十速

TM57PT46/PA46

DATA SHEET

Rev 1.0

AMENDMENT HISTORY

Version	Date	Description
V1.0	Aug, 2016	New release

CONTENTS

AMENDMENT HISTORY **2**
CONTENTS **3**
FEATURES **5**
BLOCK DIAGRAM **7**
PIN ASSIGNMENT **8**
PIN DESCRIPTIONS **9**
PIN SUMMARY **10**
FUNCTIONAL DESCRIPTION **12**
1. CPU Core **12**
1.1 Clock Scheme and Instruction Cycle 12
1.2 RAM Addressing Mode 13
1.3 Programming Counter (PC) and Stack 15
1.4 ALU and Working (W) Register 16
1.5 STATUS Register (F-Plane 03H) 17
1.6 Interrupt 19
2 Chip Operation Mode **24**
2.1 Reset 24
2.2 System Configuration Register (SYSCFG) 25
2.3 PROM Re-use ROM 26
2.4 Power-Down Mode 27
2.5 Dual System Clock 28
2.6 Dual System Clock Modes Transition 30
3 Peripheral Functional Block **33**
3.1 Watchdog (WDT) Timer/Wakeup (WKT) Timer 33
3.2 Timer0: 8-bit Timer/Counter with Pre-scale (PSC) 36
3.3 Timer1 42
3.4 PWM: (8+2) bits PWM 49
3.5 PPG (Programmable Pulse Generator) 53
3.6 Buzzer Output 58
3.7 Touch Key 59
3.8 ADC: 12-bit Analog-to-Digital Converter 63
3.9 OPA/Comparators offset voltage trimming procedures 66
3.10 System Clock Oscillator 69
4 I/O Port **70**
4.1 PA0-2 70
4.2 PA3-6, PB0-4, PD0-7 71
4.3 PA7 73

MEMORY MAP 77
F-Plane 77
R-Plane 82
INSTRUCTION SET 87
ELECTRICAL CHARACTERISTICS 99
1. Absolute Maximum Ratings 99
2. DC Characteristics 100
3. Clock Timing 101
4. Reset Timing Characteristics 101
5. OPA Electrical Characteristics 101
6. COMPARATOR Electrical characteristics 102
7. Characteristic Graphs 103
PACKAGING INFORMATION 105

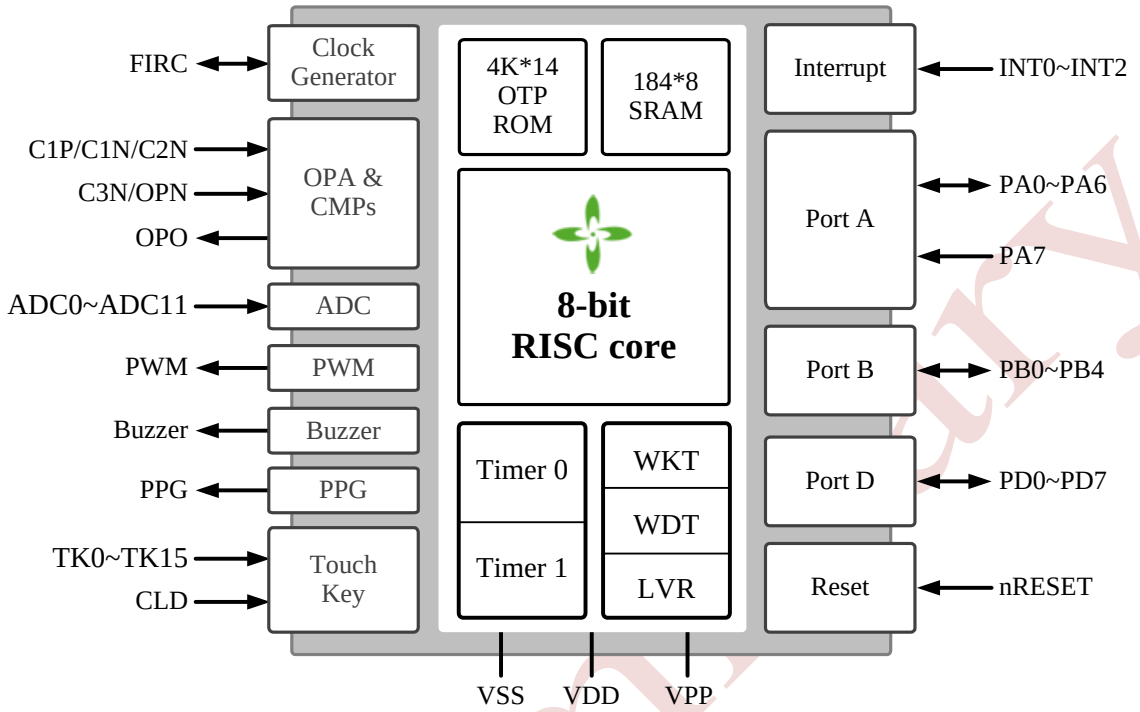
FEATURES

1. **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
   - ADC Interrupt
   - CMP1 rising or falling triggered interrupt
   - CMP2/CMP3 falling edge triggered interrupt
   - CMP4 over current duration triggered interrupt
2. **Port B individual pin low level wake up**
3. **Wake-up (WKT) Timer**
   - Clocked by built-in RC oscillator with 4 adjustable Interrupt times
     1.1 ms/2.3 ms/36 ms/145 ms @5V, 1.4 ms/2.8 ms/46 ms/182 ms @3V
4. **Watchdog Timer**
   - Clocked by built-in RC oscillator with 4 adjustable Reset Times
     145 ms/290 ms/1160 ms/2320 ms @5V, 180 ms/364 ms/1456 ms/2913 ms @3V
     Watchdog timer can be disabled/enabled in STOP mode (WDTSTP, (R0D.5))
5. **1 Independent PWM**
   - 8+2 bits, period-adjustable/duty-adjustable/Clear&Hold
   - Clock source: FIRC 8 MHz and 16 MHz which double of FIRC
6. **12-bit ADC converter with 12 input channels**
7. **Programmable Pulse Generator (PPG) function for Induction Heating**
8. **15 channel Touch Key (TM57PT46 only)**
9. **1 Operational Amplifiers with output connect to CMP4 inverted terminal**
10. **4 specified Comparators cooperate with PPG function**
11. **Reset Sources**
    - Power On Reset
    - Watchdog Reset
    - Low Voltage Reset
    - External Pin Reset
12. **Low Voltage Reset Option: LVR2.1V, LVR2.1V disable in STOP mode, LVR3.1V, and disable**

13. **Operating Voltage: Low Voltage Reset Level to 5.5V**
    - Fsys=4 MHz, 1.8V~5.5V
    - Fsys=8 MHz, 2.1V~5.5V
    - Fsys=16 MHz, 2.9V~5.5V
14. **Enhanced Power Noise Rejection.**
15. **Operating Temperature Range: -40°C to +85°C**
16. **Instruction set: 38 Instructions**
17. **Instruction Execution Time**
    - 2 oscillation clocks per instruction except branch
18. **I/O ports: Maximum 21 programmable I/O pins**
    - Pseudo-Open-Drain Output
    - Open-Drain Output
    - CMOS Push-Pull Output
    - Schmitt Trigger Input with pull-up resistor option
19. **Package Types:**
    - 24-pin SOP (300 mil)
    - 20-pin DIP (300 mil), SOP (300 mil)
    - 18-pin SOP
20. **Supported EV board on ICE**

    EV board: EV8203

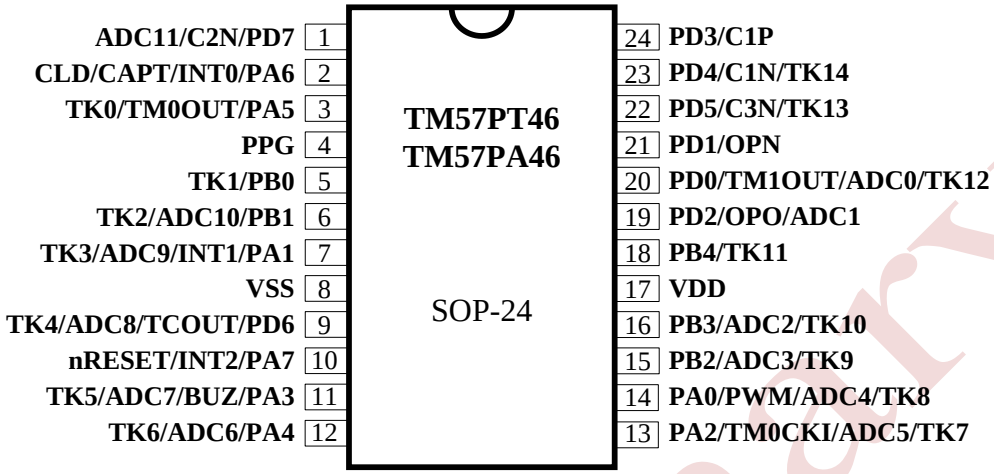
BLOCK DIAGRAM

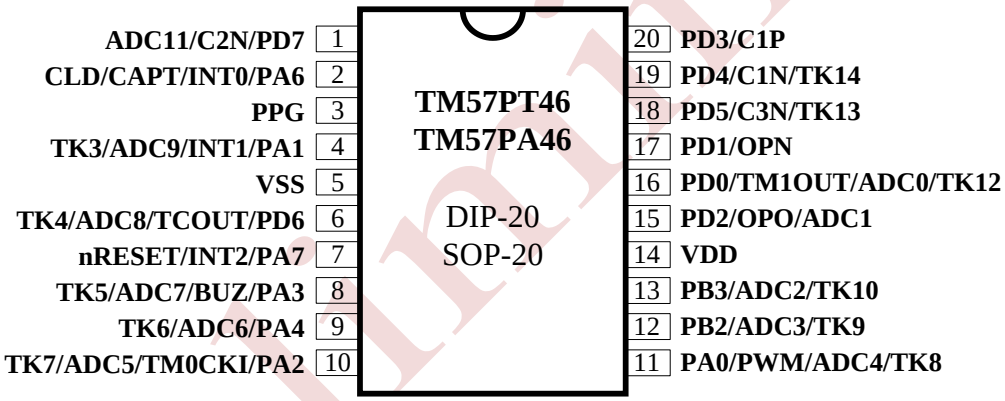


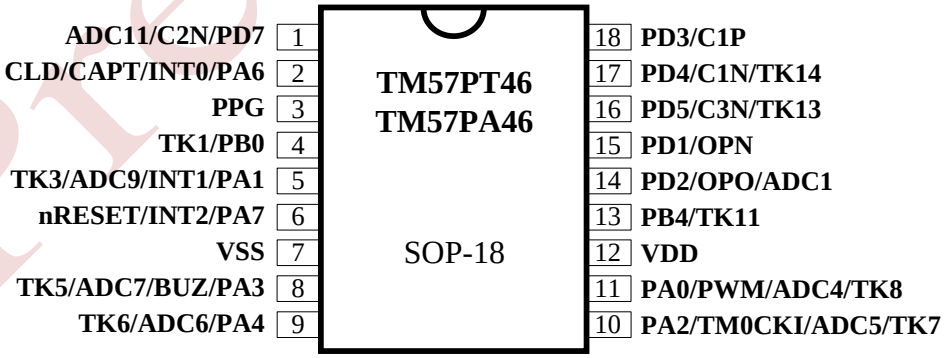
TM57PT46/PA46 Block Diagram

Note that Touch Key block is only for TM57PT46

PIN ASSIGNMENT







*** Note that TM57PA46 doesn't have TK0~TK14 and CLD pins.**

PIN DESCRIPTIONS

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–PB4 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.
PPG	O	PPG output
VPP/nRESET/ PA7	I/O	Schmitt-trigger input with pull-high configurable, External active low reset, normal stay to "high".
VDD, VSS	P	Power Voltage input pin and ground
VPP	I	PROM programming high voltage input
INT0–INT2	I	External interrupt input
PWM	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~TK14	I	Touch Key input (for TM57PT46 only)
CLD	I	Touch Key capacitor input (for TM57PT46 only)
C1P, C1N	I	Synchronous comparator CMP1 Positive/Negative inputs
C2N	I	IGBT over-voltage comparator CMP2 Negative input
C3N	I	Power over-voltage comparator CMP3 Negative input
OPN	I	Negative terminal of OPA
OPO	O	Outputs of OPA, also connects to Negative terminal of CMP4 (Over-current comparator).

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 /S-DIP	20-SOP/DIP	18-SOP/DIP			Weak Pull-up	Ext. Interrupt	O.D	P.O.D	P.P		PWM	Touch Key	ADC	PPG	MISC
1	1	1	PD7	I/O	O		O		O	PD7					
2	2	2	CLD/CAPT INT0/PA6	I/O	O	O	O		O	PA6		O			CAPT
3			TK0/ TM0OUT/PA5	I/O	O		O		O	PA5		O	O		TM0OUT
4	3	3	PPG	O					O	PPG				O	
5		4	TK1/PB0	I/O	O		O		O	PB0		O			
6			TK2/PB1/ ADC10	I/O	O		O		O	PB1		O			
7	4	5	TK3/ADC9/ INT1/PA1	I/O	O	O		O	O	PA1		O	O		
8	5	7	VSS	P						VSS					
9	6		TK4/ADC8/ TCOUT/PD6	I/O	O		O		O	PD6		O	O		TCOUT
10	7	6	nRESET/VPP INT2/PA7	I/O	O	O	O		O	*					VPP nRESET
11	8	8	TK5/ADC7/ BUZ/PA3	I/O	O		O		O	PA3		O	O		BUZ
12	9	9	TK6/ADC6/ PA4	I/O	O		O		O	PA4		O	O		
13	10	10	TK7/ADC5/ TM0CKI/PA2	I/O	O			O	O	PA2		O	O		TM0CKI
14	11	11	TK8/ADC4/ PWM/PA0	I/O	O			O	O	PA0	O	O	O		
15	12		TK9/ADC3/ PB2	I/O	O		O		O	PB2		O	O		
16	13		TK10/ADC2/ PB3	I/O	O		O		O	PB3		O	O		
17	14	12	VDD	P				O		VDD					
18		13	TK11/PB4	I/O	O		O		O	PB4		O			
19	15	14	ADC1/OPO/ PD2	I/O	O		O		O	PD2			O	O	
20	16		TK12/ADC0/ TM1OUT/PD0	I/O	O		O		O	PD0		O	O		TM1OUT

Pin Number			Pin Name	Type	GPIO					Function After Reset	Alternate Function				
					Input		Output								
24-SOP /S-DIP	20-SOP/DIP	18-SOP/DIP			Weak Pull-up	Ext. Interrupt	O.D	P.O.D	P.P		PWM	Touch Key	ADC	PPG	MISC
21	17	15	OPN/PD1	I/O	O		O	O	O	PD1				O	
22	18	16	TK13/C3N/ PD5	I/O	O		O	O	O	PD5		O		O	
23	19	17	TK14/C1N/ PD4	I/O	O		O		O	PD4		O		O	
24	20	18	TK15/C1P/ PD3	I/O	O		O		O	PD3				O	

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

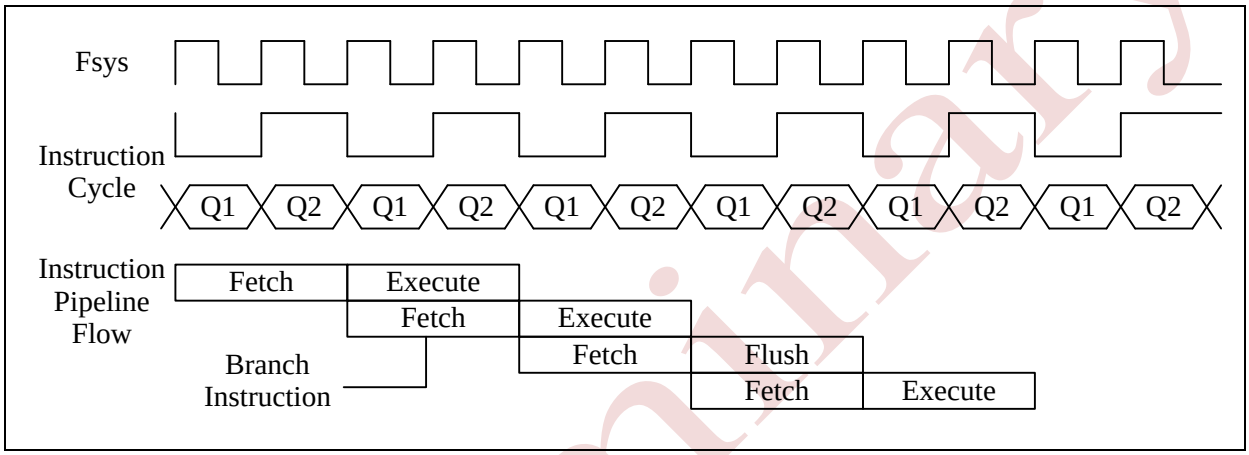
* Depends on XRSTE bit of Configword

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 is from Fast Internal RC oscillator (FIRC).

Slow-clock: The clock source is from Slow Internal RC oscillator (SIRC).

Instruction Cycle=Fsys/2

*FXT: Fast Crystal

FIRC: Fast Internal RC oscillator

*XRC: Fast or Slow External RC oscillator

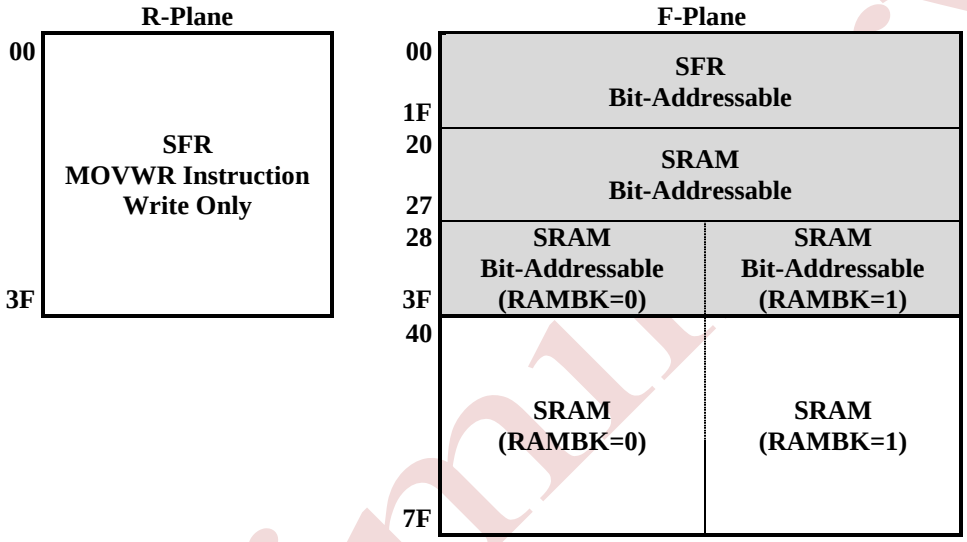
*SXT: Slow Crystal (32 KHz)

SIRC: Slow Internal RC oscillator

* TM57PT46/PA46 don't support FXT/XRC/SXT modes.

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: TM57PT46/PA46 defines 256 ROM addresses as one page, so that TM57PT46/PA46 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.

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

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

◇Example: Write immediate data into STATUS register

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

◇Example: Bit addressing set and clear STATUS register

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

◇Example: Determine the C flag by BTFSS instruction

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

◇Example: Detect WDT time out event occurrs

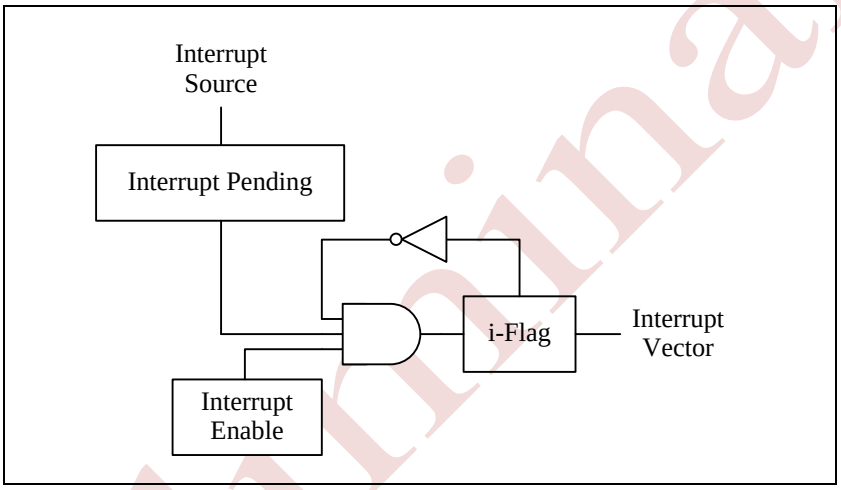
```
LOOP:
        BTFSC   STATUS, TO          ; Check the LVD flag
        GOTO    WDT_Timeout_Proc    ; If TO=1, goto WDT_Timeout_Proc
MAIN:   …….
```

1.6 Interrupt

The TM57PT46/PA46 has 1 level, 1 vector and 11 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 TM57PT46/PA46 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   xxxxxxx1B
            MOVWR   R0B             ; Set INT1 interrupt trigger as rising edge
            MOVLW   111111 0 1B
            MOVWF   INTF            ; Clear INT1 interrupt request flag
            MOVLW   000000 1 0B
            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   111111 0 1B
            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	ADCIE	-	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 **ADCIE**: ADC interrupt enable
0: disable
1: enable

F08.6 N/A

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	ADCIF	CMPIF	TM1IF	TM0IF	WKTIF	XINT2F	XINT1F	XINT0F
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	-	0	0	0	0	0	0

F09.7 **ADCIF**: ADC interrupt event pending flag
This bit is set by H/W while A/D conversion is completed, write 0 to this bit will clear this flag

F09.6 **CMPIF**: Comparators interrupt event pending flag
This bit is set by H/W while CMP1IF or CMP2IF or CMP3IF or CMP4IF is/are set, write 0s to those bits will clear this flag.
Because of the output of comparators may change at power on, so the CMPIF may not be '0'. Make sure the comparators are all in stable state then clear CMP1IF to CMP4IF before use.

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

F0E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMPIEF	CMP1IE	CMP2IE	CMP3IE	CMP4IE	CMP1IF	CMP2IF	CMP3IF	CMP4IF
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

F0E.7 **CMP1IE**: CMP1 interrupt enable
0: disable
1: enable

F0E.6 **CMP2IE**: CMP2 interrupt enable
0: disable
1: enable

F0E.5 **CMP3IE**: CMP3 interrupt enable
0: disable
1: enable

F0E.4 CMP4IE: CMP4 interrupt enable
0: disable
1: enable

F0E.3 **CMP1IF**: CMP1 interrupt event pending flag
This bit is set by H/W while CMP1 output falling/rising is happened, write 0 to this bit will clear this flag

F0E.2 **CMP2IF**: CMP2 interrupt event pending flag
This bit is set by H/W while CMP2 output falling is happened, write 0 to this bits will clear this flag

F0E.1 **CMP3IF**: CMP3 interrupt event pending flag
This bit is set by H/W while CMP3 output falling is happened, write 0 to this bit will clear this flag

F0E.0 **CMP4IF**: INT0 interrupt event pending flag
This bit is set by H/W while CMP4 output falling is happened, write 0 to this bit will clear this flag

R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	PWMOE	PWMPSC		TCOE	TM0OE	TM1OE	TM1CKS	INT1EDG
R/W	W	W		W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

R0B.0 **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 TM57PT46/PA46 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 2.1V. 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
LVR2.1	5.5V > V_{DD} > 2.2V
LVR3.0	5.5V > V_{DD} > 3.1V

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.

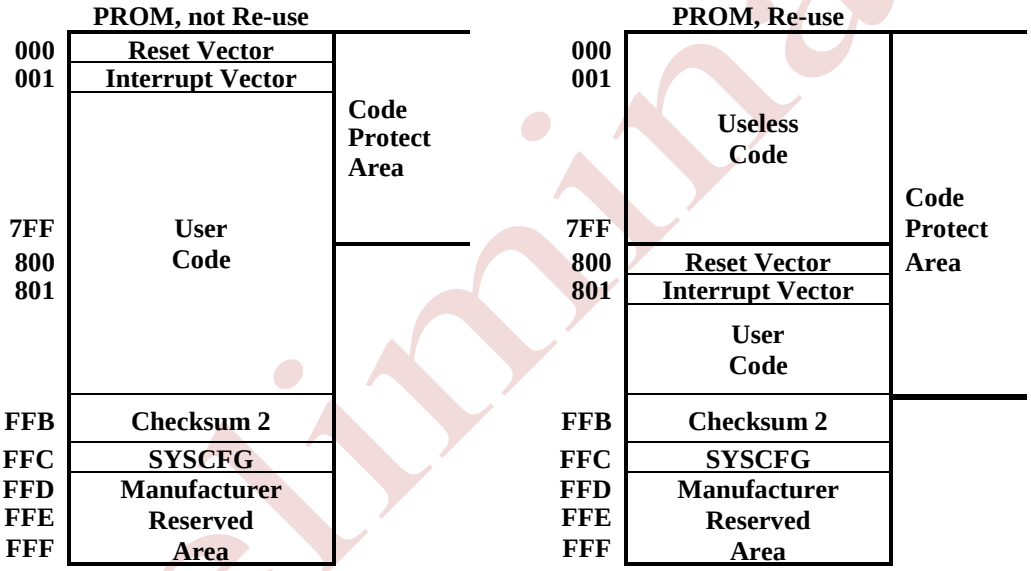
2.2 System Configuration Register (SYSCFG)

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 = 3.0V, always enable
	10	LVR = 2.1V, disable at STOP mode
	11	LVR = 2.1V; always enable
9-8	**N/A**	
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

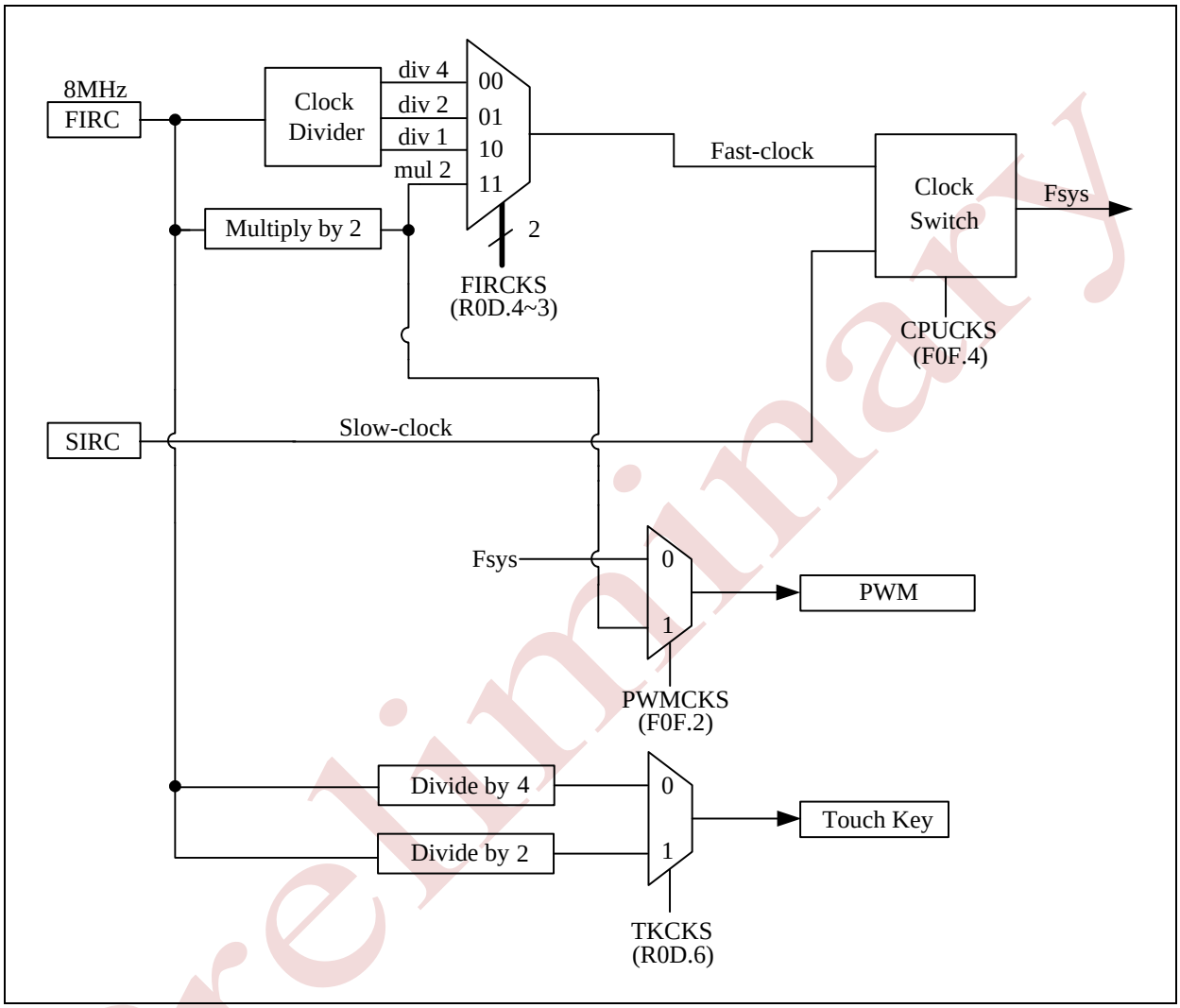
The Power-down mode of TM57PT46/PA46 has only STOP Mode. It is activated by SLEEP instruction. During the Power-down mode, the system clock and peripherals stop to minimize power consumption. The WDT is working or not depends on SYSCFG. The WKT is working or not depends on WKTIE (MF08.3). The Power-down mode can be terminated by Reset, or enabled Interrupts (External pins and WKT) and PB0-4 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 Dual System Clock

TM57PT46/PA46 is designed with dual-clock system. There are two kinds of clock source, 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, TM57PT46/PA46 enters FAST mode at power on. In FAST mode, TM57PT46/PA46 use FIRC as its CPU clock. TM57PT46/PA46 enters FAST mode by setting the CPUCKS (F0F.4=0) when it is in SLOW mode. If user wants to change to SLOW mode, because Slow-clock is always enabled, then switch to Slow-clock as CPU clock (F0F.4=1).

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 (F0F.2).

SLOW Mode:

TM57PT46/PA46 has only one type of Slow Clock, that is SIRC. User can select SIRC as its System clock by setting CPUCKS (F0F.4=1).

IDLE Mode:

The TM57PT46/PA46 does not support IDLE mode because there is no T2 exist in this model.

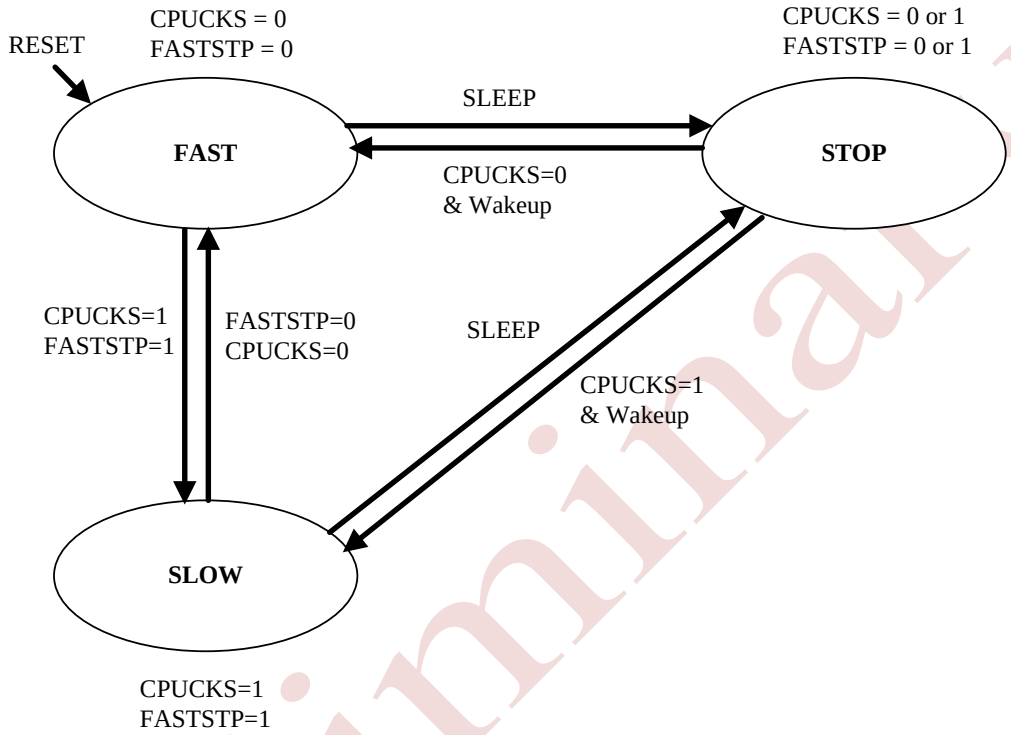
STOP Mode:

If Slow-clock is disabled, all blocks will be turned off and the TM57PT46/PA46 will enter the “STOP Mode” after executing the SLEEP instruction. STOP mode is similar to IDLE mode. The difference is all clock oscillators either Fast-clock or Slow-clock are stopped and no clocks are generated.

2.6 Dual System Clock Modes Transition

TM57PT46/PA46 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 if one of the following conditions is true:
  1. WDT is enabled (SYSCFG.6=1)
  2. Wakeup Timer (WKT) interrupt is enabled
  3. CPUCKS=1 and not in SLEEP mode
- CPUCKS (F0F.4), FASTSTP (F0F.5)

CPU Mode & Clock Functions Table:

Mode	Oscillator	Fsys	Fast-clock	Slow-clock	TM0	TM1	PWM	Wakeup event
FAST	FIRC	Fast-clock	Run	Run	Run	Run	Run	X
SLOW	SIRC	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 is Slow Internal RC (SIRC). The following steps are suggested to be executed by order when FAST mode transits to SLOW mode:

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

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

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

```
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 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 (WDTE=0, WKTIE=0)

(2) Execute SLEEP instruction

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.

STOP mode can be woken up by interrupt (INT0, INT1, INT2), WKT, or PB0-4 pins low level wake up.

◇Example: Switch operating mode to STOP mode

```
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.

(Note that TM57PT46/PA46 doesn't support FXT and SXT oscillation mode.)

	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	BUZEN	ADST	FASTSTP	CPUCKS	PWMCLR	PWMCKS	OPAPD	OPACAL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	1	0

F0F.5 **FASTSTP**: Fast-clock Enable / Disable
0: enable
1: disable

F0F.4 **CPUCKS**: System clock source select
0: Fast-clock
1: Slow-clock

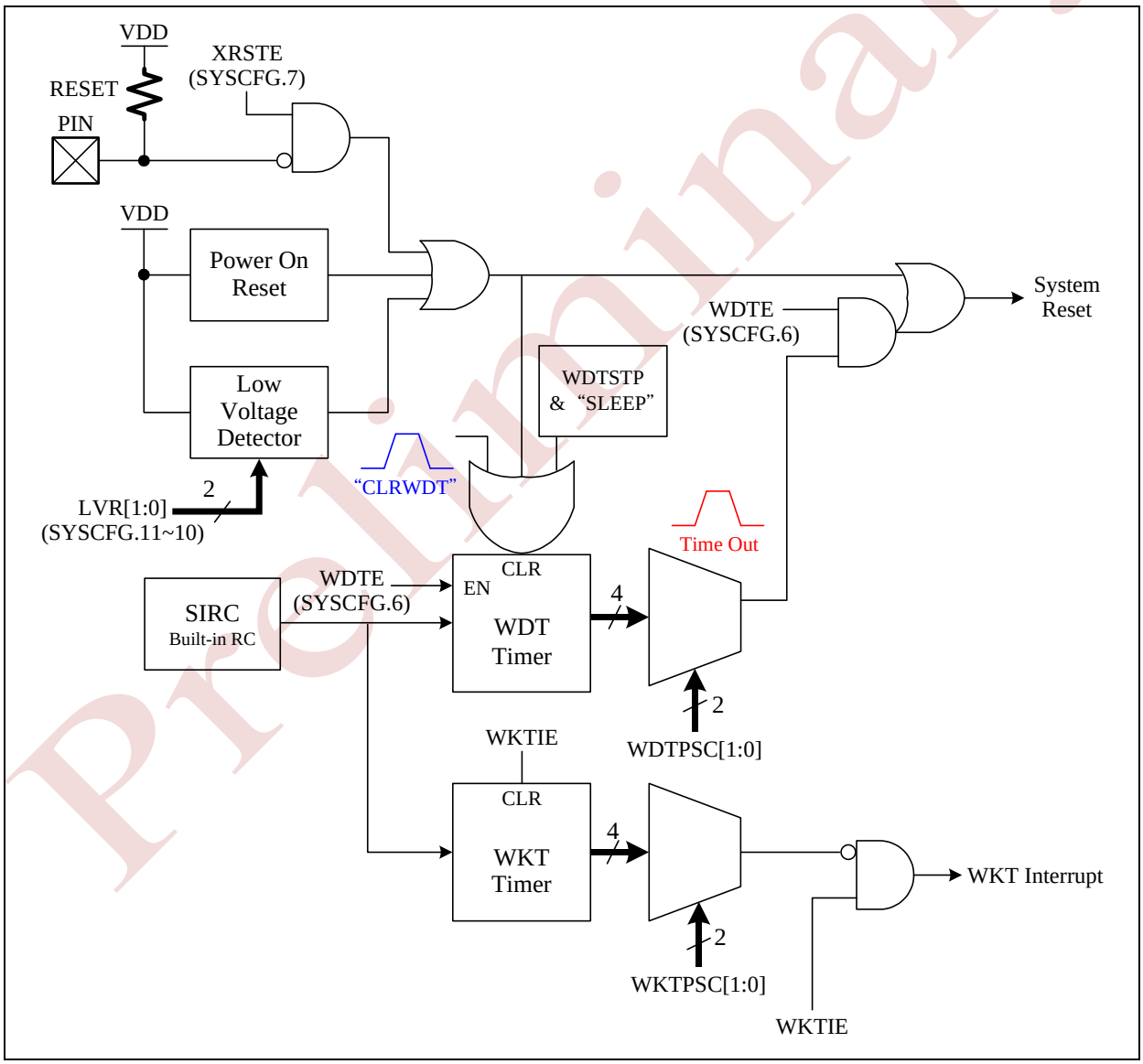
Warning: The CLKCTL (F0F) can't be set directly for CPU modes transition. It may cause the transition fail. Please refer the mentioned steps for transition in this chapter.

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 (R0C.3) 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		WDTSTP	TM1CM	FIRCKS	
R/W	–	W		W	W	W	W	W
Reset	–	0	0	0	0	0	1	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

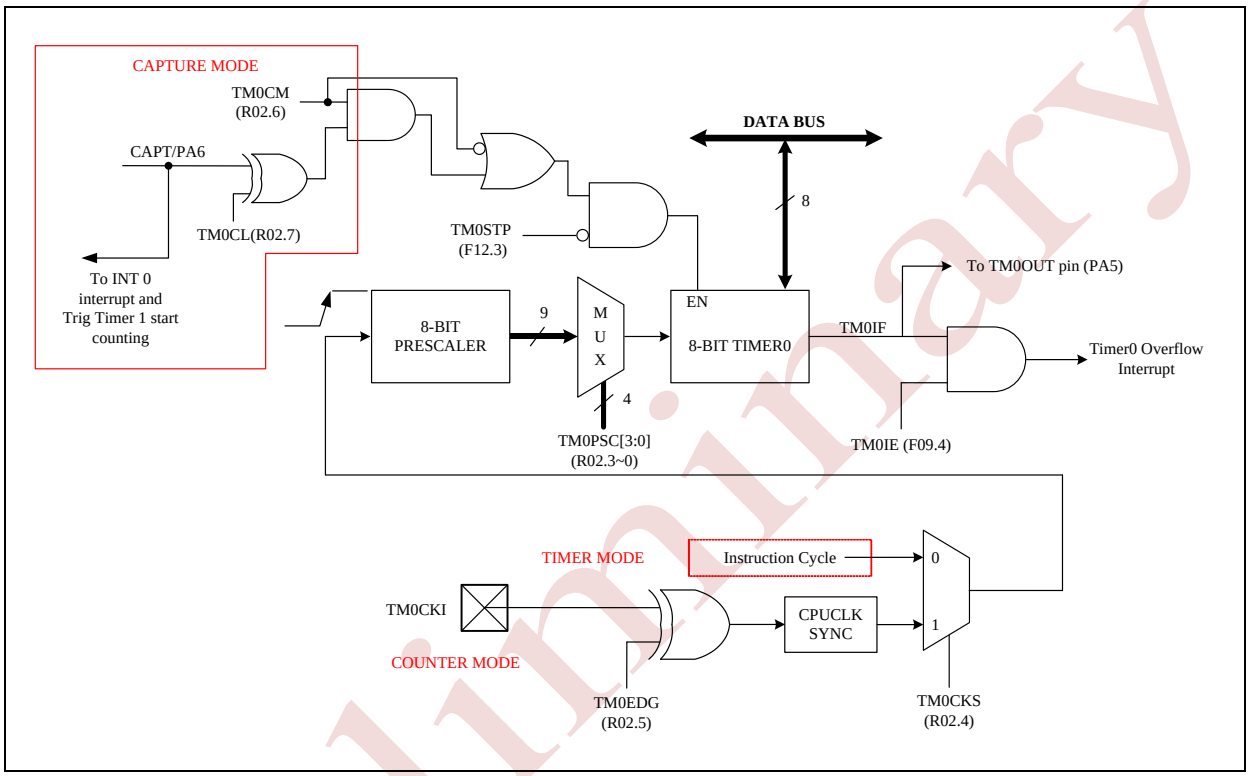
R0C.3 **WDTSTP:** WDT stops counting when in STOP mode

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)

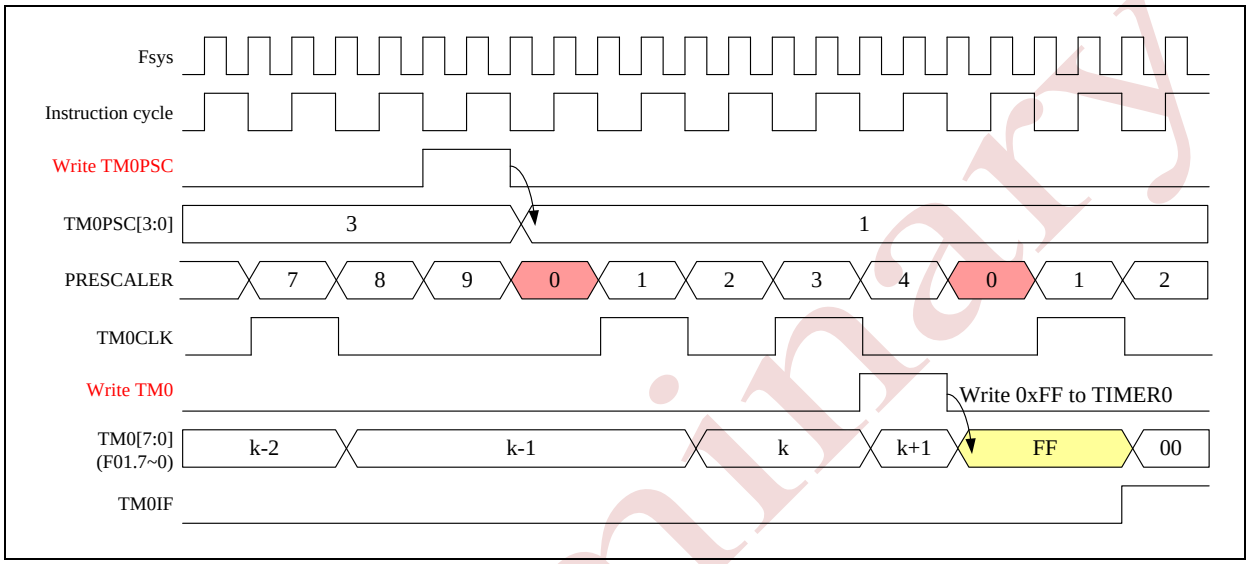
The Timer0 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatically rolls over based on the pre-scaled clock source, which can be the instruction cycle or TM0CKI (PA2) rising/falling input. The Timer0's 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 (F12.3) 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  0000010 1B
        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  00001000B
        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:

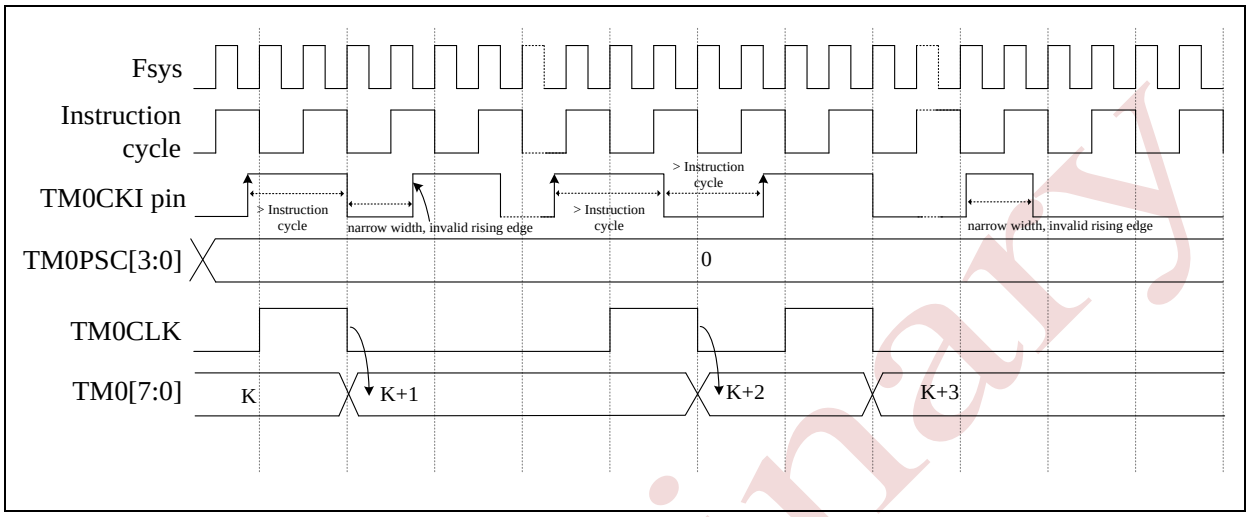
TM0 clock source is Fsys=4 MHz, Instruction cycle=2 MHz, TM0PSC=/32, TM0=156,

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	ADCIE	-	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	ADCIF	CMPIF	TM1IF	TM0IF	WKTIF	XINT2F	XINT1F	XINT0F
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	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	CALINDEX	TM1SET	TM1CLR	TM1STP	TM0STP	C1PPGEN	PPGEN	PPGSTB
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

F12.3 **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.

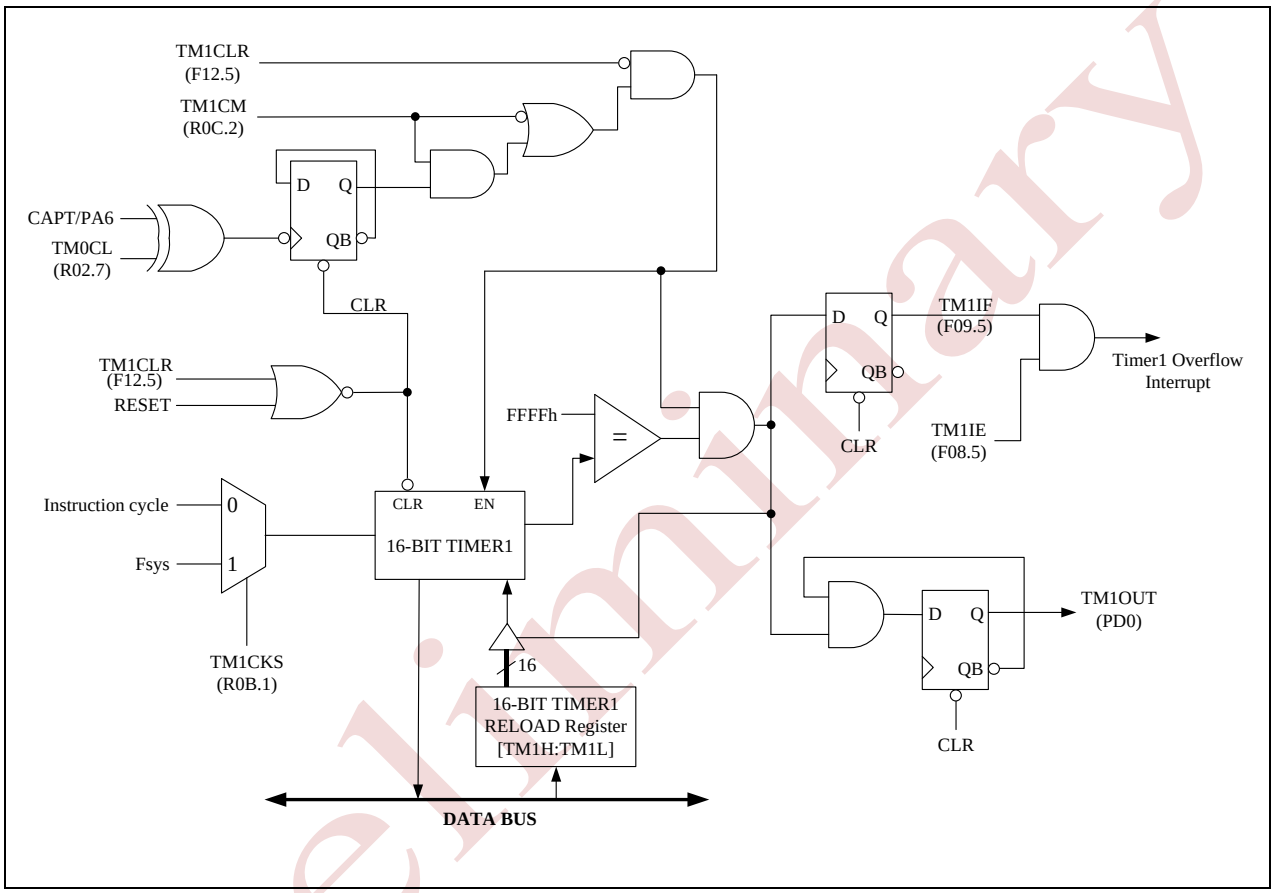
R02.5 **TM0EDG:** TM0CKI (PA2) edge selection for Timer0 prescaler count
0: TM0CKI (PA2) rising edge for Timer0 prescaler count
1: TM0CKI (PA2) falling edge for Timer0 prescaler count

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 TM1CKS (R0B.1). Setting the bit TM1CLR (F12.5) will clear Timer1 and hold Timer1 on 0000h. Setting the TM1STP (F12.4) 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 (F0A, F0B) 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”).
```

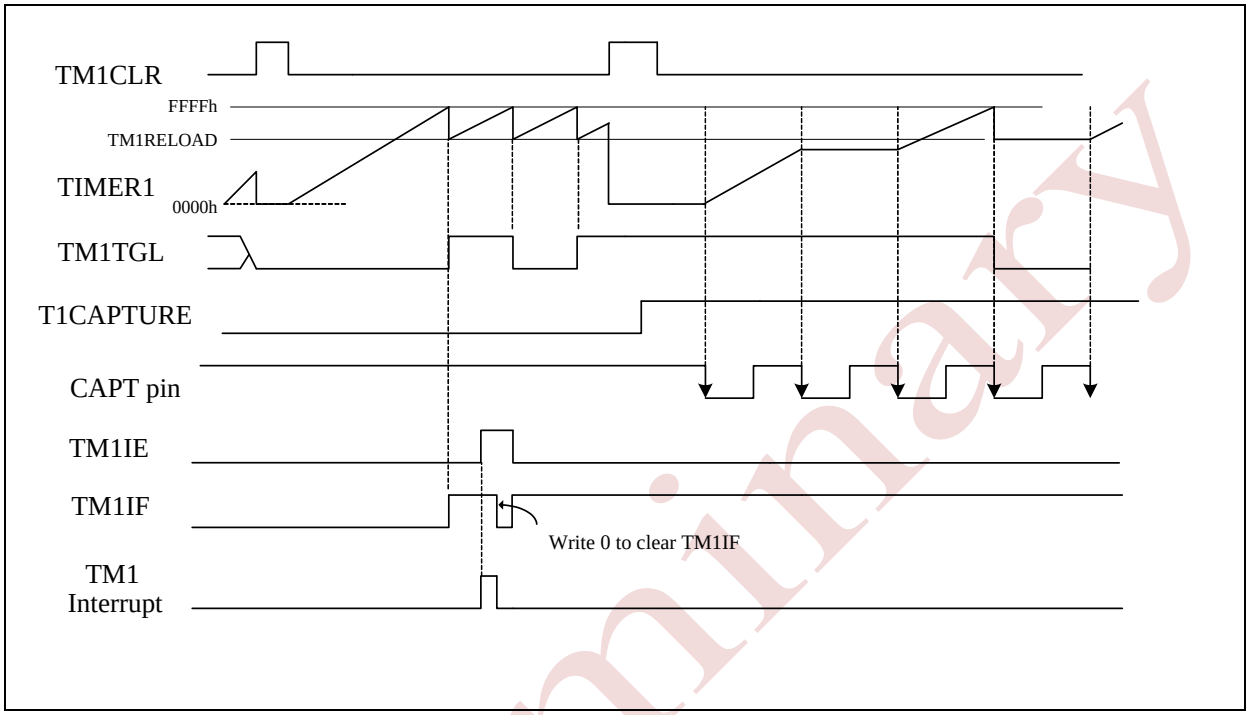
Example:

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.



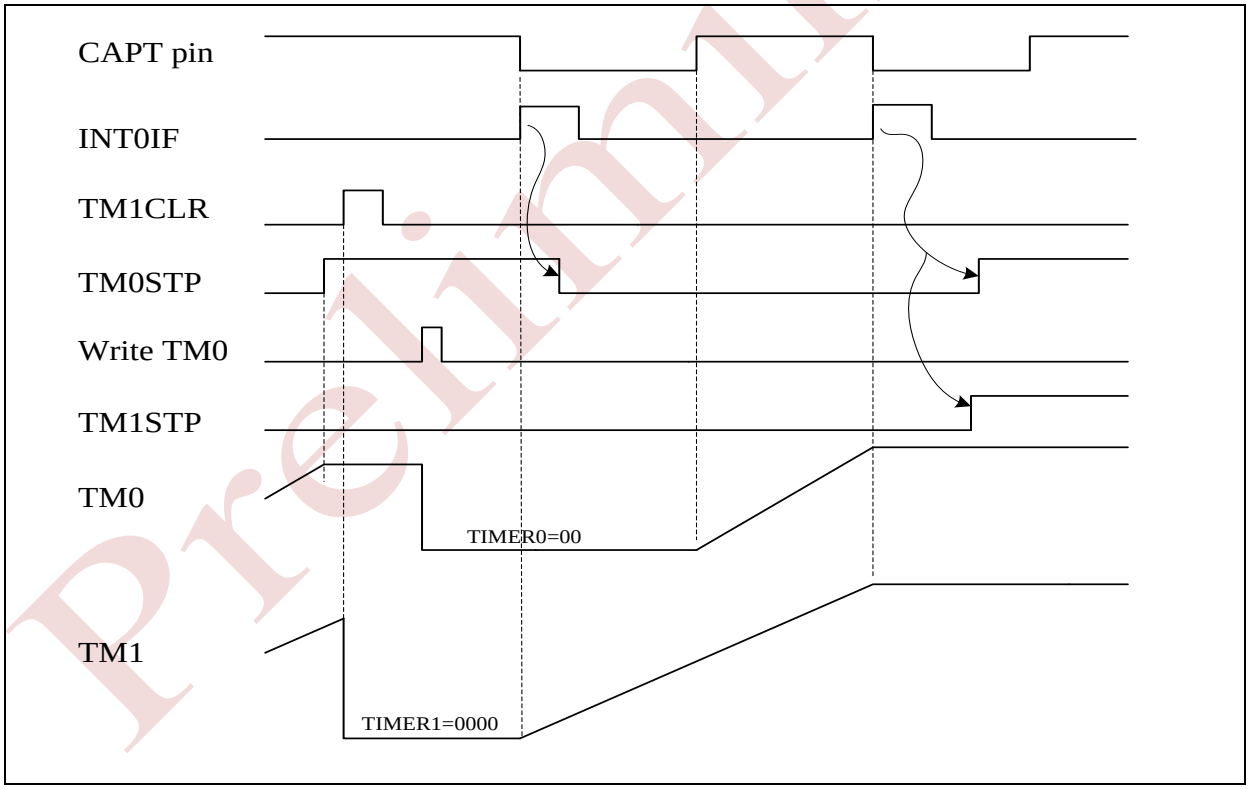
Timer1 works in Capture mode (TM0CL=0, implies CAPT falling edge)

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).

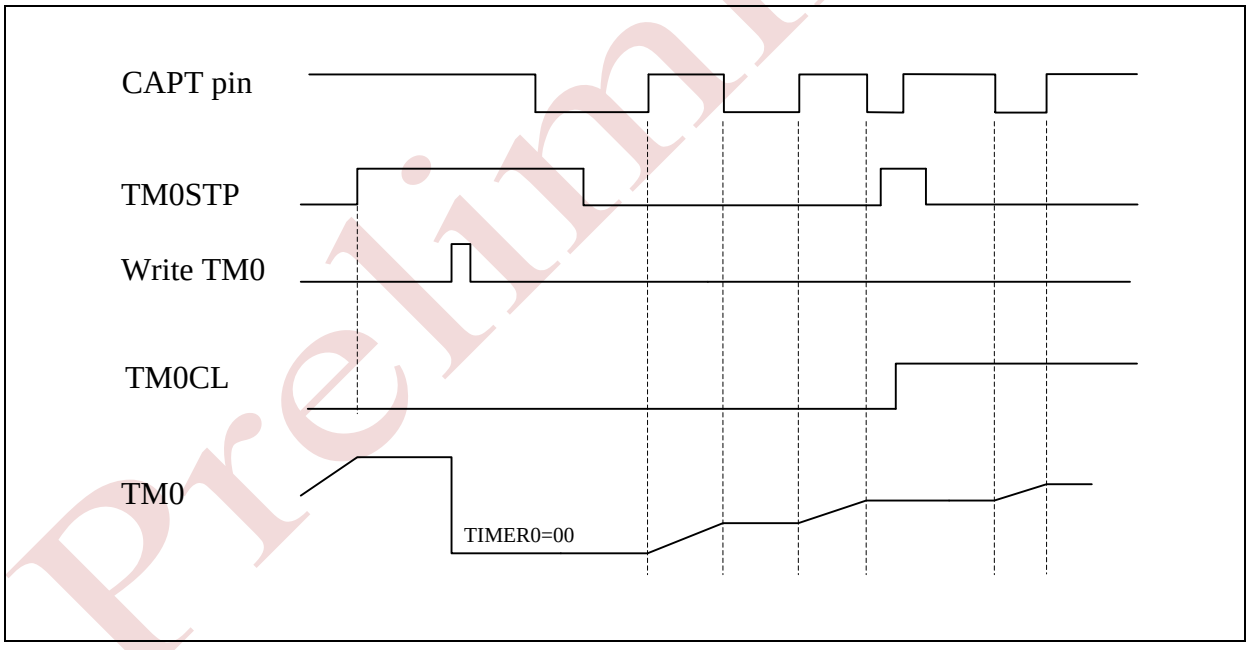


Timer0 and Timer1 are used to measure the signal on CAPT pin.

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	ADCIE	-	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

F08.5 **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	ADCIF	CMPIF	TM1IF	TM0IF	WKTIF	XINT2F	XINT1F	XINT0F
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	-	0	0	0	0	0	0

F09.5 **TM1IF**: T2 interrupt event pending flag
This bit is set by H/W while Timer1 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	CALINDEX	TM1SET	TM1CLR	TM1STP	TM0STP	C1PPGEN	PPGEN	PPGSTB
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0

F12.4 **TM1STP**: Timer1 counter stop
0: Timer1 is counting
1: Timer1 stops counting

F12.5 **TM1CLR:** Timer1 counter clear
0: Release Timer1 clear
1: Clear Timer1 to ‘0000’h and hold

F12.6 **TM1SET:** Timer1 counter set to ‘FFFF’h
0: Release Timer1 set
1: Set Timer1 to ‘FFFF'

R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	PWMOE	PWMPSC		TCOE	TM0OE	TM1OE	TM1CKS	INT1EDG
R/W	W	W		W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

R0B.2 **TM1OE**: Timer1 overflow toggle output to PD0
0: disable output TM1OUT
1: enable output TM1OUT

R0B.1 **TM1CKS:** Timer1 clock source selection
0: Instruction cycle (Fsys/2)
1: System clock (Fsys)

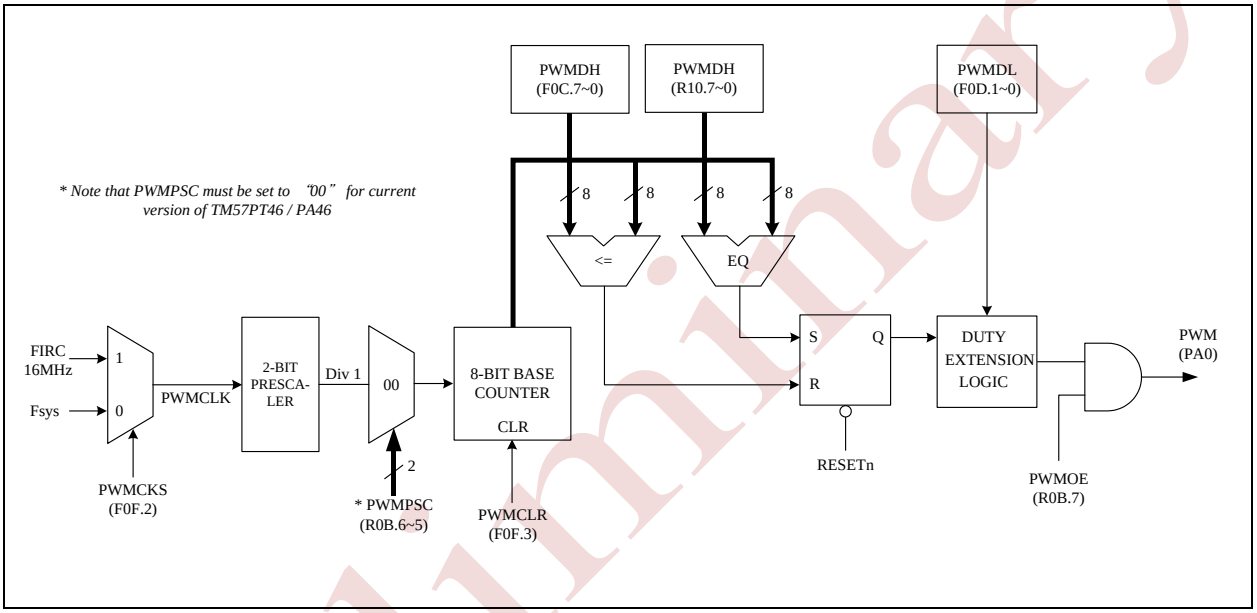
R0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0C	WKTPSC		WDTPSC		WDTSTP	TM1CM	FIRCKS	
R/W	–	W		W	W	W	W	W
Reset	–	0	0	0	0	0	1	0

R0C.2 **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.

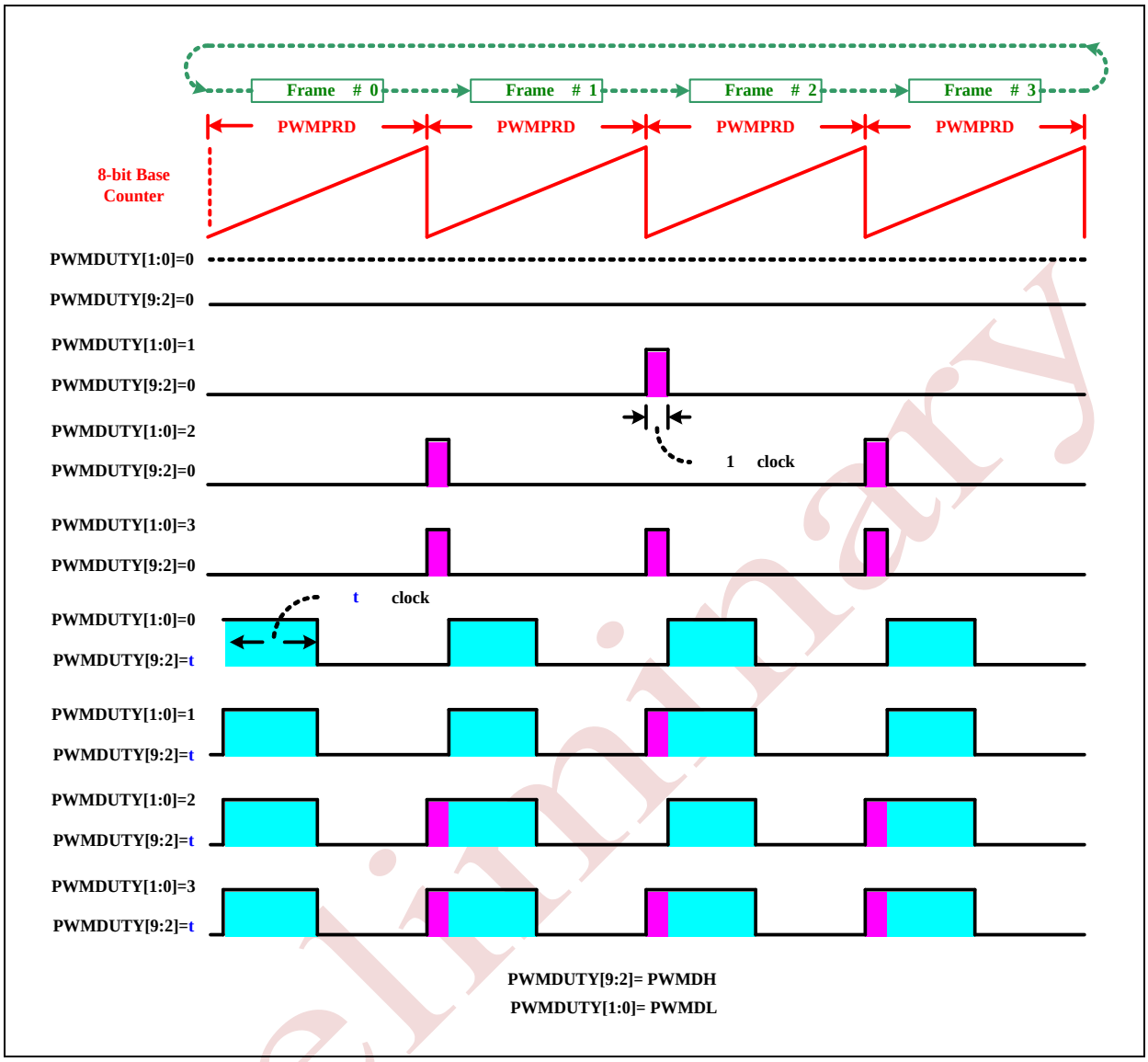
3.4 PWM: (8+2) bits PWM

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 PWMDH (F0C.7~0). When the base counter rolls over, the 2-bit LSB of PWM duty register PWMDL (F0D.1~0) decides whether to set the PWM output signal high immediately or set it high after one clock cycle delay.

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



PWM Block Diagram



PWM Timing Diagram

Example:

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

◇Example:

```
; Setup PWM0 clock prescaler.
        BCF     PWMCKS          ; PWM0 clock source = Fsys
        MOVLW   1 0000000B      ; Fsys=8 MHz, PWMOE=1
        MOVWR   R0B             ;

        MOVLW   80H
        MOVWR   PWMPRD          ; Set PWM period=80H.

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

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

        BCF     PWMCLR          ; Enable PWM0 counting
```

Example:

Fsys=8 MHz, PWMPRD=80H,

PWMDL=00H, PWMDH=20H

PWM output frequency=8 MHz/ (PWMPRD+1) =8 MHz/129=62 KHz.

PWMP output duty=32:129=24.8%.

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

F0C.7~0 **PWMDH**: PWM duty 8-bit MSB

F0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0D	PCH				-		PWMDL	
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 **PWMDL**: PWM duty 2-bit LSB

F0F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCTL	BUZEN	ADST	FASTSTP	CPUCKS	PWMCLR	PWMCKS	OPAPD	OPACAL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	1	0

F0F.3 **PWMCLR**: PWM clear and hold
0: PWM is running
1: PWM is clear and hold

F0F.2 **PWMCKS**: PWM clock selection
0: Fsys as PWM clock source
1: FIRC 16MHz as PWM clock source

R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	PWMOE	PWMPSC		TCOE	TM0OE	TM1OE	TM1CKS	INT1EDG
R/W	W	W		W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

R0B.7 PWMOE: PWM output enable
0: disable PWM output
1: enable PWM output

R0B.6~5 **PWMPSC**: PWM clock source
User code must set these 2 bits to “00” to prevent malfunction of PWM

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

R10.7~0 **PWMPRD**: PWM period data

3.5 PPG (Programmable Pulse Generator)

The PPG function can generate a 9-bit precision wide low pulse which can be stop by 3 comparator (CMP2, CMP3, and CMP4) outputs and can be re-triggered by 1 comparator output (CMP1). The PPG block diagram is below.

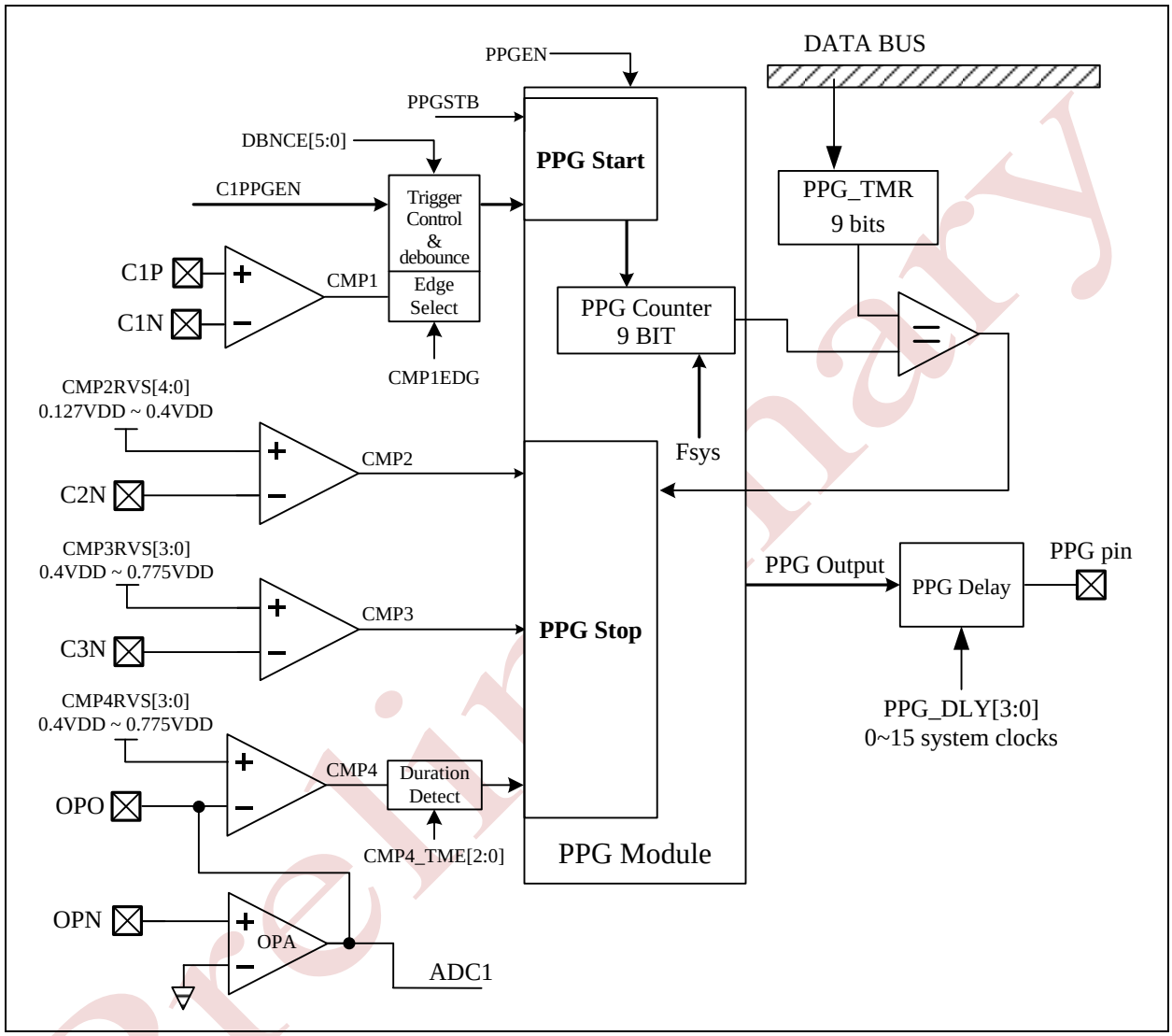


Fig 3.5.1 PPG Bock Diagram

The PPG pulse can be generated in 2 modes, Single Pulse Mode and Continuous Triggered Mode.

3.5.1 Single Pulse Mode

Single Pulse Mode is generated by software setting PPGSTB (F12.0) to 1 and clear to 0 immediately

```
……
……
    BSF     PPGSTB
    BCF     PPGSTB
    …….
    …….
```

After the executing the above 2 instructions, if C1PPGEN (F12.2) is 0, and PPGEN (F12.1) is 1, one PPG pulse will be generated whose pulse width is [PPG_TMR9, PPG_TMR] in the unit of (1/Fsys). Refer to the following diagram

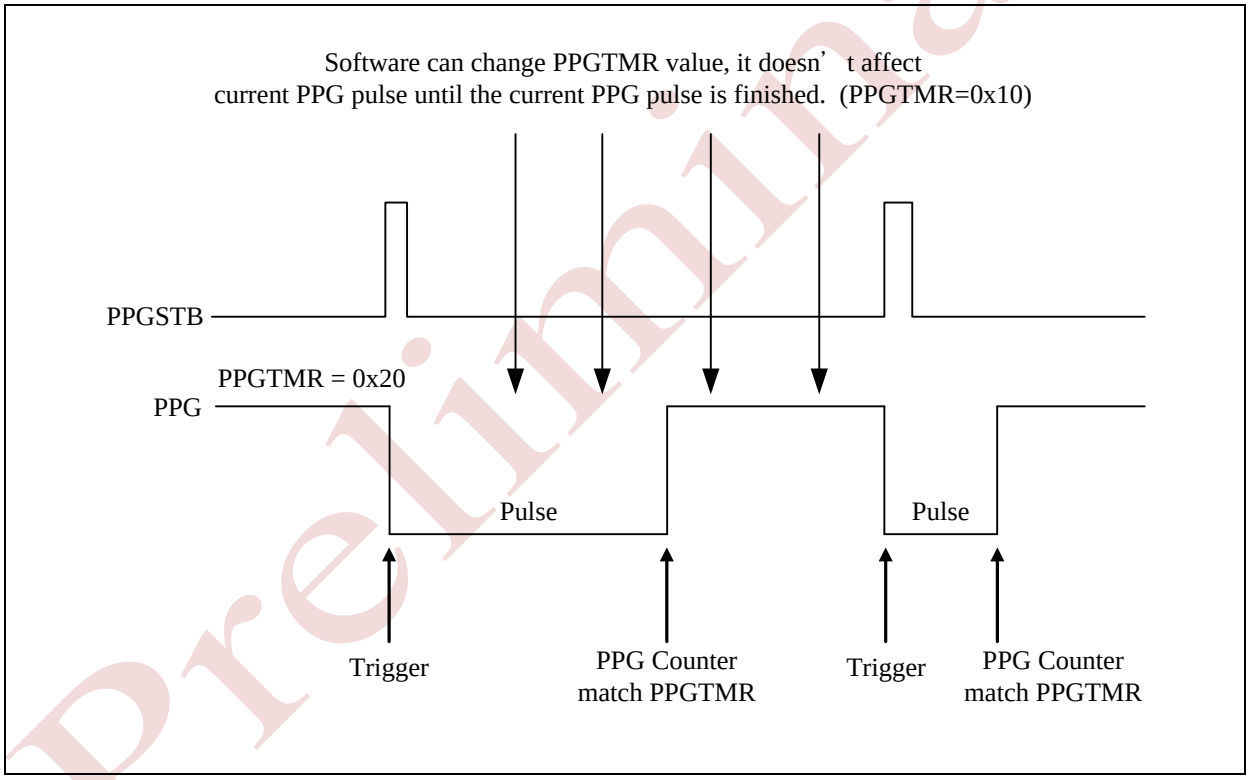


Fig 3.5.2 Single Pulse Mode

PPGTMR can be write any time even the current PPG pulse is active, but the current PPG pulse width will not be changed and it will be effective in the next PPG pulse.

3.5.2 Continuous Triggered Mode (CMP1, edge, debounce, PPG_DLY)

Another method to trigger PPG pulse is Continuous Triggered Mode which cooperate with CMP1. The positive and negative pin of CMP1 comparator is C1P and C1N, user can set CMP1EDG (R1D.6) to choose rising or falling trigger PPG pulse. In the Fig 3.5.3 we set the CMP1EDG to 0 (falling).

PPG1 also build in hardware debounce function. By setting DBNCE (R1D.5~0), if the output of CMP1 has Hi-Lo bounce within the time (DBNCE * 1/Fsys) , the bounce will be cancelled and keep the CMP1 value as previous until the Hi-Lo bounce disappear.

User can set PPGDLY[4:0] (R1A.7~4)to delay the PPG pulse output both Single Pulse Mode and Continuous Triggered Mode.

Note that there are always 6 Fsys clocks delay since CMP1 trigger PPG output even PPGDLY equals to zero in the Continuous Triggered Mode.

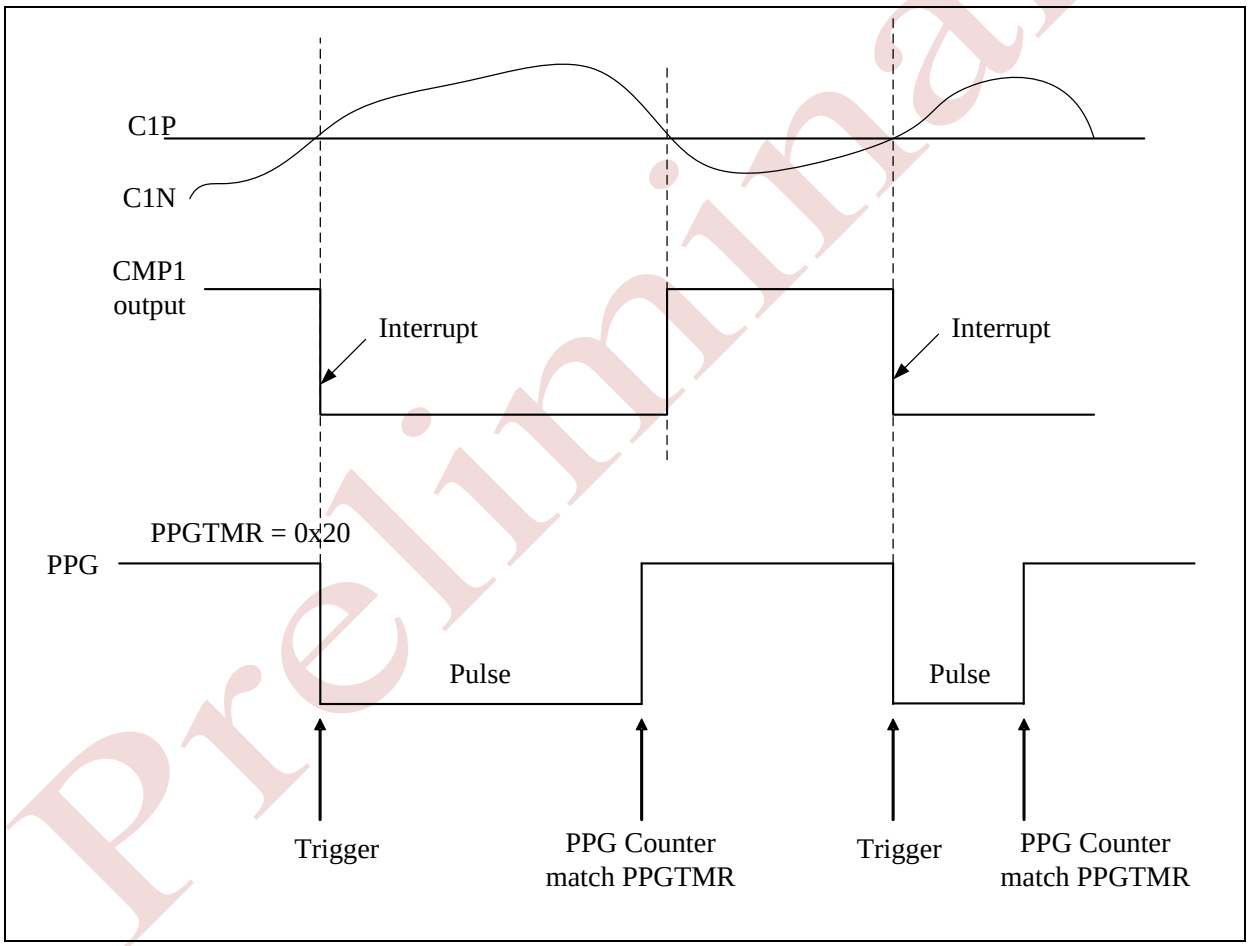


Fig 3.5.3 Continuous Triggered Mode

3.5.3 PPG Stop

PPG pulse will return to 1 when PPG counter reaches PPGTMR. In addition, there are 3 external events that generated by CMP2, CMP3, and CMP4 outputs falling to stop PPG pulse unconditionally and clear the C1PPGEN to '0' then it cannot be retriggered by CMP1 unless C1PPGEN to be set to '1' again.

The negative terminals of CMP2 and CMP3 are connected to outside voltage sources that the application want to observe, the positive terminals of CMP2 and CMP3 are connected to internal voltage divider that can be selected by CMP2RVS[4:0] and CMP3RVS[3:0]. Once the voltages of negative terminals larger than these internal reference voltages, the outputs of comparators will generate a falling edge to stop PPG pulse. Refer to Fig 3.5.4.

The negative terminal of CMP4 is directly connected to the output of OPA. The non-inverted terminal of OPA is connected to ground which means the OPA can only operate as an inverted amplifier with 2 feedback resistors. In other words, the OPA can only used to amplify a signal that all points below zero volt. Refer to Fig 3.5.5.

CMP4_TME is used to set the time duration that CMP4 must be then can generate a falling edge to stop PPG pulse. For example: if CMP4_TME = 011 (8 Fsys clocks, suppose that Fsys=8MHz), that means the CMP4 output low time should be longer than 1us to generate falling edge to stop PPG pulse.

In application, these 4 comparators can turn on hysteresis to prevent noise interfere the output of comparators. The typical value of hysteresis is about 30mV.

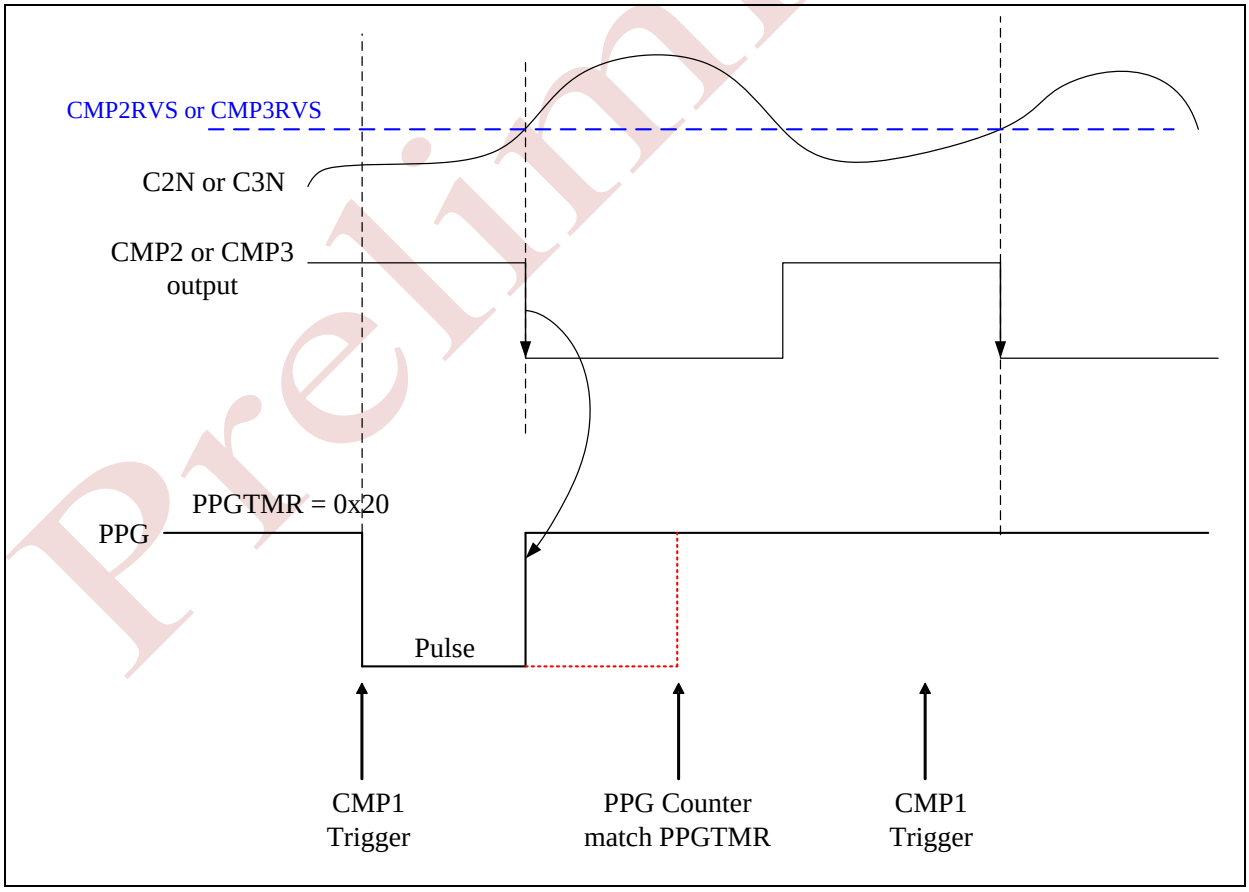


Fig 3.5.4 CMP2 and CMP3 stop PPG pulse

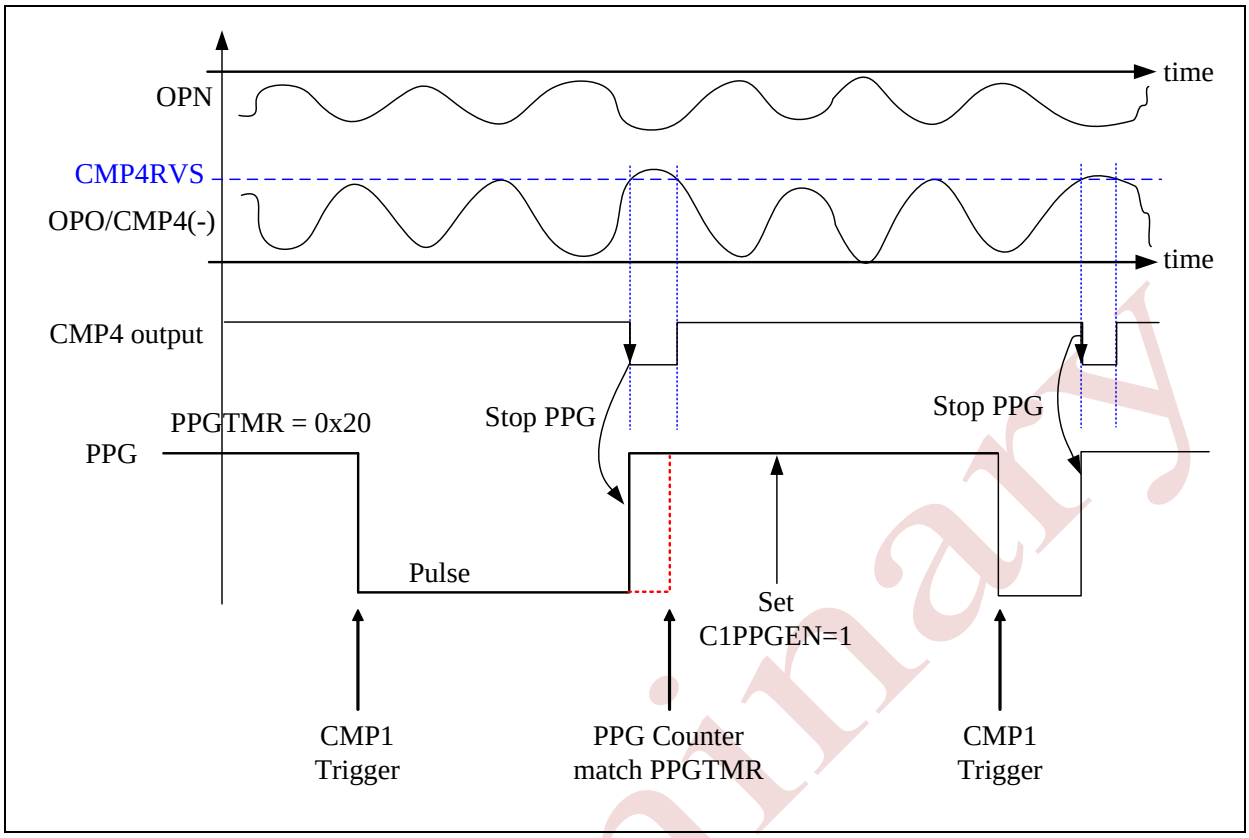
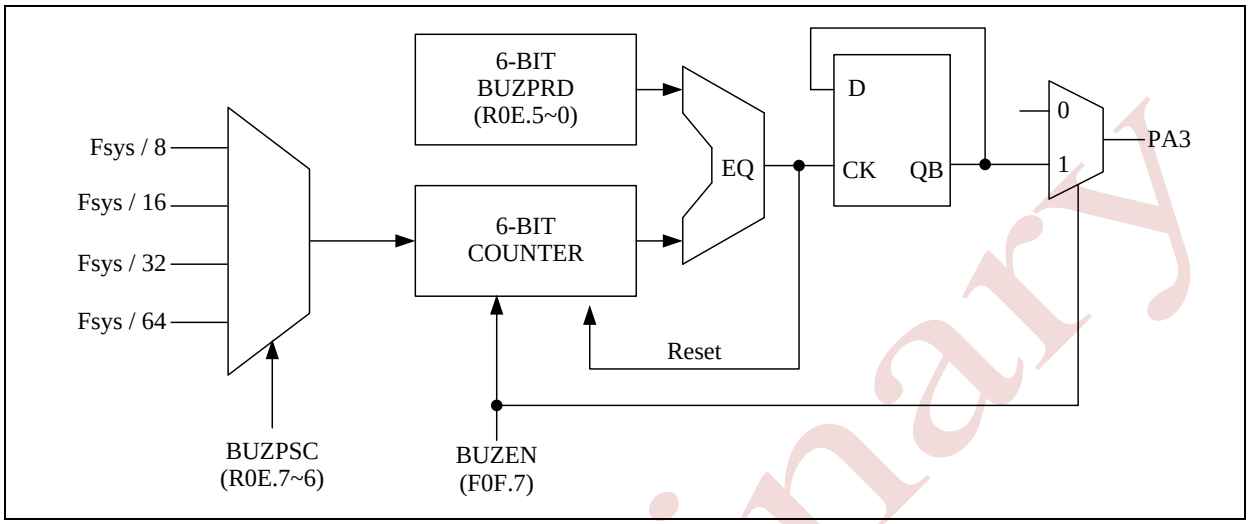


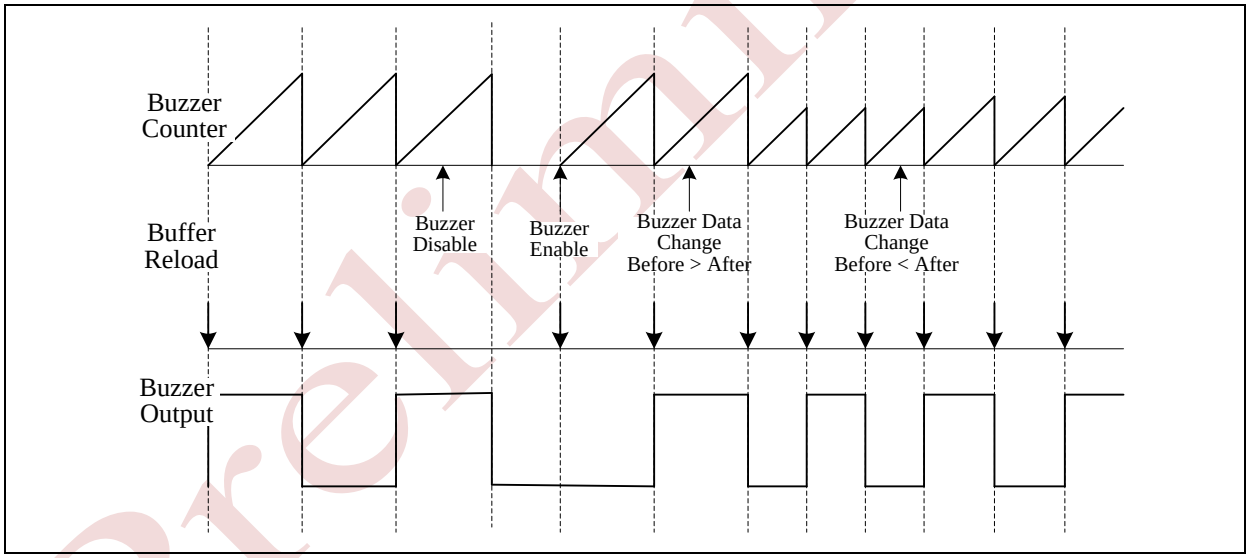
Fig 3.5.5 CMP4 and OPA stop PPG pulse

About the offset trimming of OPA and comparator please refer to corresponding chapters for details.

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]

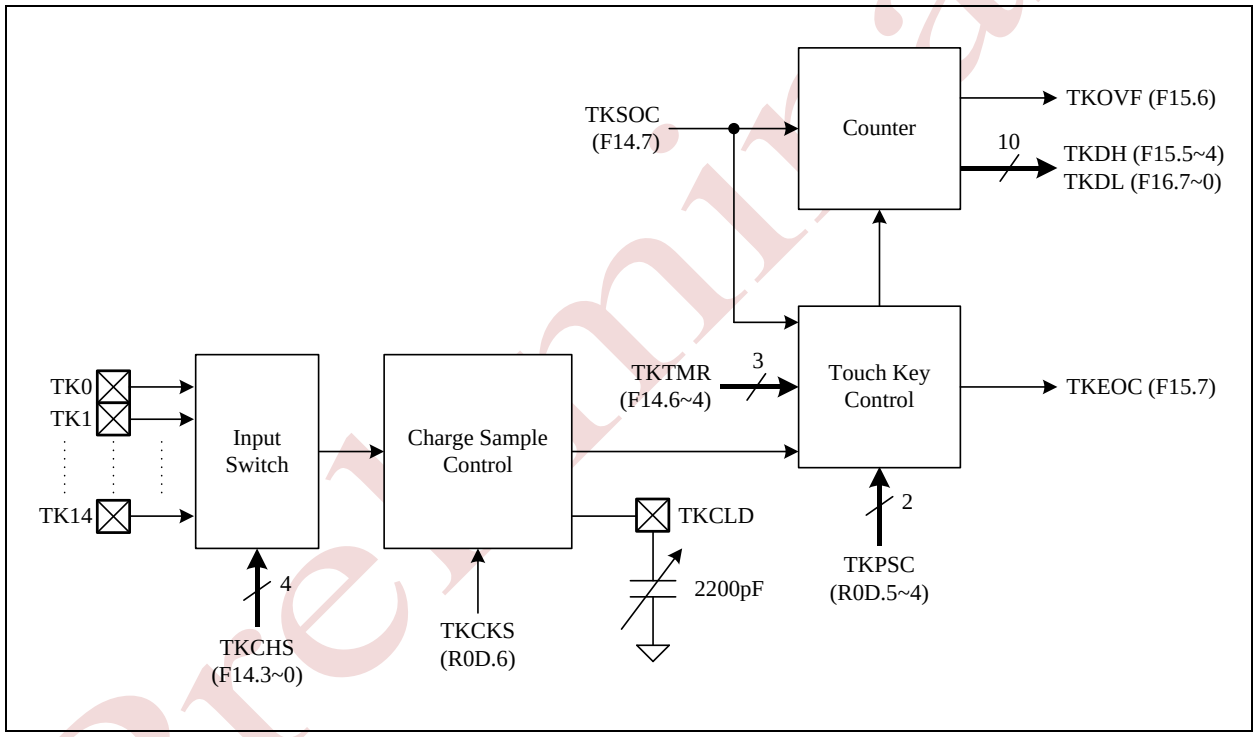
```
MOVLW   10001010B
MOVWF   F0F             ; F0F.7 (BUZEN) =1, enable Buzzer counting and output to PA3
MOVLW   10 001001B      ; R0E.7~6 (BUZPSC) =Fsys/32
MOVWR   R0E             ; R0E.5~0 (BUZPRD) =9
```

3.7 Touch Key

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

Setting the TKSOC (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 (R0D.5~4) to fit the range of TKDATA[9:0]. On the other hand, if TKOVF=0, but TKDATA[9:0] is too small, increase TKTMR or reduce TKPSC to adapting the system board circumstances. The more detailed information, refer to touch key application note.

TK15 is the standard weight that using the on-chip capacitor for software to calibration.



Touch Key Block Diagram

◇Example: Touch key channel=TK10 (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   x0xxxxxxB       ; Set PA6 as TKCLD for connecting capacitor
        MOVWR   PAE
        MOVWR   PAM             ; disable PA6(CLD) digital input

        MOVLW   x1xxxxxxB       ; disable PA6 pull high
        MOVWR   PAPUN

        MOVLW   0100 1010B
        MOVWF   F14             ; TKTMR=4, TKCHS=10 (TK10)

        MOVLW   0 1 00 0000B    ; TKPD=0
        MOVWF   F13             ; TKCKS=1 (4 MHz), TKPSC=00 (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]
```

R0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0D	TKPD	TKCKS	TKPSC		–	ADCKS		
R/W	R/W	R/W	R/W	–	R/W	W	W	W
Reset	1	1	0	0	0	0	0	0

R0D.7 **TKPD**: Touch key power down
0: Touch key running
1: Touch key power down

R0D.6 **TKCKS**: Touch key clock select
0: 2 MHz
1: 4 MHz

R0D.5~4 **TKPSC**: Touch key data prescaler, touch key data
00: divided by 1
01: divided by 2
10: divided by 4
11: divided by 8

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 (PA5)
0001: TK1 (PB0)
0010: TK2 (PB1)
0011: TK3 (PA1)
0100: TK4 (PD6)
0101: TK5 (PA3)
0110: TK6 (PA4)
0111: TK7 (PA2)
1000: TK8 (PA0)
1001: TK9 (PB2)
1010: TK10 (PB3)
1011: TK11 (PB4)
1100: TK12 (PD0)
1101: TK13 (PD5)
1110: TK14 (PD4)
1111: TK15 (Standard weight channel)

F15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCTL2	TKEOC	TKOVF	TKDH		TVOS1	TVOS2	TVOS3	TVOS4
R/W	R	R	R		R/W	R/W	R/W	R/W
Reset	1	0	0		0	0	0	0

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

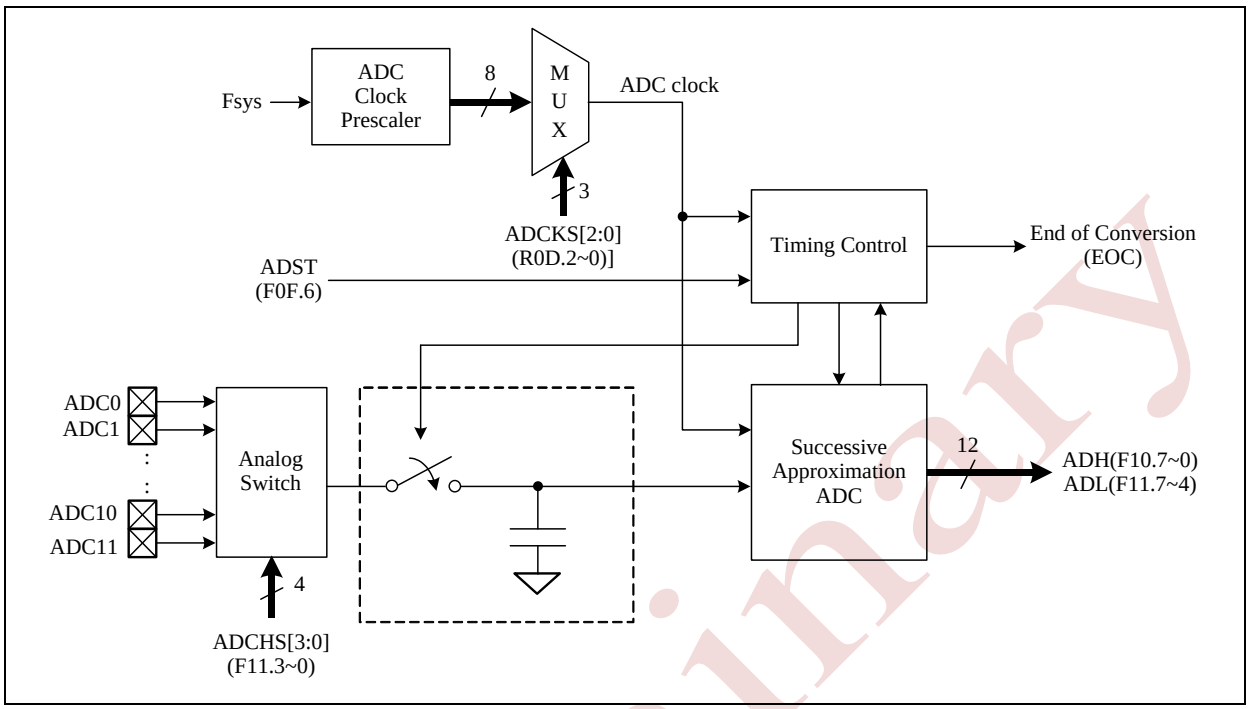
F15.6 **TKOVF**: Touch key counter overflow flag
0: not overflow
1: overflow

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

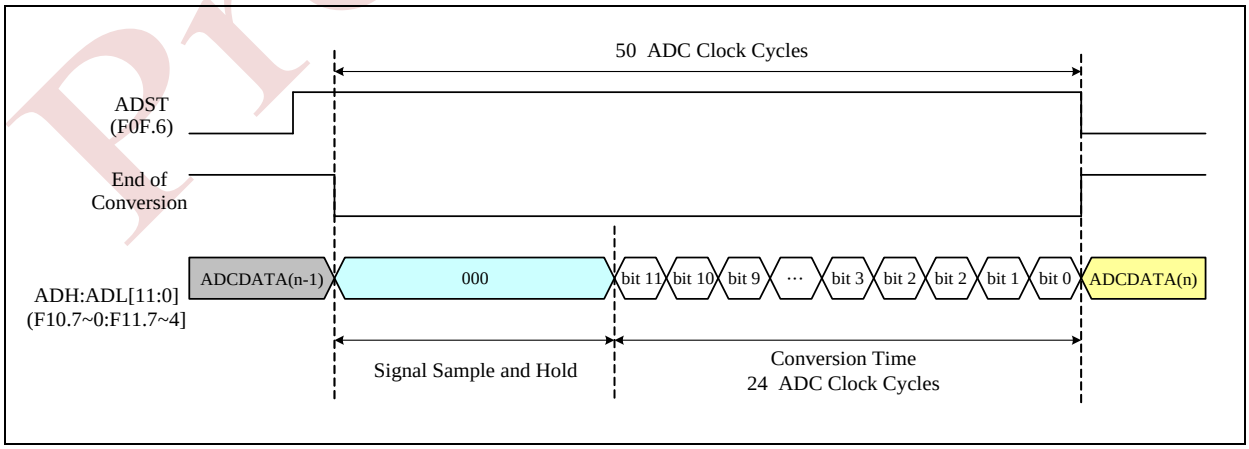
F16.7~0 **TKDL**: Touch key data LSB [7~0]

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=ADC5 (PA2).

◇Example:

```
        MOVLW  xxxxx101B     ; Fsys=8 MHz
        MOVWR  R0D           ; ADC clock prescaler/8


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

        MOVLW  00000101B
        MOVWF  F11           ; ADC channel select ADC5 (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 (PD0)
0001: ADC1 (PD2)
0010: ADC2 (PB3)
0011: ADC3 (PB2)
0100: ADC4 (PA0)
0101: ADC5 (PA2)
0110: ADC6 (PA4)
0111: ADC7 (PA3)
1000: ADC8 (PD6)
1001: ADC9 (PA1)
1010: ADC10 (PA5)
1011: ADC11 (PD7)

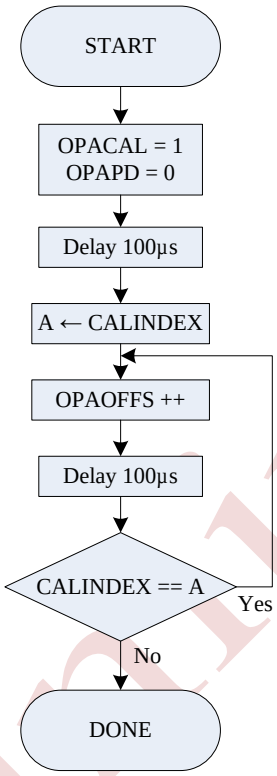
R0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0D	TKPD	TKCKS	TKPSC		–	ADCKS		
R/W	R/W	R/W	R/W	–	R/W	W	W	W
Reset	1	1	0	0	0	0	0	0

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/Comparators offset voltage trimming procedures

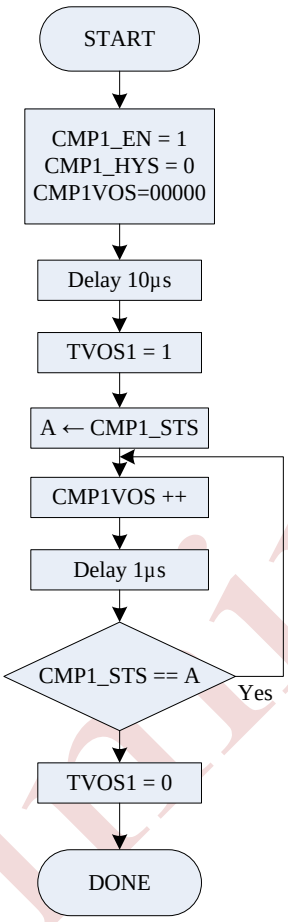
OPA must trim the offset voltage before use. The following flowchart states how to trim the offset voltage of the OPA. Note that the 100us delay time is required when OPA switch from power down mode to power on mode as well as every time the OPAOFFS is changed.



OPA trimming flowchart

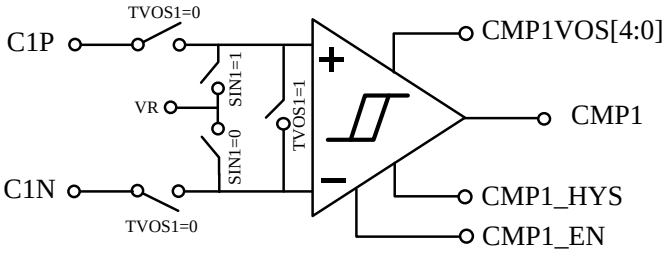
In general the trimming procedure needs to be performed once after power on.

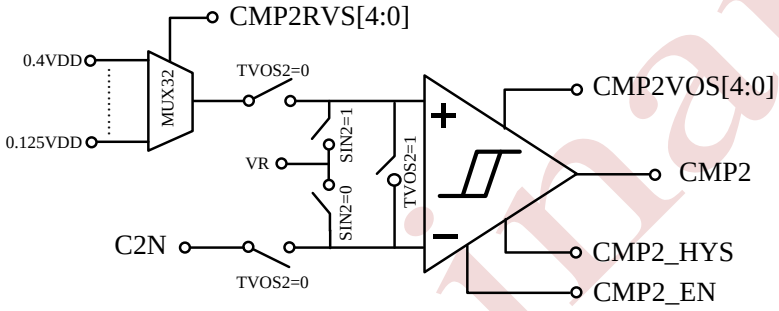
The comparators CMP1, CMP2, CMP3, and CMP4 must perform the trimming procedures before use. The following flowchart uses the name of CMP1, but the corresponding control registers of CMP2, CMP3, and CMP4 can be easily adapted.

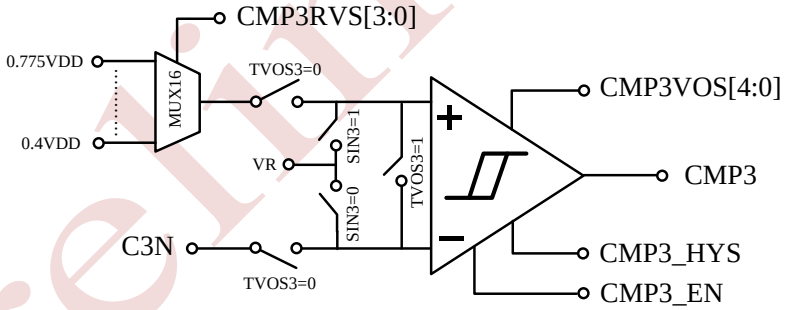


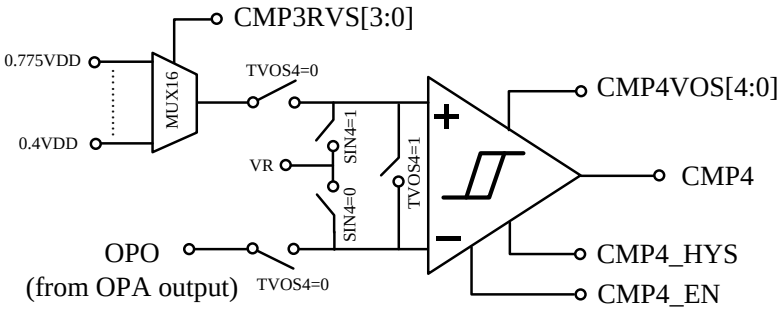
Comparator trimming flowchart

The block diagram of CMP1, CMP2, CMP3, and CMP4 is below:



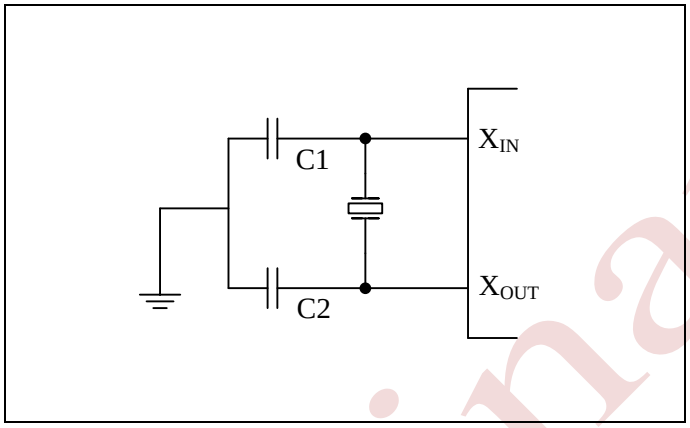




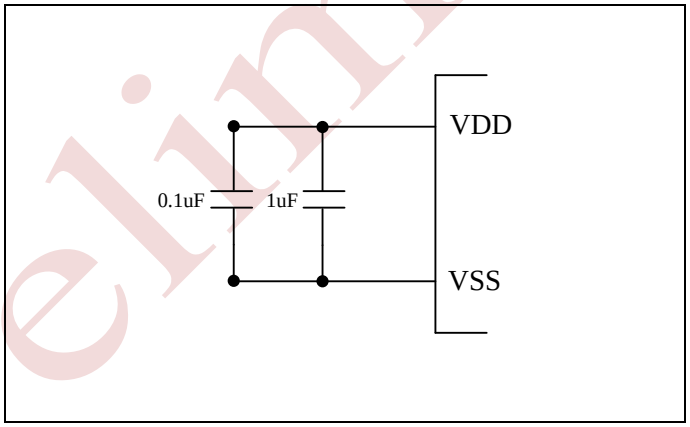


3.10 System Clock Oscillator

System clock can be operated in two different oscillation modes. Two oscillation modes are FIRC and SIRC, respectively. 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)



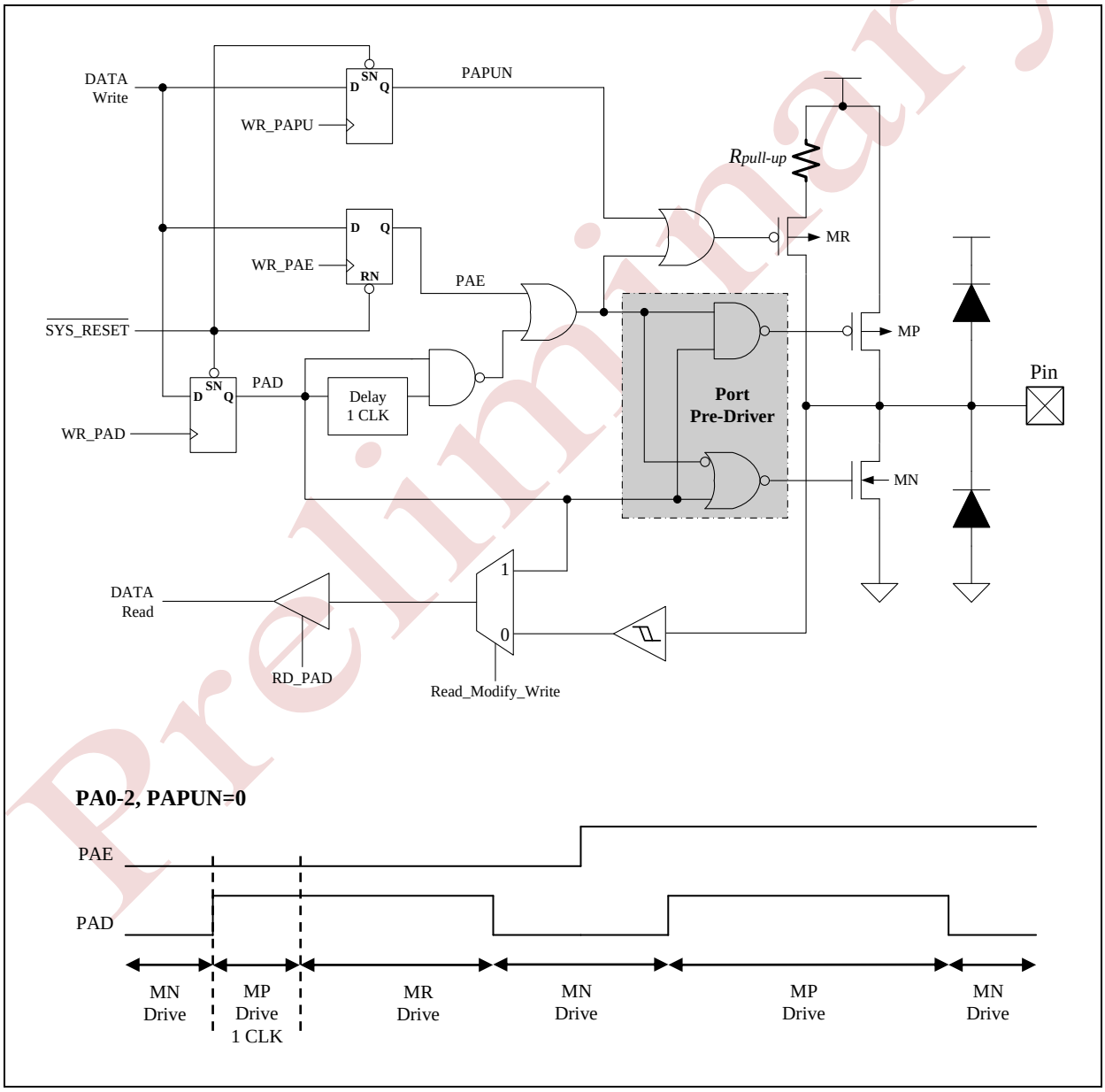
Fast Internal RC Mode

Note that TM57PT46/PA46 without XIN and XOUT pins.

4 I/O Port

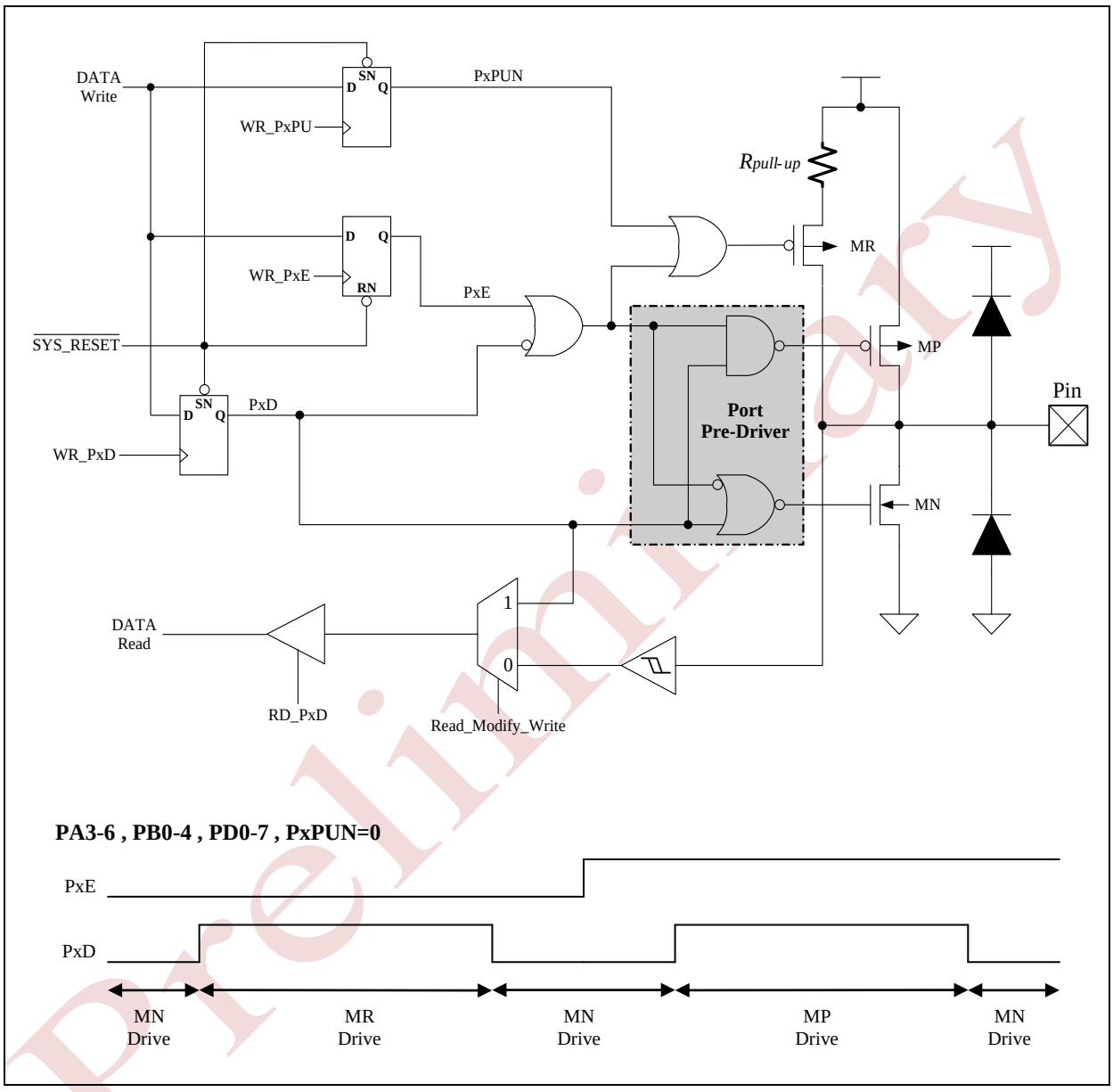
4.1 PA0-2

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, PB0-4, 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.

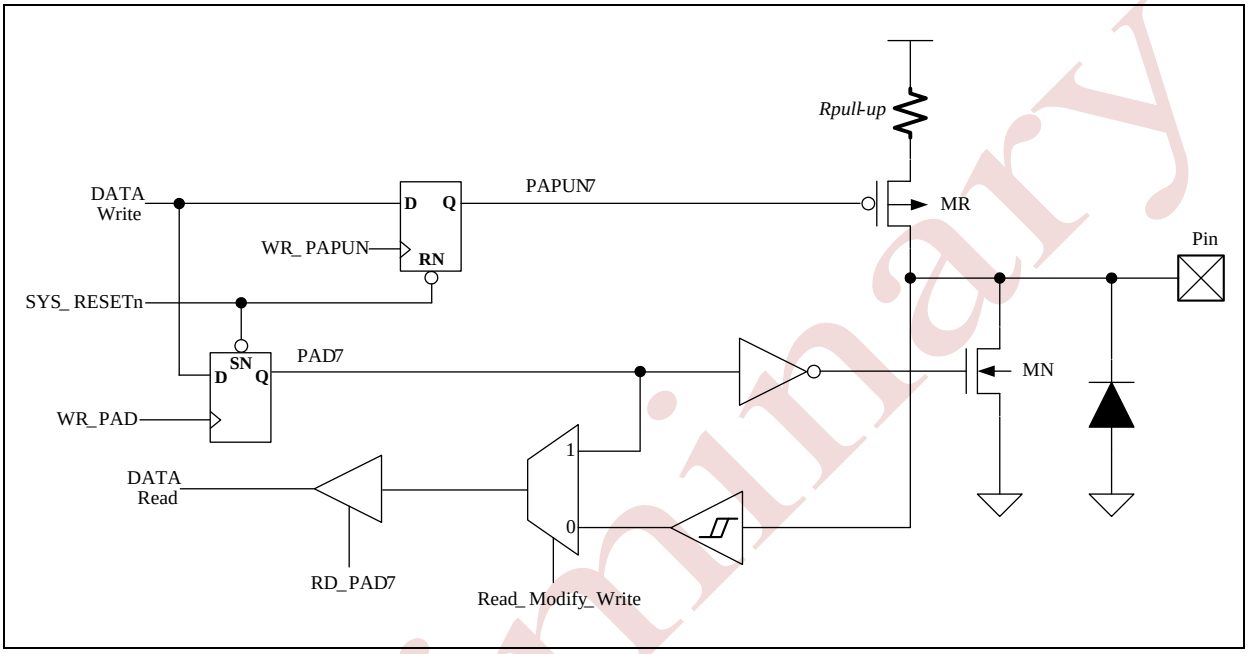
```
BCF    PAD,0
BCF    PBD,1
BCF    PDD,2  ; Set PA0, PB1 and PD2 to be "0"

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

4.3 PA7

PA7 can be only used in Schmitt-trigger input mode. The pull-up resistor is controlled by PAPUN.7 bit and the default value is enabled (i.e. PAPUN.7=0) after system reset.

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	R	R	R/W				
Reset	0	0	0	1	1	1	1	1

F06.4~0 **PBD:** PB4~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**: PA6~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

R06.4~0 **PBE**: PB4~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.

R11	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

R11.7~0 **PAM**: PA7~PA0 pin mode
0 : the pins disable I/O digital input
1: the pins enable I/O digital input

R12	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

R12.4~0 **PBM**: PB4~PB0 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
PDM	PDM							
R/W	W							
Reset	1	1	1	1	1	1	1	1

R13.7~0 PDM: PD7~PD0 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
PBWKEN	–	–	–	PBWKEN				
R/W	–	–	–	W				
Reset	–	–	–	0	0	0	0	0

R14.4~0 **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	FF	Port A output data register
(F06) PBD			**Function related to: Port B**	
PBD	06.4~0	R	-	Port B pin or “data register” state
		W	1F	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**
ADCIE	08.7	R/W	0	ADC interrupt enable 0: disable 1: enable
N/A	08.6	R/W	0	N/A
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
(F09) INTIF				**Function related to: Interrupt Flag**
ADCIF	09.7	R	-	ADC interrupt event pending flag, set by H/W while ADC complete
		W	0	0: clear this flag 1: no action
CMPIF	09.6	R	-	Comparators interrupt event pending flag, set by H/W while at least one of CMP1, CMP2, CMP3, and CMP4 output trigger PPG pulse to stop
		W	0	No action. This bit is just the combinational logic of (CMP1IF or CMP2IF or CMP3IF or CMP4IF)
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

Name	Address	R/W	Rst	Description
(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) PWMDH			**Function related to: PWM**	
PWMDH	0c.7~0	R/W	0	PWM duty 8-bit MSB
(F0D) MF0D			**Function related to: PWM, Program Counter**	
PCH	0d.7~4	R	0	Program Counter high byte, i.e. PC11~PC8
N/A	0d.3~2	R/W	0	Not used.
PWMDL	0d.1~0	R/W	0	PWM duty 2-bit LSB
(F0E) PWM1DH			**Function related to: PWM1**	
CMP1IE	0e.7	R/W	0	CMP1 falling/rising interrupt enable 0: disable 1: enable
CMP2IE	0e.6	R/W	0	CMP2 falling interrupt enable 0: disable 1: enable
CMP3IE	0e.5	R/W	0	CMP3 falling interrupt enable 0:disable 1: enable
CMP4IE	0e.4	R/W	0	CMP4 falling and remains low for CMP4_TME duration interrupt enable 0: disable 1: enable
CMP1IF	0e.3	R/W	0	CMP1 interrupt event pending flag, set by H/W when CMP1 output falling/rising edge. Write ‘0’ to clear.
CMP2IF	0e.2	R/W	0	CMP2 interrupt event pending flag, set by H/W when CMP2 output falling edge. Write ‘0’ to clear.
CMP3IF	0e.1	R/W	0	CMP3 interrupt event pending flag, set by H/W when CMP3 output falling edge. Write ‘0’ to clear.
CMP4IF	0e.0	R/W	0	CMP4 interrupt event pending flag, set by H/W when CMP4 output falling edge and remains low level for CMP4_TME duration. Write ‘0’ to clear.
(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
FASTSTP	0f.5	R/W	0	Fast-clock Enable/Disable 0: Enable 1: Disable
CPUCKS	0f.4	R/W	0	System clock (Fsys) selection 0: Fast-clock 1: Slow-clock
PWMCLR	0f.3	R/W	1	PWM counter clear 0: Release 1: Clear and hold
PWMCKS	0f.2	R/W	0	PWM0 clock source 0: Fsys 1: FIRC 16M
OPAPD	0f.1	R/W	1	OPA power down 0: OPA is enabled 1: OPA is disabled
OPACAL	0f.0	R/W	0	OPA offset calibration enable 0: Disable calibration, OPA in normal operating mode 1: Enable calibration, OPA in offset voltage calibration mode
(F10) ADCDH			**Function related to: ADC**	
ADCDH	10.7~0	R	-	ADC output data MSB[11:4]

Name	Address	R/W	Rst	Description
(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**	
CALINDEX	12.7	R	-	OPA calibration index, observing this bit toggle when OPA in calibration mode.
TM1SET	12.6	R/W	0	Timer1 counter set 0: Release 1: Set to FFFFh and hold
TM1CLR	12.5	R/W	0	Timer1 counter clear 0: Release 1: Clear to 0000H and hold
TM1STP	12.4	R/W	0	Timer1 counter stop 0: Release 1: Stop counting
TM0STP	12.3	R/W	0	Timer0 counter stop 0: Release 1: Stop counting
C1PPGEN	12.2	R/W	0	Enable CMP1 output falling/rising to trigger PPG pulse. 0: disabled 1: enable
PPGEN	12.1	R/W	0	PPG output enable. 0: disabled 1: enabled. PPGEN is cleared to '0' when PPGSTB from high to low which finish one PPG Single Pulse Mode.
PPGSTB	12.0	R/W	0	Writing a '1' and a '0' to generate single PPG pulse with PPG_TMR width and the C1TGCNT will be cleared and start counting CMP1 toggle times.
(F13) MF13			**Function related to: OPA, Touch Key**	
CMP1_STS	13.7	R	-	CMP1 output status
CMP2_STS	13.6	R	-	CMP2 output status
CMP3_STS	13.5	R	-	CMP3 output status
CMP4_STS	13.4	R	-	CMP4 output status
OPAOFFS	13.3~0	R/W	0	OPA offset tuning bits. Totally 16 steps.
(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	100	Touch key conversion time. 000=shortest, 111=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: standard weight channel 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]
TVOS1	15.3	R/W	0	CMP1 0: Normal mode 1: Trim offset mode
TVOS2	15.2	R/W	0	CMP2 0: Normal mode 1: Trim offset mode
TVOS3	15.1	R/W	0	CMP3 0: Normal mode 1: Trim offset mode
TVOS4	15.0	R/W	0	CMP4 0: Normal mode 1: Trim offset mode

Name	Address	R/W	Rst	Description
(F16) TKDL			**Function related to: Touch Key**	
TKDL	16.7~0	R	-	Touch key counter low byte TKDATA[7:0]
(F17) DPL			**Function related to: Table read**	
DPL	17.7~0	R/W	00	Low byte of DPTR.DPTR will be increased automatically when TABRH is executed.
(F18) MF18			**Function related to: Table read, PPG**	
DPH	18.3~0	R/W	0	Higher 4 bits of DPTR
PPG_TMR9	18.7	R/W	0	The bit 8 (9^{th} bit, MSb) of PPG_TMR
(F19) PPG_TMR			**Function related to: PPG**	
PPG_TMR	19.7~0	R/W	00	The lower 8 bits of PPG_TMR. PPG_TMR ranges from 0~511 in decimal.
(F1A) C1TGCNT			**Function related to: PPG**	
C1TGCNT	1a.7~0	R	00	CMP1 toggle counter. Set PPGSTB to 1 to clear this counter and start counting. The value will be held when reach 255.
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 TM57PA46, and all registers related to Touch Key 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	06.4~0	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	09.4~0	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: PWM,TM0, TM1, INT1, TCOUT**	
PWMOE	0b.7	W	0	0: PA0 as its function 1: enable PWM output to PA0 pin
PWMPSC	0b.6~5	W	0	PWM clock source is divided by User code must set these 2 bits to '00' to prevent malfunction of PWM.
TCOE	0b.4	W	0	Instruction cycle (Fsys/2) output to PD6 0: disable 1: enable
TM0OE	0b.3	W	0	Timer0 overflow toggle output to PA5 0: disable 1: enable
TM1OE	0b.2	W	0	Timer1 overflow toggle output to PD0 0: disable 1: enable
TM1CKS	0b.1	W	0	Timer1 clock source 0: Fsys/2 (instruction cycle) 1: Fsys
INT1EDG	0b.0	W	0	0: INT1 pin falling edge to trigger interrupt event 1: INT1 pin rising edge to trigger interrupt event

<table>
<tr><th>Name</th><th>Address</th><th>R/W</th><th>Rst</th><th>Description</th></tr>
<tr><td colspan="5">(R0C) MR0C 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><td colspan="2">VDD=5V</td><td>VDD=3V</td></tr>
<tr><td>00</td><td>1.1 ms</td><td>1.4 ms</td></tr>
<tr><td>01</td><td>2.3 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>145 ms</td><td>182 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><td colspan="2">VDD=5V</td><td>VDD=3V</td></tr>
<tr><td>00</td><td>145 ms</td><td>180 ms</td></tr>
<tr><td>01</td><td>290 ms</td><td>364 ms</td></tr>
<tr><td>10</td><td>1160 ms</td><td>1456 ms</td></tr>
<tr><td>11</td><td>2320 ms</td><td>2913 ms</td></tr>
</table></td></tr>
<tr><td>WDTSTP</td><td>0c.3</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>TM1CM</td><td>0c.2</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>FIRCKS</td><td>0c.1~0</td><td>W</td><td>10</td><td>FIRC clock selection
00: 2 MHz
01: 4 MHz
10: 8 MHz
11: 16 MHz</td></tr>
<tr><td colspan="5">(R0D) MR0D Function related to: Touch Key/ADC</td></tr>
<tr><td>TKPD</td><td>0d.7</td><td>W</td><td>1</td><td>Touch Key power down 0: power up 1: power down</td></tr>
<tr><td>TKCKS</td><td>0d.6</td><td>W</td><td>1</td><td>Touch key PWM clock select, "TK-clock" is
0: 2 MHz 1: 4 MHz</td></tr>
<tr><td>TKPSC</td><td>0d.5~4</td><td>W</td><td>00</td><td>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</td></tr>
<tr><td>N/A</td><td>0d.3</td><td></td><td>0</td><td>Not used</td></tr>
<tr><td>ADCKS</td><td>0d.2~0</td><td>W</td><td>000</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>
<tr><td colspan="5">(R0E) BUZCTL Function related to: Buzzer</td></tr>
<tr><td>BUZPSC</td><td>0e.7~6</td><td>W</td><td>00</td><td>Buzzer clock frequency selection
00: Fsys/8
01: Fsys/16
10: Fsys/32
11: Fsys/64</td></tr>
<tr><td>BUZPRD</td><td>0e.5~0</td><td>W</td><td>0</td><td>Buzzer Period</td></tr>
</table>

Name	Address	R/W	Rst	Description
(R0F) Reserved			**Tenx reserved**	
Reserved	0f.7~0	-	-	Tenx reserved register. Users do not write it.
(R10) PWMPRD			**Function related to: PWM**	
PWMPRD	10.7~0	W	FF	PWM Period
(R11) PAM			**Function related to: Port A**	
PAM	11.7~0	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
(R12) PBM			**Function related to: Port B**	
PBM	12.4~0	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
(R13) PDM			**Function related to Port D**	
PDM	13.7~0	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
(R14) PBWKEN			**Function related to: Wake up**	
NOISERJ	14.7	W	0	Enhance noise rejection 0: disable 1: enable
PBWKEN	14.4~0	W	00	PB4~PB0 low level wakeup 0: disable 1: enable
(R15)CMPCTL			**Function related to: Comparator**	
CMP1_EN	15.7	W	0	CMP1 : 0: disable 1: enable
CMP2_EN	15.6	W	0	CMP2 : 0: disable 1: enable
CMP3_EN	15.5	W	0	CMP3 : 0: disable 1: enable
CMP4_EN	15.4	W	0	CMP4 : 0: disable 1: enable
CMP1_HYS	15.3	W	0	CMP1 hysteresis 0: OFF 1: ON
CMP2_HYS	15.2	W	0	CMP2 hysteresis 0: OFF 1: ON
CMP3_HYS	15.1	W	0	CMP3 hysteresis 0: OFF 1: ON
CMP4_HYS	15.0	W	0	CMP4 hysteresis 0: OFF 1: ON

Name	Address	R/W	Rst	Description
(R16)CMP1OT			**Function related to: CMP1 Offset Trim**	
SIN1	16.7	W	0	When TVOS1=1, VR input to trim offset voltage from 0: negative terminal 1: positive terminal
CMP1VOS	16.4~0	W	00	CMP1 offset voltage adjustment 00000~11111
(R17)CMP2OT			**Function related to: CMP2 Offset Trim**	
SIN2	17.7	W	0	When TVOS2=1, VR input to trim offset voltage from 0: negative terminal 1: positive terminal
CMP2VOS	17.4~0	W	00	CMP2 offset voltage adjustment 00000~11111
(R18)CMP3OT			**Function related to: CMP3 Offset Trim**	
SIN3	18.7	W	0	When TVOS3=1, VR input to trim offset voltage from 0: negative terminal 1: positive terminal
CMP3VOS	18.4~0	W	00	CMP3 offset voltage adjustment 00000~11111
(R19)CMP4OT			**Function related to: CMP4 Offset Trim**	
SIN4	19.7	W	0	When TVOS4=1, VR input to trim offset voltage from 0: negative terminal 1: positive terminal
CMP4VOS	19.4~0	W	00	CMP4 offset voltage adjustment 00000~11111
(R1A)MR1A			**Function related to: PPG, CMP3**	
PPG_DLY	1a.7~4	W	0	PPG output delay selection 0000: direct output without delay 0001: 1 Fsys clock ….. 1111: 15 Fsys clocks
CMP3RVS	1a.3~0	W	0	CMP3 non-inverted terminal reference voltage selection. Ranges from 0.4VDD to 0.775VDD
(R1B)CMP2RVS			**Function related to: CMP2**	
CMP2RVS	1b.4~0	W	00	CMP2 non-inverted terminal reference voltage selection. Ranges from 0.125VDD to 0.4VDD
(R1C)CMP4CTL			**Function related to: CMP4**	
CMP4_TME	1c.7~5	W	000	CMP4 low level duration select. When CMP4 outputs low remains N system clock (Fsys), CMP4 interrupt is generated if interrupt is enabled. 000: 1 Fsys 001: 2 Fsys 010: 4 Fsys 011: 8 Fsys 100: 16 Fsys 101: 32 Fsys 110: 64 Fsys 111: 128 Fsys
CMP4RVS	1c.3~0	W	0	CMP4 non-inverted terminal reference voltage selection. Ranges from 0.4VDD to 0.775VDD
(R1D)CMP1CTL			**Function related to: CMP1**	
CMP1EDG	1d.6	W	0	CMP1 output trigger edge 0: falling 1: rising
DBNCE	1d.5~0	W	00	CMP1 debounce setting. Setting the number of Tsys (1/Fsys) time to check the bounce of CMP1 output. If the CMP1 output changes twice within the DBNCE time, the bounce will be ignored and keep the previous CMP1 output value.

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

IORLW — Inclusive OR Literal with W

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

IORWF — Inclusive OR W with "f"

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

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

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

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

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

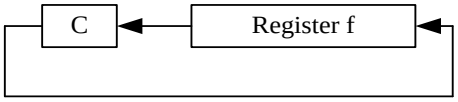
NOP	No Operation
Syntax	NOP
Operands	-
Operation	No Operation
Status Affected	-
OP-Code	00 0000 0000 0000
Description	No Operation
Cycle	1
Example	NOP -

RET	Return from Subroutine
Syntax	RET
Operands	-
Operation	PC ← TOS
Status Affected	-
OP-Code	00 0000 0100 0000
Description	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.
Cycle	2
Example	RET A : PC = TOS

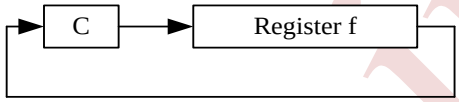
RETI	Return from Interrupt
Syntax	RETI
Operands	-
Operation	PC ← TOS, GIE ← 1
Status Affected	-
OP-Code	00 0000 0110 0000
Description	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in to the PC. Interrupts are enabled. This is a two-cycle instruction.
Cycle	2
Example	RETI A :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

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

RLF — Rotate Left "f" through Carry

Syntax	RLF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	
Status Affected	C
OP-Code	00 1101 dfff ffff
Description	The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.
Cycle	1
Example	RLF REG1, 0 B : REG1 = 1110 0110, C = 0 A : REG1 = 1110 0110 W = 1100 1100, C = 1

RRF — Rotate Right "f" through Carry

Syntax	RRF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	
Status Affected	C
OP-Code	00 1100 dfff ffff
Description	The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.
Cycle	1
Example	RRF REG1, 0 B : REG1 = 1110 0110, C = 0 A : REG1 = 1110 0110 W = 0111 0011, C = 0

SLEEP — Go into Power-down mode, Clock oscillation stops

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

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℃)

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	℃
Storage temperature	-65 to +150	

2. DC Characteristics (T_A=25℃, V_{DD}=1.1V to 5.5V)

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Operating Voltage	V_{DD}	FAST mode, 25°C, Fsys=16 MHz		3.0	–	5.5	V
		FAST mode, 25°C, Fsys=8 MHz		2.3	–	5.5	
		FAST mode, 25°C, Fsys=4 MHz		1.9	–	5.5	
		SLOW mode, 25°C, SIRC		1.5	–	5.5	
Input High Voltage	V_{IH}	All Inputs, 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 Inputs	V_{DD}=5V	–	–	0.2V_{DD}	V
			V_{DD}=3V	–	–	0.2V_{DD}	V
I/O Port Source Current	I_{OH}	All Outputs	V_{DD}=5V, V_{OH}=0.9V_{DD}	4	8	–	mA
			V_{DD}=3V, V_{OH}=0.9V_{DD}	2	4	–	
I/O Port Sink Current	I_{OL}	All Outputs	V_{DD}=5V, V_{OL}=0.1V_{DD}	10	20	–	mA
			V_{DD}=3V, V_{OL}=0.1V_{DD}	5	10	–	
Input Leakage Current (pin high)	I_{ILH}	All Inputs	V_{IN}=V_{DD}	–	–	1	µA
Input Leakage Current (pin low)	I_{ILL}	All Inputs	V_{IN}=0V	–	–	-1	
Supply Current	I_{DD}	FAST mode, LVR enable, WDT enable	V_{DD}=5V, FIRC=8 MHz	–	2.8	–	mA
			V_{DD}=3V, FIRC=8 MHz	–	1.3	–	
		SLOW mode, LVR enable	V_{DD}=5 V, SIRC	–	139	–	µA
			V_{DD}=3 V, SIRC	–	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}=2.2V	–	–	8	
LVR Reference Voltage	V_{LVR}	T_A=25°C		–	2.1	–	V
				–	3.0	–	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	–	

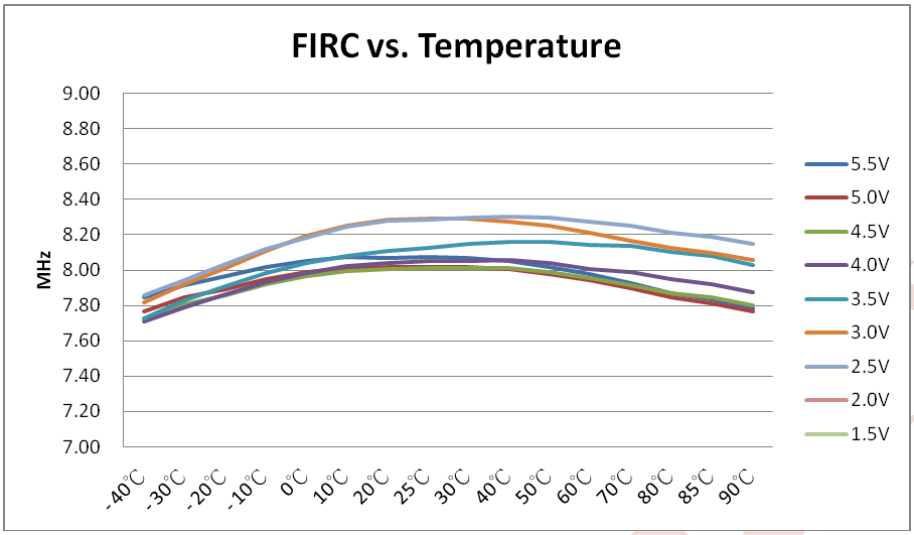
5. OPA Electrical Characteristics (VDD=5V TA = 25°C)

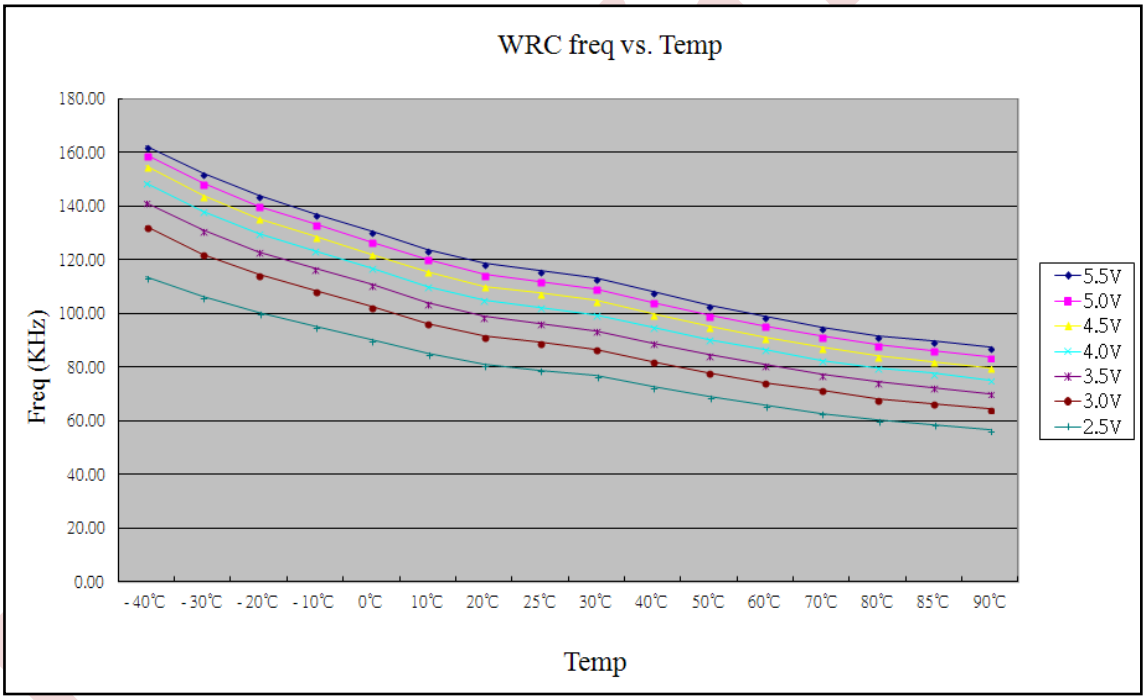
Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VOS1	Input Offset Voltage(Without calibration)				±10	mv
VOS2	Input Offset Voltage(By calibration)				±2	mv
AVOL	Large Signal Voltage Gain		60	80		dB
GBW	Gain Band Width Product	RL=1MΩ CL=100pF	0.6	2.2		MHz
CMRR	Common Mode Rejection Ratio		60	80		dB
PSRR	Power Supply Rejection Ratio		60	80		dB
SR	Slew Rate at Unity Gain	No load	0.6	1.8		V/μs

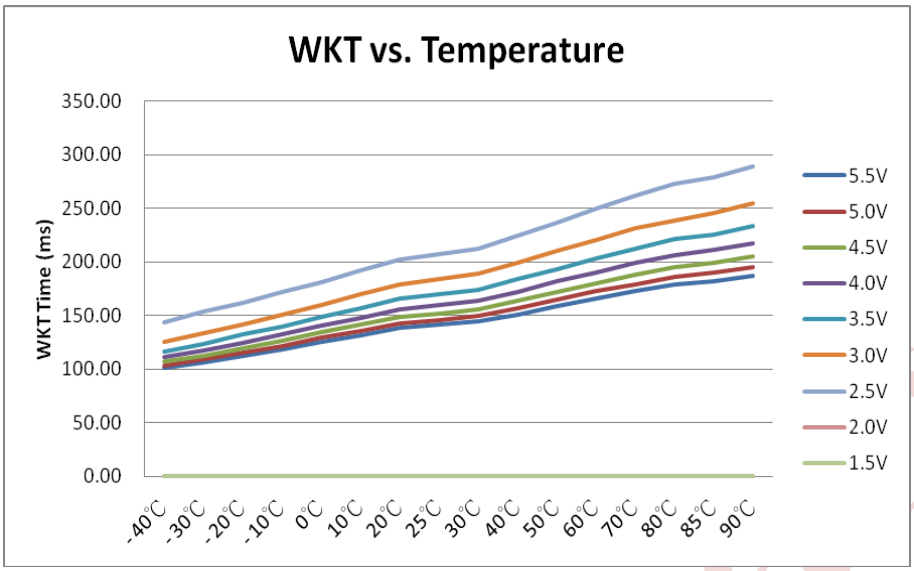
6. COMPARATOR Electrical characteristics (VDD=5V TA=25°C)

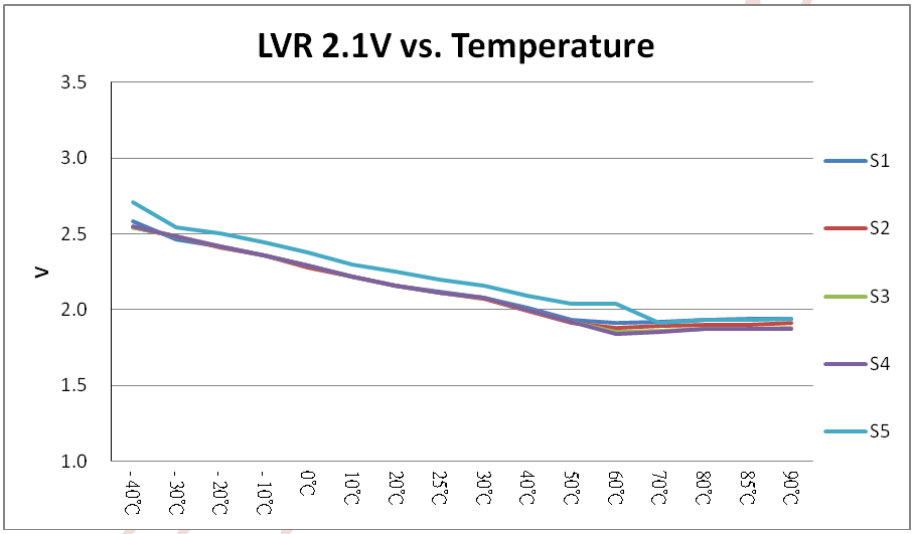
Symbol	Parameter	VDD	Condition	Min.	Typ.	Max.	Unit
	Operation Voltage	-	-	3	-	5.5	V
	Reference Voltage for Comparator	5V	-40~85 °C (±5%)	Typ. -5%	0.775* VDD	Typ. +5%	V
Vos	Analog Comparator Input Offset Voltage	5V	By calibration	-2	-	+2	mV
			Without calibration SVOSn[4:0]=10000	-15mV	-	+15mV	mV
Vcm	Analog Comparator Common Mode Voltage Range	-	-	0		VDD -1	V
tpd	Analog Comparator Response Time	-	Analog Comparator Hysteresis Disable and With 10mV overdrive			2	uS
Vhys	Analog Comparator Hysteresis Width	5V	Analog Comparator Hysteresis Enable	20	40	60	mV
	Power Consumption	5V	One Comparator		160		uA

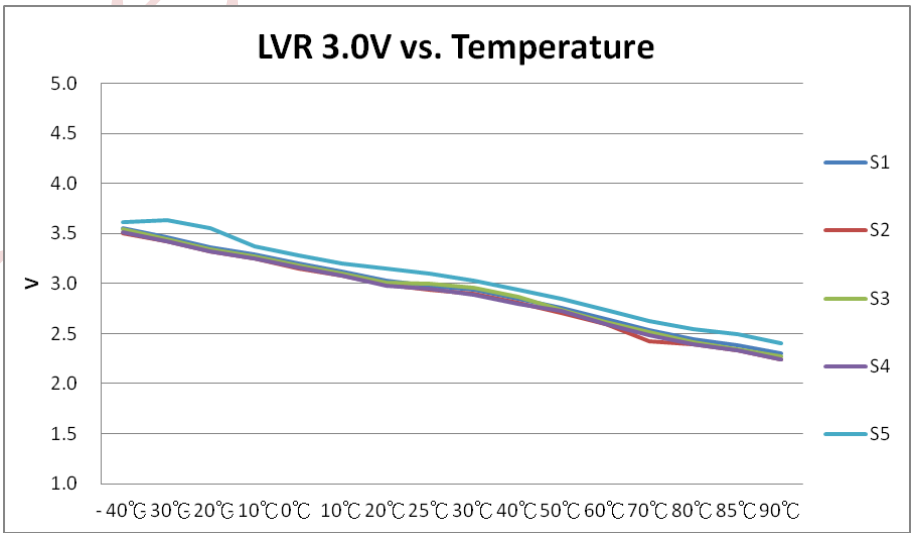
7. Characteristic Graphs






WKT vs. Temperature
WKT Time (ms)
350.00
300.00
250.00
200.00
150.00
100.00
50.00
0.00
-40℃ -30℃ -20℃ -10℃ 0℃ 10℃ 20℃ 25℃ 30℃ 40℃ 50℃ 60℃ 70℃ 80℃ 85℃ 90℃
5.5V
5.0V
4.5V
4.0V
3.5V
3.0V
2.5V
2.0V
1.5V


LVR 2.1V vs. Temperature
V
3.5
3.0
2.5
2.0
1.5
1.0
-40℃ -30℃ -20℃ -10℃ 0℃ 10℃ 20℃ 25℃ 30℃ 40℃ 50℃ 60℃ 70℃ 80℃ 85℃ 90℃
S1
S2
S3
S4
S5


LVR 3.0V vs. Temperature
V
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
-40℃ -30℃ -20℃ -10℃ 0℃ 10℃ 20℃ 25℃ 30℃ 40℃ 50℃ 60℃ 70℃ 80℃ 85℃ 90℃
S1
S2
S3
S4
S5

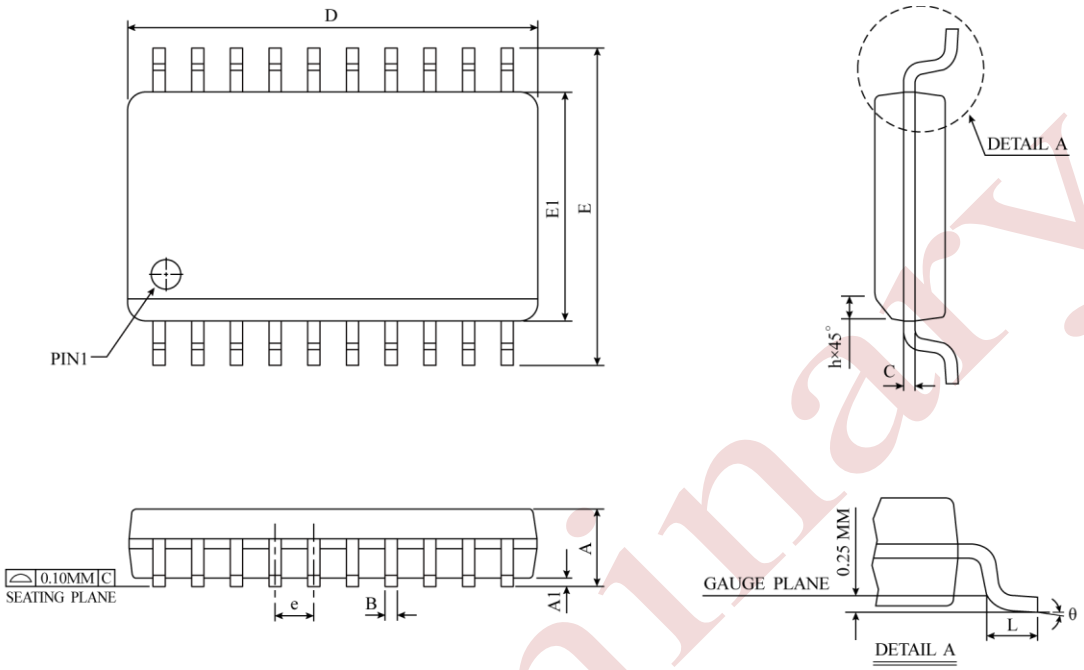
PACKAGING INFORMATION

The ordering information:

Ordering number	Package
TM57PT46-OTP	Wafer / Dice blank chip
TM57PT46-COD	Wafer / Dice with code
TM57PT46-OTP-20	SOP 18-pin (300mil)
TM57PT46-OTP-21	SOP 20-pin (300mil)
TM57PT46-OTP-05	DIP 20-pin (300mil)
TM57PT46-OTP-22	SOP 24-pin (300mil)
TM57PA46-OTP	Wafer/Dice blank chip
TM57PA46-COD	Wafer/Dice with code
TM57PA46-OTP-20	SOP 18-pin (300mil)
TM57PA46-OTP-21	SOP 20-pin (300mil)
TM57PA46-OTP-05	DIP 20-pin (300mil)
TM57PA46-OTP-22	SOP 24-pin (300mil)

Package Information

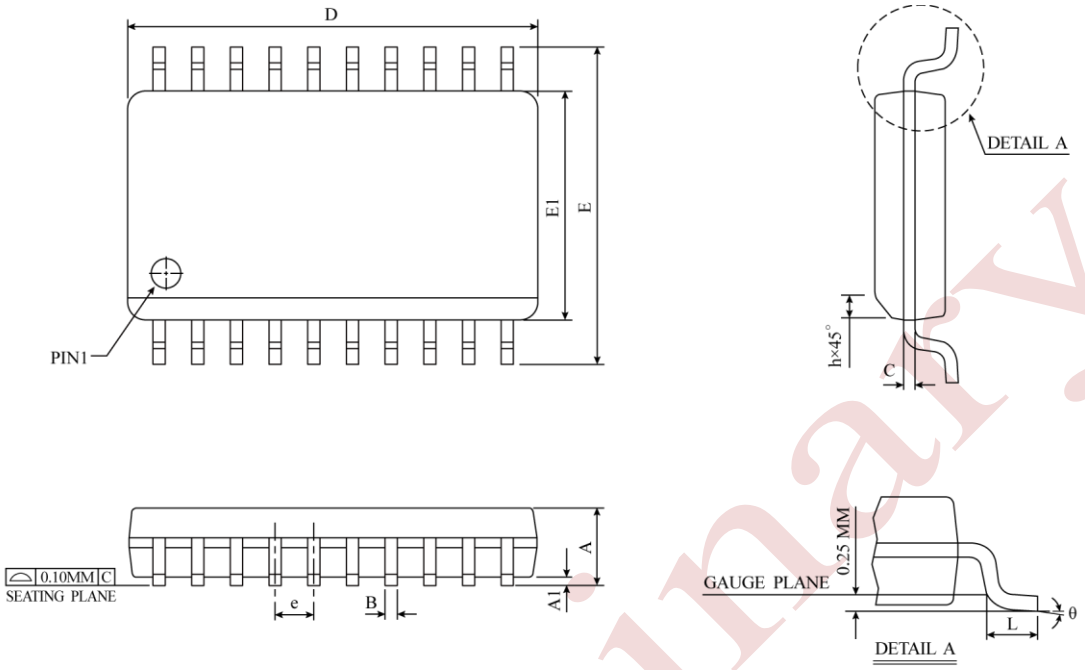
- SOP-18 (300mil) Package Dimension



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.

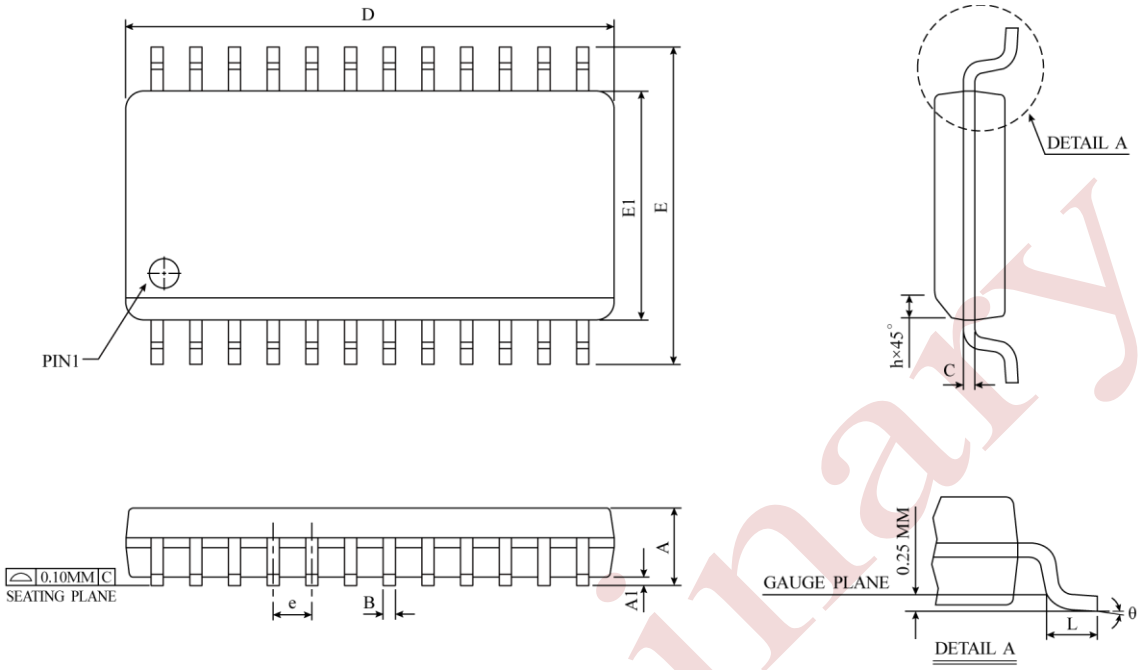
- SOP-20 (300mil) Package Dimension



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.

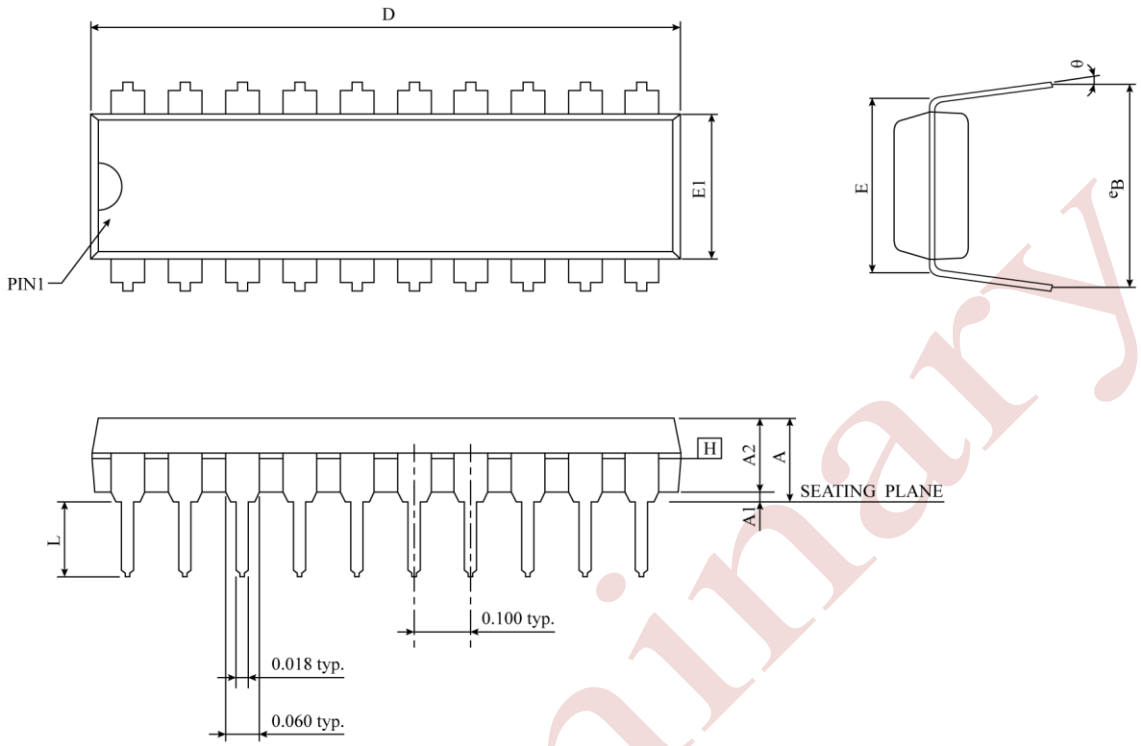
- SOP-24 (300mil) Package Dimension



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.

- DIP-20 (300 mil) 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.