十速

TM57PA25B

DATA SHEET

Rev 2.0

AMENDMENT HISTORY

Version	Date	Description
V1.0	Nov, 2011	New release
V1.1	Mar, 2012	1. Modify Features section. 2. Modify Block Diagram. 3. Modify Pin Assignment diagram. 4. Modify Pin Descriptions. 5. Modify Functional Descriptions section. 6. Modify Chip Operation Mode section. 7. Modify I/O Port section. 8. Modify Peripheral Function Block. 9. Modify Memory Map. 10. Modify Electrical Characteristics.
V1.2	Aug, 2012	1. Modify the figure and description about ADC and Internal Bandgap Reference Voltage. 2. Modify ADC Electrical Characteristics. 3. Modify the program code. 4. Modify the Memory Map description. 5. Modify DC Characteristics. 6. Modify Bandgap Ref. Voltage. 7. Modify the sequence of the register description. 8. Modify PROM description. 9. Modify SYSCFG description.
V1.3	Oct, 2012	1. Modify sample code. 2. Modify PAM, PBM, PDM descriptions. 3. Modify VBG vs. VDD figure.
V1.4	Apr, 2013	1. Modify Feature/Interrupt 2. Modify Timer0 description 3. Modify Capture Mode description 4. Modify Example 5. Modify Clock Timing/FIRC Freq. 6. Re-edit PWMA chapter
V1.5	May, 2013	1. Modify PWM0, PWMA description 2. Modify Bandgap description 3. Add body TM57PA25 4. PROM description 5. Modify PWMA block diagram 6. Modify ADC description 7. Example code 8. Add TM57PA25 VBG spec. 9. Modify Ordering Information
V1.6	Jul, 2013	1. Add Bandgap reference. 2. Add supported EV board on ICE. 3. Modify system block diagram. 4. Modify Pin Assignment.

		5. PWM pin description. 6. Add Pin Summary section. 7. Modify PROM description. 8. Modify PWM0 block diagram and description. 9. Modify PWMA clock. 10. Modify ADC description. 11. Modify R-Plane: PWM0/PWMA clock description. 12. Add TM57PA21B/25B description. 13. Add PWMAP, PWMAN high drive/sink description. 14. Modify LVR description. 15. Modify Ordering Information.
V1.7	Sept, 2013	1. Modify body to TM57PA21A, TM57PA21B, TM57PA25B. 2. Add AVREF pin. 3. Modify Pin assignment. 4. Chip name change. 5. Modify Package information.
V1.8	Sep, 2014	1. Remove 21A and 21B (doc headline and text) 2. Block diagram & Pinout 3. Remove AVREF 4. Modify VBG spec 5. Modify ordering information
V1.9	Dec, 2015	1. Modify Operating Voltage (p8) 2. Add LVR selection table (p19) 3. Modify DC (p82) 4. Add LVR vs. Temperature (p86)
V2.0	Aug, 2016	1. Add PWMAPSC issue and PWMA block diagram modified 2. PWMA example code modified 3. PWMAPSC description modified

CONTENTS

FEATURES

1. **ROM: 2K x 14 bits OTP**
2. **RAM: 176 x 8 bits**
3. **STACK: 6 Levels**
4. **System Oscillation Sources (Fsys)**
   - Fast-clock
     - FXT (Fast Crystal): 1M~24 MHz
     - FIRC (Fast Internal RC): 2/4/8/16 MHz
     - XRC (External R, External C): 10K~3 MHz
   - Slow-clock
     - SXT (Slow Crystal): 32768 Hz
     - XRC (External R, External C): 10K~3 MHz
     - SIRC (Slow Internal RC): 150K/37.5K/9.4K/2.3 KHz @5V; 116K/29K/7.25K/1.8 KHz @3V
5. **Dual System Clock**
   - FIRC + SIRC
   - FIRC + SXT
   - FIRC + XRC
   - FXT + SIRC
   - XRC + SIRC
6. **Power Saving Operation Mode**
   - FAST mode: Slow-clock can be disabled or enabled
   - SLOW mode: Fast-clock stops, CPU is running
   - IDLE mode: Slow-clock is running, CPU stops, T2 is running
   - STOP mode: All Clocks stop, Wake-up Timer is disabled or enabled
7. **Operation Voltage and Speed: VDD=2.2V @4 MHz**

8. 3 Independent Timers

- Timer0
  - 8-bit timer divided by 1~256 pre-scaler option, Counter/Interrupt/Stop function
  - Capture – high duty or low duty (pulse width measurement)
  - Overflow and Toggle out
- Timer1
  - 16-bit timer with two pre-scalers, Counter/Interrupt/Stop/Clear&Hold/Set/Reload function
  - Capture – period time
  - Overflow and Toggle out
- T2
  - 15-bit timer with 4 interrupt interval time options
  - IDLE mode wake-up timer or used as one simple 15-bit time base
  - Clock source: SXT/XRC/SIRC

9. 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/T2/WKT (wake-up) Interrupts
- ADC/PWM0 Interrupt

10. PB[7:0] individual pin low level wake up

11. Wake-up (WKT) Timer

- Clocked by built-in RC oscillator with 4 adjustable Interrupt times
  0.9 ms/1.9 ms/30 ms/122 ms @5V, 1.2 ms/2.4 ms/38 ms/152 ms @3V

12. Watchdog Timer

- Clocked by built-in RC oscillator with 4 adjustable Reset Time
  120 ms/240 ms/950 ms/1920 ms @5V, 150 ms/300 ms/1180 ms/2340 ms @3V
  Watchdog timer can be disabled/enabled in STOP mode (WDTSTP, (R0E.5))

13. **2 Independent PWMs**
    - PWM0:
      - 8-bit with 1~8 pre-scalers, period-adjustable/duty-adjustable/Clear&Hold
      - Clock source: System clock (Fsys)
      - With differential output pair
      - Non-overlap durations adjustable
    - PWMA:
      - 8+2 bits, period-adjustable/duty-adjustable/Clear&Hold
      - Clock sources: IRC 16 MHz or system clock (Fsys)
      - With differential output pair
      - Non-overlap durations adjustable
      - PWMAP and PWMAN are high drive/sink pins
14. **12-bit ADC converter with 11 input channels and 1 internal band gap reference on channel-11**
    - Internal Bandgap Reference Voltage 1.26V ±5% on Channel-11.
15. **Reset Sources**
    - Power On Reset/Watchdog Reset/Low Voltage Reset/External Pin Reset
16. **Low Voltage Reset Option: LVR2.0V, LVR2.0V disable in STOP mode, LVR2.9V**
17. **Operating Voltage: Low Voltage Reset Level to 5.5V**
    - Fsys=4 MHz, 2.2V~5.5V
    - Fsys=8 MHz, 2.4V~5.5V
    - Fsys=12 MHz, 2.7V~5.5V
    - Fsys=16 MHz, 3.1V~5.5V
    - Fsys=24 MHz, 4.0V~5.5V
18. **Enhanced Power Noise Rejection.**
19. **Operating Temperature Range: -40°C to +85°C**
20. **Instruction set: 36 Instructions**
21. **Instruction Execution Time**
    - 2 oscillation clocks per instruction except branch
22. **I/O ports: Maximum 18 programmable I/O pins**
    - Pseudo-Open-Drain Output
    - Open-Drain Output
    - CMOS Push-Pull Output
    - Schmitt Trigger Input with pull-up resistor option

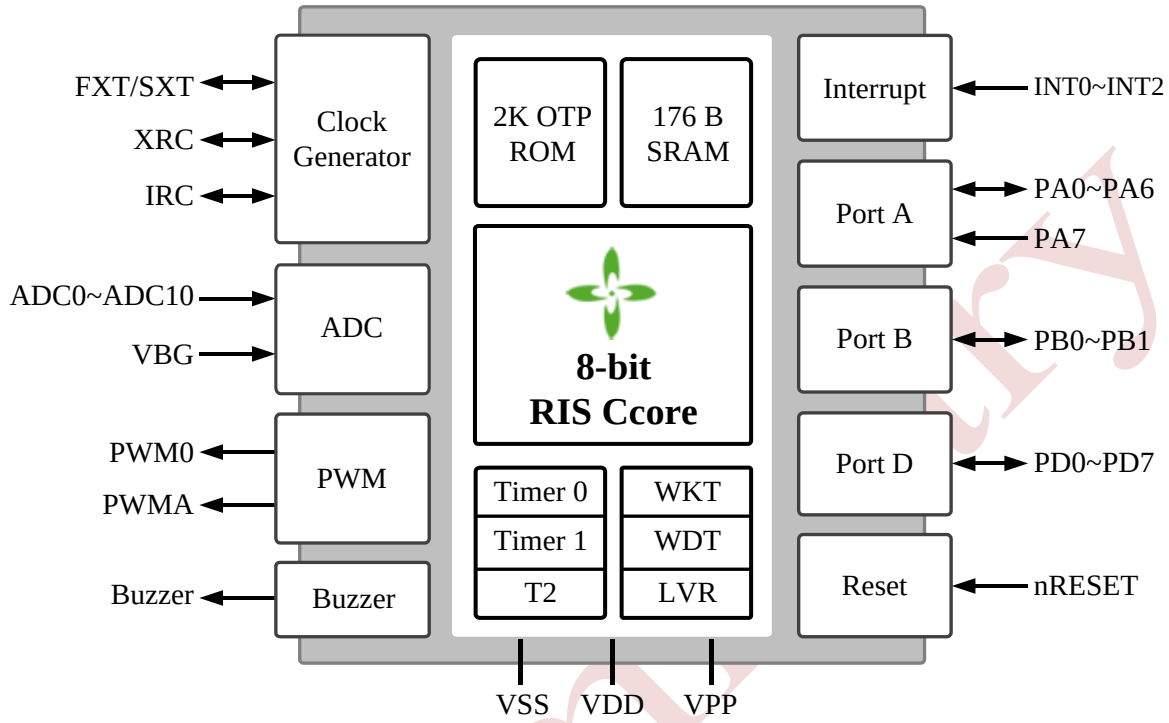
23. Package Types:

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

24. Supported EV board on ICE

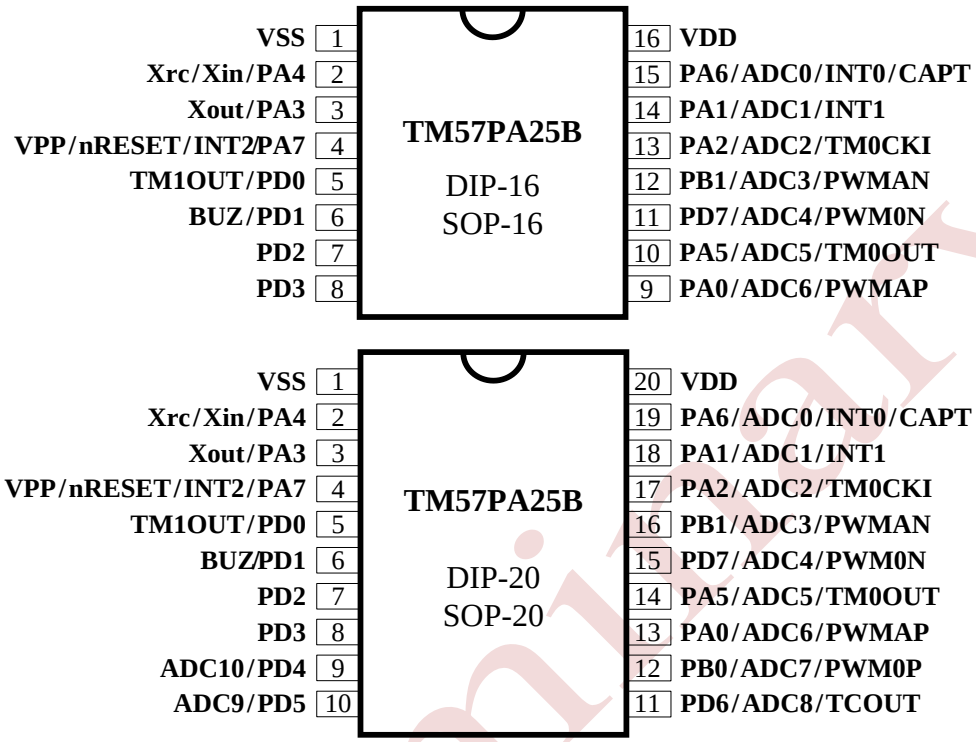
EV board: EV2784

BLOCK DIAGRAM



TM57PA25B Block Diagram

PIN ASSIGNMENT



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-PB1	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.
VPP/nRESET/PA7	I	Schmitt-trigger input with pull-high configurable, External active low reset, normal stay to "high".
Xin, Xout	–	Crystal/Resonator oscillator connection for system clock.
Xrc	–	External RC oscillator connection for system clock.
VDD, VSS	P	Power Voltage input pin and ground
VPP	I	PROM programming high voltage input
INT0–INT2	I	External interrupt input
PWM0N PWM0P PWMAP PWMAN	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
AD0~AD10	I	A/D converter input

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							
20-SOP/DIP	16-SOP/DIP			Weak Pull-up	Ext. Interrupt	O.D	P.O.D	P.P		PWM	Touch Key	ADC	MISC
1	1	VSS	P										
2	2	Xrc/Xin/PA4	I/O			O		O	PA4				Xrc/Xin
3	3	Xout/PA3	I/O			O		O	PA3				Xout
4	4	VPP/nRESET/INT2/PA7	I		O				PA7				nRESET
5	5	TM1OUT/PD0	I/O			O		O	PD0				TM1OUT
6	6	BUZ/PD1	I/O			O		O	PD1				BUZ
7	7	PD2	I/O			O		O	PD2				
8	8	PD3	I/O			O		O	PD3				
9	-	ADC10/PD4	I/O			O		O	PD4			O	
10	-	ADC9/PD5	I/O			O		O	PD5			O	
11	-	PD6/ADC8/TCOUT	I/O			O		O	PD6			O	TCOUT
12	-	PB0/ADC7/PWM0P	I/O			O		O	PB0	O		O	
13	9	PA0/ADC6/PWMAP	I/O				O	O	PA0	O		O	
14	10	PA5/ADC5/TM0OUT	I/O			O		O	PA5			O	TM0OUT
15	11	PD7/ADC4/PWM0N	I/O			O		O	PD7	O		O	
16	12	PB1/ADC3/PWMAN	I/O			O		O	PB1	O		O	
17	13	PA2/ADC2/TM0CKI	I/O				O	O	PA2			O	TM0CKI
18	14	PA1/ADC1/INT1	I/O		O		O	O	PA1			O	
19	15	PA6/ADC0/INT0/CAPT	I/O		O	O		O	PA6			O	CAPT
20	16	VDD	P										

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

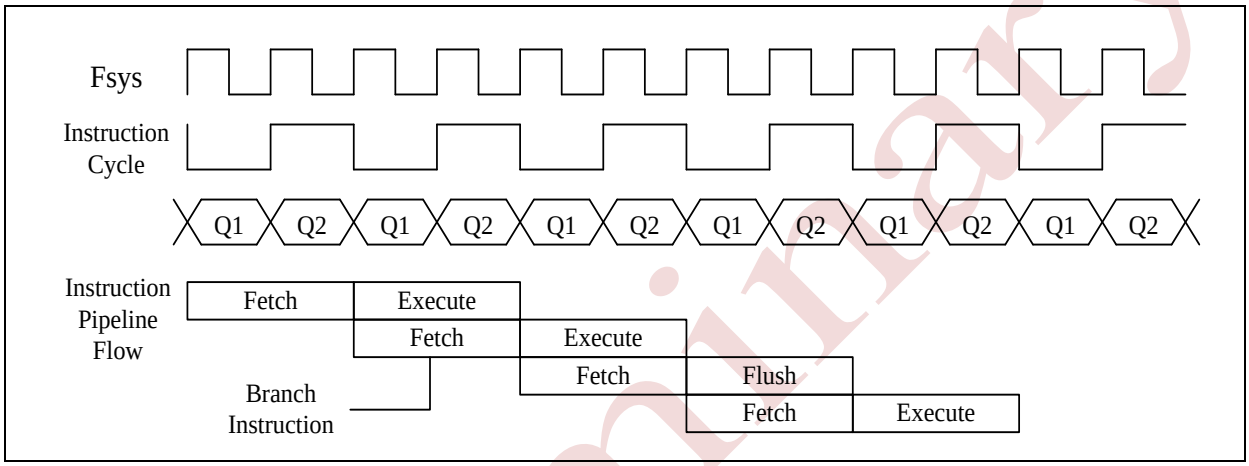
PWMAP and PWMAN are high drive/sink pins.

FUNCTIONAL DESCRIPTION

1. CPU Core

1.1 Clock Scheme and Instruction Cycle

The system clock is internally divided by two to generate Q1 state and Q2 state for each instruction cycle. The Programming Counter (PC) is updated at Q1 and the instruction is fetched from program ROM and latched into the instruction register in Q2. It is then decoded and executed during the following Q1-Q2 cycle. Branch instructions take two cycles since the fetch instruction is 'flushed' from the pipeline, while the new instruction is being fetched and then executed.



Terminology definitions:

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

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

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

Instruction Cycle = Fsys / 2

FXT: Fast Crystal

FIRC: Fast Internal RC oscillator

XRC: Fast or Slow External RC oscillator

SXT: Slow Crystal (32 KHz)

SIRC: Slow Internal RC oscillator

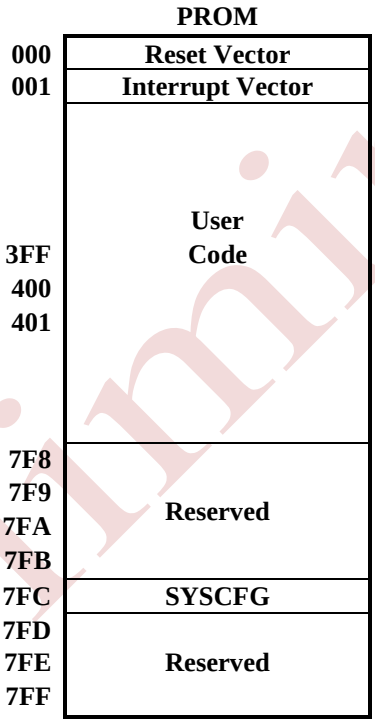
1.2 Program ROM (PROM)

The PROM of this device is 2K words. The addresses from 7F8 to 7FF are reserved by tenx. The address 7FC stores the system configuration bits (SYSCFG).

For TM57PA25B, the 7FA and 7F8 store the VBG value under VDD=3.6V while the 7FB and 7F9 stores the VBG value under VDD=5.0V.

These four values are programmed by tenx, they can be read by calling these addresses and the 'retlw 40h', for example, will be executed and W will load the corresponding value.

When PROTECT (SYSCFG.13) is 0, that means protect is enabled. The address from 000 to 3FF will be protected and the low-byte of the ROM code will be protected and read as 0s, however, the high 6 bits still can be shown by tenx writer.



1.3 System Configuration Register (SYSCFG)

The System Configuration Register (SYSCFG) is located at ROM address 7FCh. The SYSCFG determines the option for initial condition of MCU. It is written by PROM Writer only. User can select clock source, LVR threshold voltage and chip operation mode by SYSCFG register. The default value of SYSCFG is 14'b11_1111_111x_xxxx. The 13th bit of SYSCFG is code protection selection bit. If this bit is 0, the data in PROM will be protected, when user reads PROM.

Bit	**13~0**	
Default Value	**11_1111_111x_xxxx**	
Bit	Description	
13	**PROTECT**: Code protection selection	
	0	Enable
	1	Disable
12	**Reserved**	
11-10	**LVR**: Low Voltage Reset Mode	
	00	Disable
	01	2.9V, always enable
	10	2.0V, disable in STOP mode
	11	2.0V, always enable
9-8	**CLKT**: Clock Source Types	
	00	XRC
	01	FIRC
	10	SXT
	11	FXT
7	**XRSTE**: External Pin Reset Enable	
	0	Disable
	1	Enable
6	**WDTE**: WDT Reset Enable	
	0	Disable
	1	Enable
5-0	**Reserved**	

Bit13: PROTECT option

Protect code option is Program ROM (PROM) protection. When protect code option is enabled, the PROM code does not read PROM content.

Bit9-8: CLKT option

The register option is in the boot of FAST mode system clock source.

FXT: Fast Crystal

SXT: Slow Crystal (32 KHz)

FIRC: Fast Internal RC oscillator, 2/4/8/16 MHz, can be set by R0E.3~2

XRC: Fast External RC oscillator

Bit7: XRSTE option

The reset pin is shared with the general input pin (PA7) controlled by SYSCFG[7] option.

- Reset: The reset pin is external reset function. When falling edge trigger occurs, the system will be reset.
- PA7: Set reset pin to general input pin (PA7). The external reset function is disabled.

1.4 Program Counter (PC) and Stack

The Programming Counter is 11-bit wide capable of addressing a 2K x 14 OTP ROM. As a program instruction is executed, the PC will contain the address of the next program instruction to be executed. The PC value is normally increased by one except the followings. The Reset Vector (000h) and the Interrupt Vector (001h) are provided for PC initialization and Interrupt. For CALL/GOTO instructions, PC loads 11 bits address from instruction word. For RET/RETI/RETLW instructions, PC retrieves its content from the top level STACK. For the other instructions updating PC [7:0], the PC [10:8] keeps unchanged. The STACK is 11-bit wide and 6-level in depth. The CALL instruction and hardware interrupt will push STACK level in order. While the RET/RETI/RETLW instruction pops the STACK level in order.

◇ Example: To look up the PROM data located in "TABLE".

```
            ORG     000H        ; Reset Vector
            GOTO    START

START:
            MOVLW   00H
            MOVWF   INDEX       ; Set lookup table's address.

LOOP:
            MOVFW   INDEX       ; Move index value to W register.
            CALL    TABLE       ; To look up data, W=55H.
            …..
            INCF    INDEX,1     ; Increment the index address for next address
            …..
            GOTO    LOOP        ; Go to LOOP label.

TABLE:
            ADDWF   PCL,1       ; Add the W register with register PCL, the result is stored back
                                ; in PCL register.
            RETLW   55H         ; W=55h when returns
            RETLW   56H         ; W=56h when returns
            RETLW   58H         ; W=58h when returns
            …..
```

1.5 Reset (000H)

This device can be RESET in four ways.

- Power-On-Reset
- Low Voltage Reset (LVR) (SYSCFG bit-11-10)
- External Pin Reset (PA7) (SYSCFG bit-7)
- Watchdog Reset (WDT) (SYSCFG bit-5)

After Power-On-Reset, all system and peripheral control registers are then set to their default hardware Reset values. The clock source, LVR level and chip operation mode are selected by the SYSCFG register value. The Low Voltage Reset features static reset when supply voltage is below a threshold level. There are three threshold levels can be selected. The LVR's operation mode is defined by the SYSCFG register. If operating frequency is faster than 16 MHz, selection LVR 2.9V is recommended. The External Pin Reset and Watchdog Reset can be disabled or enabled by the SYSCFG register. These two resets also set all the control registers to their default reset value. The TO/PD flags is not affected by these resets.

VR Selection Table

LVR Threshold Level	Consider the operating voltage to choose LVR
LVR2.9	5.5V > V_{DD} > 3.3V or V_{DD}=5.0V
LVR2.0	V_{DD} is wide voltage range

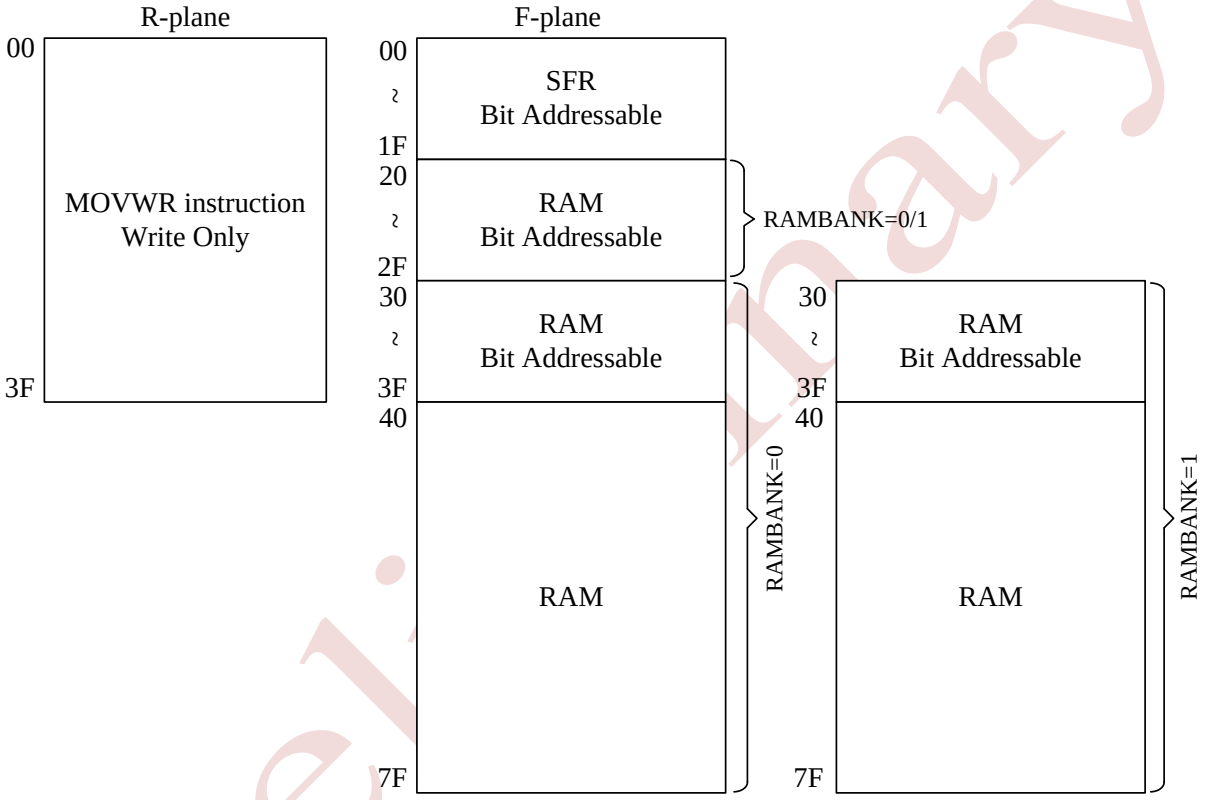
Different Fsys have different system minimum operating voltage, reference to Operating Voltage of DC characteristics, if current system voltage is lower than minimum operating voltage and lower LVR is selected, then the system maybe enter dead-band and error occur.

◇Example: Defining Reset Vector

```
        ORG     000H
        GOTO    START       ; Jump to user program address.

        ORG     010H

START:

        …                   ; 010H, The head of user program
        …
```

1.6 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 copies the W-register's content to R-Plane registers by direct addressing mode. The lower locations of F-Plane are reserved for the SFR. Above the SFR is General Purpose Data Memory, implemented as static RAM. F-Plane can be addressed directly or indirectly. Indirect Addressing is made by INDF register. The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a pointer). The first half of F-Plane is bit-addressable, while the second half of F-Plane is not bit-addressable.



◇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 data register
```

◇Example: Move the immediate data 55H to W register and F-Plane location 20H.

```
MOVLW  55H    ; Move immediate 55H into W register
MOVWF  20H    ; Get the content of W and save in F-Plane location 20H
```

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

```
MOVFW  20H    ; Get the content of F-Plane location 20H and save in W
```

◇Example: Indirectly addressing mode with FSR/INDF register (F-Plane 04H / 00H).

```
MOVLW  20H
MOVWF  FSR    ; Move immediate 20H into FSR register
MOVLW  55H
MOVWF  INDF   ; Use data pointer FSR writes a data into F-Plane location 20H
              ; 55H into F-Plane 20H
INCF   FSR,1  ; Increment the index address for next address
MOVFW  INDF   ; Use data pointer FSR reads a data from F-Plane location 21H
              ; W data from F-Plane 21
```

1.7 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.

1.8 STATUS Register (F-Plane 03H)

This register contains the arithmetic status of ALU and the reset status. The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. It is recommended, therefore, that only BCF, BSF and MOVWF instructions are used to alter the STATUS register because these instructions do not affect those bits.

STATUS	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reset Value	**0**	**0**	**0**	**0**	**0**	**0**	**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**	
4	**TO**: Time Out Flag 0: after Power On Reset, LVR Reset, or CLRWDT/SLEEP instruction 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,0      ; Set C=1
BSF    03H,5         ; Selection RAM Bank1
BCF    STATUS,0      ; Clear C=0
BCF    03H,5         ; Selection RAM Bank0
```

◇Example: Determine the C flag by BTFSS instruction.

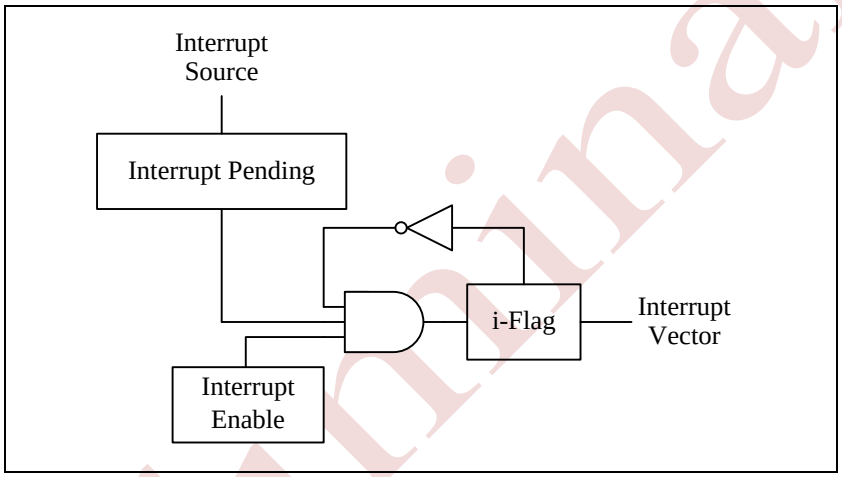
```
BTFSS  STATUS,0      ; Check the carry flag
GOTO   LABEL_1       ; If C=0, go to LABEL_1
GOTO   LABEL_2       ; If C=1, go to LABEL_2
```

1.9 Interrupt

This device has 1 level, 1 vector and eight 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 TM57PA25B 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.



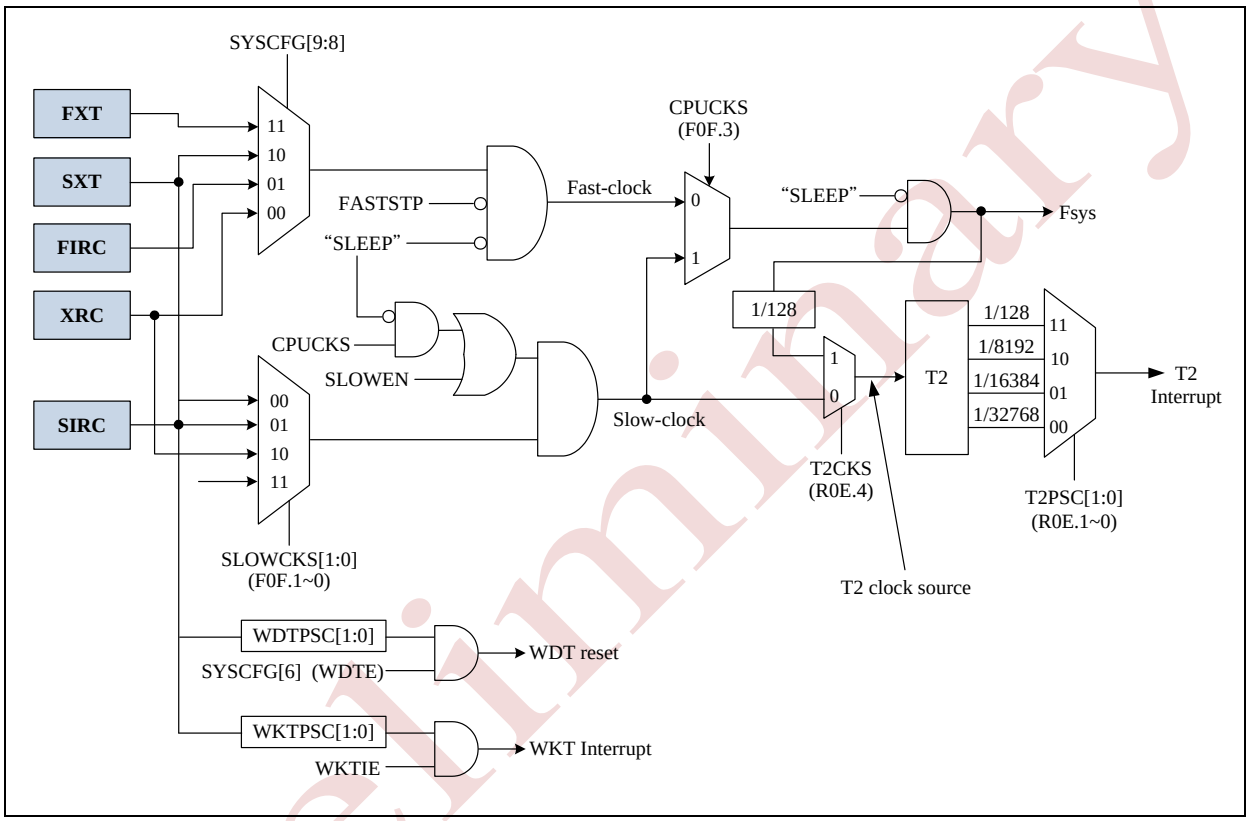
◇Example: Setup INT1 (PA1) interrupt request and rising edge trigger.

```
        ORG     000H                ; Reset vector
        GOTO    START               ; Go to user program address
        ORG     01H                 ; All interrupt vector
        GOTO    INT_SUBROUTINE      ; If INT1(PA1) input occurs, trigger rising edge
        ORG     02H

START:
        MOVLW   11111101B
        MOVWR   PAPU                ; Enable INT0 (PA1) input pull up resistor
        MOVLW   00001000B
        MOVWR   R0C                 ; Set INT1 interrupt trigger as rising edge
        MOV     11111101B
        MOVWF   INTIF               ; Clear INT0 interrupt request flag
        BSF     INT1IE              ; Enable INT0 interrupt.

MAIN:
        …
        GOTO    MAIN

INT_SUBROUTINE:
        …                           ; Push routine to Save W and STATUS data to buffers
        BTFSS   INT1IF              ; Check INT1IF bit
        GOTO    EXIT_INT            ; INT1IF=0, exit interrupt vector
        …                           ; INT1 interrupt service routine
        MOVLW   11111101B
        MOVWF   INTIF               ; Clear INT0IF bit
        GOTO    EXIT_INT

EXIT_INT:
        …                           ; POP Routine W and STATUS data from buffers
        RETI
```

2. Chip Operation Mode

2.1 Dual System Clock

TM57PA25B is designed with dual-clock system. There are five kinds of clock source, i.e. FXT (Fast Crystal) Clock, SXT (Slow Crystal) Clock, XRC (External RC) Clock, SIRC (Slow Internal RC) Clock and FIRC (Fast Internal RC). Each clock source can be applied to CPU kernel as system clock. When in IDLE mode, only Slow-clock can be configured to keep oscillating to provide clock source to T2 block. Refer to the figure below.



Clock Scheme Block Diagram

FAST mode

After power on or reset, TM57PA25B enters FAST mode. In FAST mode, the chip can select FXT, XRC or FIRC as its CPU clock is set by SYSCFG[9:8] setting. Besides, firmware can also enable or disable the Slow-clock for the T2 system operating.

In this mode, the program is executed using Slow-clock as CPU clock (Fsys). The Timer0, Timer1, PWM0, PWMA blocks are also driven by Slow-clock. T2 can also be driven by Slow-clock by setting T2CKS=1 and CPUCKS=0.

SLOW mode

After power on reset, user must set SLOWCKS[1:0] to select Slow-clock type (SXT, XRC or SIRC) as its CPU clock when running at FAST Mode. In SLOW mode, the Fast-clock is stopped and Slow-clock is enabled for power saving. All peripheral blocks (Timer0, Timer1, PWM0, PWMA, etc...) clock sources are Slow-clock in the SLOW mode.

IDLE mode

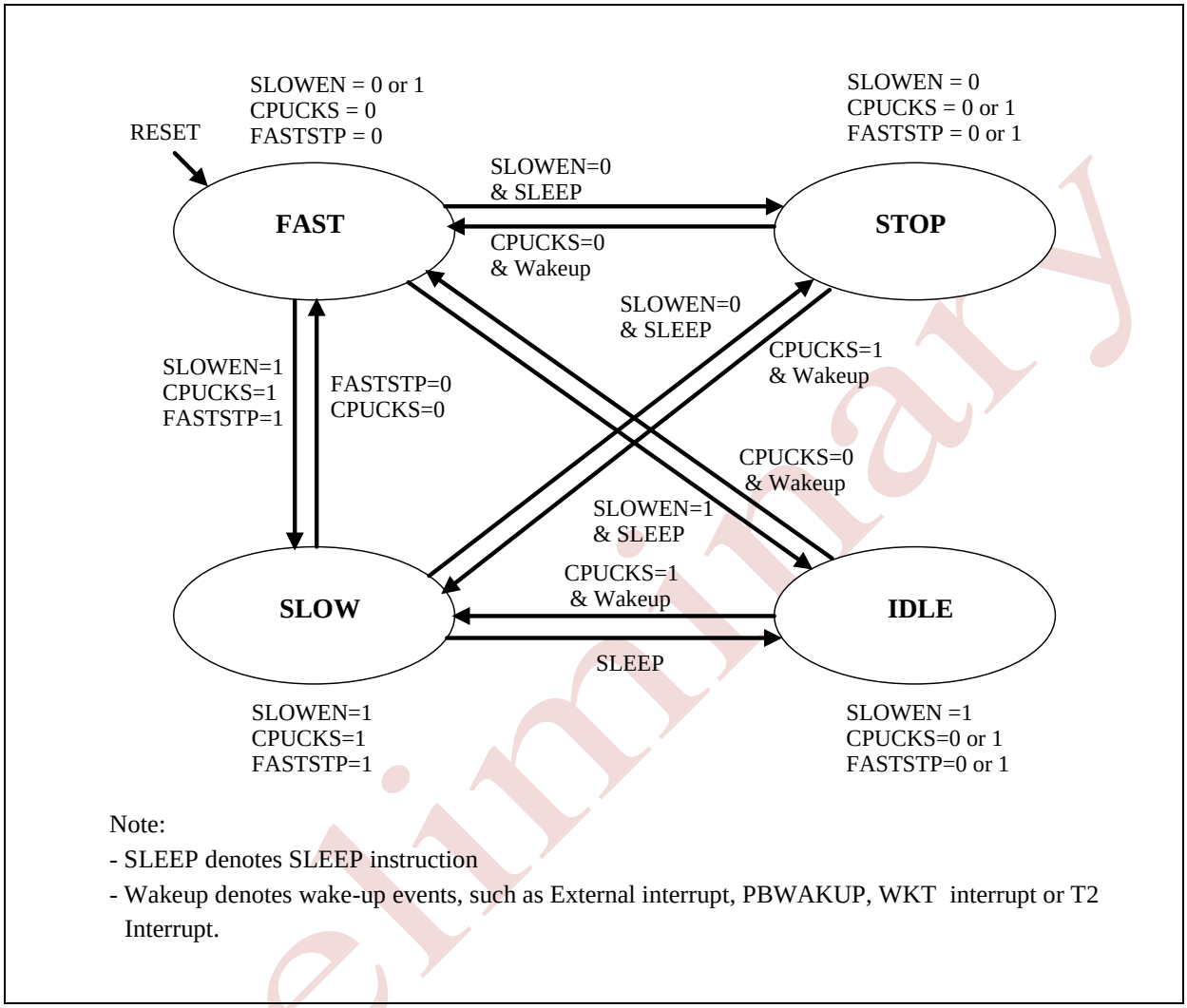
If Slow-clock is enabled and T2CKS=0 before executing the SLEEP instruction, the TM57PA25B enters the IDLE mode. In this mode, the Slow-clock will continue running to provide clock to T2 block. CPU stop fetching code and all blocks are stop except T2 related circuits.

STOP mode

If Slow-clock is disabled before executing the SLEEP instruction, every block is turned off and the TM57PA25B enters the STOP mode. STOP mode is similar to IDLE mode. The difference is all clock oscillators either Fast-clock or Slow-clock is power down and no clock is generated.

2.2 Dual System Clock Modes Switching

TM57PA25B is operated in one of four modes: FAST mode, SLOW mode, IDLE mode, and STOP mode.



Note:
- SLEEP denotes SLEEP instruction
- Wakeup denotes wake-up events, such as External interrupt, PBWAKUP, WKT interrupt or T2 Interrupt.

CPU Operation Block Diagram

Mode	Oscillator	Fsys	Fast-clock	Slow-clock	TM0/TM1	T2	Wakeup event
FAST	FIRC, FXT, SXT, XRC	Fast-clock	Run	Set by SLOWEN bit	Run	Run	X
SLOW	SXT, XRC, SIRC	Slow-clock	Set by FASTSTP	Run	Run	Run	X
IDLE	SXT, XRC, SIRC	Stop	Stop	Run	Stop	Run	WKT/IO/PB/T2
STOP	SIRC(1)	Stop	Stop	Stop	Stop	Stop	WKT/IO/PB

(1) If WDT or WKT function is enabled

PA3/PA4 IO setting notes in dual clock mode

Note: In Slow-clock modes, PA3 and PA4 must be set as input pull-up mode to prevent the input pin floating when Slow-clock selects SXT or XRC mode. PA3 and PA4 IO setting list is as shown below.

	Fast-clock	Slow-clock	PAD3	PAE3	PAPUN3	PAD4	PAE4	PAPUN4
1	FIRC	SIRC	※	※	※	※	※	※
1	FIRC	SXT	1	0	0	1	0	0
2	FIRC	XRC	※	※	※	1	0	0
3	FXT	SIRC	※	※	※	※	※	※
4	XRC	SIRC	※	※	※	※	※	※

※：Don't care

- **FAST mode switches to SLOW mode**

FAST mode can be chosen by SYSCFG [9:8] when equals to 11 (FXT), 00 (XRC), or 01 (FIRC). The following steps are suggested to be executed by order when FAST mode switches to SLOW mode:

(1) Enable Slow-clock (SLOWEN=1)

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

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

◇Example: Switch FAST mode to SLOW mode.

```
MOVLW  01100001B
MOVWF  F0F          ; Slow-clock type=SIRC 2 KHz.
BSF    SLOWEN       ; Enable Slow-clock.
NOP
BSF    CPUCKS       ; Fsys=Slow-clock.
BSF    FASTSTP      ; Disable Fast-clock.
```

- **SLOW mode switches to FAST mode**

SLOW mode can be enabled by SLOWEN bit and CPUCKS bit in F0F register of F-Plane. The following steps are suggested to be executed by order when SLOW mode switches to FAST mode:

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

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

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

Note: Stop Slow-clock (SLOWEN=0) is optional. Slow-clock can keep oscillating to provide T2 counter block in FAST mode.

◇Example: Switch SLOW mode to FAST mode (The Fast-clock stop).

```
MOVLW  0000110 0B
MOVWR  R0E          ; Fast-clock type=FIRC 16 MHz
BCF    FASTSTP      ; Enable Fast-clock.
NOP
BCF    CPUCKS       ; Fsys=Fast-clock
BC     SLOWEN       ; Disable Slow-clock
```

- **IDLE mode setting**

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

(1) Enable Slow-clock (SLOWEN=1)

(2) Switch T2 clock source to Slow-clock (T2CKS=0)

(3) Execute SLEEP instruction

IDLE mode can be waken up by INT, PBWAKUP, WKT interrupt and T2 interrupt.

◇Example: Switch FAST/SLOW mode to IDLE mode.

```
BSF    SLOWEN       ; Enable Slow-clock.
MOVLW  0000 0000B
MOVWR  R0E          ; T2 Clock source=Slow-clock.
SLEEP               ; Enter IDLE mode.
```

- **STOP mode setting**

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

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

(2) Execute SLEEP instruction

STOP mode can be waken up by INT, PBWAKUP and WKT interrupt.

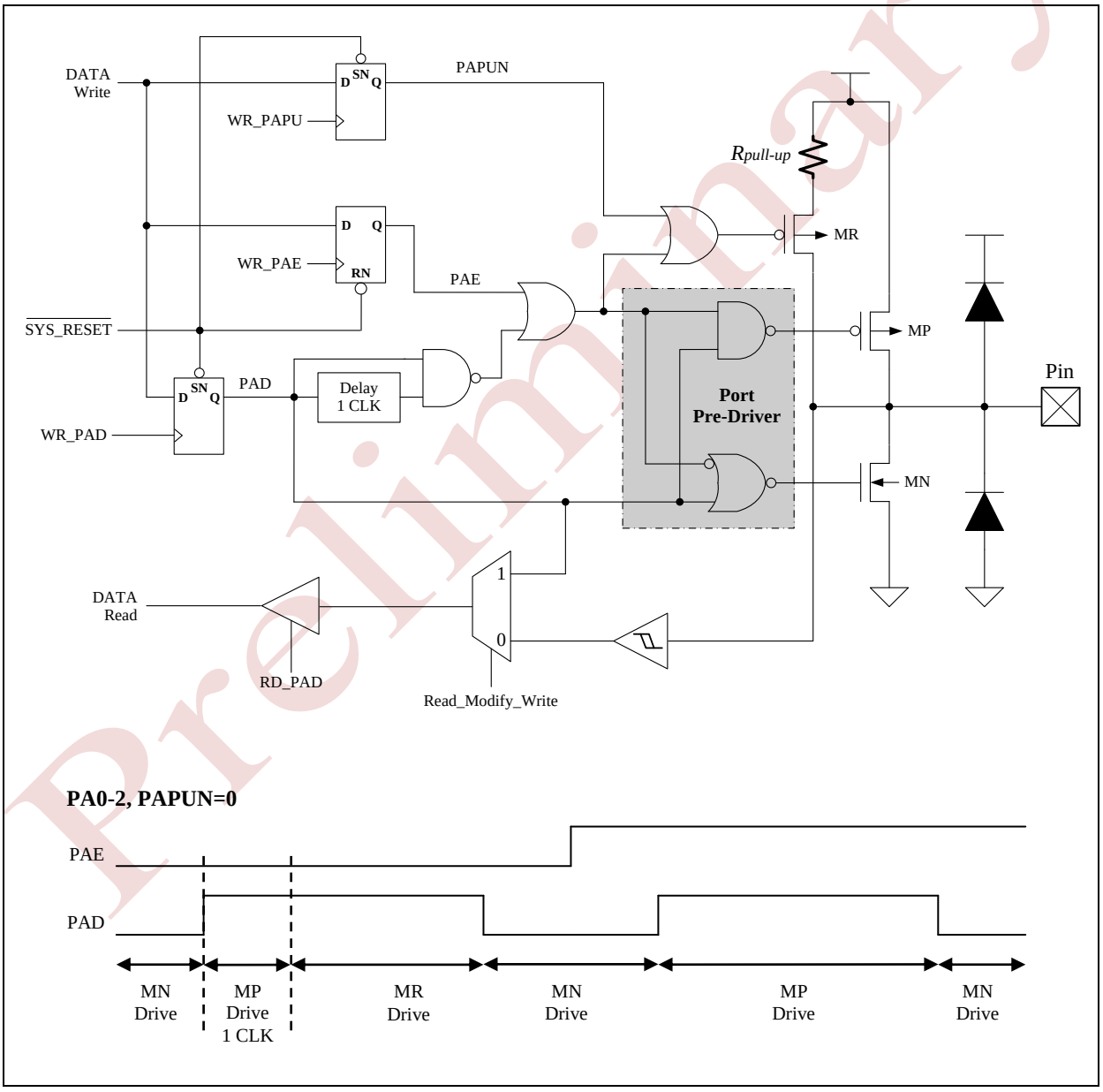
◇Example: Switch FAST/SLOW mode to STOP mode.

```
BCF    SLOWEN       ; Disable Slow-clock.
SLEEP               ; Enter STOP mode.
```

3. I/O Port

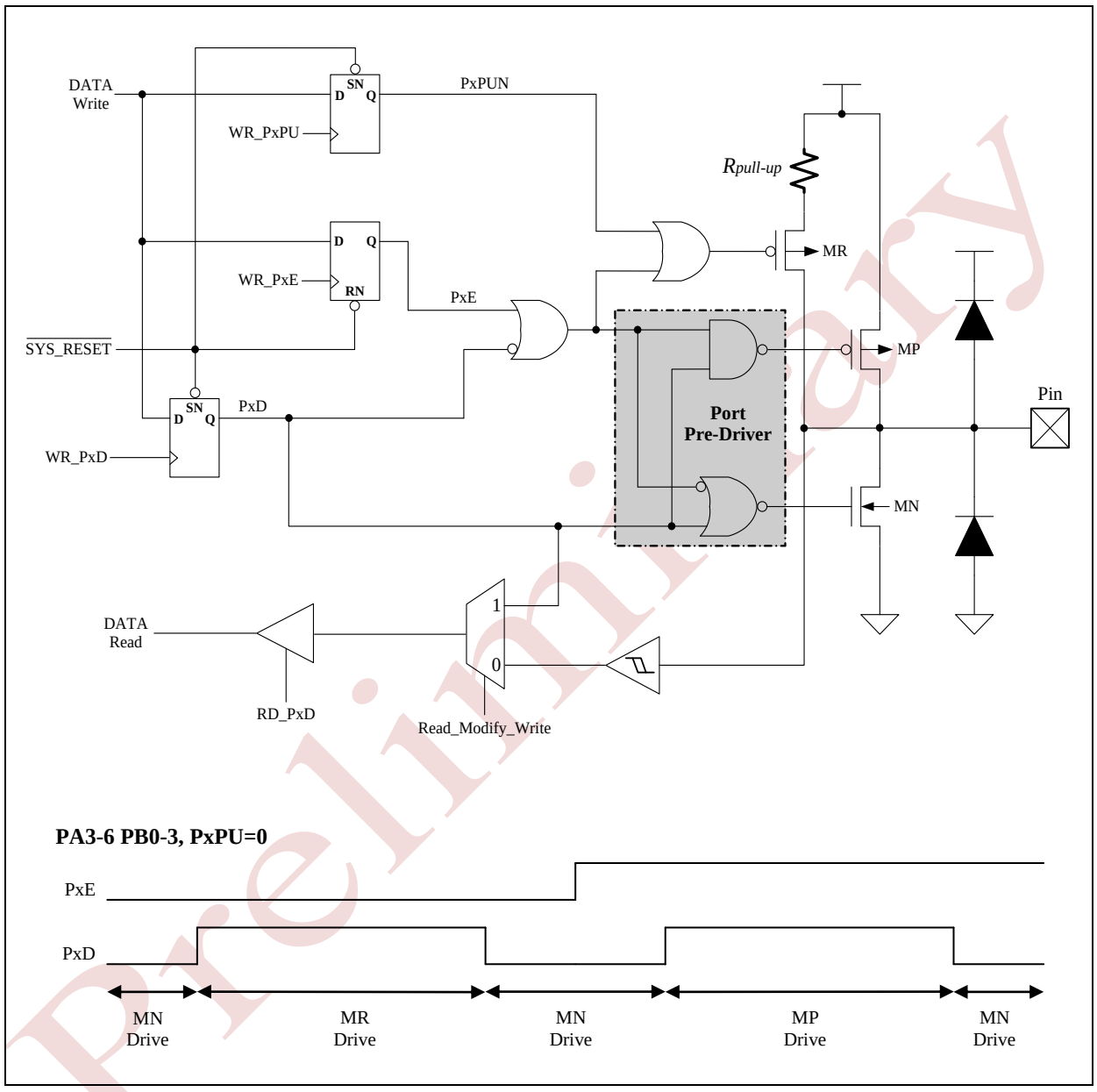
3.1 PA0-2

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.



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

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



◇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  000B
MOVWR  PAE          ; Set PA2-PA0 as pseudo-open-drain mode

MOVLW  000B
MOVWF  PAD          ; PA2~PA0 output low level
```

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

```
MOVLW  111B
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
```

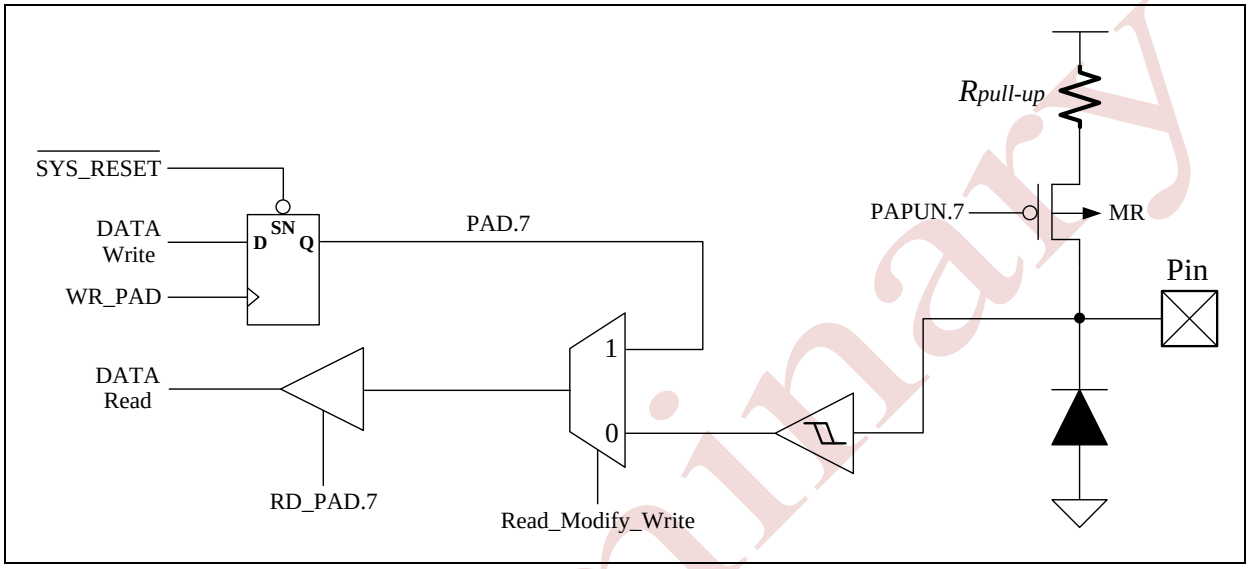
◇Example: Write one bit data to output port.

```
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”
```

3.3 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.

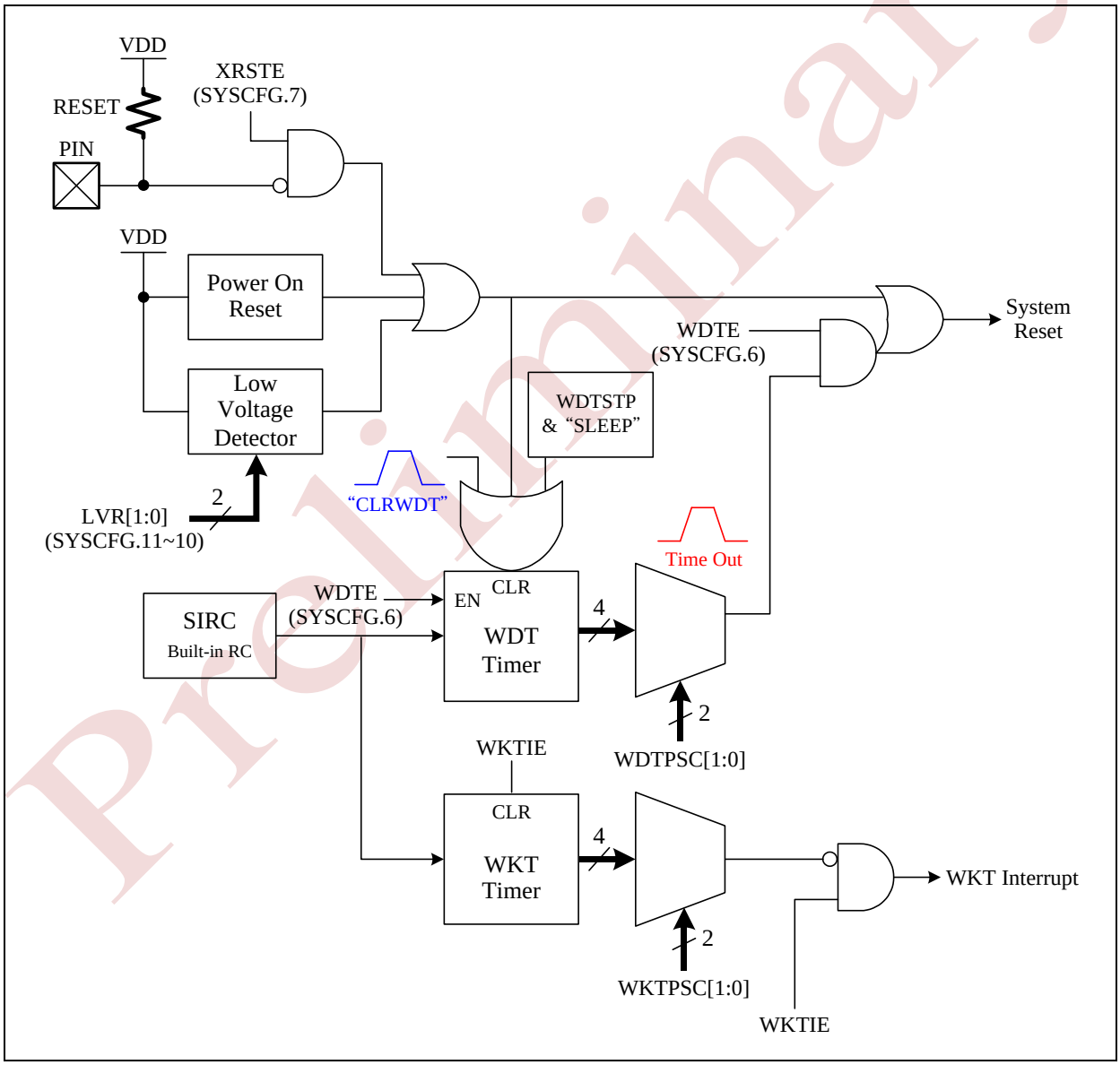
```
BTFSS   PAD,7
GOTO    LOOP_A      ; If PA7=0.
GOTO    LOOP_B      ; If PA7=1.
```

4. Peripheral Functional Blocks

4.1 Watchdog (WDT)/Wakeup (WKT) Timer

The WDT and WKT share the same internal RC Timer (IRC). The overflow period of WDT, WKT can be selected by individual both four options (WDTPSC[1:0], 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. Set WDTSTP (R0E.5) to '1' can let WDT timer stop counting after executing SLEEP instruction, i.e. WDTSTP=0 WDT timer always keeps counting even if the SLEEP instruction is executed.

The WKT timer is an interval time, if WKT time is up, 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

Watchdog clear is controlled by CLRWDT instruction and moving any value into WDTCLR is done by watchdog timer.

◇Example: Clear watchdog timer by CLRWDT instruction.

```
MAIN:
        …                       ; Execute program
        CLRWDT                  ; Execution of CLRWDT instruction
        …
        GOTO    MAIN
```

◇Example: Clear watchdog timer by writing WDTCLR register.

```
MAIN:
        …                       ; Execute program
        MOVWF   WDTCLR          ; Write any value into WDTCLR register
        …
        GOTO    MAIN
```

◇Example: Setup WDT time and disable after executing SLEEP instruction.

```
        MOVLW   00100000B
        MOVWR   R0E             ; Select WDT Time out  120 ms @5V (default 240 ms)
                                ; Setup WDT disable in IDLE/STOP mode (default is enable)
        SLEEP
```

◇Example: Set WKT period and interrupt function.

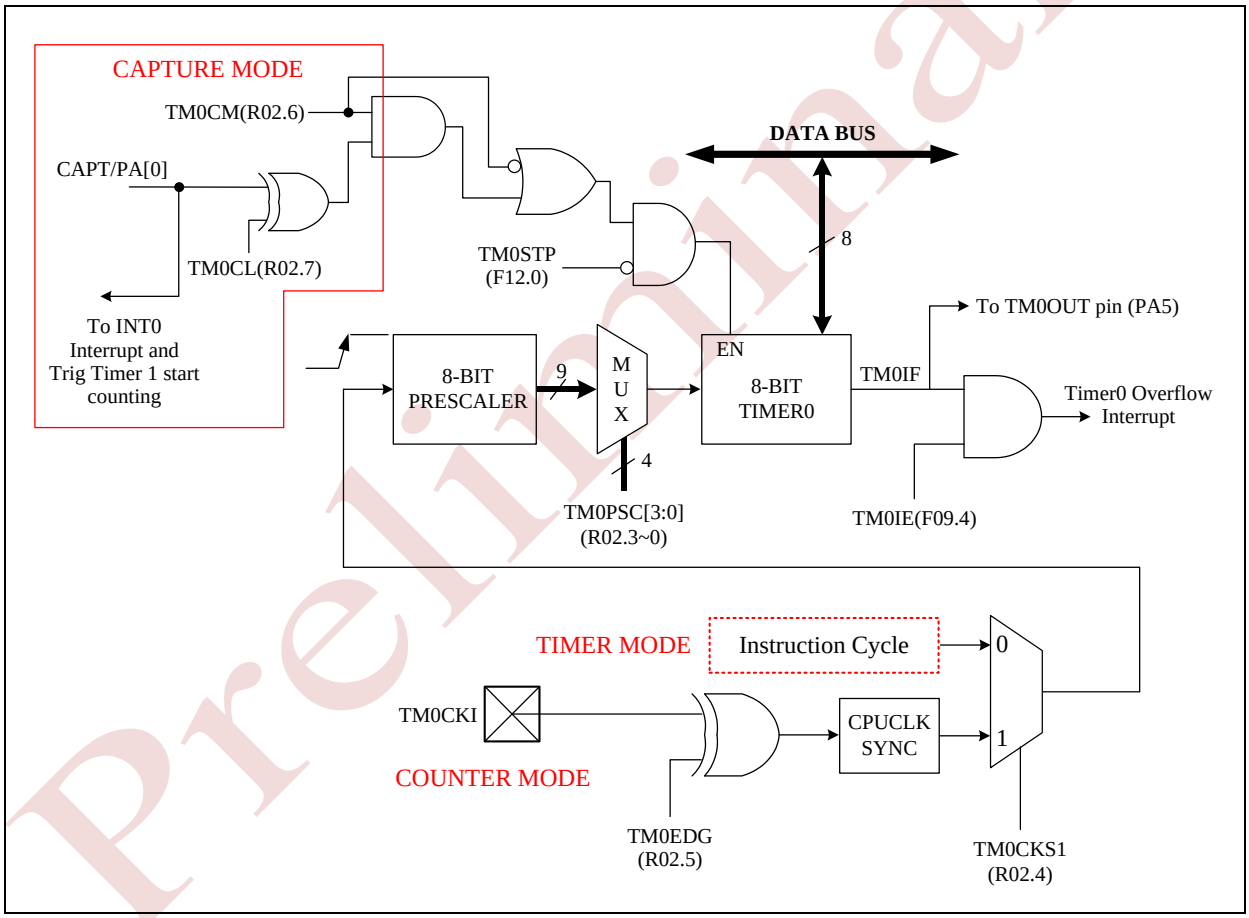
```
        MOVLW   00010B
        MOVWR   R0B             ; Select WKT period=30 ms @5V (default 122 ms)
        MOVLW   11110111B       ; Clear WKT interrupt request flag
        MOVWF   INTIF
        BSF     WKTIE           ; Enable WKT interrupt function
```

4.2 Timer0

The Timer0 is an 8-bit wide register of F-Plane 01h (TM0). 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 increase rate is determined by "Timer0 Pre-Scale" (TM0PSC) register in R-Plane. The Timer0 can generate interrupt flag (TM0IF) when it counts to rolls over if Timer0 Interrupt enable (TM0IE) is set. Timer0 can be stopped counting if the TM0STP bit is set. T0OUT is an output signal that toggles when Timer0 overflows.

Timer0 can be configured as Capture mode. If TM0CM bit is set to "1", Timer0 will not count until the CAPT pin (i.e. PA0) is high level (TM0CL=0) or low level (TM0CL=1).

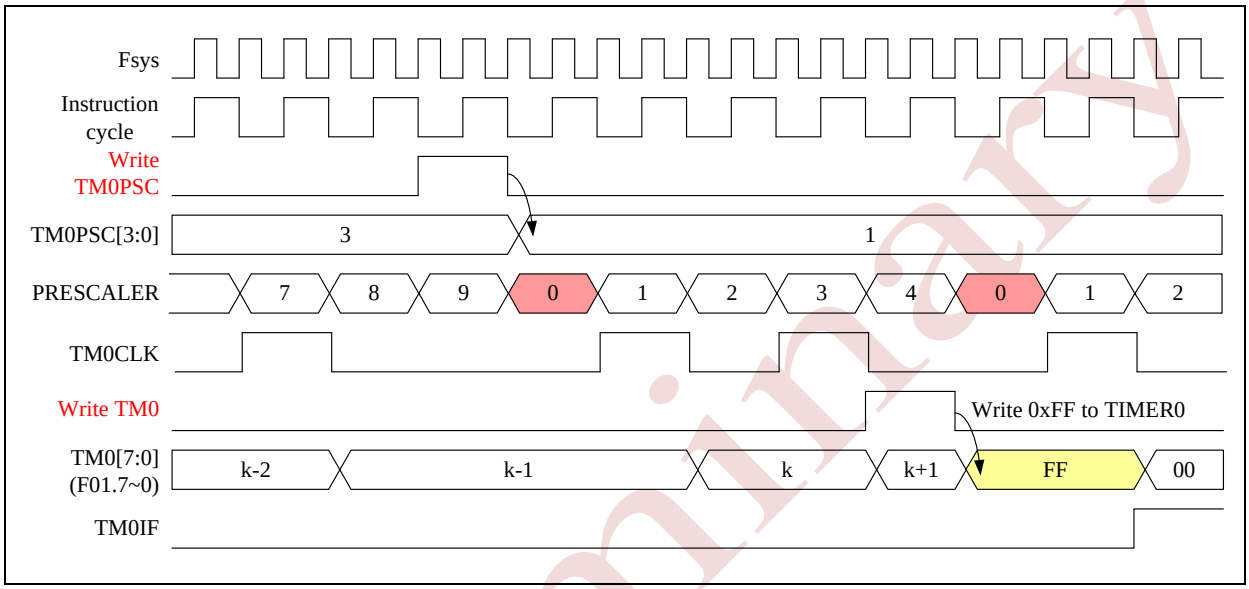
Timer0 can also be used to measure the pulse width and period capture on CAPT pin. This function needs the Timer1 and INT0 external interrupt. Software control details will be discussed step by step.



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

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

; Set T0OUT pin function.
        MOVLW  00000100B
        MOVWR  R0B          ; 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:

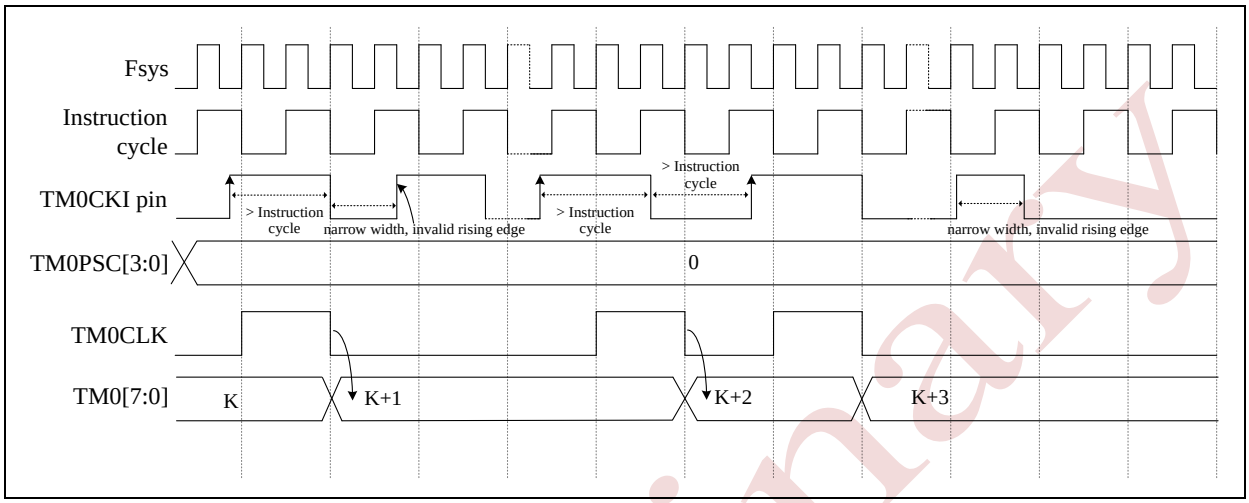
TM0 clock source is Fsys=4 MHz, Instruction cycle=2 MHz, TM0PSC=/32, TM0=156,

TM0OUT output frequency=2 MHz/32/ (256-156) =2 MHz/32/100=312.5 Hz

T0OUT 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)

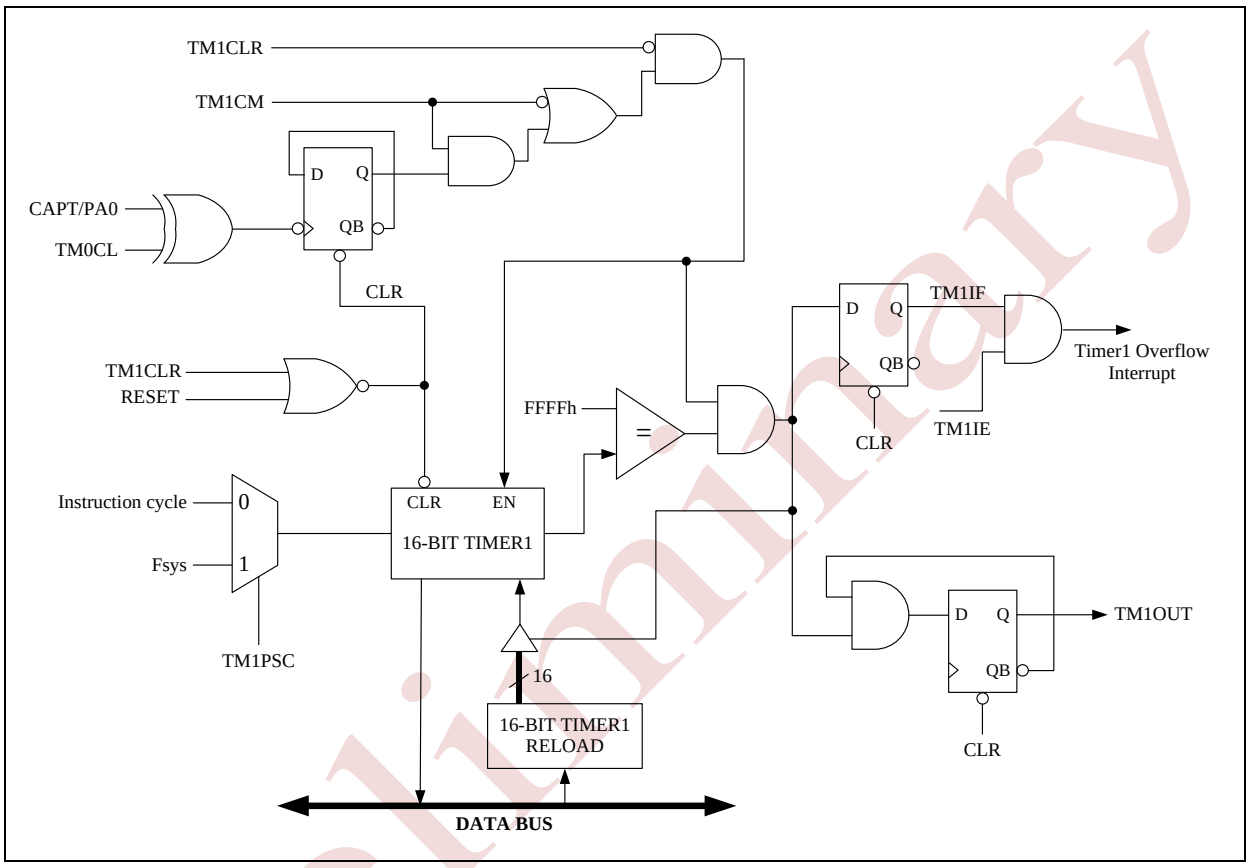
◇Example:

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

```
; Setup TM0 clock source from TM0CKI pin (PA2) and divide.
        MOVLW  0010000B
        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
```

4.3 Timer1

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



Timer1 Block Diagram

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

◇Example:

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

```
; Setup TM1 clock source and divider.
        MOVLW  00000101B
        MOVWR  R0C          ; Bit0=1: Select TM1 clock source=Fsys
                            ; Bit1=0: Select TM1 mode=Timer mode
                            ; Bit2=1: Enable TM1OUT function pin (PD0)

; Set TM1 timer.
        BSF    TM0STP       ; Stop TM1 counting (Default "0")
        BCF    TM0SET
        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”)
```

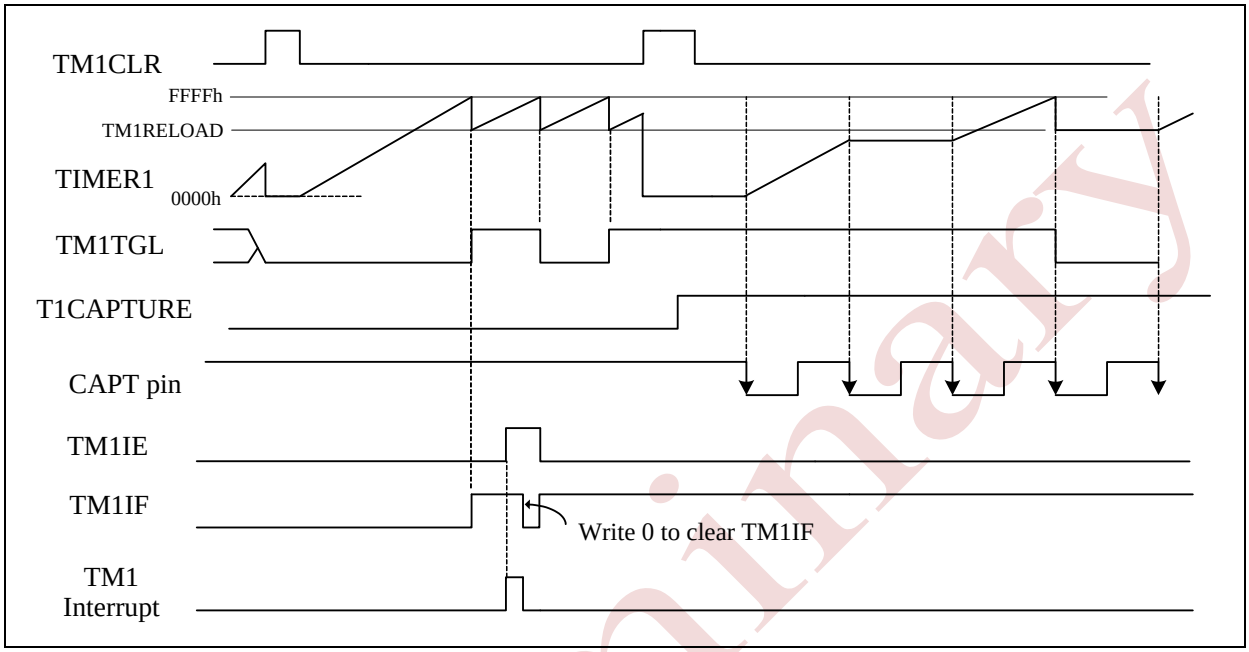
Example:

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 us.

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.



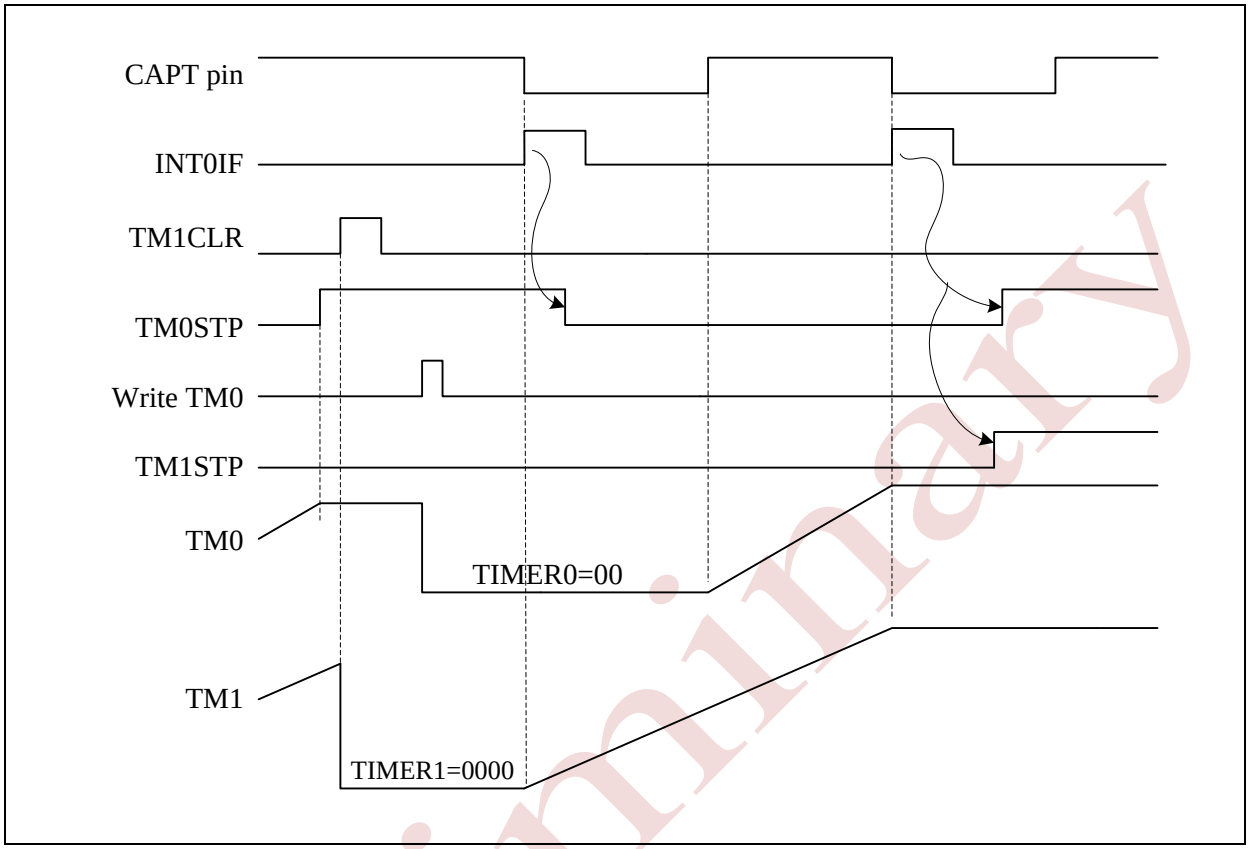
Timer1 works in Capture mode (TM0CL=0, implies CAPT falling edge)

- 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 PA0 (CAPT, INT0). Suppose that:

- TM0CKS=0, Timer0 prescaler increases per instruction cycle.
- T0CAPTURE=1, T1CAPTURE=1. Timer0 and Timer1 work in Capture mode.
- PA0 pin (CAPT pin) interrupts every falling edge. TM0CL=0, **Timer1** start/hold in turn when PA0 pin (CAPT pin) falling edge is coming. **Timer0** start counting when PA0 pin (CAPT pin) in logic '1' level, and hold the Timer0 value when PA0 pin (CAPT pin) in logic '0' level.
- Timer1 is used to measure the signal period, Timer0 is used to measure the PA0 (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 PA0 (CAPT pin) signal's period and duty cycle (TM0CL=0).

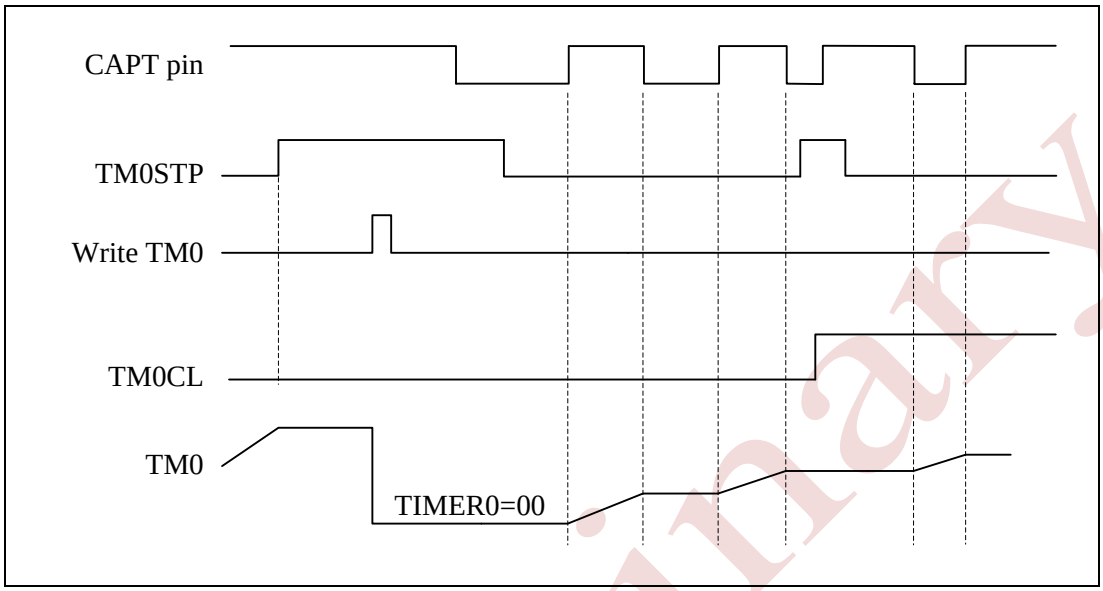


Timer0 and Timer1 are used to measure the signal on CAPT pin.

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 PA0 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 return to 0 and hold the counting value. Timer1 also stops counting and holds the value.
5. PA0 interrupt is generated again, firmware stops Timer1 and Timer0 to read the period and duty cycle.

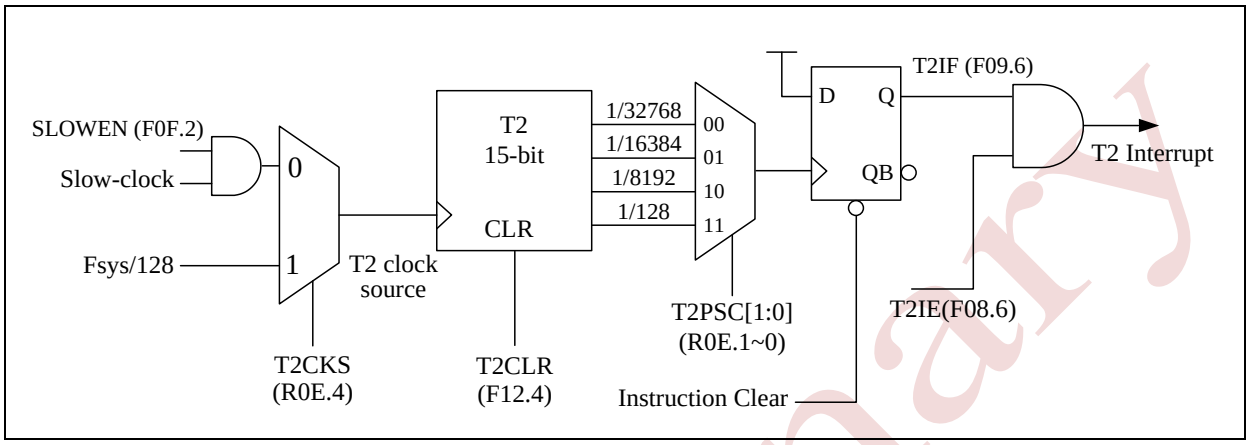
It is not necessary to use both Timer0 and Timer1. If only the duty cycle (CAPT high time) needs to be measured, there is no need to use Timer1 to measure the period. In such case, user can set the TM0CM=1 and TM1CM=0. Timer0 is counting up only when CAPT pin is '1'. Note that the internal prescaler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

4.4 T2: 15-bit Timer

The T2 is a 15-bit counter and the clock sources are from either Fsys/128 or Slow-clock. It is used to generate time base interrupt and T2 counter block clock. The T2 content cannot be read by instructions. It generates interrupt flag T2IF (F09.6) with the clock divided by 32768/16384/8192/128 depends on T2PSC[1:0] (R0E.1~0) register bits. The following figure shows the block diagram of T2.



T2 Block Diagram

Example:

[CPU running at Fast mode, Fsys=FIRC 4 MHz]

◇Example:

Setup T2 configuration.

```
; Setup T2 clock source and divider
        MOVLW  00010100B    ; Fsys=4 MHz
        MOVWR  R0E          ; T2 clock source=Fsys/128
                            ; Divided by 32768
        BSF    T2CLR        ; Stop T2 counting

; Enable T2 timer and interrupt function.
        MOVLW  10111111B    ; Clear T2 request interrupt flag
        MOVWF  INTIF
        BSF    T2IE         ; Enable T2 interrupt function
        BCF    T2CLR        ; Enable T2 counting (Default "0")
```

Example:

T2 clock source is Fsys/128=4 MHz/128=31250 Hz, T2PSC=/32768,

T2 clock frequency=31250/32768=0.95 Hz

Example:

[CPU running at Slow mode, Fsys=Slow-clock=SXT 32768 Hz]

◇Example:

Setup CPU running at Slow mode.

```
MOVLW  00000000B
MOVWF  F0F          ; Slow-clock type=SXT (Default “01”)
BSF    SLOWEN       ; Enable slow-clock
NOP
BSF    CPUCKS       ; Select Fsys=Slow-clock
BSF    FASTSTP      ; Stop Fast-clock

; Setup T2 clock source and divider
MOVLW  00000000B    ; Fsys=32K
MOVWR  R0E          ; T2 clock source=Slow-clock=32 KHz
                    ; Divided by 32768

BSF    T2CLR        ; Stop T2 counting

; Enable T2 timer and interrupt function
MOVLW  10111111B    ; Clear T2 request interrupt flag
MOVWF  INTIF
BSF    T2IE         ; Enable T2 interrupt function
BCF    T2CLR        ; Enable T2 counting (Default “0”)
```

Example:

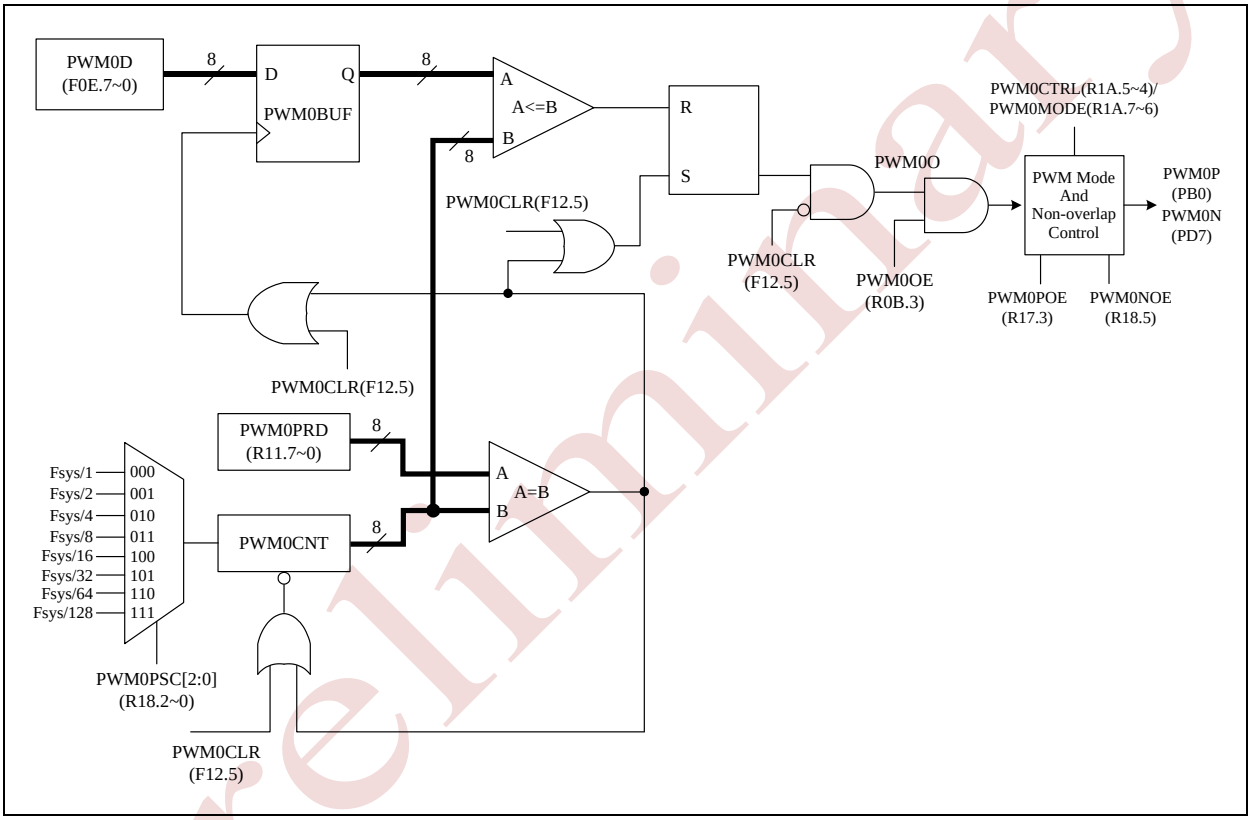
T2 clock source is Slow-clock=32768 Hz, T2PSC=/32768,

T2 frequency=32768 Hz/32768=1 Hz

4.5 PWM0: 8-bit PWM

The chip has a built-in 8-bit PWM generator. The source clock comes from Fcpuclk divided by 1, 2, 4, 8, 16, 32, 64, and 128. The PWM0 duty cycle can be changed with writing to PWM0D (F0E), writing to PWM0D (F0E) will not change the current PWM0 duty until the current PWM0 period completes. When current PWM0 period is finish, the new value of PWM0D (F0E) will be updated to the PWM0BUF.

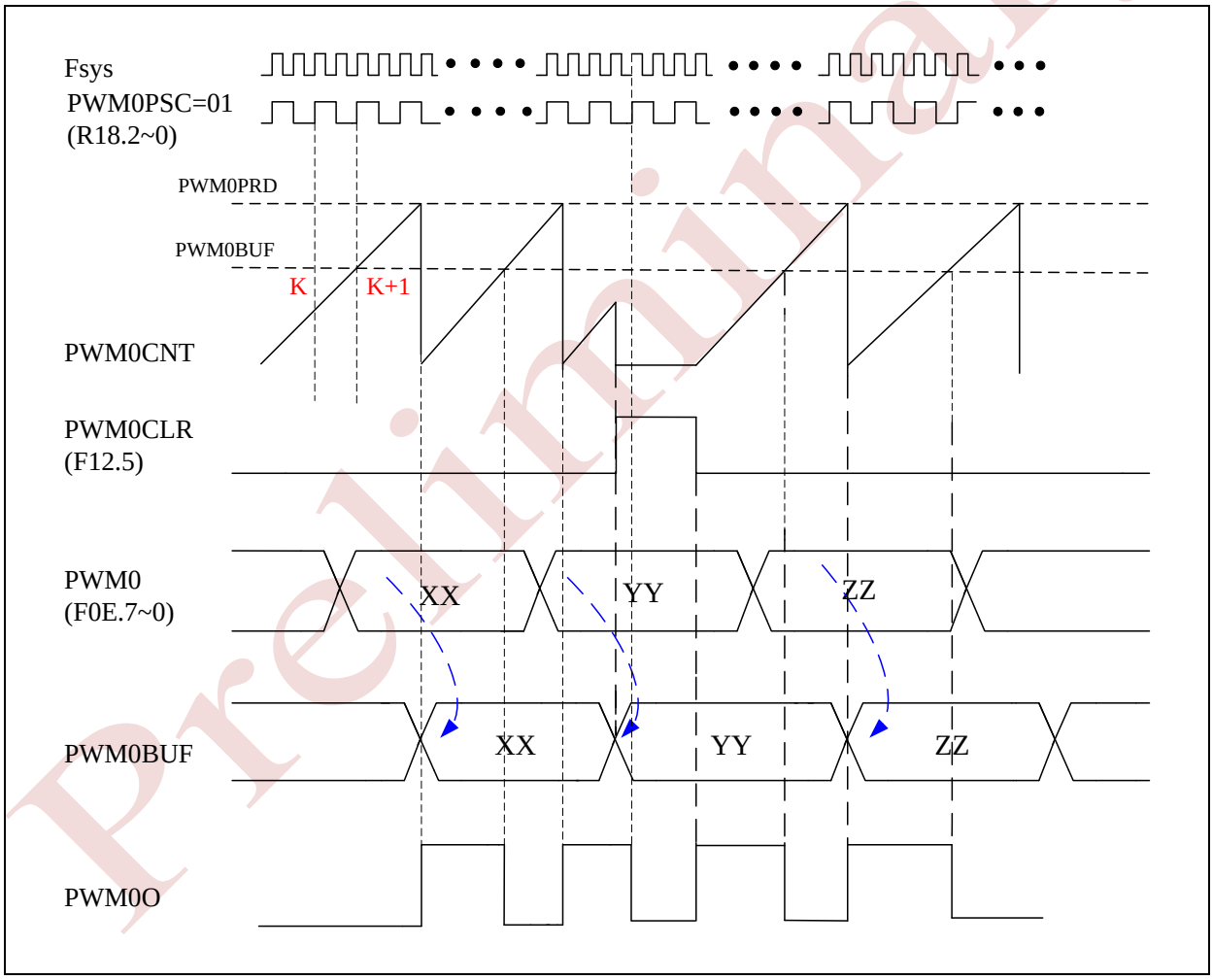
The PWM0 has a pair of differential output. PWM0P and PWM0N can be individually output to PB0 and PD7 by PWM0POE (R17.3) and PWM0NOE (R18.5) if PWM0OE (R0B.3) is set to 1. Setting the PWM0CLR (F12.5) bit will clear the PWM0 counter and load the PWM0D (F0E) to PWM0BUF, PWM0CLR (F12.5) bit must be cleared so that the PWM0 counter can count. The following figure shows the block diagram of PWM0.



PWM0 Block Diagram

The following figure shows the PWM0 waveforms. When PWM0CLR (F12.5) bit is set to '1', the PWM0 output is cleared to '0' no matter what its current status is. Once the PWM0CLR (F12.5) bit is cleared to '0', the PWM0 output is set to '1' to begin a new PWM cycle. PWM0 output will be '0' when PWM0CNT is greater than or equals to PWM0BUF. PWM0CNT keeps counting up when equals to PWM0PRD (R11), the PWM0 output is set to '1' again.

The PWM0 period can be set by writing period value to PWM0PRD (R11) register. Note that changing the PWM0PRD (R11) immediately changes the PWM0PRD (R11) values in the Figure that is different from PWM0D (F0E) which has PWM0BUF to update the duty at the end of current period. The Programmer must pay attention to the current time to change PWM0PRD (R11) by observing the following figure. There is a digital comparator that compares the PWM0CNT and PWM0PRD (R11), if PWM0CNT is larger than PWM0PRD (R11) after setting the PWM0PRD (R11), a fault long PWM cycle will be generated because PWM0CNT must count to overflow then keep counting PWM0PRD (R11) to finish the cycle.



PWM0 waveform

Example:

[CPU running at Fast mode, Fsys=4 MHz]

◇Example:

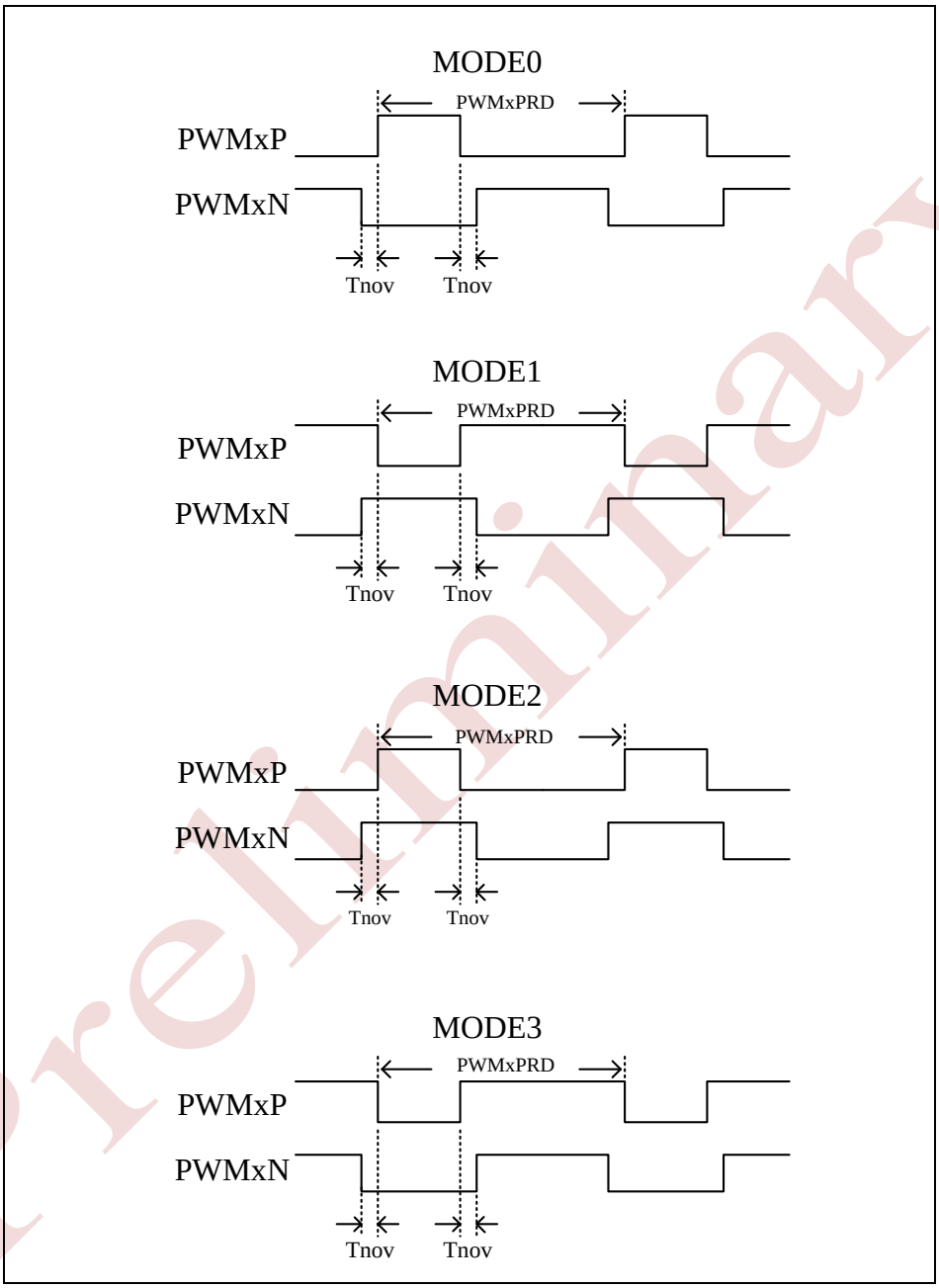
```
; Setup PWM0 clock prescaler
        MOVLW  00000010B     ; Fsys=4 MHz
        MOVWR  R18           ; PWM0 clock source Fsys/4=4 MHz/4=1 MHz

        MVOLW  00000000B
        MOVWR  R1A           ; Setup PWM0 mode

        MOVLW  FFH
        MOVWR  PWM0PRD       ; Set PWM0 period=FFH

        MOVLW  80H
        MOVWF  PWM0D         ; Set PWM0 duty=80H/FFH=50%

        MOVLW  08H
        MOVWR  R0B           ; Enable PWM0 OUT (PB0 or PD7)
        MOVLW  08H
        MOVWR  R17           ; Enable PWM0P out to PB0
        MOVLW  00100010B
        MOVWR  R18           ; Enable PWM0N out to PD7

        BCF    PWM0CLR       ; Enable PWM0 counting
```

Example:

Fsys=4 MHz, PWM0PSC=/4, PWM0PRD=FFH, PWM0D=80H,

PWM0 output frequency=4 MHz/ (PWM0PRD+1) =4 MHz/256=15.625 KHz.

PWM0 can be output via PWM0P and PWM0N with four different modes. The edges of the PWM pulse can be separated with 4 different time non-overlap clocks intervals (Tnov), 0s, 1 Fsys clock, 2 Fsys clocks, and 4 Fsys clocks which are selected by PWM0CTRL (R1A.5~4). The default output form is MODE0. The waveforms of the four output modes are shown below.



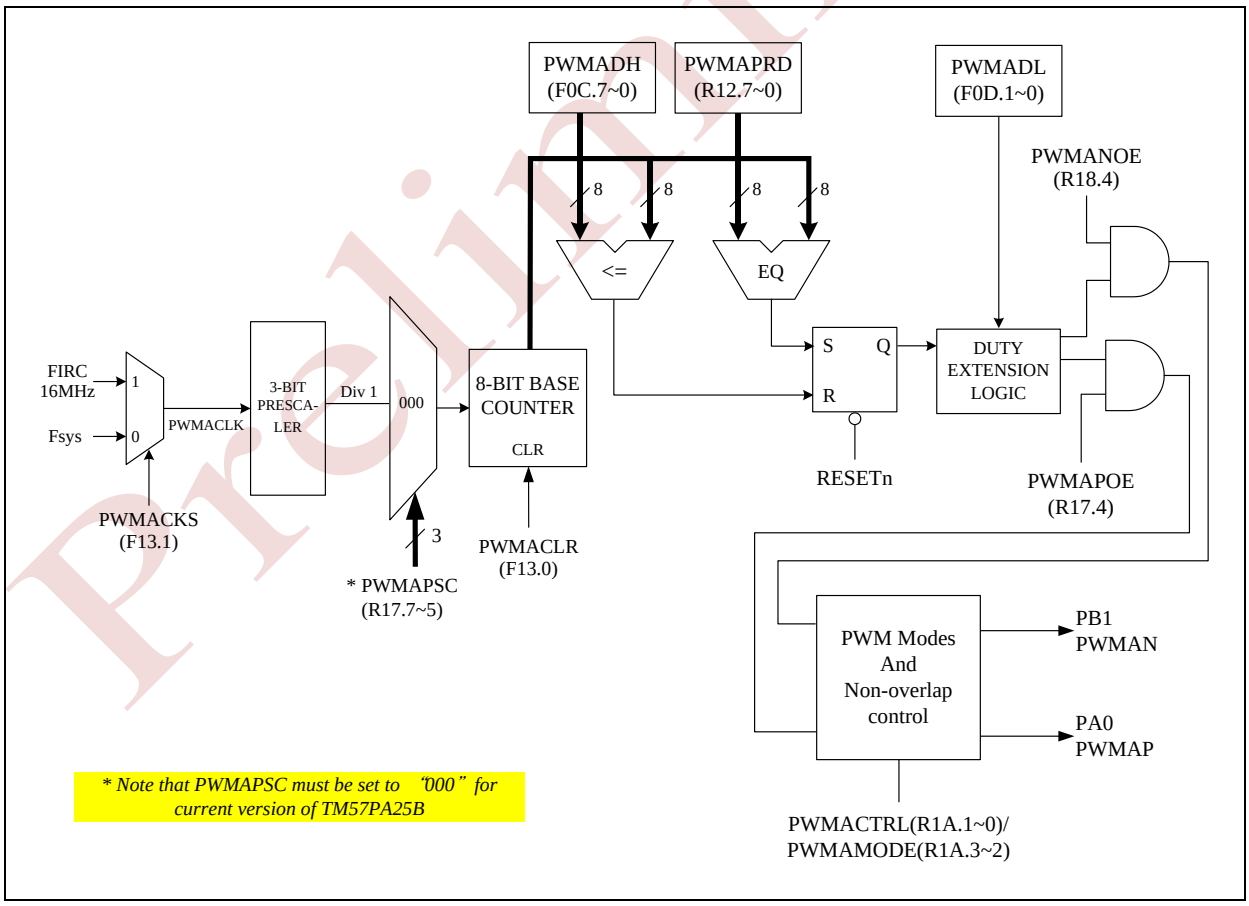
PWM0/PWMA Waveform Modes

4.6 PWMA: (8+2) bits PWM

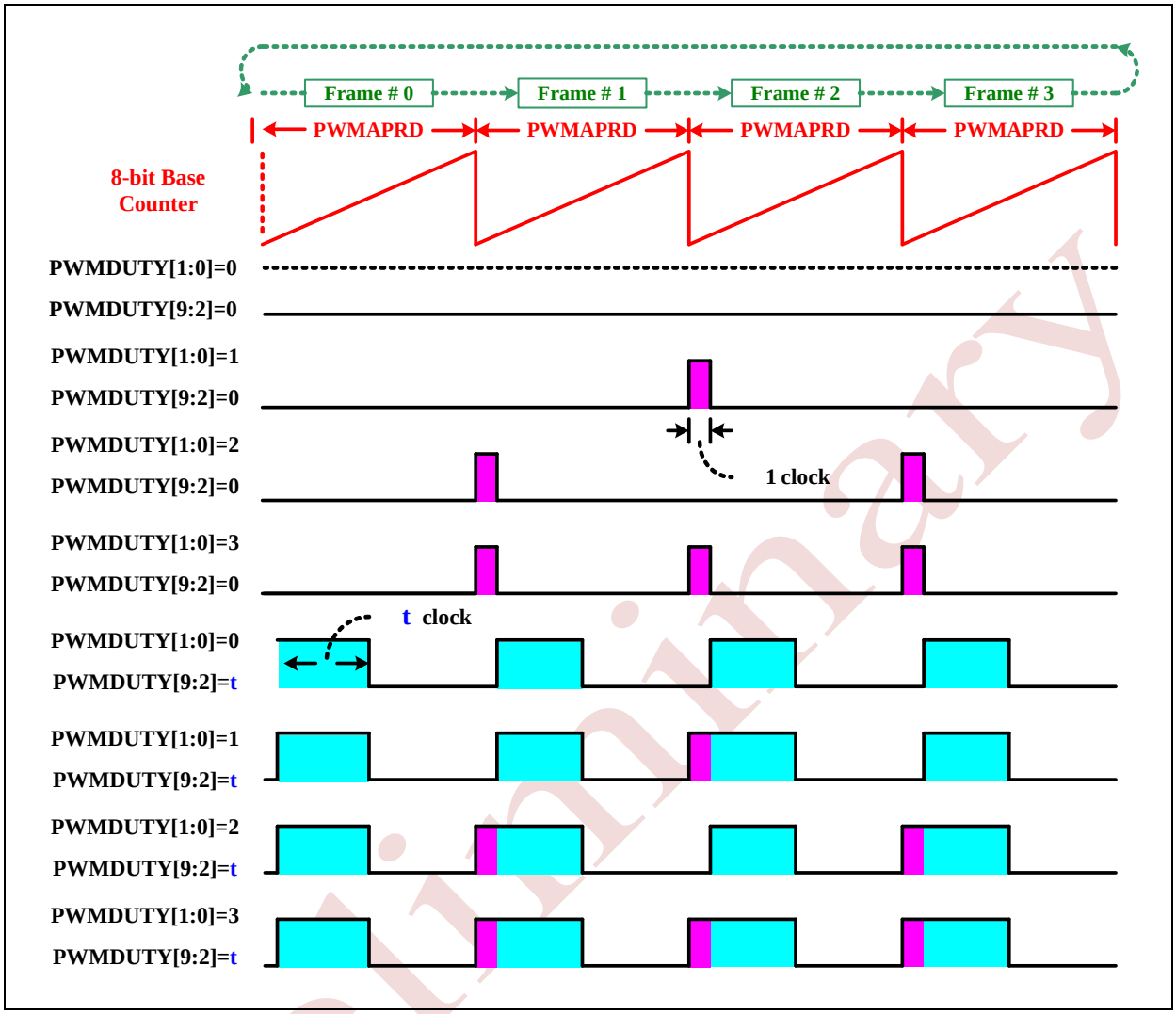
The PWM can generate various frequency waveform with 1024 duty resolution based on PWMACLK, which can select Fsys or FIRC 16 MHz, decided by PWMACKS (F13.1). A spread LSB technique allows PWMA to run its frequency at "PWMACLK divided by 256" instead of "PWMACLK divided by 1024", which means the PWM is 4 times faster than normal. The advantage of higher PWM frequency is that the post RC filter can transform the PWM signal to more stable DC voltage level. The PWM output signal reset to low level whenever the 8-bit base counter matches the 8-bit MSB of PWM duty register PWMADH (F0C.7~0). When the base counter rolls over, the 2-bit LSB of PWM duty register PWMADL (F0D.1~0) decides whether to set the PWM output signal high immediately or set it high after one clock cycle delay.

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

PWMAPSC is not be implemented in this version, user must set PWMAPSC to "000" to prevent malfunction.



PWMA Block Diagram



PWMA 8+2 Timing Diagram

Example:

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

◇Example:

```
; Setup PWMA clock prescaler
        BCF     PWMACKS     ; PWMA clock source=Fsys
        MOVLW   00000000B   ; Fsys=8 MHz
        MOVWR   R17         ;

        MVOLW   00000000B
        MOVWR   R1A         ; Setup PWMA mode

        MOVLW   80H
        MOVWR   PWMAPRD     ; Set PWMA period=80H

        MOVLW   0000B
        MOVWF   F0D         ; Set PWMADL duty=00H

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

        MOVLW   00010000B
        MOVWR   R0B         ; Enable PWMA OUT (PA0 or PB1)
        MOVLW   00010000B
        MOVWR   R17         ; Enable PWMAP out to PA0
        MOVLW   00010000B
        MOVWR   R18         ; Enable PWMAN out to PB1

        BCF     PWMACLR     ; Enable PWMA counting
```

Example:

Fsys=8 MHz, PWMACLK=4 MHz, PWMAPSC=/1, PWMAPRD=80H,

PWMADL=00H, PWMADH=20H

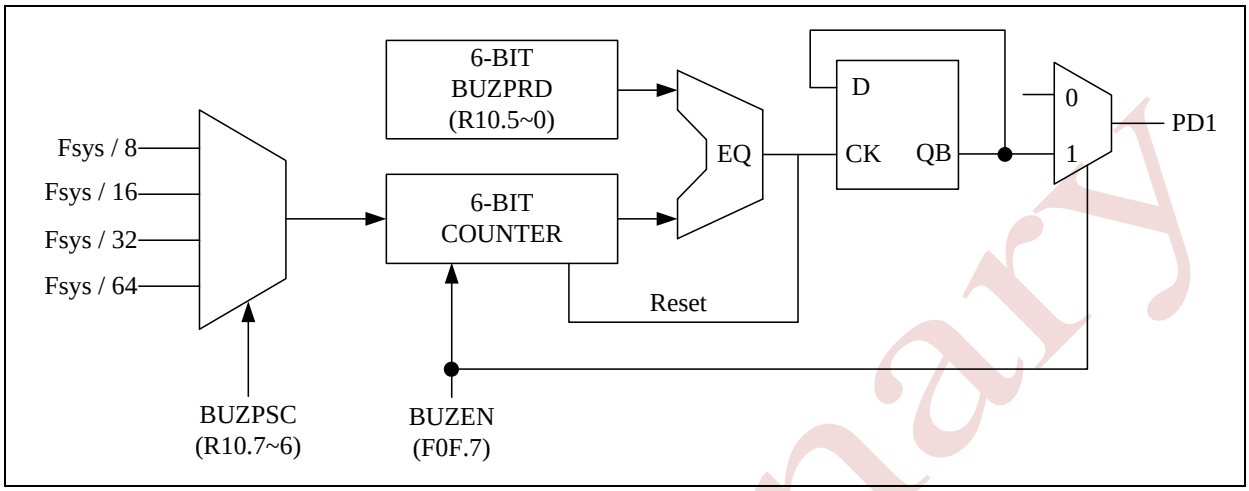
PWMA output frequency=4 MHz/1/ (PWMPROD+1) =4 MHz/1/129=31008 Hz.

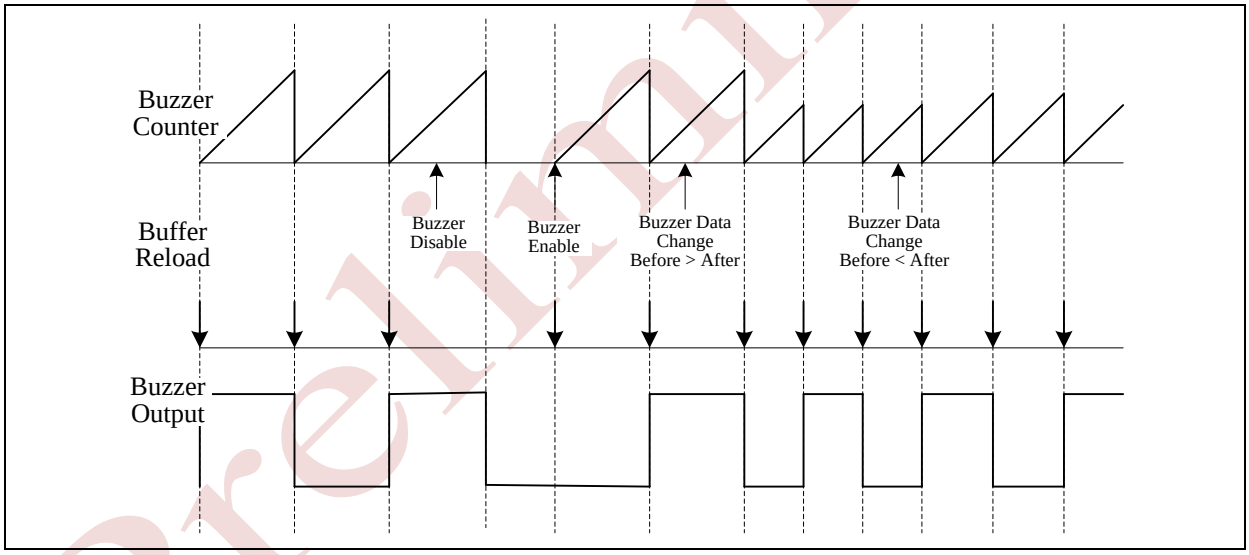
PWMAP output duty=32:129=24.8%.

PWMA can be output via PWMAP and PWMAN with four different modes. The edges of the PWM pulse can be separated with 4 different time non-overlap clocks intervals, 0s, 1 PWMACLK, 2 PWMACLKs, and 4 PWMACLKs which are selected by PWMACTRL (R1A.1~0). The default output form is MODE0. The waveforms of the four output modes are shown in the previous section "**PWM0/PWMA Waveform Modes**" diagram.

4.7 BUZZER Output

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





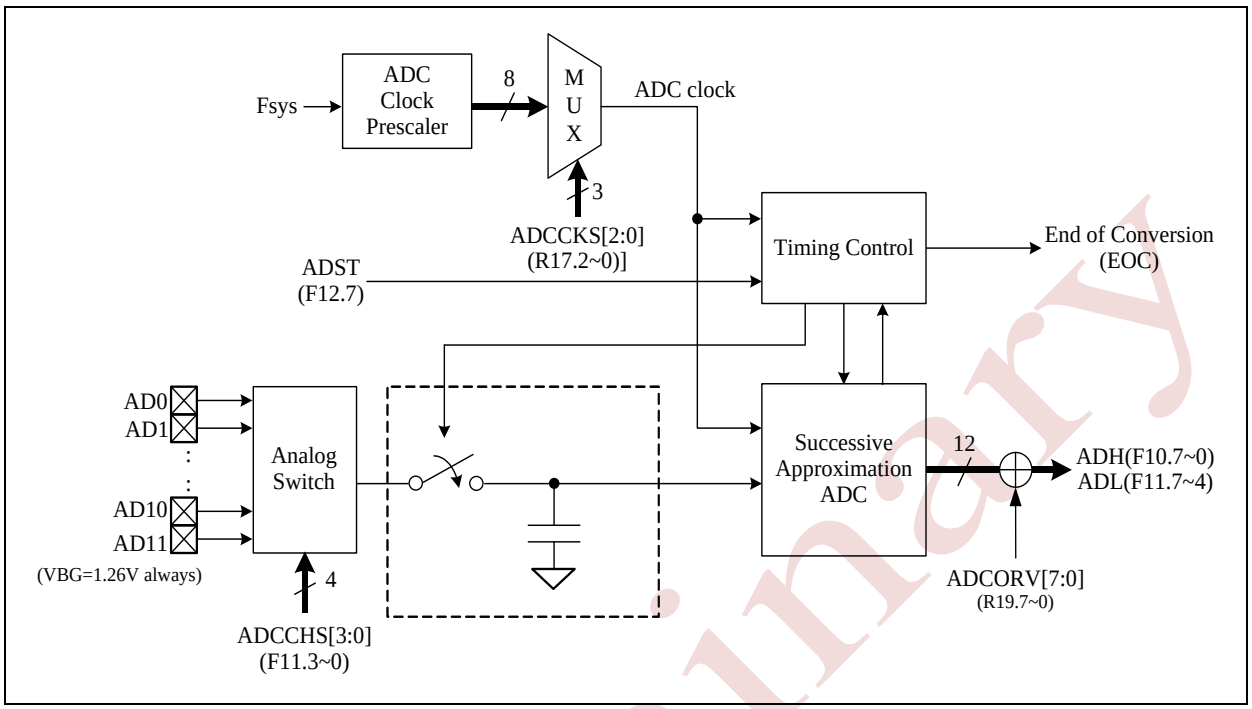
Frequency calculation is as follows. F_{BZ}= (Fsys/BUZPSC) / (BUZPRD+1) /2

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

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

```
MOVLW   10000000B
MOVWF   F0F         ; F0F.7 (BUZEN) =1, enable Buzzer counting and output to PD1
MOVLW   10001001B   ; R10.7~6 (BUZPSC) =Fsys/32
MOVWR   R10         ; R10.5~0 (BUZPRD) =9
```

4.8 Analog-to-Digital Converter and Internal Bandgap Reference Voltage



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(R17.2~0) to choose a proper ADC clock frequency, which must be less than 1 MHz. User then launches the ADC conversion by setting the ADST (F12.7) control bit. After end of conversion, H/W automatic clears the ADST (F12.7) bit. User can poll this bit to know the conversion status. The PAM (R13.7~0), PBM (R14.1~0), PDM (R15.7~0) 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.

AD11 is connected to internal bandgap reference voltage VBG, the voltage spec is 1.26V.

For TM57PA25B, it is 1.26V±5%. The ROM address 7FAh and 7F8h store the VBG voltage tested by tenx when VDD=3.6V and the address 7FBh and 7F9h store the value under VDD=5.0V. The two addresses are pre-stored RETLW xxh which value is tested by tenx. The value is obtained by A/D converter and is 12-bit wide.

7FAh stores the highest 8-bit of VBG A/D value; 7F8h stores the lowest 4-bit of VBG A/D value with high nibble equals 0000.

7FBh stores the highest 8-bit of VBG A/D value; 7F9h stores the lowest 4-bit of VBG A/D value with high nibble equals 0000.

***CAUTION**: Note that when AD11 is selected, the ADC clock must be carefully set to be under 1 MHz to ensure the ADC conversion is correct.*

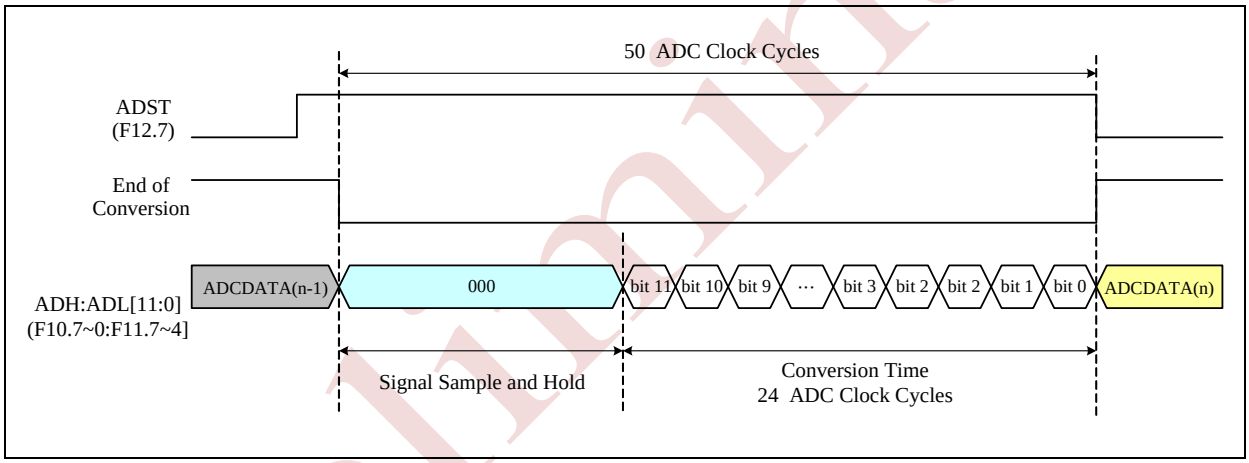
The 7F8h to 7FBh addresses will not be over-written by writer. User can assume the reasonable value to it while in development phase with an ICE (In-Circuit Emulator).

Example:

```
CALL    7FBh
MOVWF   20h             ; W=40h and F20=40h
……..

.org    7F8h
RETLW   09h             ; User code assumes any values will not affect real value
RETLW   08h             ; User code assumes any values will not affect real value
RETLW   59h             ; User code assumes any values will not affect real value
RETLW   40h             ; User code assumes any values will not affect real value
```

The above example code means the VBG value are 599h when VDD=3.6V and 408h when VDD=5.0V.

The A/D conversion timing diagram



Example:

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

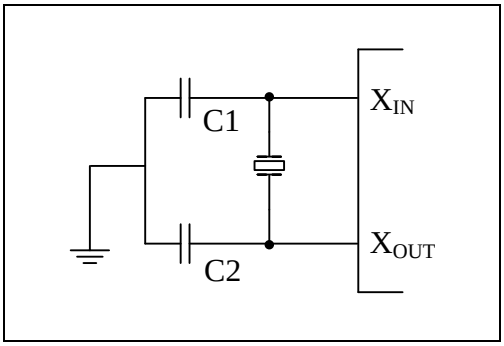
ADC clock frequency=1 MHz, ADC channel=ADC2 (PA2).

◇Example:

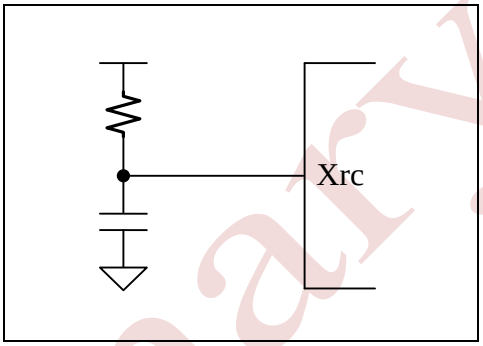
```
        MOVLW  00000101B     ; Fsys=8 MHz
        MOVWR  R17           ; ADC clock prescaler/8

        MVOLW  00000000B
        MOVWR  R18           ; Disable ADC corrected
        MOVWR  R19           ; ADC corrected value=0

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

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

        BSF    ADST          ; ADC start conversion

WAIT_ADC:
        BTFSC  ADST          ; Wait ADC conversion
        GOTO   WAIT_ADC

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

4.9 System Clock Oscillator

System clock can be operated in four different oscillation modes, which is selected by setting the CLK in the SYSCFG register. In Slow/Fast Crystal (SXT/FXT) mode, a crystal or ceramic resonator is connected to the Xin and Xout pins to establish oscillation. In external RC (XRC) mode, the external resistor and capacitor determine the oscillation frequency. In the fast internal RC (FIRC) mode, the on-chip oscillator generates 16/8/4/2 MHz system clock, which is controlled by register FIRCKS[1:0] (R0E.3~2) bits.



External Oscillator Circuit (Crystal or Ceramic)



External RC Oscillator

MEMORY MAP

F-Plane

Name	Adr	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: TM0**
TM0	01.7~0	R/W	0	Timer0 counting byte
(F02) PCL				**Function related to: PROGRAM COUNT**
PCL	02.7~0	R/W	0	Programming Counter LSB[7~0]
(F03) STATUS				**Function related to: STATUS**
GB0	03.7	R/W	0	General purpose bit 0
GB1	03.6	R/W	0	General purpose bit 1
RAMBK	03.5	R/W	0	RAM Bank Selection
TO	03.4	R	0	WDT timeout flag, cleared by PWRST, 'SLEEP' or 'CLRWDT' instruction
PD	03.3	R	0	Sleep mode flag, set by 'SLEEP', cleared by 'CLRWDT' instruction
Z	03.2	R/W	0	Zero flag
DC	03.1	R/W	0	Decimal Carry flag
C	03.0	R/W	0	Carry flag
(F04) MF04				**Function related to: RAM W/R**
GB2	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 state
PAD	05.6~0	R	-	Port A pin or "data register" state
		W	7F	Port A output data register
(F06) PBD				**Function related to: Port B**
PBD	06.1~0	R	-	Port B pin or "data register" state
		W	03	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
(F08) INTIE				**Function related to: Interrupt Enable**
ADCIE	08.7	R/W	0	ADC interrupt enable, 1=enable, 0=disable
T2IE	08.6	R/W	0	T2 interrupt enable, 1=enable, 0=disable
TM1IE	08.5	R/W	0	Timer1 interrupt enable, 1=enable, 0=disable
TM0IE	08.4	R/W	0	Timer0 interrupt enable, 1=enable, 0=disable
WKTIE	08.3	R/W	0	Wakeup Timer interrupt enable, 1=enable, 0=disable Set to 0 will clear & disable WKT timer
INT2IE	08.2	R/W	0	INT2 pin (PA7) interrupt enable, 1=enable, 0=disable
INT1IE	08.1	R/W	0	INT1 pin (PA1) interrupt enable, 1=enable, 0=disable
INT0IE	08.0	R/W	0	INT0 pin (PA6) interrupt enable, 1=enable, 0=disable

Name	Adr	R/W	Rst	Description
(F09) INTIF				**Function related to: Interrupt Flag**
ADCIF	09.7	R	-	ADC interrupt event pending flag, set by A/D conversion done
		W	0	write 0: clear this flag; write 1: no action
T2IF	09.6	R	-	T2 interrupt event pending flag, set by H/W while T2 overflows
		W	0	write 0: clear this flag; write 1: no action
TM1IF	09.5	R	-	Timer1 interrupt event pending flag, set by H/W while Timer1 overflows
		W	0	write 0: clear this flag; write 1: no action
TM0IF	09.4	R	-	Timer0 interrupt event pending flag, set by H/W while Timer0 overflows
		W	0	write 0: clear this flag; write 1: no action
WKTIF	09.3	R	-	WKT interrupt event pending flag, set by H/W while WKT time out
		W	0	write 0: clear this flag; write 1: no action
INT2IF	09.2	R	-	INT2 interrupt event pending flag, set by H/W at INT2 pin's falling edge
		W	0	write 0: clear this flag; write 1: no action
INT1IF	09.1	R	-	INT1 interrupt event pending flag, set by H/W at INT1 pin's falling edge
		W	0	write 0: clear this flag; write 1: no action
INT0IF	09.0	R	-	INT0 interrupt event pending flag, set by H/W at INT0 pin's f/r edge
		W	0	write 0: clear this flag; write 1: no action
(F0A) TM1L				**Function related to: Timer1 count**
TM1L	0a.7~0	R	-	Timer1 counting low byte
		W	0	Timer1 reload data low byte
(F0B) TM1H				**Function related to: Timer1 count**
TM1H	0b.7~0	R	-	Timer1 counting high byte
		W	0	Timer1 reload data high byte
(F0C) PWMADH				**Function related to: PWMA**
PWMADH	0c.7~0	R/W	0	PWMA duty 8-bit MSB
(F0D) MF0D				**Function related to: PWMA/PWM0/INT2**
PWM0IF	0d.3	R	-	PWM0 interrupt event pending flag, set by H/W at PWM0 period done
		W	0	write 0: clear this flag; write 1: no action
PWM0IE	0d.2	R/W	0	PWM0 interrupt enable, 1=enable, 0=disable
PWMADL	0d.1~0	R/W	0	PWMA duty 2-bit LSB
(F0E) PWM0D				**Function related to: PWM0**
PWM0D	0e.7~0	R/W	0	PWM0 duty 8-bit

Name	Adr	R/W	Rst	Description
(F0F) MF0F				**Function related to: BUZZER/CPUCLK**
BUZEN	0f.7	R/W	0	Buzzer function, 1=enable, 0=disable
SIRCKS	0f.6~5	R/W	3	SIRC clock selection (Hz) 00: 150K 01: 37K 10: 9K 11: 2K (not precisely)
FASTSTP	0f.4	R/W	0	Fast-clock Enable/Disable 0: Enable 1: Disable
CPUCKS	0f.3	R/W	0	System clock (Fsys) selection 0: Fast-clock 1: Slow-clock
SLOWEN	0f.2	R/W	0	If CPUCKS=1, this SLOWEN bit is invalid, Slow-clock keep oscillating If CPUCKS=0, Set 1 to enable Slow-clock oscillate, Clear 0 to stop Slow-clock oscillating
SLOWCKS	0f.1~0	R/W	0	Slow-clock type 00: SXT 01: SIRC (see SIRCKS[6~5]) 10: XRC 11: Reserved
(F10) ADH				**Function related to: ADC**
ADH	10.7~0	R	-	ADC output data MSB[11:4]
(F11) ADCTL				**Function related to: ADC**
ADL	11.7~4	R	-	ADC output data LSB [3:0]
ADCHS	11.3~0	R/W	0	ADC channel select 0000: ADC0 (PA6) 0110: ADC6 (PA0) 0001: ADC1 (PA1) 0111: ADC7 (PB0) 0010: ADC2 (PA2) 1000: ADC8 (PD6) 0011: ADC3 (PB1) 1001: ADC9 (PD5) 0100: ADC4 (PD7) 1010: ADC10 (PD4) 0101: ADC5 (PA5) 1011: VBG (Built-in)
(F12) MF12				**Function related to: ADC/CPUCLK/PWM0/T2/TM1/TM0**
ADST	12.7	R/W	0	ADC start bit. 0 :H/W clear after end of conversion 1: ADC start conversion,
KICKE	12.6	R/W	1	Speedup SXT warmup, Clear 0 to save power after SXT oscillate stable
PWM0CLR	12.5	R/W	1	PWM0 counter clear 0:Release 1:Clear and hold
T2CLR	12.4	R/W	1	T2 counter clear 0:Release 1:Clear and hold ,
TM1SET	12.3	R/W	0	Timer1 counter set 0:Release 1:Set to FFFFh and hold
TM1CLR	12.2	R/W	0	Timer1 counter clear 0:Release 1:Clear to 0000H and hold
TM1STP	12.1	R/W	0	Timer1 counter stop 0:Release 1:Stop counting,
TM0STP	12.0	R/W	0	Timer0 counter stop 0:Release 1:Stop counting,
(F13) MF13				**Function related to: VBG REF/PWMACKS/PWMACLR**
VBGEN	13.2	R/W	0	VBG reference 0:Disable 1:Enable
PWMACKS	13.1	R/W	0	PWMA clock source 0:Fsys 1:FIRC 16M
PWMACLR	13.0	R/W	1	PWMA counter 0: Release 1:Clear and hold.
User Data Memory				
SRAM	20~2f	R/W	-	RAM Common Area (16 bytes)
	30~7F	R/W	-	RAM BANK0 area (RAMBK=0, 80 bytes)
	30~7F	R/W	-	RAM BANK1 area (RAMBK=1, 80 bytes)

R-Plane

Name	Adr	R/W	Rst	Description
(R02) TM0CTL				**Function related to: TM0**
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 Mode (i.e. 'SLEEP' instruction)
(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	06.1~0	W	0	Each bit controls its corresponding pin, if the bit is 0: the pin is open-drain output or Schmitt-trigger input 1: the pin is CMOS push-pull output
(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	Adr	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/XRC/PWM/T0OUT output 1: the pin pull up resistor is disabled
(R09) PBPUN				**Function related to: Port B**
PBPUN	09.1~0	W	03	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) 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. pins are working for PWM/TM1OUT/TCOUT output 1: the pin pull up resistor is disabled
(R0B) MR0B				**Function related to: PWMA/PWM0/TM0/WKT**
PWMAOE	0b.4	W	0	PWMA output to PA0/PB1 pin enable, see R-Plane R17.4 and R18.4
PWM0OE	0b.3	W	0	PWM0 output to PB0/PD7 pin enable, see R-Plane R17.3 and R18.5
TM0OE	0b.2	W	0	Timer0 match toggle out to PA5 pin enable
WKTPSC	0b.1~0	W	11	WKT period VDD=5V / VDD=3V 00: 0.9 ms / 1.2 ms 01: 1.9 ms, / 2.4 ms 10: 30 ms, / 38 ms 11: 122 ms / 152 ms
(R0C) MR0C				**Function related to: Instruction Cycle Output**
TCOPSC	0c7~6	W	0	Instruction Cycle output prescaler. 0:Fsys/2, 1:Fsys/4, 2:Fsys/8, 3:Fsys/16
TCOE	0c.5	W	0	Enable Instruction Cycle output to PD6 pin (TCOUT)
	0c.4	-	-	Reserved
INT1EDG	0c.3	W	0	0: INT1 pin falling edge to trigger interrupt event 1: INT1 pin rising edge to trigger interrupt event
TM1OE	0c.2	W	0	enable Timer1 overflow toggle output to PD0 pin (TM1OUT)
TM1CM	0c.1	W	0	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
TM1PSC	0c.0	W	0	Timer1 prescaler. 1: Fsys 0:Instruction cycle
(R0D) RESERVED				**tenx reserved**
Reserved	0d.7~0	W	0F	*DO NOT write this register!! Writing this register may cause system error!*

Name	Adr	R/W	Rst	Description		
(R0E) MR0E				**Function related to: WDT/T2/FIRC**		
WDTPSC	0e.7~6	W	01	WDT time out.		
				VDD=5V		VDD=3V
				00	120 ms	150 ms
				01	240 ms	300 ms
				10	950 ms	1180 ms
				11	1920 ms	2340 ms
WDTSTP	0e.5	W	0	WDT disable in IDLE/STOP mode, If WDTE=0, this WDTSTP bit is invalid		
				1	stop counting	
				0	always counting	
T2CKS	0e.4	W	0	T2 clock source. 1: Fsys/128 0: Slow-clock		
FIRCKS	0e.3~2	W	01	FIRC clock select. 00: 2 MHz 01: 4 MHz 10: 8 MHz 11: 16 MHz		
T2PSC	0e.1~0	W	00	T2 prescaler. T2 clock source divided by - 00: 32768 01: 16384 10: 8192 11: 128		
(R0F) RESERVED				**tenx reserved**		
Reserved	0f.7~0	W	0	*DO NOT write this register!! Writing this register may cause system error!!*		
(R10) BUZCTL				**Function related to: BUZZER**		
BUZPSC	10.7~6	W	0	Fsys divided by 00: /8 01: /16 10: /32 11: /64		
BUZPRD	10.5~0	W	0	Buzzer Period		
(R11) PWM0PRD				**Function related to: PWM0**		
PWM0PRD	11.7~0	W	FF	PWM0 period		
(R12) PWMAPRD				**Function related to: PWMA**		
PWMAPRD	12.7~0	W	FF	PWMA period		
(R13) PAM				**Function related to: Port A/ADC**		
PAM	13.7~0	W	FF	Each bit controls its corresponding pin 0 : disable I/O digital input to save power when ADC channels are selected 1 : enable I/O digital input		
(R14) PBM				**Function related to: Port B/ADC**		
PBM	14.1~0	W	03	Each bit controls its corresponding pin 0 : disable I/O digital input to save power when ADC channels are selected 1 : enable I/O digital input		
(R15) PDM				**Function related to: Port/ADC**		
PDM	15.7~0	W	FF	Each bit controls its corresponding pin 0 : disable I/O digital input to save power when ADC channels are selected 1 : enable I/O digital input		

Name	Adr	R/W	Rst	Description
(R16) PBWKEN				**Function related to: Port B/WAKE UP**
PBWKEN	16.1~0	W	0	1: PB1~0 wake up enable 0:disable
(R17) MR17				**Function related to: ADC/PWM0/PWMA**
PWMAPSC	17.7~5	W	0	PWMA clock source: User code must set these 3 bits to '000' to prevent malfunction of PWMA.
PWMAPOE	17.4	W	0	1: enable PWMAP output to PA0, 0: PA0 function
PWM0POE	17.3	W	0	1: enable PWM0P output to PB0, 0: PB0 function
ADCKS	17.2~0	W	0	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
(R18) MR18				**Function related to: PWM0/ADC/PWMA/PWM0**
PWM0NOE	18.5	W	0	1: enable PWM0N output to PD7, 0: PD7 function
PWMANOE	18.4	W	0	1: enable PWMAN output to PB1, 0: PB1 function
ADCOR	18.3	W	0	1: ADC value is corrected, 0: ADC value is not be corrected
PWM0PSC	18.2~0	W	0	PWM0 clock source prescaler 000: Fsys 001: Fsys/2 010: Fsys/4 011: Fsys/8 100: Fsys/16 101: Fsys/32 110: Fsys/64 111: Fsys/128
(R19) ADCORV				**Function related to: ADC**
ADCORV	19.7~0	W	0	When ADCCOR=1, the ADC value will be ADCQ+bit[6:4] when ADCQ < 1024, or ADCQ-bit[2:0] when ADC > 2048

Name	Adr	R/W	Rst	Description
(R1A) MR1A				**Function related to: PWMA/PWM0**
PWM0MODE	1a.7~6	W	0	PWM0 differential output mode 00: Mode 0, 01: Mode 1, 10: Mode 2, 11: Mode 3
PWM0CTRL	1a.5~4	W	0	00: original PWM0, 01: non-overlap 1 Fsys clock 10: non-overlap 2 Fsys clocks 11: non-overlap 4 Fsys clocks
PWMAMODE	1a.3~2	W	0	PWMA differential output mode 00: Mode 0, 01: Mode 1, 10: Mode 2, 11: Mode 3
PWMACTRL	1a.1~0	W	0	00: original PWMA, 01: non-overlap 1 PWMACLK 10: non-overlap 2 PWMACLKs 11: non-overlap 4 PWMACLKs
(R1B) MR1B				**Function related to: POWER NOISE**
RESERVED	1b.7~1	W	0	TENX reserved bits. Make sure all bits are zero. DO NOT write 1s!!
EFTEN1	1b.0	W	0	Enable EFT enhance operation mode. 1: Enable 0:Disable

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
DC	Decimal Carry Flag
PC	Program Counter
TOS	Top Of Stack
GIE	Global Interrupt Enable Flag (i-Flag)
[]	Option Field
( )	Contents
.	Bit Field
B	Before
A	After
←	Assign direction

Mnemonic		Op Code	Cycle	Flag Affect	Description
Byte-Oriented File Register Instruction					
ADDWF	f,d	00 0111 dfff ffff	1	C, DC, Z	Add W and "f"
ANDWF	f,d	00 0101 dfff ffff	1	Z	AND W with "f"
CLRF	f	00 0001 1fff ffff	1	Z	Clear "f"
CLRW		00 0001 0100 0000	1	Z	Clear W
COMF	f,d	00 1001 dfff ffff	1	Z	Complement "f"
DECF	f,d	00 0011 dfff ffff	1	Z	Decrement "f"
DECFSZ	f,d	00 1011 dfff ffff	1 or 2	-	Decrement "f", skip if zero
INCF	f,d	00 1010 dfff ffff	1	Z	Increment "f"
INCFSZ	f,d	00 1111 dfff ffff	1 or 2	-	Increment "f", skip if zero
IORWF	f,d	00 0100 dfff ffff	1	Z	OR W with "f"
MOVFW	f	00 1000 0fff ffff	1	-	Move "f" to W
MOVWF	f	00 0000 1fff ffff	1	-	Move W to "f"
MOVWR	r	00 0000 00rr rrrr	1	-	Move W to "r"
RLF	f,d	00 1101 dfff ffff	1	C	Rotate left "f" through carry
RRF	f,d	00 1100 dfff ffff	1	C	Rotate right "f" through carry
SUBWF	f,d	00 0010 dfff ffff	1	C, DC, Z	Subtract W from "f"
SWAPF	f,d	00 1110 dfff ffff	1	-	Swap nibbles in "f"
TESTZ	f	00 1000 1fff ffff	1	Z	Test if "f" is zero
XORWF	f,d	00 0110 dfff ffff	1	Z	XOR W with "f"
Bit-Oriented File Register Instruction					
BCF	f,b	01 000b bbff ffff	1	-	Clear "b" bit of "f"
BSF	f,b	01 001b bbff ffff	1	-	Set "b" bit of "f"
BTFSC	f,b	01 010b bbff ffff	1 or 2	-	Test "b" bit of "f", skip if clear
BTFSS	f,b	01 011b bbff ffff	1 or 2	-	Test "b" bit of "f", skip if set
Literal and Control Instruction					
ADDLW	k	01 1100 kkkk kkkk	1	C, DC, Z	Add Literal "k" and W
ANDLW	k	01 1011 kkkk kkkk	1	Z	AND Literal "k" with W
CALL	k	10 kkkk kkkk kkkk	2	-	Call subroutine "k"
CLRWDT		00 0000 0000 0100	1	TO, PD	Clear Watch Dog Timer
GOTO	k	11 kkkk kkkk kkkk	2	-	Jump to branch "k"
IORLW	k	01 1010 kkkk kkkk	1	Z	OR Literal "k" with W
MOVLW	k	01 1001 kkkk kkkk	1	-	Move Literal "k" to W
NOP		00 0000 0000 0000	1	-	No operation
RET		00 0000 0100 0000	2	-	Return from subroutine
RETI		00 0000 0110 0000	2	-	Return from interrupt
RETLW	k	01 1000 kkkk kkkk	2	-	Return with Literal in W
SLEEP		00 0000 0000 0011	1	TO, PD	Go into standby mode, Clock oscillation stops
XORLW	k	01 1111 kkkk kkkk	1	Z	XOR Literal "k" with W

ADDLW — Add Literal "k" and W

Syntax	ADDLW k
Operands	k : 00h ~ FFh
Operation	(W) ← (W) + k
Status Affected	C, DC, Z
OP-Code	01 1100 kkkk kkkk
Description	The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.
Cycle	1
Example	ADDLW 0x15 B : W = 0x10 A : W = 0x25

ADDWF — Add W and "f"

Syntax	ADDWF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	(destination) ← (W) + (f)
Status Affected	C, DC, Z
OP-Code	00 0111 dfff ffff
Description	Add the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Cycle	1
Example	ADDWF FSR, 0 B : W = 0x17, FSR = 0xC2 A : W = 0xD9, FSR = 0xC2

ANDLW — Logical AND Literal "k" with W

Syntax	ANDLW k
Operands	k : 00h ~ FFh
Operation	(W) ← (W) AND 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 Timer ← 00h	
Status Affected	TO, PD	
OP-Code	00 0000 0000 0100	
Description	CLRWDT instruction clears the Watchdog Timer	
Cycle	1	
Example	CLRWDT	B : WDT counter = ? A : WDT counter = 0x00

COMF — Complement "f"

Syntax	COMF f [,d]
Operands	f : 00h ~ 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 : TABLE ADDWF PCL, 1 RETLW k1 RETLW k2 : RETLW kn
	B : W = 0x07 A : W = value of k8

RLF — Rotate Left "f" through Carry

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

RRF — Rotate Right "f" through Carry

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

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

SUBWF	Subtract W from "f"	
Syntax	SUBWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(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 REG1, 0	B : REG1 = 0xA5 A : REG1 = 0xA5, W = 0x5A

TESTZ	Test if "f" is zero	
Syntax	TESTZ f	
Operands	f : 00h ~ 7Fh	
Operation	Set Z flag if (f) is 0	
Status Affected	Z	
OP-Code	00 1000 1fff ffff	
Description	If the content of register 'f' is 0, Zero flag is set to 1.	
Cycle	1	
Example	TESTZ REG1	B : REG1 = 0, Z = ? A : REG1 = 0, Z = 1

XORLW	Exclusive OR Literal with W	
Syntax	XORLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) XOR k	
Status Affected	Z	
OP-Code	01 1111 kkkk kkkk	
Description	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.	
Cycle	1	
Example	XORLW 0xAF	B : W = 0xB5 A : W = 0x1A

XORWF	Exclusive OR W with "f"	
Syntax	XORWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (W) XOR (f)	
Status Affected	Z	
OP-Code	00 0110 dfff ffff	
Description	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	XORWF REG, 1	B : REG = 0xAF, W = 0xB5 A : REG = 0x1A, W = 0xB5

ELECTRICAL CHARACTERISTICS

1. Absolute Maximum Ratings (T_A=25℃)

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}=2.0 V to 5.5 V unless otherwise specified)

Parameter	Sym	Conditions		Min	Typ	Max	Unit
Operating Voltage	V_{DD}	FAST mode, 25℃, F_{OSC}=24 MHz		4.0	–	5.5	V
		FAST mode, 25℃, F_{OSC}=20 MHz		3.4	–	5.5	
		FAST mode, 25℃, F_{OSC}=16 MHz		3.1	–	5.5	
		FAST mode, 25℃, F_{OSC}=12MHz		2.7	–	5.5	
		FAST mode, 25℃, F_{OSC}=8 MHz		2.4	–	5.5	
		FAST mode, 25℃, F_{OSC}=4 MHz		2.2	–	5.5	
Input High Voltage	V_{IH}	All Input, except PA7	V_{DD}=5 V	0.6 V_{DD}	–	V_{DD}	V
			V_{DD}=3 V	0.6 V_{DD}	–	V_{DD}	V
		PA7	V_{DD}=5 V	0.7 V_{DD}	–	V_{DD}	V
			V_{DD}=3 V	0.7 V_{DD}	–	V_{DD}	V
Input Low Voltage	V_{IL}	All Input, except PA7	V_{DD}=5 V	0	–	0.2 V_{DD}	V
			V_{DD}=3 V	0	–	0.2 V_{DD}	V
		PA7	V_{DD}=5 V	0	–	0.4 V_{DD}	V
			V_{DD}=3 V	0	–	0.3 V_{DD}	V
Output High Current	I_{OH}	All Output	V_{DD}=5 V, V_{OH}=4.5 V	3	6	–	mA
			V_{DD}=3 V, V_{OH}=2.7 V	1.5	3	–	
Output Low Current	I_{OL}	All Output	V_{DD}=5 V, V_{OL}=0.5 V	10	20	–	mA
			V_{DD}=3 V, V_{OL}=0.3 V	4.5	9	–	
PWMAP/PWMAN High Current	I_{OHD}	PA0, PB1	V_{DD}=5 V, V_{OH}=4.5 V	6	11		mA
			V_{DD}=3 V, V_{OH}=2.7 V	3	5		
PWMAP/PWMAN Low Current	I_{OHS}	PA0, PB1	V_{DD}=5 V, V_{OL}=0.5 V	30	40		mA
			V_{DD}=3 V, V_{OL}=0.3 V	10	18		
Input Leakage Current (pin high)	I_{ILH}	All Input	V_{IN}=V_{DD}	–	–	1	μA
Input Leakage Current (pin low)	I_{ILL}	All Input	V_{IN}=0 V	–	–	-1	μA
Supply Current	I_{DD}	FAST mode, LVR enable, WKT enable No load All I/O pins with internal pull-up	V_{DD}=5 V, FXT=16 MHz		5.0	6.5	mA
			V_{DD}=3 V, FXT=16 MHz		2.3	3.0	
			V_{DD}=5 V, FXT=8 MHz		2.9	3.7	
			V_{DD}=3 V, FXT=8 MHz		1.5	1.9	
			V_{DD}=5 V, FXT=4 MHz		1.7	2.2	
			V_{DD}=3 V, FXT=4 MHz		1.0	1.3	
			V_{DD}=5 V, FIRC=16 MHz		4.5	5.8	
			V_{DD}=3 V, FIRC=16 MHz		2.2	2.8	
			V_{DD}=5 V, FIRC=8 MHz		2.6	3.4	

Parameter	Sym	Conditions		Min	Typ	Max	Unit
Supply Current	I_{DD}	FAST mode, LVR enable, WKT enable No load All I/O pins with internal pull-up	V_{DD}=3 V, FIRC=8 MHz		1.3	1.7	mA
			V_{DD}=5 V, FIRC=4 MHz		1.5	2.0	
			V_{DD}=3 V, FIRC=4 MHz		0.8	1.0	
			V_{DD}=5 V, FIRC=2 MHz		1	1.3	
			V_{DD}=3 V, FIRC=2 MHz		0.5	0.65	
Supply Current	I_{DD}	SLOW mode, LVR enable, WKT enable No loads, All I/O pins with internal pull-up.	V_{DD}=5 V, SXT=32 KHz		80	100	µA
			V_{DD}=3 V, SXT=32 KHz		14	18	
			V_{DD}=5 V, SIRC=32KHz	–	26	33	
			V_{DD}=3 V, SIRC=32KHz	–	8	10	
			V_{DD}=5 V, SIRC=2KHz	–	18	23	
			V_{DD}=3 V, SIRC=2KHz	–	5	7	
		IDLE mode, LVR enable, No loads, All I/O pins with internal pull-up.	V_{DD}=5 V, SXT=32 KHz	–	57	74	
			V_{DD}=3 V, SXT=32 KHz	–	6	8	
			V_{DD}=5 V, SIRC=32KHz	–	10	13	
			V_{DD}=3 V, SIRC=32KHz	–	3	4	
		IDLE mode, LVR disable No loads, All I/O pins with internal pull-up.	V_{DD}=5 V, SXT= 32 KHz	–	56.5	73	
			V_{DD}=3 V, SXT=32 KHz	–	5.5	7	
			V_{DD}=5 V, SIRC=32KHz	–	9	12	
			V_{DD}=3 V, SIRC=32KHz	–	2	3	µA
		STOP mode, LVR enable	V_{DD}=5 V	–	1	2	
			V_{DD}=3 V	–	0.5	1	
		STOP mode, LVR disable	V_{DD}=5 V	–	–	0.1	
			V_{DD}=3 V	–	–	0.1	
Pull-Up Resistor	R_P	V_{IN}=0 V Ports A/B/D	V_{DD}=5 V V_{DD}=3 V	–	60 120	–	KΩ
		V_{IN}=0 V PA7	V_{DD}=5 V V_{DD}=3 V	–	60 133	–	KΩ

3. Clock Timing (T_A= -40℃ to +85℃)

Parameter	Condition			Min	Typ	Max	Unit
External RC Frequency (*)	V_{DD}=3V	R=4.7K	C=20 pF	2.0	2.8	3.6	MHz
		R=10K	C=100 pF	0.6	0.8	1.0	
		R=100K	C=300 pF	0.04	0.05	0.06	
	V_{DD}=5V	R=4.7K	C=20 pF	2.3	3.3	4.3	
		R=10K	C=100 pF	0.47	0.67	0.87	
		R=100K	C=300 pF	0.02	0.03	0.04	
FIRC Frequency (**)	25℃, V_{DD}=3 ~ 5.5 V (±3%)			-3%	4	+3%	
	25℃, V_{DD}=2.5 ~ 3 V (±5%)			-5%	4	+5%	
	-40℃ ~ 85℃, V_{DD}=2.5 ~ 5.5 V			-7%	4	+7%	

(*) Note that the values of the passive components vary over 5% in general cases.

(**) FIRC frequency can be selected to 2 MHz, 4 MHz, 8 MHz, and 16 MHz.

4. Reset Timing Characteristics (T_A=25℃)

Parameter	Conditions	Min	Typ	Max	Unit
RESET Input Low width	Input V_{DD}=5 V ±10%	3	–	–	μs
WDT time	V_{DD}=5 V, WDTPSC=00	110	150	190	ms
	V_{DD}=3 V, WDTPSC=00	80	120	160	ms
WKT time	V_{DD}=5 V, WKTPSC=00	0.7	0.9	1.2	ms
	V_{DD}=3 V, WKTPSC=00	0.9	1.2	1.6	ms
CPU start up time	V_{DD}=5 V	–	14	–	ms

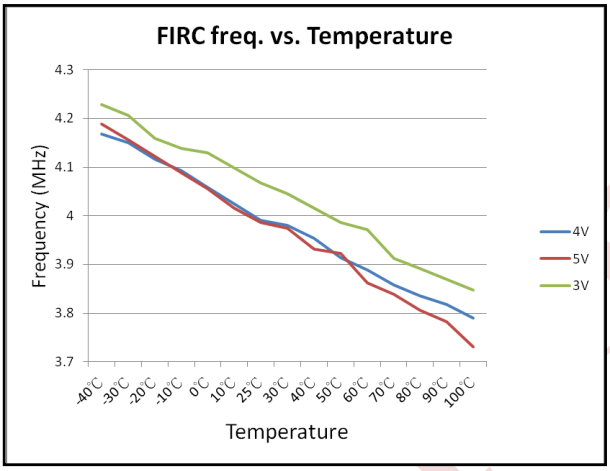
5. LVR Circuit and VBG (Bandgap Reference Voltage) Characteristics (T_A=25℃)

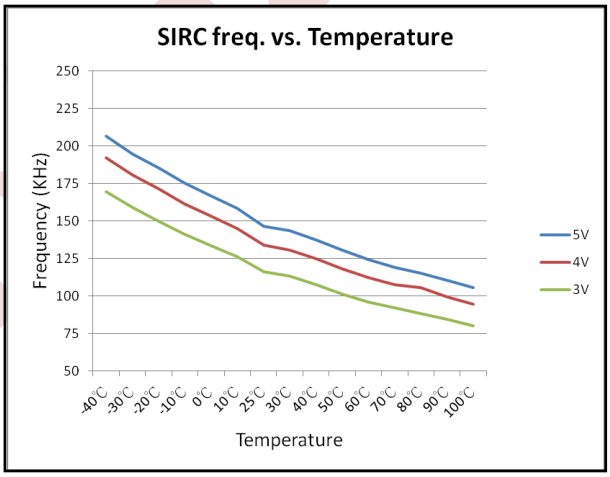
Parameter	Symbol	Min	Typ	Max	Unit
LVR Reference Voltage	V_{LVR}	1.90	1.95	2.00	V
		2.80	2.85	2.90	
LVR Hysteresis Voltage	V_{HYST}	–	±0.1	–	V
Low Voltage Detection time	t_{LVR}	100	–	–	μs
Bandgap Ref. Voltage	V_{BG}	1.20	1.26	1.32	V

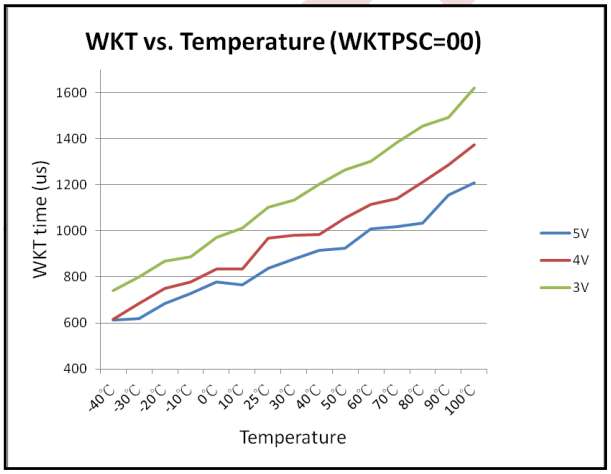
6. ADC Electrical Characteristics (T_A=25℃, VDD=2.0V to 5.5V, VSS=0V)

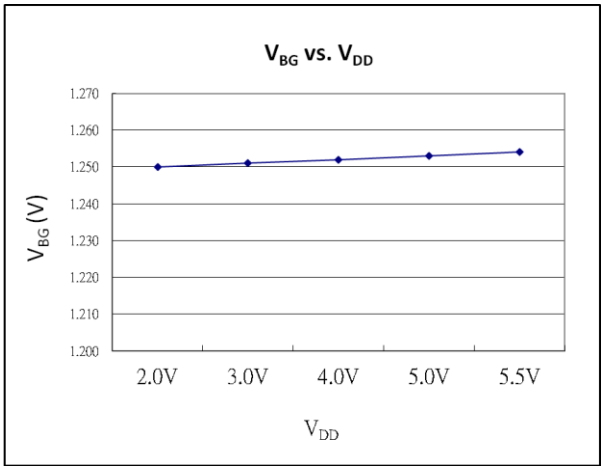
Parameter	Conditions	Min	Typ	Max	Units
Total Accuracy	V_{DD}=5.12V, V_{SS}=0V	–	±2.5	±5	LSB
Integral Non-Linearity		–	±3.2	±6	
Max Input Clock (f_{ADC})	–	–	–	2	MHz
Conversion Time (for ADC0 ~ ADC10)	$f_{ADC(max)}$ =2 MHz	–	25	–	μs
Conversion Time (for V_{BG})	$f_{ADC(max)}$ =1MHz		50		μs
Input Voltage	–	V_{SS}	–	V_{DD}	V

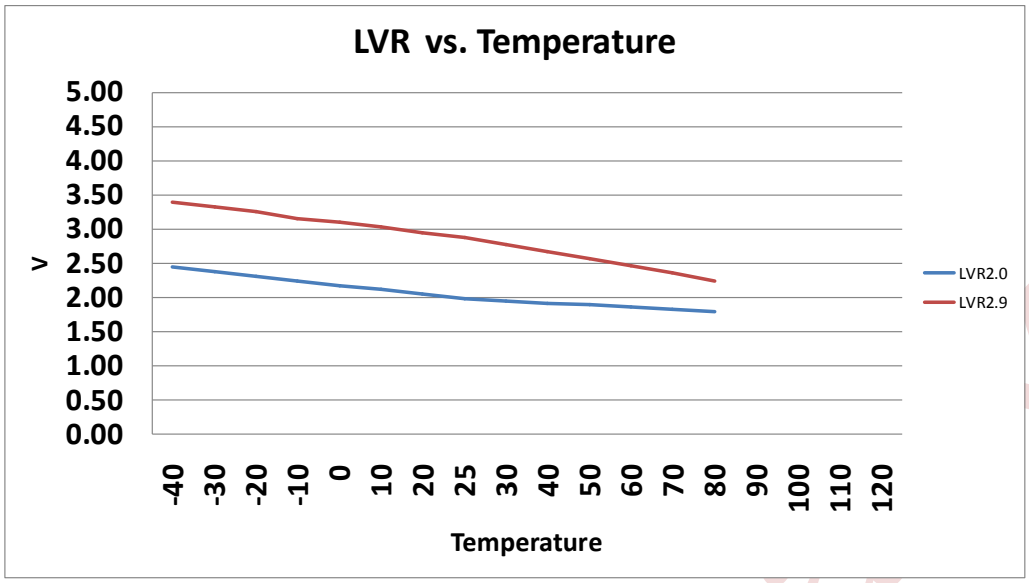
7. Characteristic Graph

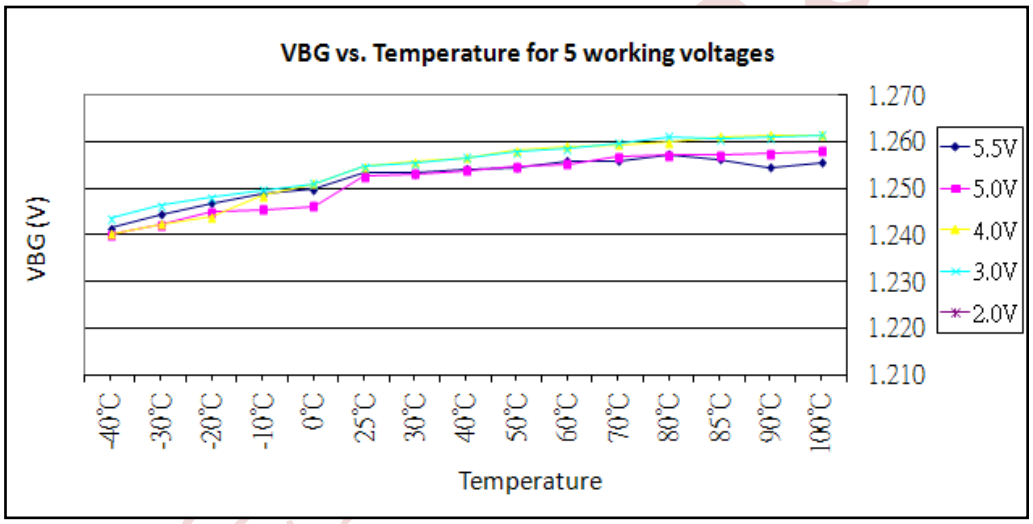












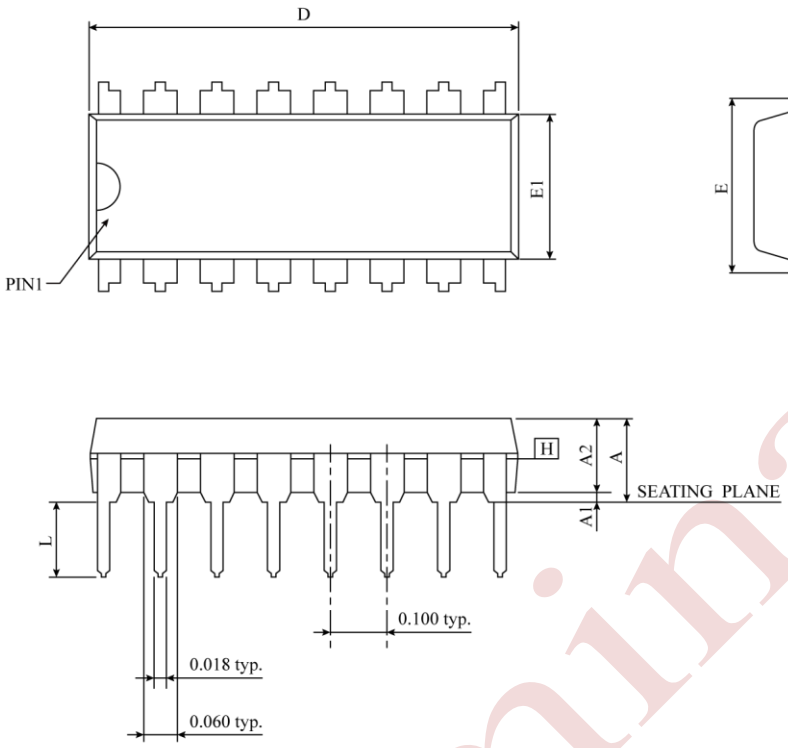
Note: Curve may vary chip by chip. Tenx guarantees that VBG ranges within the spec.

PACKAGING INFORMATION

The ordering information:

Ordering number	Package
TM57PA25B-OTP	Wafer/Dice blank chip
TM57PA25B-COD	Wafer/Dice with code
TM57PA25B-OTP-03	DIP 16-pin (300 mil)
TM57PA25B-OTP-16	SOP 16-pin (150 mil)
TM57PA25B-OTP-05	DIP 20-pin (300 mil)
TM57PA25B-OTP-21	SOP 20-pin (300 mil)

16-DIP Package Dimension

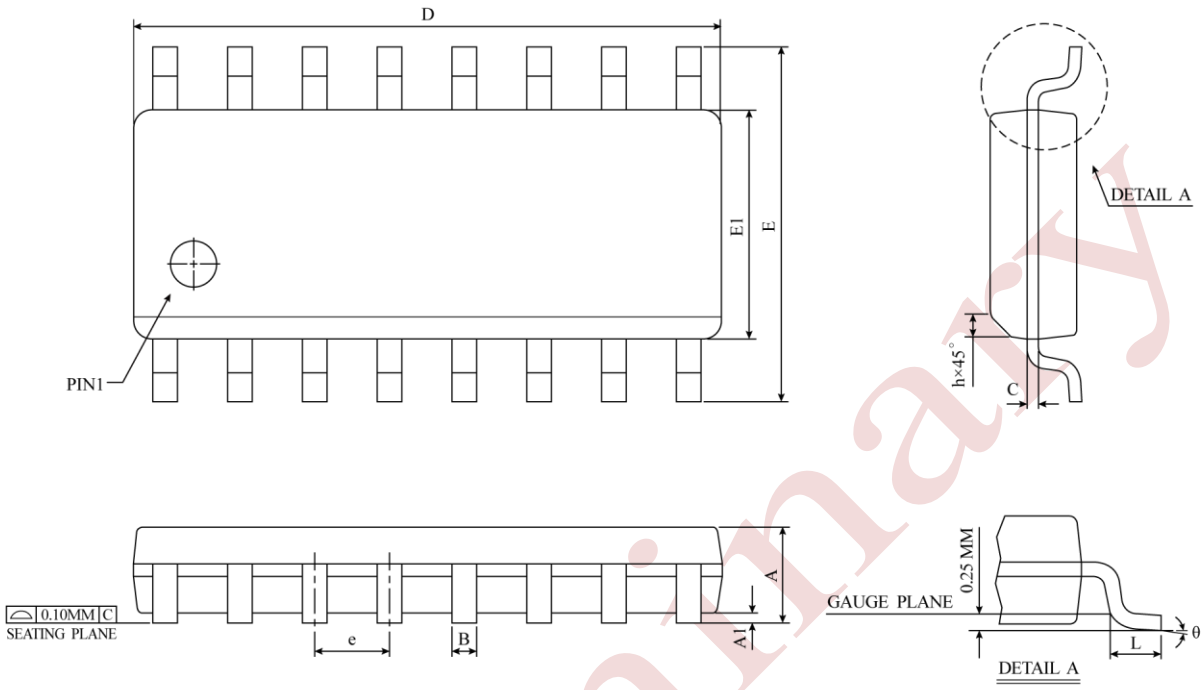


SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	4.369	-	-	0.172
A1	0.381	0.673	0.965	0.015	0.027	0.038
A2	3.175	3.302	3.429	0.125	0.130	0.135
D	18.669	19.177	19.685	0.735	0.755	0.775
E	7.620 BSC			0.300 BSC		
E1	6.223	6.350	6.477	0.245	0.250	0.255
L	2.921	3.366	3.810	0.115	0.133	0.150
^{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 (BB)					

NOTES：

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

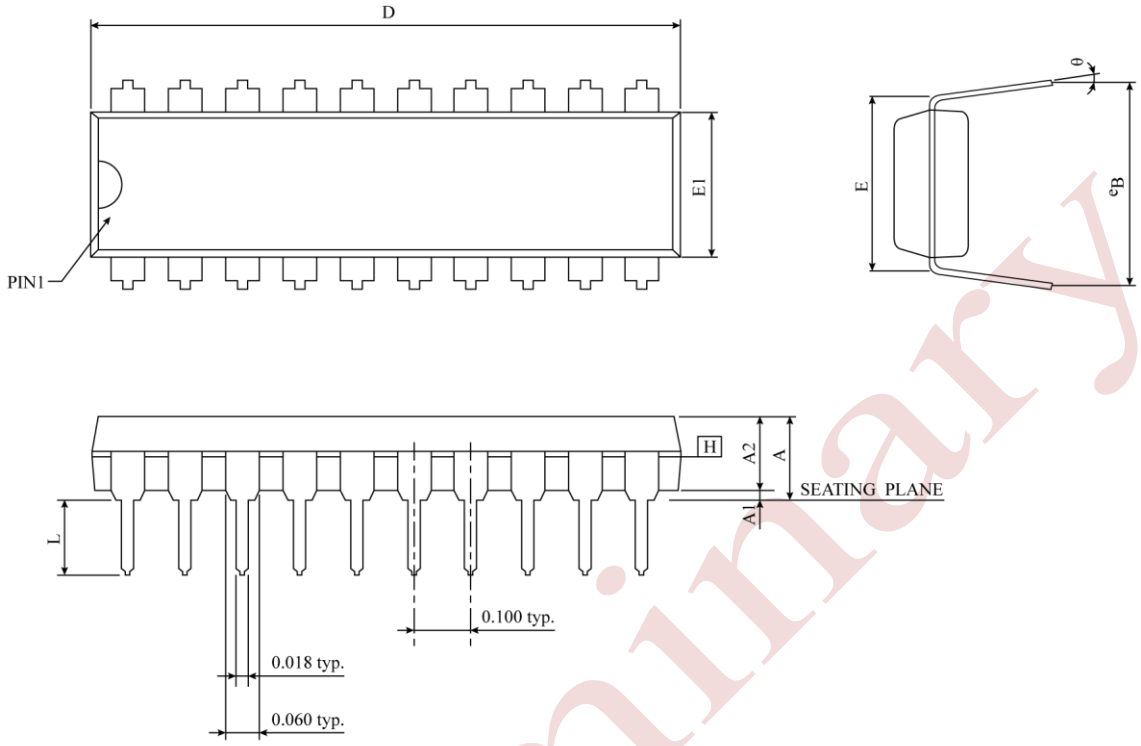
16-SOP Package Dimension



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.55	1.75	0.0532	0.0610	0.0688
A1	0.10	0.18	0.25	0.0040	0.0069	0.0098
B	0.33	0.42	0.51	0.0130	0.0165	0.0200
C	0.19	0.22	0.25	0.0075	0.0087	0.0098
D	9.80	9.90	10.00	0.3859	0.3898	0.3937
E	5.80	6.00	6.20	0.2284	0.2362	0.2440
E1	3.80	3.90	4.00	0.1497	0.1536	0.1574
e	1.27 BSC			0.050 BSC		
h	0.25	0.38	0.50	0.0099	0.0148	0.0196
L	0.40	0.84	1.27	0.0160	0.0330	0.0500
θ	0°	4°	8°	0°	4°	8°
JEDEC	MS-012 (AC)					

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

20-DIP Package Dimension

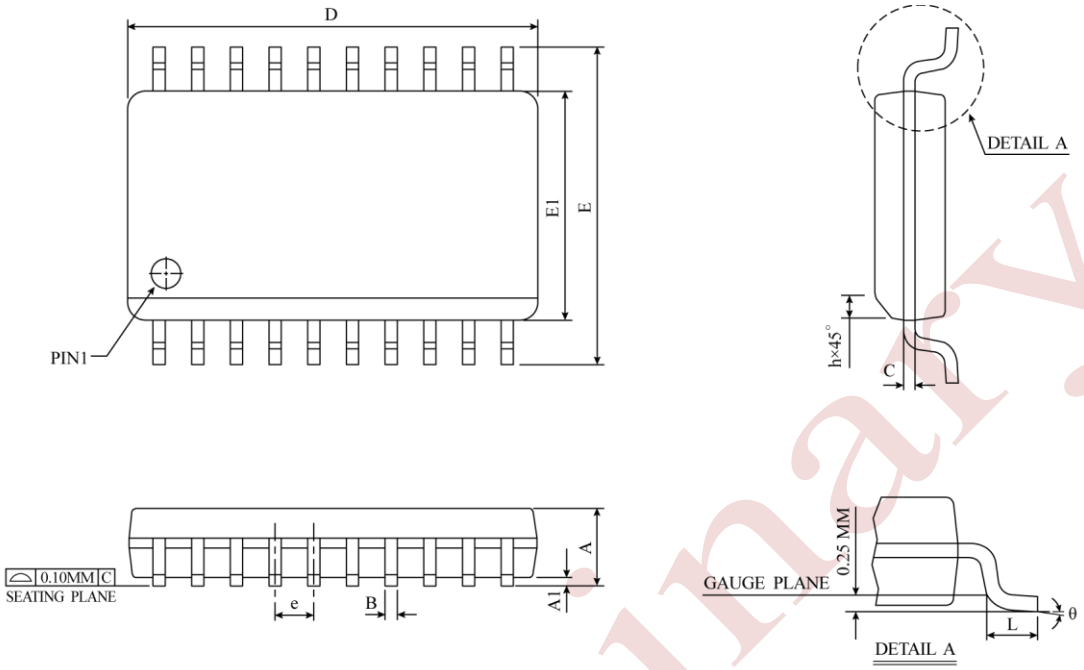


SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	4.445	-	-	0.175
A1	0.381	-	-	0.015	-	-
A2	3.175	3.302	3.429	0.125	0.130	0.135
D	25.705	26.061	26.416	1.012	1.026	1.040
E	7.620	7.747	7.874	0.300	0.305	0.310
E1	6.223	6.350	6.477	0.245	0.250	0.255
L	3.048	3.302	3.556	0.120	0.130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

NOTES：

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

20-SOP Package Dimension



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

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