set to "0". If SYSCFG[7] = "1", then PA7 pin is external reset pin function.

```
BTFSS   PAD,7
GOTO    LOOP_A        ; If PA7=0.
GOTO    LOOP_B        ; If PA7=1.
```

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	0	0	1	1	1	1	1	1

F06.7~0 **PBD:** PB5~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	1	1	1	1	1	1

F07.1~0 **PDD:** PD7~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
PAE	PAE							
R/W	W							
Reset	0	0	0	0	0	0	0	0

R05.7~0 **PAE**: PA7~PA0 Pin CMOS output enable
0 : For PA2-PA0, the pins are Pseudo-open-drain output or Schmitt-trigger input.
For PA3-PA7, the pins are open-drain output or Schmitt-trigger input
1 : the pins are CMOS push-pull output except PA7. PA7 can only be open-drain output mode.

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

R06.5~0 **PBE**: PB5~PB0 Pin CMOS output enable
0 : the pins are open-drain output or Schmitt-trigger input
1: the pins are CMOS push-pull output.

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

R07.7~0 PDE: PD7~PD0 Pin CMOS output enable
0 : the pins are open-drain output or Schmitt-trigger input
1: the pins are CMOS push-pull output.

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

R08.7~0 **PAPUN**: PA7~PA0 pin pull-high enable
0 : the pins are pull-high
1: the pins are not pull-high

R09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBPUN	–		PBPUN					
R/W	–		W					
Reset	0	0	1	1	1	1	1	1

R09.5~0 **PBPUN**: PB5~PB0 Pin pull-high enable
0 : the pins are pull-high
1: the pins are not pull-high.

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

R0A.7~0 **PDPUN**: PD7~PD0 Pin pull-high enable
0 : the pins are pull-high
1: the pins are not pull-high.

R12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAM	PAM							
R/W	W							
Reset	1	1	1	1	1	1	1	1

R12.7~0 **PAM**: PA7~PA0 pin mode
0 : the pins disable I/O digital input
1: the pins enable I/O digital input

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

R08.7~0 **PBM**: PB5~PB0 pin mode
0 : the pins disable I/O digital input
1: the pins enable I/O digital input

R14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDM	PDM							
R/W	W							
Reset	1	1	1	1	1	1	1	1

R14.7~0 PDM: PD7~PD0 pin mode
0 : the pins disable I/O digital input
1: the pins enable I/O digital input

R15	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

R15.5~0 **PBWKEN:** PB5~PB0 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 Counter**
PCL	02.7~0	R/W	0	Programming Counter LSB[7~0]
(F03) STATUS				**Function related to: STATUS**
GBIT1	03.7	R/W	0	General purpose bit 1
GBIT0	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**
GBIT2	04.7	R/W	0	General purpose bit 2
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	06.5~0	R	-	Port B pin or “data register” state
		W	FF	Port B output data register
(F07) PDD				**Function related to: Port D**
PDD	07.7~0	R	-	Port D pin or “data register” state
		W	FF	Port D output data register

Name	Address	R/W	Rst	Description
(F08) INTIE				**Function related to: Interrupt Enable**
PWM0IE	08.7	R/W	0	PWM0 interrupt enable 0: disable 1: enable
PWM1IE	08.6	R/W	0	PWM1 interrupt enable 0: disable 1: enable
TM1IE	08.5	R/W	0	Timer1 interrupt enable 0: disable 1: enable
TM0IE	08.4	R/W	0	Timer0 interrupt enable 0: disable 1: enable
WKTIE	08.3	R/W	0	WKT interrupt enable 0: disable 1: enable
INT2IE	08.2	R/W	0	INT2 (PA7) pin interrupt enable 0: disable 1: enable
INT1IE	08.1	R/W	0	INT1 (PA1) pin interrupt enable 0: disable 1: enable
INT0IE	08.0	R/W	0	INT0 (PA6) pin interrupt enable 0: disable 1: enable

Name	Address	R/W	Rst	Description
(F09) INTIF				**Function related to: Interrupt Flag**
PWM0IF	09.7	R	-	PWM0 interrupt event pending flag, set by H/W while PWM0 period match
		W	0	0: clear this flag 1: no action
PWM1IF	09.6	R	-	PWM1 interrupt event pending flag, set by H/W while PWM1 period match
		W	0	0: clear this flag 1: no action
TM1IF	09.5	R	-	Timer1 interrupt event pending flag, set by H/W while Timer1 overflows
		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
WKTIF	09.3	R	-	WKT interrupt event pending flag, set by H/W while WKT time out
		W	0	0: clear this flag 1: no action
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
(F0A) TM1L				**Function related to: Timer1**
TM1L	0a.7~0	R/W	0	(Read) Timer1 counter low byte. (Write) Timer1 reload low byte
(F0B) TM1H				**Function related to: Timer1**
TM1H	0b.7~0	R/W	0	(Read) Timer1 counter high byte. (Write) Timer1 reload high byte
(F0C) PWM0DH				**Function related to: PWM0**
PWM0DH	0c.7~0	R/W	0	PWM0 duty 8-bit MSB
(F0D) MF0D				**Function related to: PWM0, PWM1, Program Counter**
PCH	0d.7~4	R	0	Program Counter high byte, i.e. PC11~PC8
PWM1DL	0d.3~2	R/W	0	PWM1 duty 2-bit LSB
PWM0DL	0d.1~0	R/W	0	PWM0 duty 2-bit LSB
(F0E) PWM1DH				**Function related to: PWM1**
PWM1DH	0e.7~0	R/W	0	PWM1 duty 8-bit MSB

Name	Address	R/W	Rst	Description
(F0F) MF0F				**Function related to: Buzzer, ADC, CPU clock**
BUZEN	0f.7	R/W	0	Buzzer function, 1=enable, 0=disable
ADST	0f.6	R/W	0	ADC start bit. 0 :H/W clear after end of conversion 1: ADC start conversion
-	0f.5	R	0	N/A
FASTSTP	0f.4	R/W	0	Fast-clock Enable/Disable 0: Enable 1: Disable
CPUCKS	0f.3	R/W	0	System clock (Fsys) selection 0: Fast-clock 1: Slow-clock
SLOWEN	0f.2	R/W	0	If CPUCKS =1, this SLOWEN bit is invalid, Slow-clock keeps oscillating If CPUCKS =0, set 1 to enable Slow-clock oscillate, clear 0 to stop Slow-clock oscillating
-	0f.1	R	0	N/A
SLOWCKS	0f.0	R/W	1	Slow-clock type 0: SXT 1: SIRC
(F10) ADCDH				**Function related to: ADC**
ADCDH	10.7~0	R	-	ADC output data MSB[11:4]
(F11) MF11				**Function related to: ADC**
ADCDL	11.7~4	R	-	ADC output data LSB [3:0]
ADCHS	11.3~0	R/W	0	ADC channel select 0000: ADC0 0110: ADC6 0001: ADC1 0111: ADC7 0010: ADC2 1000: ADC8 0011: ADC3 1001: ADC9 0100: ADC4 1010: ADC10 0101: ADC5 1011: ADC11
(F12) MF12				**Function related to: PWM0, PWM1, Timer0, Timer1**
PWM0CLR	12.7	R/W	1	PWM0 counter clear 0: Release 1: Clear and hold
PWM1CLR	12.6	R/W	1	PWM1 counter clear 0: Release 1: Clear and hold
PWM0CKS	12.5	R/W	0	PWM0 clock source 0: Fsys 1: FIRC 16M
PWM1CKS	12.4	R/W	0	PWM1 clock source 0: Fsys 1: FIRC 16M
TM1SET	12.3	R/W	0	Timer1 counter set 0: Release 1: Set to FFFFh and hold
TM1CLR	12.2	R/W	0	Timer1 counter clear 0: Release 1: Clear to 0000H and hold
TM1STP	12.1	R/W	0	Timer1 counter stop 0: Release 1: Stop counting
TM0STP	12.0	R/W	0	Timer0 counter stop 0: Release 1: Stop counting

Name	Address	R/W	Rst	Description
(F13) MF13				**Function related to: OPA, Touch Key**
OPA0EN	13.7	R/W	0	OPA0 control 0: disable 1: enable
OPA1EN	13.6	R/W	0	OPA1 control 0: disable 1: enable
OP1IPSEL	13.5	R/W	0	OPA1 Non-inverted pin input selection 0: from PD3 1: from output of OPA0
-	13.4	-	0	Reserved
TKPD	13.3	R/W	1	Touch Key power down 0: power up 1: power down
TKCKS	13.2	R/W	1	Touch key PWM clock select, "TK-clock" is 0: 2 MHz 1: 4 MHz
TKPSC	13.1~0	R/W	0	Touch key counter data prescaler. Touch key prescaler divided by 0: TK-clock 1: TK-clock/2 2: TK-clock/4 3: TK-clock/8
(F14) TKCTL1				**Function related to: Touch Key**
TKSOC	14.7	R/W	0	Touch key start of conversion, rising edge to start
TKTMR	14.6~4	R/W	4	Touch key conversion time. 0=shortest, 7=longest
TKCHS	14.3~0	R/W	0	Touch key channel select, TKCHS[3:0]= 0000: TK0 0110: TK6 1100: TK12 0001: TK1 0111: TK7 1101: TK13 0010: TK2 1000: TK8 1110: TK14 0011: TK3 1001: TK9 1111: TK15 0100: TK4 1010: TK10 0101: TK5 1011: TK11
(F15) TKCTL2				**Function related to: Touch Key**
TKEOC	15.7	R	1	Touch key end of conversion, 1: end of conversion 0: conversion is in process
TKOVF	15.6	R	0	Touch key counter overflow
TKDH	15.5~4	R	-	Touch key counter high byte TKDATA[9:8]
(F16) TKDL				**Function related to: Touch Key**
TKDL	16.7~0	R	-	Touch key counter low byte TKDATA[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)

Note that the Touch Key function is always be power down when the body is TM57PA45, and MF13.3~0, TKCTL1, TKCTL2, TKDL functions would not affect the internal Touch Key function which is disabled permanently !

R-Plane

Name	Address	R/W	Rst	Description
(R02) TM0CTL				**Function related to: Timer0**
TM0CL	02.7	W	0	Timer0 Capture Mode Level 0: High level capture 1: Low level capture
TM0CM	02.6	W	0	Timer0 Mode 0: Timer/Counter Mode Clock source from TM0PSC (set R02.3~0) TM0CKI (set R02.4) 1: Capture Mode Clock source from CAPT pin
TM0EDG	02.5	W	0	Timer0 prescaler counting edge for TM0CKI pin 0: rising edge 1: falling edge
TM0CKS	02.4	W	0	Timer0 prescaler clock source 0: Instruction cycle 1: TM0CKI pin (PA2 pin)
TM0PSC	02.3~0	W	0	Timer0 prescaler. Timer0 prescaler clock source divided by 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1xxx: /256
(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) PAE				**Function related to: Port A**
PAE	05.6~3	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
	05.2~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is pseudo-open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
(R06) PBE				**Function related to: Port B**
PBE	06.5~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
(R07) PDE				**Function related to: Port D**
PDE	07.7~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output

Name	Address	R/W	Rst	Description
(R08) PAPUN				**Function related to: Port A**
PAPUN	08.7~0	W	7F	Each bit controls its corresponding pin, if the bit is 0: the pin pull up resistor is enabled, except a. the pin's output data register (PAD) is 0 b. the pin's CMOS push-pull mode is chosen (PAE=1) c. the pin is working for FXT/SXT/PWMs/TM0OUT/TM1OUT/TCOUT/Buzzer output 1: the pin pull up resistor is disabled
(R09) PBPUN				**Function related to: Port B**
PBPUN	09.5~0	W	3F	Each bit controls its corresponding pin, if the bit is 0: the pin pull up resistor is enabled, except a. the pin's output data register (PBD) is 0 b. the pin's CMOS push-pull mode is chosen (PBE=1) c. the pin is working for FXT/SXT/PWMs/TM0OUT/TM1OUT/TCOUT/Buzzer output 1: the pin pull up resistor is disabled
(R0A) PDPUN				**Function related to: Port D**
PDPUN	0a.7~0	W	FF	Each bit controls its corresponding pin, if the bit is 0: the pin pull up resistor is enabled, except a. the pin's output data register (PDD) is 0 b. the pin's CMOS push-pull mode is chosen (PDE=1) c. the pin is working for FXT/SXT/PWMs/TM0OUT/TM1OUT/TCOUT/Buzzer output 1: the pin pull up resistor is disabled
(R0B) MR0B				**Function related to: PWM0/PWM1**
PWM0POE	0b.7	W	0	0: PA0 as its function 1: enable PWM0P output to PA0 pin
PWM0NOE	0b.6	W	0	0: PD7 as its function 1: enable PWM0N output to PD7
PWM1POE	0b.5	W	0	0: PB0 as its function 1: enable PWM1P output to PB0
PWM1NOE	0b.4	W	0	0: PB1 as its function 1: enable PWM1N output to PB1
PWM0PSC	0b.3~2	W	00	PWM0 clock source is divided by Users must set these 2 bits to "00"
PWM1PSC	0b.1~0	W	00	PWM1 clock source is divided by User must set these 2 bits to "00"

<table>
<tr><th>Name</th><th>Address</th><th>R/W</th><th>Rst</th><th>Description</th></tr>
<tr><td colspan="4">(R0C) MR0C</td><td>Function related to : WDT/WKT/Timer0/Timer1/TCOUT</td></tr>
<tr><td>WKTPSC</td><td>0c.7~6</td><td>W</td><td>11</td><td>WKT Period
<table>
<tr><th colspan="2">VDD=5V</th><th>VDD=3V</th></tr>
<tr><td>00</td><td>1.1 ms</td><td>1.4 ms</td></tr>
<tr><td>01</td><td>2.2 ms</td><td>2.8 ms</td></tr>
<tr><td>10</td><td>36 ms</td><td>46 ms</td></tr>
<tr><td>11</td><td>144 ms</td><td>184 ms</td></tr>
</table></td></tr>
<tr><td>WDTPSC</td><td>0c.5~4</td><td>W</td><td>01</td><td>WDT Period
<table>
<tr><th colspan="2">VDD=5V</th><th>VDD=3V</th></tr>
<tr><td>00</td><td>144 ms</td><td>284 ms</td></tr>
<tr><td>01</td><td>289 ms</td><td>367 ms</td></tr>
<tr><td>10</td><td>1155 ms</td><td>1469 ms</td></tr>
<tr><td>11</td><td>2312 ms</td><td>2939 ms</td></tr>
</table></td></tr>
<tr><td>TM1CKS</td><td>0c.3</td><td>W</td><td>0</td><td>Timer1 clock source
0: Fsys/2 (instruction cycle)
1: Fsys</td></tr>
<tr><td>TM0OE</td><td>0c.2</td><td>W</td><td>0</td><td>Timer0 overflow toggle output to PA5
0: disable
1: enable</td></tr>
<tr><td>TCOE</td><td>0c.1</td><td>W</td><td>0</td><td>Instruction cycle (Fsys/2) output to PD6
0: disable
1: enable</td></tr>
<tr><td>TM1OE</td><td>0c.0</td><td>W</td><td>0</td><td>Timer1 overflow toggle output to PD0
0: disable
1: enable</td></tr>
<tr><td colspan="4">(R0D) MR0D</td><td>Function related to : INT1/Timer1/WDT/FIRC/ADC</td></tr>
<tr><td>INT1EDG</td><td>0d.7</td><td>W</td><td>0</td><td>0: INT1 pin falling edge to trigger interrupt event
1: INT1 pin rising edge to trigger interrupt event</td></tr>
<tr><td>TM1CM</td><td>0d.6</td><td>W</td><td>0</td><td>Timer1 Mode
0:Timer Mode (source form TM1PSC clock out)
1:Capture Mode (source from CAPT pin), measure CAPT pin period time
between successive rising or falling edges</td></tr>
<tr><td>WDTSTP</td><td>0d.5</td><td>W</td><td>0</td><td>WDT disable in STOP mode
If WDTE=0, this bit is don’t care.
0: stop counting WDT in STOP mode
1: always counting WDT in STOP mode</td></tr>
<tr><td>FIRCKS</td><td>0d.4~3</td><td>W</td><td>01</td><td>FIRC clock selection
00: 2 MHz
01: 4 MHz
10: 8 MHz
11: 16 MHz</td></tr>
<tr><td>ADCKS</td><td>0d.2~0</td><td>W</td><td>111</td><td>ADC clock frequency selection
000: Fsys/256
001: Fsys/128
010: Fsys/64
011: Fsys/32
100: Fsys/16
101: Fsys/8
110: Fsys/4
111: Fsys/2</td></tr>
</table>

Name	Address	R/W	Rst	Description
(R0E) BUZCTL				**Function related to: Buzzer**
BUZPSC	0e.7~6	W	00	Buzzer clock frequency selection 00: Fsys/8 01: Fsys/16 10: Fsys/32 11: Fsys/64
BUZPRD	0e.5~0	W	0	Buzzer Period
(R0F) Reserved				**Tenx reserved**
Reserved	0f.7~0	-	-	Tenx reserved register. Users do not write it.
(R10) PWM0PRD				**Function related to: PWM0**
PWM0PRD	10.7~0	W	FF	PWM0 Period
(R11) PWM1PRD				**Function related to: PWM1**
PWM1PRD	11.7~0	W	FF	PWM1 Period
(R12) PAM				**Function related to: Port A**
PAM	12.7~0	W	FF	Each bit control its corresponding pin 0: disable I/O digital input to save power when ADC channels are selected 1: enable I/O digital input
(R13) PBM				**Function related to: Port B**
PBM	13.5~0	W	3F	Each bit control its corresponding pin 0: disable I/O digital input to save power when ADC channels are selected 1: enable I/O digital input
(R14) PDM				**Function related to Port D**
PDM	14.7~0	W	FF	Each bit control its corresponding pin 0: disable I/O digital input to save power when ADC channels are selected 1: enable I/O digital input
(R15) PBWKEN				**Function related to: Wake up**
PBWKEN	15.5~0	W	00	PB5~PB0 low level wakeup 0: disable 1: enable
(R16) PWMCTL				**Function related to: PWM0/PWM1**
PWM1MODE	16.7~6	W	0	PWM1 differential output mode 00: Mode 0, 01: Mode 1, 10: Mode 2, 11: Mode 3
PWM1DTC	16.5~4	W	0	00: original PWM1, 01: non-overlap 1 PWM1 clock 10: non-overlap 2 PWM1 clocks 11: non-overlap 4 PWM1 clocks
PWM0MODE	16.3~2	W	0	PWM0 differential output mode 00: Mode 0, 01: Mode 1, 10: Mode 2, 11: Mode 3
PWM0DTC	16.1~0	W	0	00: original PWM0, 01: non-overlap 1 PWM0 clock 10: non-overlap 2 PWM0 clocks 11: non-overlap 4 PWM0 clocks

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
ANDLW	k	01 1101 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 1010 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
XORLW	k	01 1111 kkkk kkkk	1	Z	XOR Literal "k" with W
TABRH		00 0000 0101 1000	2	-	Lookup ROM high data to W
TABRL		00 0000 0101 0000	2	-	Lookup ROM low data to 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) ← ($\bar{f}$)	
Status Affected	Z	
OP-Code	00 1001 dfff ffff	
Description	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	COMF REG1, 0	B : REG1 = 0x13 A : REG1 = 0x13, W = 0xEC

DECF — Decrement "f"

Syntax	DECF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (f) - 1	
Status Affected	Z	
OP-Code	00 0011 dfff ffff	
Description	Decrement register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	DECF CNT, 1	B : CNT = 0x01, Z = 0 A : CNT = 0x00, Z = 1

DECFSZ — Decrement "f", Skip if 0

Syntax	DECFSZ f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (f) - 1, skip next instruction if result is 0	
Status Affected	-	
OP-Code	00 1011 dfff ffff	
Description	The contents of register 'f' are decremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, then a NOP is executed instead, making it a 2 cycle instruction.	
Cycle	1 or 2	
Example	LABEL1 DECFSZ CNT, 1 GOTO LOOP CONTINUE	B : PC = LABEL1 A : CNT = CNT – 1 if CNT = 0, PC = CONTINUE if CNT ≠ 0, PC = LABEL1 + 1

GOTO — Unconditional Branch

Syntax	GOTO k	
Operands	k : 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        B : PC = LABEL1
        GOTO  LOOP            A : CNT = CNT + 1
        CONTINUE                  if CNT = 0, PC = CONTINUE
                                  if CNT ≠ 0, PC = LABEL1 + 1
```

IORLW — Inclusive OR Literal with W

Syntax	IORLW k
Operands	k : 00h ~ FFh
Operation	(W) ← (W) OR k
Status Affected	Z
OP-Code	01 1010 kkkk kkkk
Description	The contents of the W register are OR'ed with the eight-bit literal 'k'. The result is placed in the W register.
Cycle	1
Example	IORLW 0x35 — B : W = 0x9A A : W = 0xBF, Z = 0

IORWF — Inclusive OR W with "f"

Syntax	IORWF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	(destination) ← (W) OR k
Status Affected	Z
OP-Code	00 0100 dfff ffff
Description	Inclusive OR the W register with register 'f'. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.
Cycle	1
Example	IORWF RESULT, 0 — B : RESULT = 0x13, W = 0x91 A : RESULT = 0x13, W = 0x93, Z = 0

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

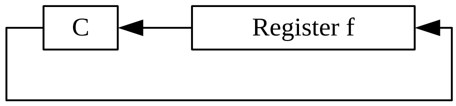
NOP	No Operation	
Syntax	NOP	
Operands	-	
Operation	No Operation	
Status Affected	-	
OP-Code	00 0000 0000 0000	
Description	No Operation	
Cycle	1	
Example	NOP	-

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

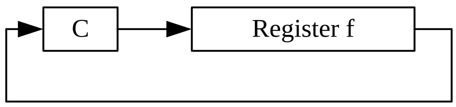
RETI	Return from Interrupt	
Syntax	RETI	
Operands	-	
Operation	PC ← TOS, GIE ← 1	
Status Affected	-	
OP-Code	00 0000 0110 0000	
Description	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in to the PC. Interrupts are enabled. This is a two-cycle instruction.	
Cycle	2	
Example	RETI	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 : TABLE ADDWF PCL, 1 RETLW k1 RETLW k2 : RETLW kn	B : W = 0x07 A : W = value of k8

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 — -

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

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

TABRL Return DPTR low byte to W

Syntax	TABRL
Operands	-
Operation	(W) ← ROM[DPTR] low byte content, Where DPTR={DPH[max:8],DPL[7:0]} After TABRL is executed, DPTR ← DPTR+1 automatically
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   DPL                 ; Where DPL is F-plane register
MOVLW   (TAB1>>8)&0xFF
MOVWF   DPH                 ; Where DPH is F-plane register

                            ; DPTR=0234H
TABRH                       ; W=0x37
TABRL                       ; W=0x89, DPTR=0235H
TABRH                       ; W=0x22
TABRL                       ; W=0x77, DPTR=0236H


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

TABRH Return DPTR high byte to W

Syntax	TABRH
Operands	-
Operation	(W) ← ROM[DPTR] high byte content, Where DPTR={DPH[max:8],DPL[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

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}=2.0V to 5.5V unless otherwise specified)

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Operating Voltage	V_{DD}	FAST mode, 25°C, Fsys=24 MHz		4.2	–	5.5	V
		FAST mode, 25°C, Fsys=16 MHz		3.3	–	5.5	
		FAST mode, 25°C, Fsys=8 MHz		2.4	–	5.5	
		FAST mode, 25°C, Fsys=4 MHz		2.2	–	5.5	
		SLOW mode, 25°C, SIRC		1.7	–	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}	4	8	–	mA
			V_{DD}=3V, V_{OH}=0.9V_{DD}	2	4	–	
PWM Ports Source Current (TM57PT45C/PA45C)	I_{OH}	PA0, PD7 PB0, PB1	V_{DD}=5V, V_{OH}=0.9V_{DD}	13	27		
			V_{DD}=3V, V_{OH}=0.9V_{DD}	5	10		
I/O Port Sink Current	I_{OL}	All Output, except PA7	V_{DD}=5V, V_{OL}=0.1V_{DD}	10	20	–	mA
			V_{DD}=3V, V_{OL}=0.1V_{DD}	5	10	–	
		PA7	V_{DD}=5V, V_{OL}=0.1V_{DD}	15	30	–	
			V_{DD}=3V, V_{OL}=0.1V_{DD}	6	12	–	
		High sink current pins	V_{DD}=5V, V_{OL}=0.1V_{DD}	20	40	–	
			V_{DD}=3V, V_{OL}=0.1V_{DD}	10	20	–	
PWM Ports Sink Current (TM57PT45C/PA45C)	I_{OL}	PA0, PD7 PB0, PB1	V_{DD}=5V, V_{OL}=0.1V_{DD}	20	40		
			V_{DD}=3V, V_{OL}=0.1V_{DD}	10	20		
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	–	3.5	–	mA
			V_{DD}=3V, FXT=12MHz	–	1.7	–	
			V_{DD}=5V, FXT=8 MHz	–	2.6	–	
			V_{DD}=3V, FXT=8 MHz	–	1.2	–	
			V_{DD}=5V, FXT=4 MHz	–	1.6	–	
			V_{DD}=3V, FXT=4 MHz	–	0.7	–	
			V_{DD}=5V, FIRC=8 MHz	–	2.4	–	
			V_{DD}=3V, FIRC=8 MHz	–	1.3	–	
		SLOW mode, LVR enable	V_{DD}=5V, SXT=32 KHz	–	127	–	μA
			V_{DD}=3V, SXT=32 KHz	–	38	–	
			V_{DD}=5 V, SIRC, CPUPSC=11	–	139	–	
			V_{DD}=3 V, SIRC, CPUPSC=11	–	44	–	
		STOP mode, LVR enable	V_{DD}=5V	–	1.0	–	
			V_{DD}=3V	–	0.4	–	
		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	–	–	12	MHz
			V_{DD}=2.1V	–	–	8	
			V_{DD}=1.6V	–	–	4	
LVR Reference Voltage	V_{LVR}	T_A=25°C		–	2.0	–	V
				–	2.9	–	V
LVR Hysteresis Voltage	V_{HYST}	T_A=25°C		–	±0.1	–	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	–	62	–	KΩ
			V_{DD}=3V		113		
		V_{IN}=0 V PA7	V_{DD}=5V	–	53	–	
			V_{DD}=3V		109		

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	

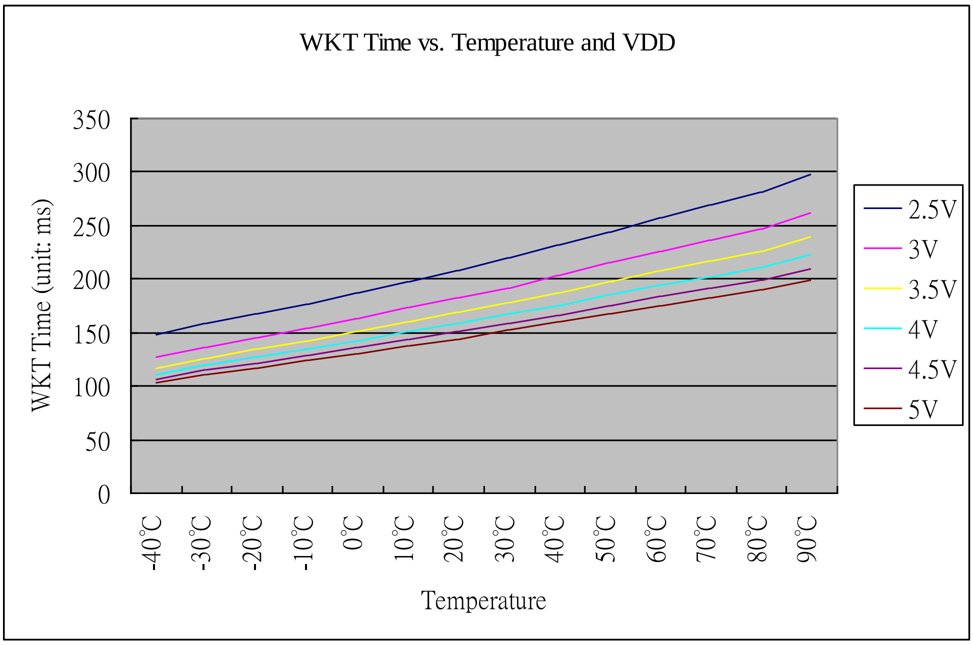
4. Reset Timing Characteristics (T_A= -40°C to +85°C, V_{DD}=3V to 5V)

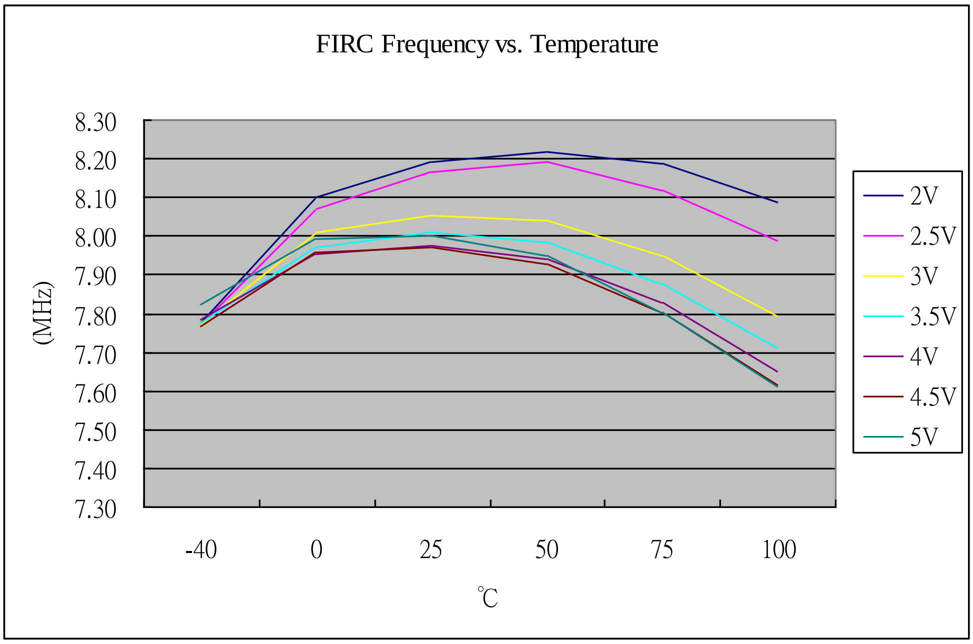
Parameter	Conditions	Min	Typ	Max	Unit
RESET Input Low width	Input V_{DD}=5 V ±10 %	3	–	–	μs
WDT wakeup time	V_{DD}=5V, WDTPSC=00	–	19	–	ms
	V_{DD}=3V, WDTPSC=00	–	24	–	
CPU start up time	V_{DD}=5V	–	19	–	ms
	V_{DD}=3V	–	24	–	

5. OPA Electrical Characteristics (VDD=5V, TA=25°C)

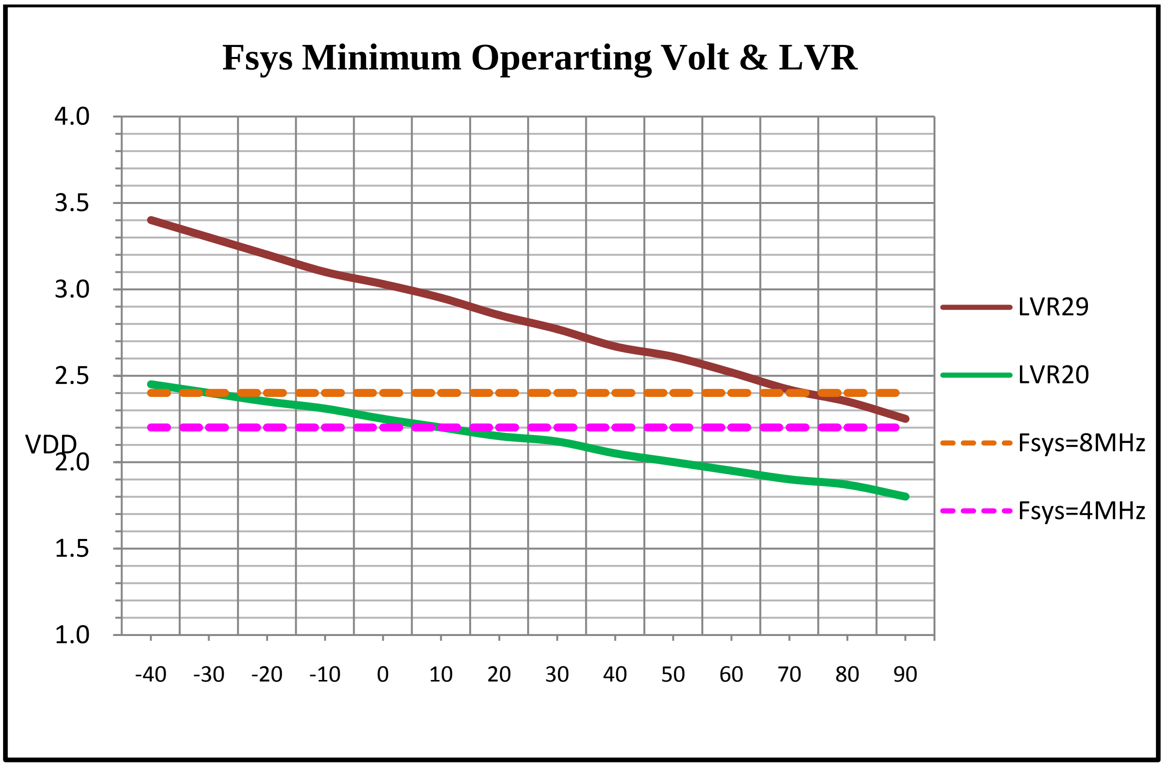
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VOS0	Input Offset Voltage for op0	Vo=1.4V			5	mv
VOS1	Input Offset Voltage for op1	Vo=1.4V			5	mv
AVOL	Large Signal Voltage Gain	RL=1MΩ CL=60pF		80		dB
GBW	Gain Band Width Product	RL=1MΩ CL=60pF		1		MHz
CMRR	Common Mode Rejection Ratio	Vo=1.4V Vi=0V		80		dB
PSRR	Power Supply Rejection Ratio	Vo=1.4V		60		dB
ICC	Supply Current Per Single Amplifier	Av=1 Vo=1.4V No load		40	60	μA
SR	Slew Rate at Unity Gain	No load		0.3		V/μs
Φm	Phase Margin at Unity Gain	RL=1MΩ CL=60pF		30		Degree
IOH	Output Source Current			-500		μA
IOL	Output Sink Current			500		μA

6. Characteristic Graphs





Fsys Minimum Operating Voltage & LVR relationship

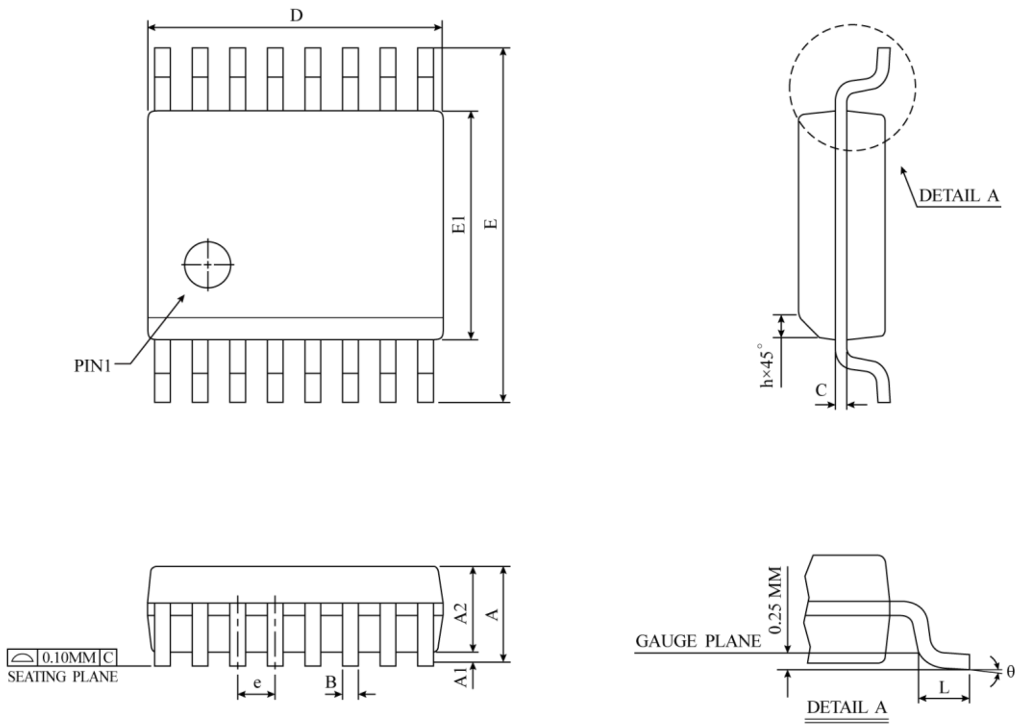


PACKAGING INFORMATION

The ordering information:

Ordering number	Package
TM57PT45-OTP TM57PA45-OTP TM57PT45C-OTP TM57PA45C-OTP	Wafer/Dice blank chip
TM57PT45-COD TM57PA45-COD TM57PT45C-COD TM57PA45C-COD	Wafer/Dice with code
TM57PT45-OTP-26 TM57PA45-OTP-26 TM57PT45C-OTP-26 TM57PA45C-OTP-26	SSOP 16-pin (150mil)
TM57PT45-OTP-05 TM57PA45-OTP-05 TM57PT45C-OTP-05 TM57PA45C-OTP-05	DIP 20-pin (300 mil)
TM57PT45-OTP-21 TM57PA45-OTP-21 TM57PT45C-OTP-21 TM57PA45C-OTP-21	SOP 20-pin (300 mil)
TM57PT45-OTP-22 TM57PA45-OTP-22 TM57PT45C-OTP-22 TM57PA45C-OTP-22	SOP 24-pin (300 mil)
TM57PT45-OTP-28 TM57PA45-OTP-28 TM57PT45C-OTP-28 TM57PA45C-OTP-28	SSOP 24-pin (150 mil)

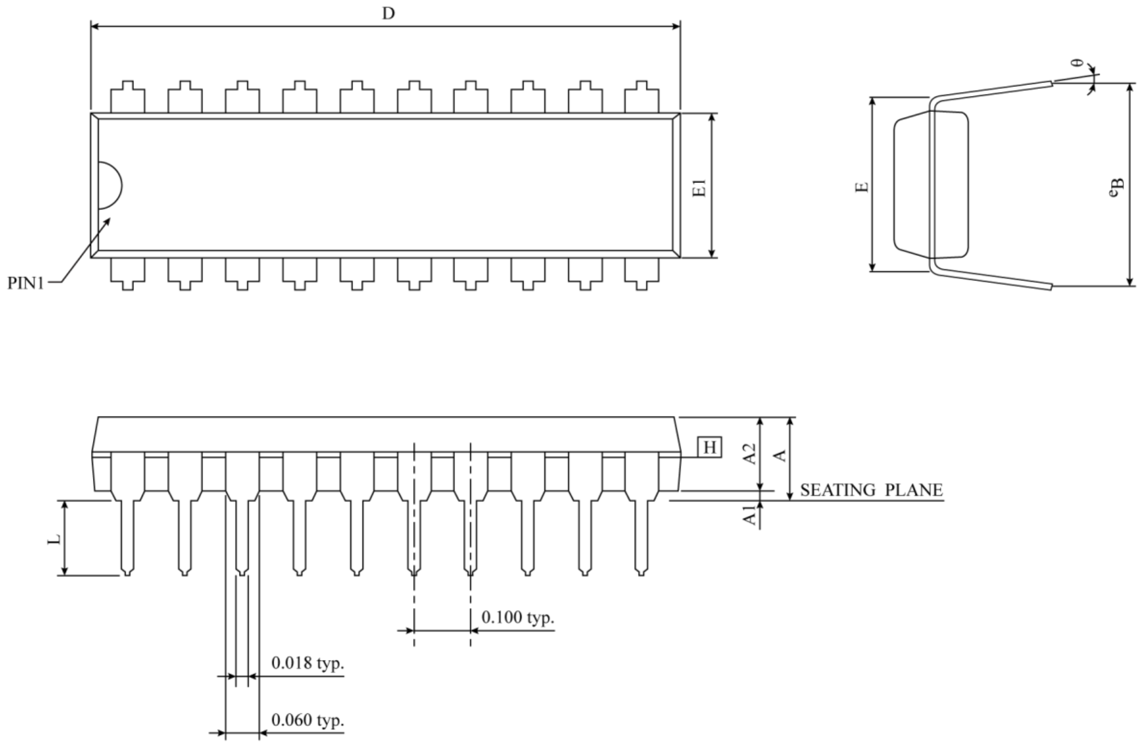
16-SSOP (150mil) Package Dimension



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.55	1.75	0.053	0.061	0.069
A1	0.10	0.18	0.25	0.004	0.007	0.010
A2	-	-	1.50	-	-	0.059
B	0.20	0.25	0.30	0.008	0.010	0.012
C	0.18	0.22	0.25	0.007	0.009	0.010
D	4.80	4.90	5.00	0.189	0.193	0.197
E	5.79	6.00	6.20	0.228	0.236	0.244
E1	3.81	3.90	3.99	0.150	0.154	0.157
e	0.635 BSC			0.025 BSC		
L	0.41	0.84	1.27	0.016	0.033	0.050
θ	0°	4°	8°	0°	4°	8°
JEDEC	M0-137 (AB)					

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

20-DIP Package Dimension

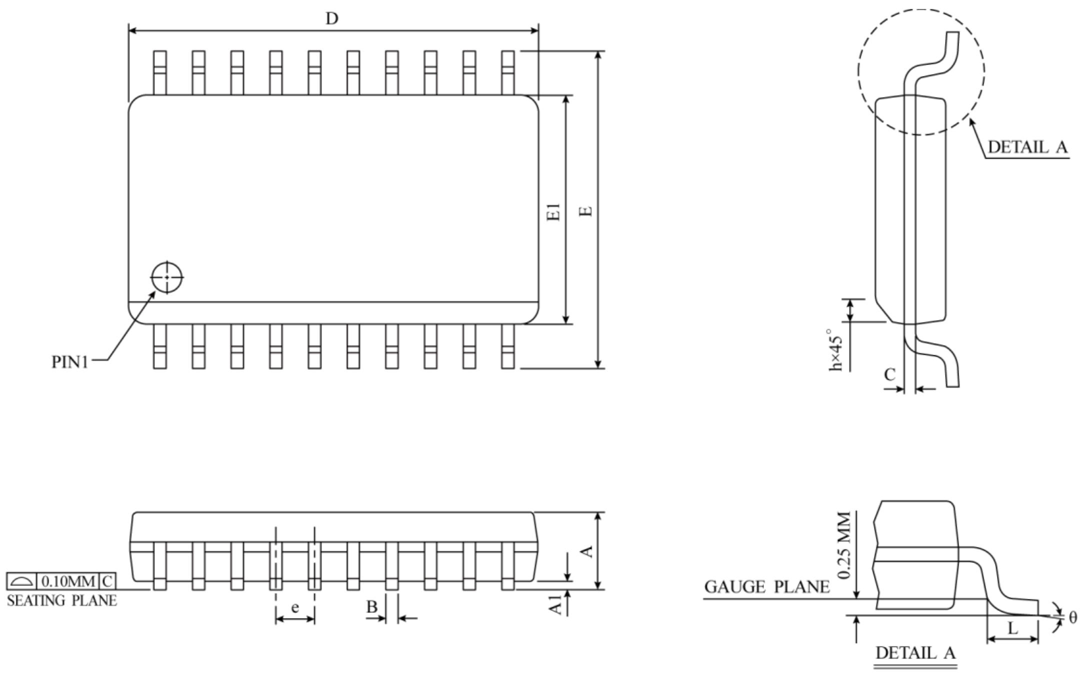


SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	4.445	-	-	0.175
A1	0.381	-	-	0.015	-	-
A2	3.175	3.302	3.429	0.125	0.130	0.135
D	25.705	26.061	26.416	1.012	1.026	1.040
E	7.620	7.747	7.874	0.300	0.305	0.310
E1	6.223	6.350	6.477	0.245	0.250	0.255
L	3.048	3.302	3.556	0.120	0.130	0.140
e_B	8.509	9.017	9.525	0.335	0.355	0.375
θ	0°	7.5°	15°	0°	7.5°	15°
JEDEC	MS-001 (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.

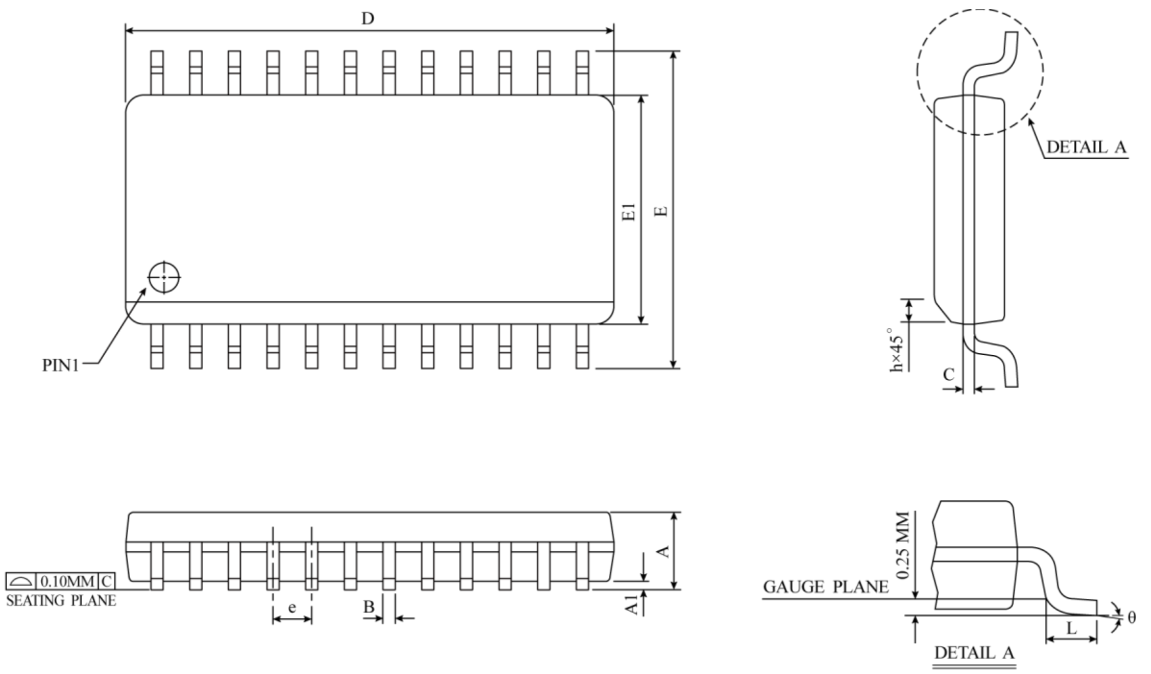
20-SOP Package Dimension



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

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

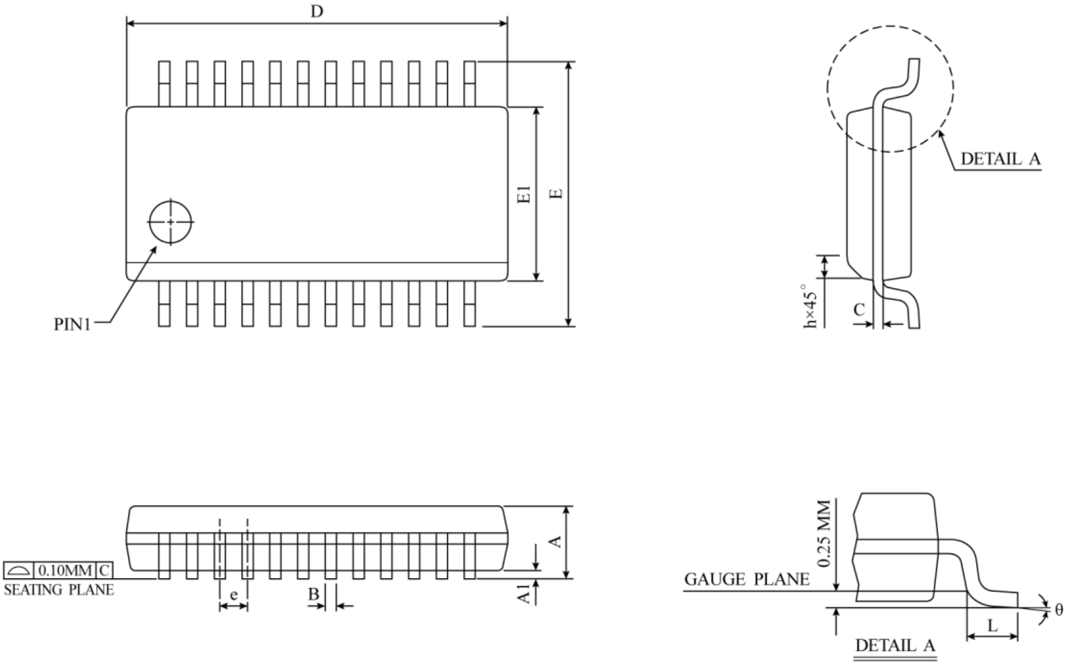
24-SOP Package Dimension



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	2.35	2.50	2.65	0.0926	0.0985	0.1043
A1	0.10	0.20	0.30	0.0040	0.0079	0.0118
B	0.33	0.42	0.51	0.0130	0.0165	0.0200
C	0.23	0.28	0.32	0.0091	0.0108	0.0125
D	15.20	15.40	15.60	0.5985	0.6063	0.6141
E	10.00	10.33	10.65	0.3940	0.4425	0.4910
E1	7.40	7.50	7.60	0.2914	0.2953	0.2992
e	1.27 BSC			0.050 BSC		
h	0.25	0.50	0.75	0.0100	0.0195	0.0290
L	0.40	0.84	1.27	0.0160	0.0330	0.0500
θ	0°	4°	8°	0°	4°	8°
JEDEC	MS-013 (AD)					

⚠ * 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.

24-SSOP (150mil) Package Dimension



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.55	1.75	0.053	0.061	0.069
A1	0.10	0.18	0.25	0.004	0.007	0.010
A2	-	-	1.50	-	-	0.059
B	0.20	0.25	0.30	0.008	0.010	0.012
C	0.18	0.22	0.25	0.007	0.009	0.010
D	8.56	8.65	8.74	0.337	0.341	0.344
E	5.79	6.00	6.20	0.228	0.236	0.244
E1	3.81	3.90	3.99	0.150	0.154	0.157
e	0.635 BSC			0.025 BSC		
L	0.41	0.84	1.27	0.016	0.033	0.050
θ	0°	4°	8°	0°	4°	8°
JEDEC	M0-137 (AE)					

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