TM55M8428/8428T/8228

DATA SHEET

Rev 0.93

AMENDMENT HISTORY

Version	Date	Description
0.90	2019/04/16	New release.
0.91	2019/06/13	1. P26,28,37,73,76,81 : Added restrictions on the use of SLEEP/CLRWDT instructions 2. P17: Added example of BANKSEL/BANKISEL 3. P15: Modify spelling errors
0.92	2019/07/15	1. P53,56,70: Added priority description for PWM output enable
0.93	2019/12/09	1. P10,12: Add SOP-16 Pin Assigment and Pin Summary 2. P92, 95: Add Package Information for SOP-16

CONTENTS

AMENDMENT HISTORY **2**
CONTENTS **3**
FEATURES **5**
BLOCK DIAGRAM **8**
PIN ASSIGNMENT **9**
PIN DESCRIPTIONS **11**
PIN SUMMARY **12**
FUNCTIONAL DESCRIPTION **13**
1. CPU Core **13**
1.1 Program ROM (PROM) 13
1.2 System Configuration Register (SYSCFG) 14
1.3 Page Locker Function 15
1.4 RAM Addressing Mode 16
1.5 Programming Counter (PC) and Stack 19
2. Reset **23**
2.1 Power on Reset 23
2.2 Low Voltage Reset 23
2.3 External Pin Reset 23
2.4 Watchdog Timer Reset 24
3. Clock Circuitry and Operation Mode **25**
3.1 System Clock 25
3.2 Dual System Clock Modes Transition 27
3.3 System Clock Oscillator 30
4. Interrupt **31**
5. I/O Port **33**
5.1 PA0-6, PB0-1, PD0-7 33
5.2 PA7 36
6. Peripheral Functional Block **37**
6.1 Watchdog (WDT) /Wakeup (WKT) Timer 37
6.2 Timer0 40
6.3 Timer1 44
6.4 T2:15-bit Timer 47
6.5 PWM0: (8+2) bits PWM 50
6.6 PWM1A/PWM1B/PWM1C: 8 bits PWMs 55
6.7 Analog-to-Digital Converter 58
6.8 Touch Key 61

MEMORY MAP 64
INSTRUCTION SET 73
ELECTRICAL CHARACTERISTICS 86
1. Absolute Maximum Ratings 86
2. DC Characteristics 86
3. Clock Timing 87
4. Reset Timing Characteristics 87
5. LVR Circuit Characteristics 87
6. ADC Electrical Characteristics 88
7. Characteristic Graphs 89
PACKAGING INFORMATION 92

FEATURES

1. **ROM: 2K x 14 bits MTP (Multi Time Programmable ROM) with Page Locker function**
2. **RAM: 256 x 8 bits**
3. **STACK: 8 Levels**
4. **System Oscillation Sources (Fsys) :**
   - Fast-clock
     - FIRC (Fast Internal RC) : 16 MHz
   - Slow-clock
     - SIRC (Slow Internal RC) : 65 KHz @VCC=5V (EV8230: 108K@5V)
5. **System Clock Prescaler:**
   - System Oscillation Sources can be divided by 1/2/4/8 as System Clock (Fsys)
6. **Dual System Clock:**
   - FIRC+SIRC
7. **Power Saving Operation Mode**
   - FAST Mode: Slow-clock can be disabled or enabled, Fast-clock keeps CPU running
   - SLOW Mode: Fast-clock can be disabled or enabled, Slow-clock keeps CPU running
   - IDLE Mode: Fast-clock and CPU stop. Slow-clock, T2, or Wake-up Timer keep running
   - STOP Mode: All clocks stop, T2 and Wake-up Timer stop
8. **3 Independent Timers**
   - Timer0
     - 8-bit timer divided by 1~256 pre-scaler option, Reload/Interrupt/Stop function
   - Timer1
     - 8-bit timer divided by 1~256 pre-scaler option, Reload/Interrupt/Stop function
     - 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: Slow-clock (SIRC), Fsys/128

9. Interrupt

- Three External Interrupt pins
  - 1 pin are falling edge wake-up triggered & interrupts
  - 2 pins is rising or falling edge wake-up triggered & interrupt
- Timer0/Timer1/T2/WKT (wake-up) Interrupts
- TK (Touch Key) /ADC Interrupt
- Individual Interrupt Vector

10. Wake-up (WKT) Timer

- Clocked by built-in RC oscillator with 4 adjustable interrupt times

  17 ms/34 ms/68 ms/136 ms @VCC=3V, 16 ms/32 ms/64 ms/128 ms @VCC=5V
  (EV8230: 11 ms/22 ms/43 ms/86 ms @VCC=3V, 10ms/19 ms/39 ms/77 ms @VCC=5V)

11. Watchdog Timer

- Clocked by built-in RC oscillator with 4 adjustable reset times

  140 ms/280 ms/1120 ms/2240 ms @VCC=3V, 128 ms/256 ms/1024 ms/2048 ms @VCC=5V
  (EV8230: 90 ms/179 ms/719 ms/1440 ms @ 3V, 77 ms/154 ms/616 ms/1232 ms @5V)

- Watchdog timer can be disabled/enabled in STOP mode

12. PWMx4

- PWM0:
  - 8+2 bits, duty-adjustable, period-adjustable controlled PWM
  - PWM0 clock source: Fast-clock or FIRC 16 MHz/32MHz, with 1~64 pre-scalers
  - Complementary PWM output (PWM0P, PWM0N)
  - Non-overlap time durations adjustable: (0~8)*(PWMCLK)
- PWM1A/PWM1B/PWM1C:
  - 8 bits, duty-adjustable (Independent) , period-adjustable controlled (Shared) PWM x3
  - PWM1A/1B/1C clock source (Shared) : Fast-clock or FIRC 16 MHz/32MHz, with 1~64 pre-scalers

13. 12-bit ADC Converter with 14 input channels and 1 internal reference voltage

- Internal Bandgap reference voltage1.25V ±3% @25°C, VCC=3V~5V
- ADC reference voltage=VCC

14. Reset Sources

- Power On Reset/Watchdog Reset/Low Voltage Reset/External Pin Reset

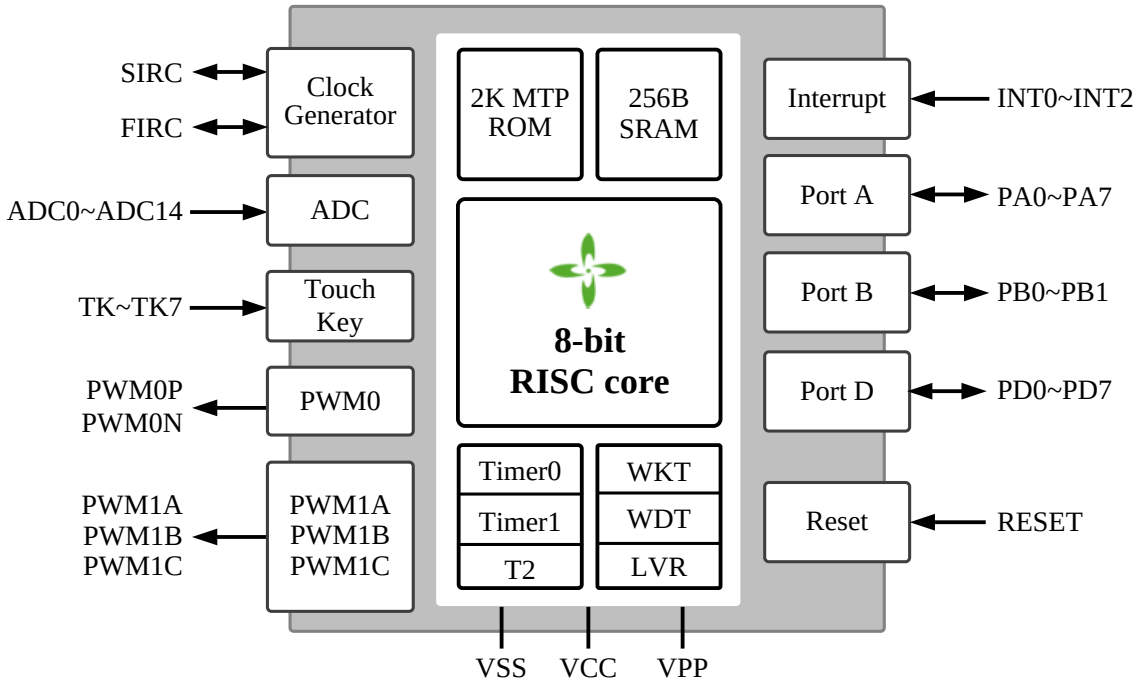
15. Low Voltage Reset (LVR) /Low Voltage Detection Flag (LVD) Option:

- 4-Level Low Voltage Reset: 2.3V/2.8V/3.6V/4.2V
- 3-Level Low Voltage Detection Flag: 2.8V/3.6V/4.2V (when LVR = 2.3V)

16. **8-Channel Touch Key**
    - Interrupt/Wake-up CPU while key is pressed
    - 3-bit TK reference clock capacitor adjustment
    - 12-bit TK scan length adjustment
17. **Operating Voltage:**
    - Fsys= 2 MHz, LVR ~5.5V
    - Fsys=16 MHz, 2.8V~5.5V
18. **Operating Temperature Range : -40°C to + 85°C**
19. **Table Read Instruction: 14-bit ROM data lookup table**
20. **Instruction set: 39 Instructions**
21. **Instruction Execution Time**
    - 2 system clocks (Fsys) per instruction except branch
22. **I/O ports: Maximum 18 programmable I/O pins**
    - Open-Drain Output
    - CMOS Push-Pull Output
    - Schmitt Trigger Input with pull-up resistor option
23. **Programming connectivity support 5-wire (ICP) or 8-wire program**
24. **Page Locker Size: 512W/640W/768W …./1920W by 125 words step**
25. **Package Types:**
    - SOP-20/DIP-20/SOP-16/DIP-16/SOP-8
26. **Supported EV board on ICE**

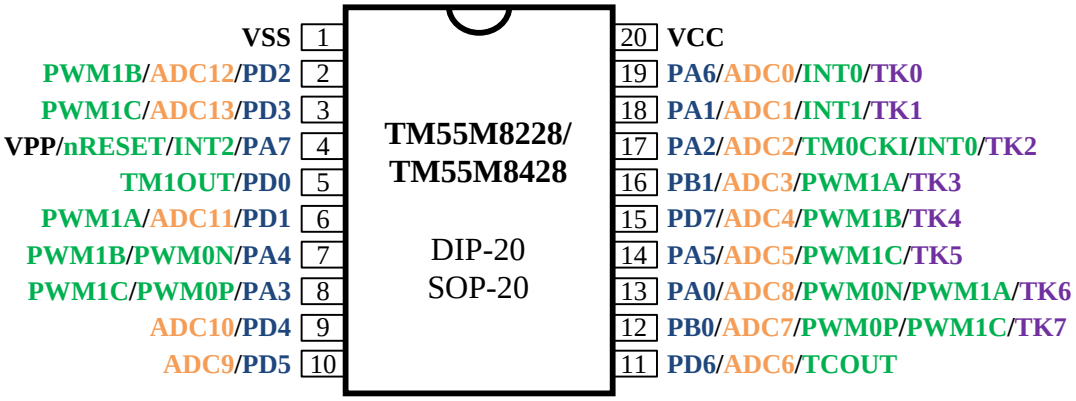
    EV board: EV8230

BLOCK DIAGRAM

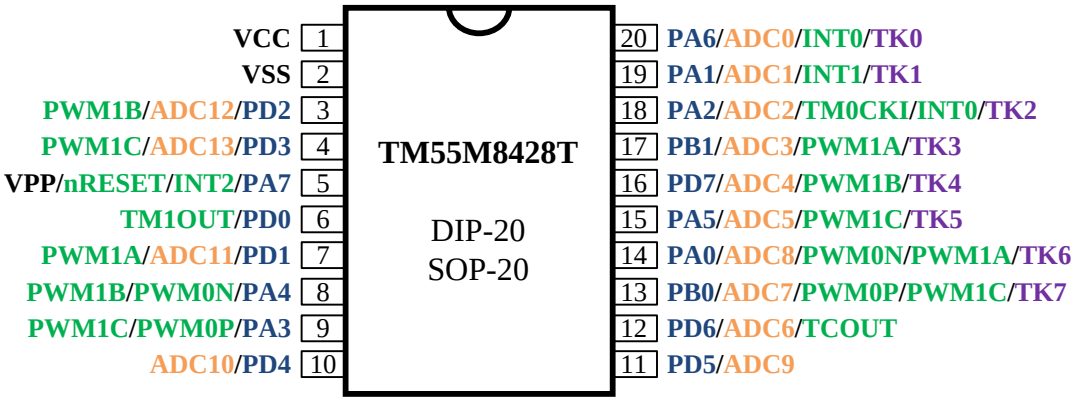


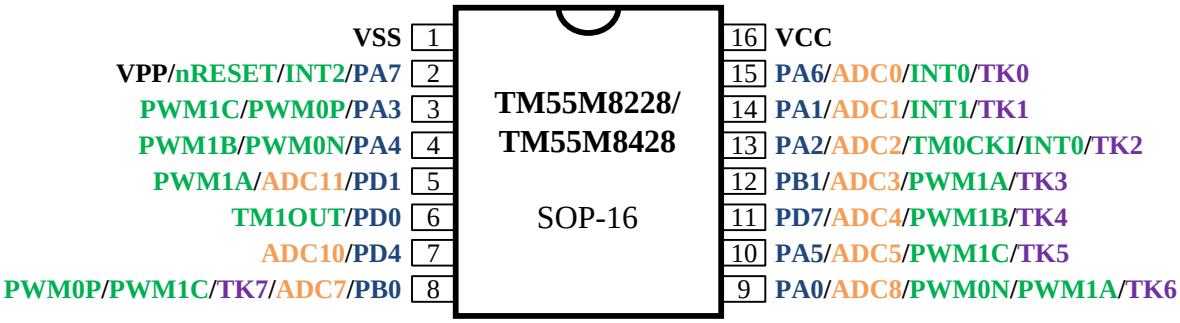
TM55M8228 Block Diagram

PIN ASSIGNMENT

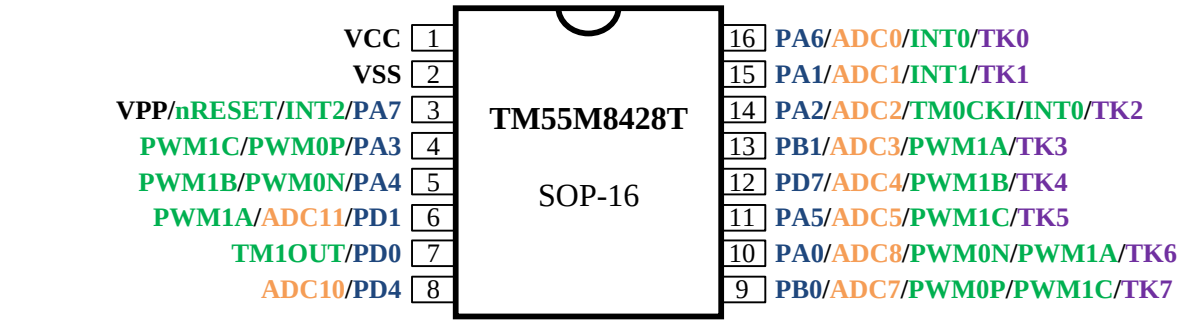


TKx : Only **TM55M8428** supports this feature





TKx : Only **TM55M8428** supports this feature



PIN DESCRIPTIONS

Name	In/Out	Pin Description
PA0-PA6 PB0-PB1 PD0-PD7	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS "**push-pull**" output or "**open-drain**" output. Pull-up resistors are assignable by software.
PA7	I/O	Bit-programmable I/O port for Schmitt-trigger input, or "**open-drain**" output. Schmitt-trigger input with pull-high
nRESET	I	External active low reset, internal pull-high
VCC, VSS	P	Power Voltage input pin and ground
VPP	I	PROM programming high voltage input
INT0–INT2	I	External interrupt input
TM0CKI	I	Timer0's input in counter mode
TM1OUT	O	Timer1 match output, TM1OUT toggles when Timer1 overflow occurs.
PWM0P,PWMN	O	(8+2) bit PWM0 output
PWM1A	O	8 bit PWM1A output
PWM1B	O	8 bit PWM1B output
PWM1C	O	8 bit PWM1C output
ADC14~ADC0	I	A/D channels input
TK0-TK7	I	Touch key input (M8428 only)
TCOUT	O	Post-prescaler Instruction Cycle (Fsys/2) output

Programming pins:

Normal mode: VCC/VSS/PA0/PA1/PA2/PA3/PA4/PA7 (VPP)

ICP mode: VCC/VSS/PA0/PA1/PA7(VPP) -When using ICP (In-circuit Program) mode, the PCB needs to remove all components of PA0, PA1, PA7.

PIN SUMMARY

20-Pin		16-Pin		Pin Name	Type	GPIO				Function After Reset	Alternate Function			
						Input		Output						
TM55M8428/8228	TM55M8428T	TM55M8428/8228	TM55M8428T			Wake up	Ext. Interrupt	O.D	P.P		PWM	TK	ADC	MISC
20	1	16	1	**VCC**	P									
1	2	1	2	**VSS**	P									
2	3			**PWM1B/ADC12/PD2**	I/O			○	○	PD2			○	
3	4			**PWM1C/ADC13/PD3**	I/O			○	○	PD3			○	
4	5	2	3	**VPP/nRESET/INT2/PA7**	I/O	○	○	○		PA7				nRESET
5	6	6	7	**TM1OUT/PD0**	I/O			○	○	PD0				TM1OUT
6	7	5	6	**PWM1A/ADC11/PD1**	I/O			○	○	PD1			○	
7	8	4	5	**PWM1B/PWM0N/PA4**	I/O			○	○	PA4				
8	9	3	4	**PWM1C/PWM0P/PA3**	I/O			○	○	PA3				
9	10	7	8	**ADC10/PD4**	I/O			○	○	PD4	○		○	
10	11			**ADC9/PD5**	I/O			○	○	PD5			○	
11	12			**PD6/ADC6/TCOUT**	I/O			○	○	PD6			○	TCOUT
12	13	8	9	**PB0/ADC7 /PWM0P/PWM1C/TK7**	I/O			○	○	PB0	○	○	○	
13	14	9	10	**PA0/ADC8 /PWM0N/PWM1A/TK6**	I/O			○	○	PA0	○	○	○	
14	15	10	11	**PA5/ADC5/PWM1C/TK5**	I/O			○	○	PA5		○	○	
15	16	11	12	**PD7/ADC4/PWM1B/TK4**	I/O			○	○	PD7		○	○	
16	17	12	13	**PB1/ADC3/PWM1A/TK3**	I/O			○	○	PB1		○	○	
17	18	13	14	**PA2/ADC2 /TM0CKI/INT0/TK2**	I/O	○	○	○	○	PA2		○	○	TM0CKI
18	19	14	15	**PA1/ADC1/INT1/TK1**	I/O	○	○	○	○	PA1		○	○	
19	20	15	16	**PA6/ADC0/INT0/TK0**	I/O	○	○	○	○	PA6		○	○	

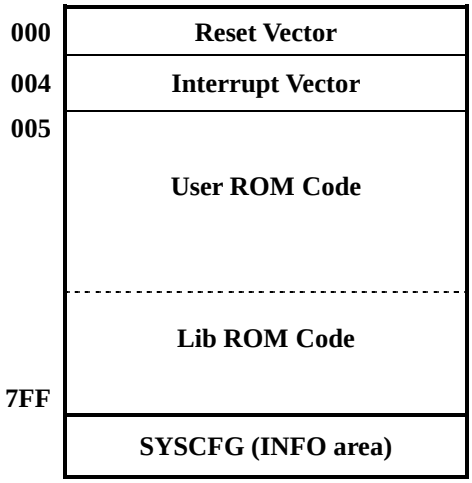
Symbol：P.P. = Push-Pull Output
O.D. = Open Drain
SYS = by SYSCFG bit

FUNCTIONAL DESCRIPTION

1. CPU Core

1.1 Program ROM (PROM)

The MTP Program ROM of this device is 2K words, with an extra INFO area to store the SYSCFG. The ROM can be written multi-times and can be read as long as the PROTECT and LPROT bits of SYSCFG are not set. The SYSCFG can be read no matter PROTECT or LPROT is set, but PROTECT bit can be cleared only when the User ROM Code area is erased, and LPROT bit can be cleared only when the Lib ROM Code area is erased. That is, unprotect the PROTECT or LPROT bit needs to erase the corresponding ROM area. If LPROT bit is set, The ROM can still be written multi-times in the User ROM Code area to update user ROM code again by writer, but the Lib ROM Code area will not be read or written again by writer until the LPROT bit is cleared. On the other hand, if PORTECT bit is set, the user ROM code area will not be read by writer, and the user ROM code can't be updated until the PORTECT bit is cleared.



1.1.1 Reset Vector (000H)

After reset , system will restart the program counter (PC) at the address 000h, all registers will revert to the default value

1.1.2 Interrupt Vector (004H)

When an interrupt occurs, the program counter (PC) will be pushed onto the stack and jumps to address 004H.

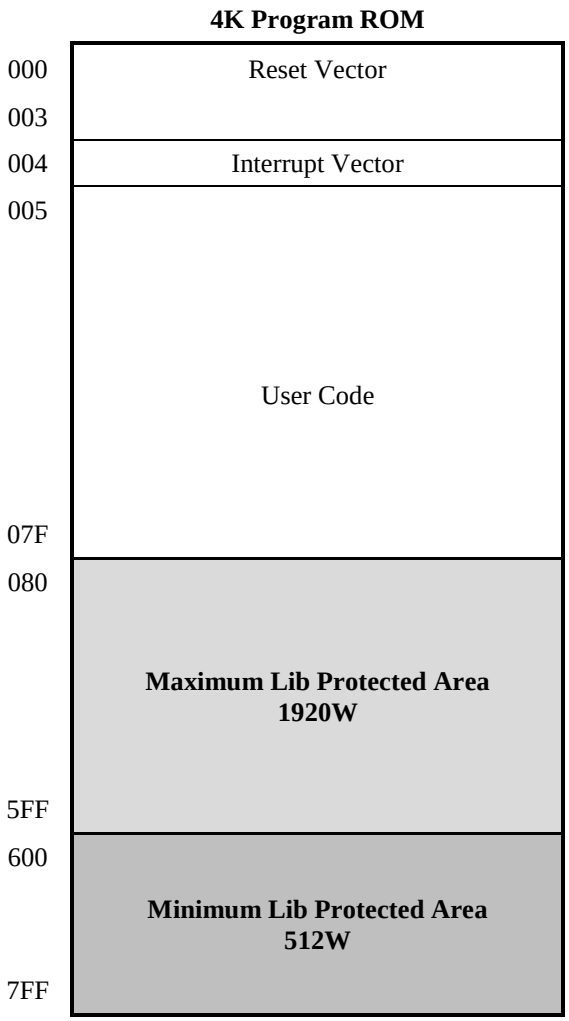
1.2 System Configuration Register (SYSCFG)

The System Configuration Register (SYSCFG) is located at MTP INFO area, it contains two 13bits registers (CFGWL/CFGWH) . The SYSCFG determines the option for initial condition of MCU. It is written by PROM Writer only. User can select LVR operation Mode and chip operation mode by SYSCFG register.The 13th bit of CFGWH is code protection selection bit. If this bit is 1, the data in PROM will be protected, when user reads PROM.

Bit		**13~0**	
Default Value		**11111111111111**	
Bit		Description	
CFGWL	13	**LPROT**: Lib Code protection selection	
		1	Enable
		0	Disable
	12~9	**LSIZE**: Lib Size selection	
		1100	1920W
		0001	512W
		0000	No use Page locker function
	8~0	Tenx Reserved	
CFGWH	13	**PROTECT**: Code protection selection	
		1	Enable
		0	Disable
	12	**XRSTE**: External Pin (PA7) Reset Enable	
		1	Enable
		0	Disable (PA7 as input I/O pin)
	11-10	**LVR**: Low Voltage Reset Mode	
		11	LVR 4.2V (without LVD function)
		10	LVR 3.6V (without LVD function)
		01	LVR 2.8V (without LVD function)
		00	LVR 2.3V (with LVD function)
	9-8	**WDTE**: WDT Reset Enable	
		11	Always Enable
		10	Enable in FAST/SLOW mode, Disable in IDLE/STOP mode
		0X	Disable
	7-0	Tenx Reserved	

1.3 Page Locker Function

TM55M8228/8428 support Page Locker function. By setting LPROT (CFGWL.13) , user can choose whether to turn it on. If the user A (library code provider) turns this function on, the user A (library code provider) can select different size (512~2304W) of lib protected area by LSIZE (CFGWL.12~9) . In lib protected area, the user B (firmware developer) can't read ROM code by TABRL/TABRH instruction or in any other way. By using the TICE99IDE tool, the user A can provide a protected lib code for the user B to use, but the user B does not know its details, and the user B still can continue to complete the main code in the unprotected area.



LSIZE	Lib Protected Area
1920	(80H~7FFH)
1792	(100H~7FFH)
1664	(180H~7FFH)
1536	(200H~7FFH)
1408	(280H~7FFH)
1280	(300H~7FFH)
1152	(380H~7FFH)
1024	(400H~7FFH)
896	(480H~7FFH)
768	(500H~7FFH)
640	(580H~7FFH)
512	(600H~7FFH)

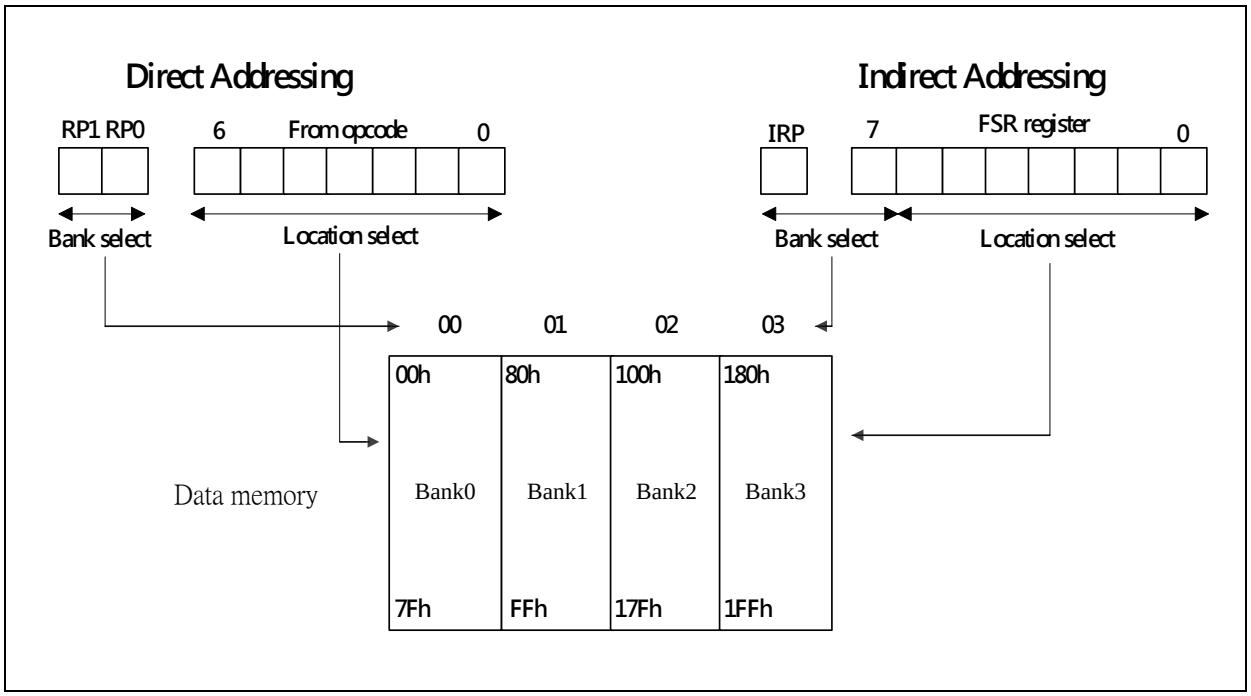
1.4 RAM Addressing Mode

There are one Data Memory Plane in CPU, F-Plane. The F-Plane is partitioned into four banks. Each bank extends up to 7Fh (128bytes). The lower locations of each bank are reserved for Special Function Registers (SFR). Above the SFR are General Purpose Registers, implemented as static RAM. All implemented banks contain Special Function Registers. Some frequently used Special Function Registers from one bank may be mirrored in another bank for code reduction and quicker access.

Bits RP1 and RP0 (STATUS[6:5]) are the bank select bits.

RP1 : RP0	BANK
00	0
01	1
10	2
11	3

F-Plane can be addressed directly or indirectly. The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing. Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself, indirectly (FSR = '0') will read 00h. Writing to the INDF register indirectly results in a no operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>). Refer to the figure below.



Direct / Indirec addressing

	BANK0 00~7Fh		BANK1 80h~FFh		BANK2 100h~17Fh		BANK3 180h~1FFh
00h	**INDF**	80h	**INDF**	100h	**INDF**	180h	**INDF**
01h	TM0	81h	OPTION	101h	TM0	181h	OPTION
02h	**PCL**	82h	**PCL**	102h	**PCL**	182h	**PCL**
03h	**STATUS**	83h	**STATUS**	103h	**STATUS**	183h	**STATUS**
04h	**FSR**	84h	**FSR**	104h	**FSR**	184h	**FSR**
05h	PAD	85h		105h	TESTREG	185h	DPL
06h	PBD	86h		106h		186h	DPH
07h	PDD	87h		107h		187h	
08h		88h		108h		188h	
09h		89h		109h	LVRPD	189h	
0Ah	**PCLATH**	8Ah	**PCLATH**	10Ah	**PCLATH**	18Ah	**PCLATH**
0Bh	**INTIE**	8Bh	**INTIE**	10Bh	**INTIE**	18Bh	**INTIE**
0Ch	INTIF	8Ch	PAMODH	10Ch		18Ch	
0Dh		8Dh	PAMODL	10Dh		18Dh	
0Eh		8Eh		10Eh		18Eh	
0Fh	CLKCTL	8Fh	PBMODL	10Fh		18Fh	IRCF
10h	TM0RLD	90h	PDMODH	110h		190h	
11h	TM0CTL	91h	PDMODL	111h		191h	
12h	TM1	92h	PWM0PRD	112h		192h	
13h	TM1RLD	93h	PWM0DH	113h		193h	
14h	TM1CTL	94h	PWM0DL	114h		194h	
15h	T2CTL	95h	PWM0CTL	115h		195h	
16h	MF016	96h	PWMCTL	116h		196h	
17h	ADH	97h	PWM1PRD	117h		197h	
18h	ADCTL	98h	PWM1AD	118h		198h	
19h	MF019	99h	PWM1BD	119h		199h	
1Ah	TKDL	9Ah	PWM1CD	11Ah		19Ah	
1Bh	TKDH	9Bh	MF09B	11Bh		19Bh	
1Ch	TKTMRL	9Ch	MF09C	11Ch		19Ch	
1Dh	TKTMRH	9Dh		11Dh		19Dh	
1Eh	TKCON0	9Eh		11Eh		19Eh	
1Fh	TKCON1	9Fh		11Fh		19Fh	
20h ~ RAM Bank0 area ~ (80 Bytes) 6Fh		A0h ~ RAM Bank1 area ~ (80 Bytes) EFh		120h ~ RAM Bank2 area ~ (80 Bytes) 16Fh		1A0h ~ ~ 1EFh	
70h ~ 7Fh	**common area 16 Bytes**	F0h ~ FFh	**accesses 70h~7Fh**	170h ~ 17Fh	**accesses 70h~7Fh**	1F0h ~ 1FFh	**accesses 70h~7Fh**

◇Example: read/write register by using direct addressing

```
BCF     RP1         ;
BSF     RP0         ; set RP1=0,RP0=1  =>Bank1
MOVFW   PWM1PRD     ; read PWM1PRD (Bank1) to W
.
BSF     RP1         ;
BCF     RP0         ; set RP1=1,RP0=0  =>Bank2
MOVLW   037H        ; W=37H
MOVWF   LVRPD       ; LVRPD (Bank2) = W = 037H
```

◇Example: read/write register by using direct addressing and BANKSEL

```
BANKSEL PWM1PRD     ; Set RP1/RP0 to the bank where PWM1PRD is located
MOVFW   PWM1PRD     ; read PWM1PRD (Bank1) to W
.
BANKSEL LVRPD       ; Set RP1/RP0 to the bank where LVRPD is located
MOVLW   037H        ; W=37H
MOVWF   LVRPD       ; LVRPD (Bank2) = W = 037H
```

◇Example: read/write register by using indirect addressing

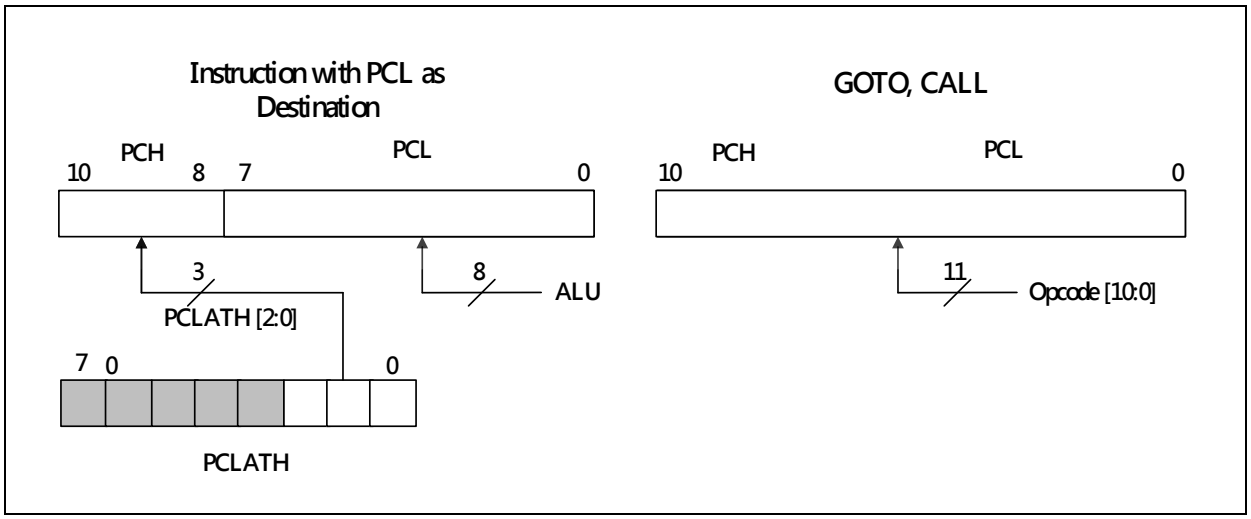
```
BCF     RP1         ; IRP=0  =>Bank0/1
MOVLW   PWM1PRD     ; W=97H
MOVWF   FSR         ; FSR = W =97H
MOVFW   INDF        ; read SFR PWM1PRD(97h) to W
.
BSF     IRP         ; IRP1=1 =>Bank2/3
MOVLW   09H         ; W=09H
MOVWF   FSR         ; FSR = W =09H
MOVLW   037H        ; W=37H
MOVWF   INDF        ; LVRPD (109H) = W = 037H
```

◇Example: read/write register by using indirect addressing and BANKISEL

```
BANKISEL PWM1PRD    ;Set IRP to the bank where PWM1PR is located
MOVLW   PWM1PRD     ; W=97H
MOVWF   FSR         ; FSR = W =97H
MOVFW   INDF        ; read SFR PWM1PRD(97h) to W
.
BANKISEL LVRPD      ; Set IRP to the bank where LVRPD is located
MOVLW   09H         ; W=09H
MOVWF   FSR         ; FSR = W =09H
MOVLW   037H        ; W=37H
MOVWF   INDF        ; LVRPD (109H) = W = 037H
```

1.5 Programming Counter (PC) and Stack

The Programming Counter is 11-bit wide capable of addressing a 2Kx14 MTP ROM. The low byte comes from the PCL register, which is a readable and writable register. The upper bits (PC[10:8]) are not readable, but are indirectly writable through the PCLATH register. On any RESET, the upper bits of the PC will be cleared. 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 (004h) 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. Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC[10:8] bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the program counter to be changed by writing the desired upper 3 bits to the PCLATH register. When the lower 8 bits are written to the PCL register, all 11 bits of the program counter will change to the values contained in the PCLATH register and those being written to the PCL register.



The STACK is 11-bit wide and 8-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.

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

◇ Example: To look up the PROM data located “TABLE” & “TABLE2”.

```
            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 lookup data, W=55H.
            …..
            INCF        INDEX, 1                ; Increment the index address for next address
            …..
            GOTO        LOOP                    ; Go to LOOP label.
            …..
            BANKSEL     DPH
            MOVLW       (TABLE2 >>8) & 0xff
            MOVWF       DPH                     ; DPH register (F186.2~0)
            MOVLW       (TABLE2) & 0xff
            MOVWF       DPL                     ; DPL register (F185.7~0)
            TABRL                               ; W=86H
            TABRH                               ; W=19H
            …..

TABLE:
            ADDWF       PCL, 1                  ; Add the W with PCL, the result back in PCL.
            RETLW       55H                     ; W=55h when return
            RETLW       56H                     ; W=56H when return
            RETLW       58H                     ; W=58H when return
            …..
            ORG         368H

TABLE2:
            .DT  0x1986, 0x3719, 0x2983…        ; 14-bit ROM data
```

1.5.1 ALU and Working (W) Register

The ALU is 8-bit wide and capable of addition, subtraction, shift and logical operations. In two-operand instructions, typically one operand is the W register, which is an 8-bit non-addressable register used for ALU operations. The other operand is either a file register or an immediate constant. In single operand instructions, the operand is either W register or a file register. Depending on the instruction executed, the ALU may affect the values of Carry (C) , Digit Carry (DC) , and Zero (Z) Flags in the STATUS register. The C and DC flags operate as a/Borrow and/Digit Borrow, respectively, in subtraction.

Note: /Borrow represents inverted of Borrow register.

/Digit Borrow represents inverted of Digit Borrow register.

1.5.2 STATUS Register (F-Plane 03H/83H/103H/183H)

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.

<table>
<tr><td>STATUS</td><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td>Reset Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R</td><td>R</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Bit</td><td colspan="8">Description</td></tr>
<tr><td>7</td><td colspan="8">IRP: Register Bank Select bit (used for indirect addressing)
0 = Bank 0,1 (00h - FFh)
1 = Bank 2,3 (100h - 1FFh)</td></tr>
<tr><td>6-5</td><td colspan="8">RP1:RP0: Register Bank Select bits (used for direct addressing)
00 = Bank 0 (00h - 7Fh)
01 = Bank 1 (80h - FFh)
10 = Bank 2 (100h - 17Fh)
11 = Bank 3 (180h - 1FFh)
Each bank is 128 bytes</td></tr>
<tr><td>4</td><td colspan="8">TO: Time Out Flag
0: after Power On Reset, LVR Reset, or CLRWDT/SLEEP instruction
1: WDT time out occurs</td></tr>
<tr><td>3</td><td colspan="8">PD: Power Down Flag
0: after Power On Reset, LVR Reset, or CLRWDT instruction
1: after SLEEP instruction</td></tr>
<tr><td>2</td><td colspan="8">Z: Zero Flag
0: the result of a logic operation is not zero
1: the result of a logic operation is zero</td></tr>
<tr><td rowspan="3">1</td><td colspan="8">DC: Decimal Carry Flag or Decimal / Borrow Flag</td></tr>
<tr><td colspan="4">ADD instruction</td><td colspan="4">SUB instruction</td></tr>
<tr><td colspan="4">0: no carry
1: a carry from the low nibble bits of the result occurs</td><td colspan="4">0: a borrow from the low nibble bits of the result occurs
1: no borrow</td></tr>
<tr><td rowspan="3">0</td><td colspan="8">C: Carry Flag or/Borrow Flag</td></tr>
<tr><td colspan="4">ADD instruction</td><td colspan="4">SUB instruction</td></tr>
<tr><td colspan="4">0: no carry
1: a carry occurs from the MSB</td><td colspan="4">0: a borrow occurs from the MSB
1: no borrow</td></tr>
</table>

◇ Example: Write immediate data into STATUS register.

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

◇ Example: Bit addressing set and clear STATUS register.

```
BSF     STATUS, 0   ; Set C=1.
BSF     RP1
BSF     RP0         ; Selection Bank3
BCF     STATUS, 0   ; Clear C=0.
BCF     RP1
BCF     RP0         ; Selection Bank0
```

◇ Example: Determine the C flag by BTFSS instruction.

```
BTFSS   STATUS, 0   ; Check the carry flag
GOTO    LABEL_1     ; If C=0, goto label_1
GOTO    LABEL_2     ; If C=1, goto label_2
```

2. Reset

This device can be RESET in four ways.

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

Resets can be caused by Power on Reset (POR) , External Pin Reset (XRST) , Watchdog Timer Reset (WDTR) , or Low Voltage Reset (LVR) . The CFGWH controls the Reset functionality. After Reset, the SFRs are returned to their default value, the program counter (PC) is cleared , and the system starts running from the reset vector 000H place. The TO and PD flags at status register (STATUS) are indicate system reset status.

2.1 Power on Reset

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.

2.2 Low Voltage Reset

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. See the following LVR Selection Table; user must also consider the lowest operating voltage of operating frequency.

LVR Selection Table:

LVR level	Operating voltage
LVR2.3	5.5V > VCC > 2.3V
LVR2.8	5.5V > VCC > 2.8V
LVR3.6	5.5V > VCC > 3.6V
LVR4.2	5.5V > VCC > 4.2V or V_{CC} =5.0V

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

2.3 External Pin Reset

The External Pin Reset can be disabled or enabled by the SYSCFG register. It needs to keep at least 2 SIRC clock cycle long to be seen by the chip. XRST also set all the control registers to their default reset value. The TO/PD flags are not affected by these resets.

2.4 Watchdog Timer Reset

WDT overflow Reset can be disabled or enabled by the SYSCFG register. It runs in Fast/Slow mode and runs or stops in IDLE/STOP mode. WDT overflow speed can be defined by WDTPSC SFR. WDT is cleared by device Reset or CLRWDT SFR bit WDT overflow Reset also set all the control registers to their default reset value. The TO/PD flags are not affected by these resets.

◇ Example: Defining Reset Vector

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

        ORG     010H

START:
        …                       ; 010H, The head of user program
        …
        GOTO    START
```

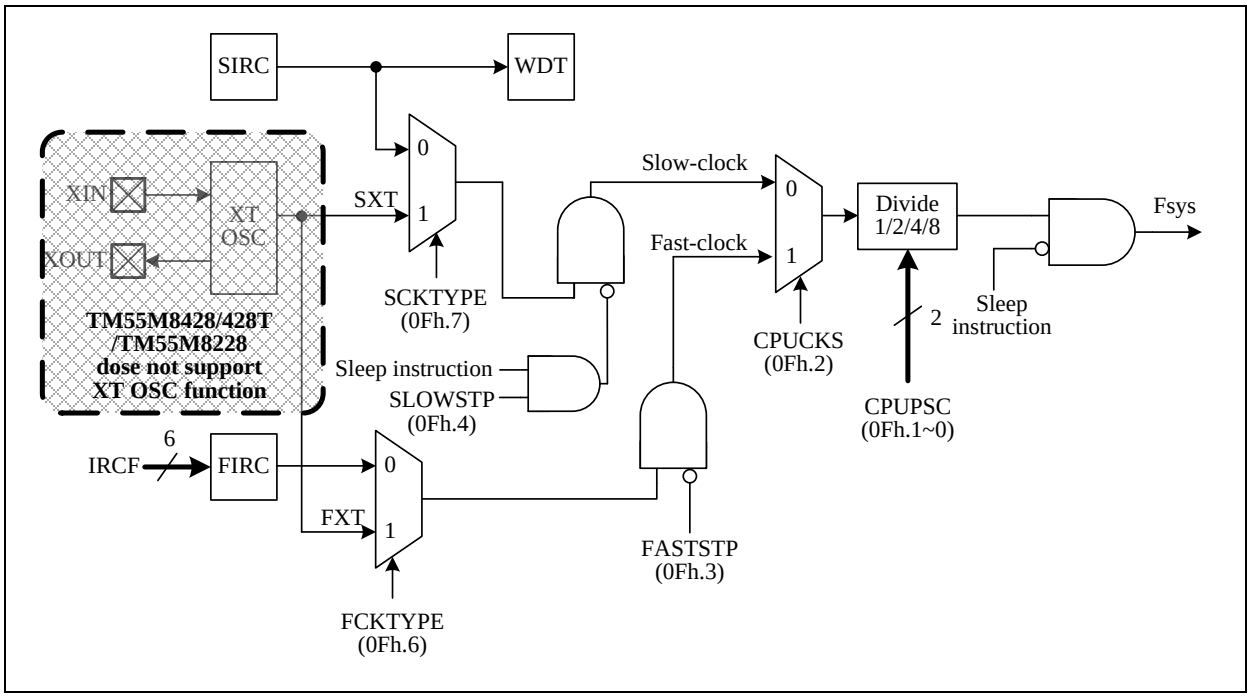
3. Clock Circuitry and Operation Mode

3.1 System Clock

The device is designed with dual-clock system. There are two kinds of clock source, i.e. SIRC (Slow Internal RC) , 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.

After Reset, the device is running at Slow mode with 80 KHz SIRC. S/W should select the proper clock rate for chip operation safety. The higher V_{CC} allows the chip to run at a higher System clock frequency. In a typical condition, an 16 MHz System clock rate requires V_{CC} > 2.8V..

The CLKCTL (0B) SFR controls the System clock operating. H/W automatically blocks the S/W abnormally setting for this register. S/W can only change the Slow-clock type in Fast mode and change the Fast-clock type in Slow mode. Never to write both FASTSTP=1 & CPUCKS=1. It is recommended to write this SFR bit by bit.



Clock Scheme Block Diagram

The frequency of FIRC (Fast Internal RC) can be adjusted by IRCF (18Fh) . When IRCF=00h, frequency is the lowest. When IRCF=3Fh, frequency is the highest. With this function, we can adjust the frequency of FIRC after power on. Each IC may have different default value of IRCF, to make sure the frequency of FIRC=16 MHz after Power on Reset.

FAST Mode:

In this mode, the program is executed using Fast-clock as CPU clock (Fsys) . The Timer0, Timer1 blocks are also driven by Fast-clock, The PWM0/PWM1 block can driven by FIRC16M, FIRC32M or Fsys. T2 can be driven by Slow-clock or Fsys/128 by setting T2CKS (15h.2).

SLOW Mode:

After power-on or reset, device enters SLOW mode, the default Slow-clock is SIRC. In this mode, the Fast-clock can stopped (by FASTSTP=1, for power saving) or running (by FASTSTP=0) , and Slow-clock is enabled. All peripheral blocks (Timer0, Timer1etc...) clock sources are Slow-clock in the SLOW mode.

IDLE Mode:

If Slow-clock is enabled (SLOWSTP=0) and T2CKS=0 before executing the SLEEP instruction, the CPU enters the IDLE mode. In this mode, the Slow-clock source keeps T2 block running. CPU stop fetching code and all blocks are stop except T2 related circuits. Idle mode is terminated by Reset or enabled Interrupts wake up.

Another way to keep clock oscillation in IDLE mode is setting WKTIE=1 before executing the SLEEP instruction. In such condition, the WKT keeps working and wake up CPU periodically.

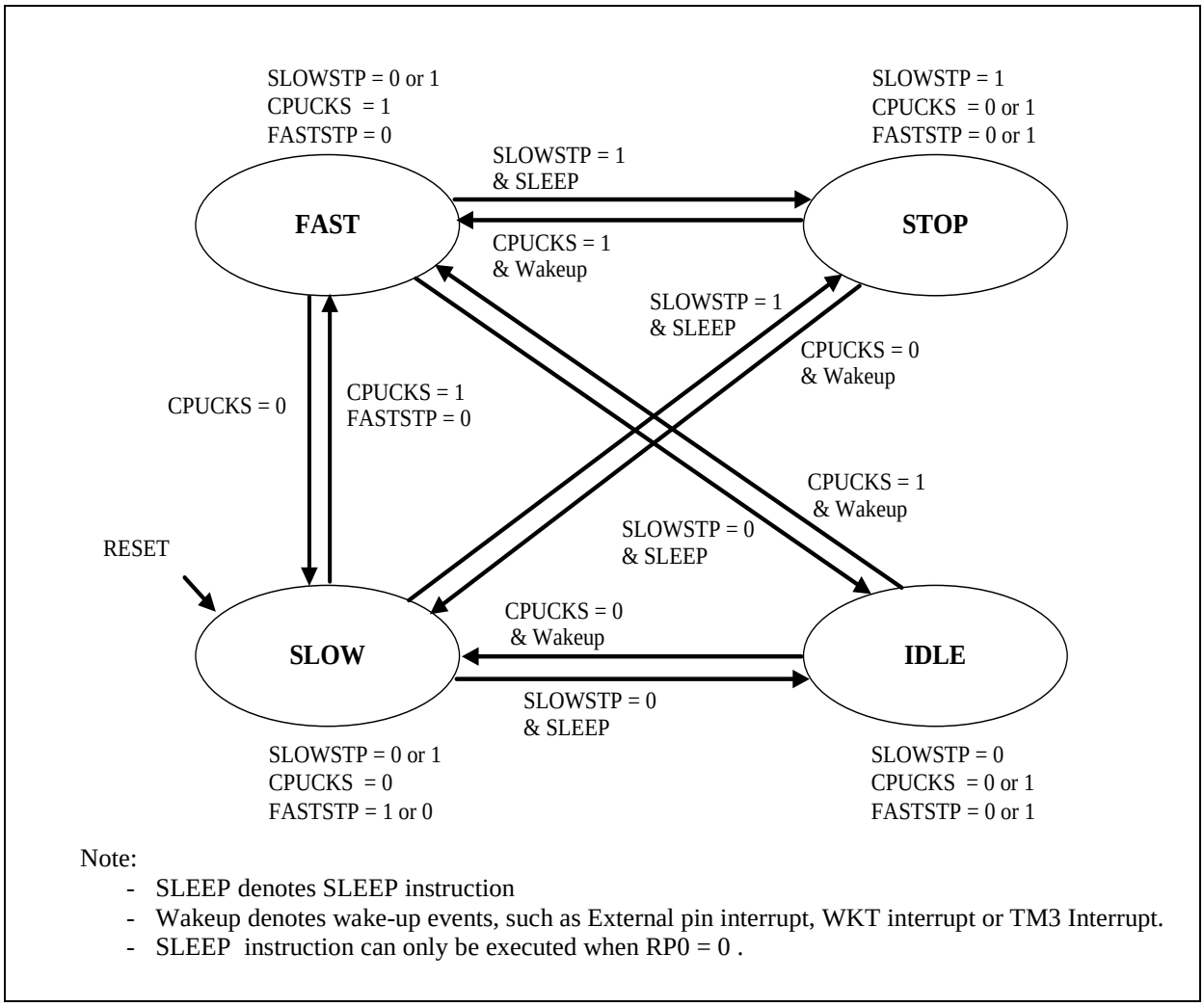
T2 and WKT/WDT are independent and have their own control registers. It is possible to keep both T2 and WKT working and wake-up in the IDLE mode.

STOP Mode:

If Slow-clock and WKT/WDT are disabled before executing the SLEEP instruction, every block is turned off and the device 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.

3.2 Dual System Clock Modes Transition

The device 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 pin interrupt, WKT interrupt or TM3 Interrupt.
- SLEEP instruction can only be executed when RP0 = 0 .

CPU Operation Block Diagram

CPU Mode & Clock Functions Table:

Mode	Oscillator	Fsys	Fast-clock	Slow-clock	TM0/TM1	T2	Wakeup event
FAST	FIRC	Fast-clock	Run	Set by SLOWSTP	Run	Run	X
SLOW	SIRC	Slow-clock	Set by FASTSTP	Run	Run	Run	X
IDLE	SIRC	Stop	Stop	Run	Stop	Run	WKT/IO/T2
STOP	Stop	Stop	Stop	Stop	Stop	Stop	IO

- **FAST mode switches to SLOW mode**

  The following steps are suggested to be executed by order when FAST mode switches to SLOW mode:

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

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

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

  ◇ Example: Switch FAST mode to SLOW mode.

```
BANKSEL   CLKCTL
MOVLW     00x101xxB
MOVWF     CLKCTL        ; Slow-clock type=SIRC
BCF       SLOWSTP       ; Enable Slow-clock.
NOP
BCF       CPUCKS        ; Fsys=Slow-clock.
BSF       FASTSTP       ; Disable Fast-clock.
```

- **SLOW mode switches to FAST mode**

  SLOW mode can be enabled by CPUCKS=0 in CLKCTL 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=1)

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

```
BANKSEL   CLKCTL
MOVLW     00001000B
MOVWF     CLKCTL        ; Fast-clock=FIRC
BCF       FASTSTP       ; Enable Fast-clock.
NOP
BSF       CPUCKS        ; Fsys=Fast-clock
```

- **IDLE mode Setting**

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

(1) Enable Slow-clock (SLOWSTP=0) or WKT(WKTIE=1)

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

(3) Execute SLEEP instruction

IDLE mode can be waken up by External interrupt, WKT interrupt and TM3 interrupt.

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

```
BANKSEL   CLKCTL
BCF       SLOWSTP      ; Enable Slow-clock.
MOVLW     0000xxxxB
MOVWF     T2CTL        ; T2 Clock source=Slow-clock. T2PSC=div 32768
BCF       RP0          ; SLEEP  instruction can only be executed when RP0 = 0
SLEEP                  ; Enter IDLE mode.
```

STOP Mode Setting

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

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

(2) Stop WKT/WDT (WKTIE=0, WDTE=10 or 0X)

(3) Execute SLEEP instruction

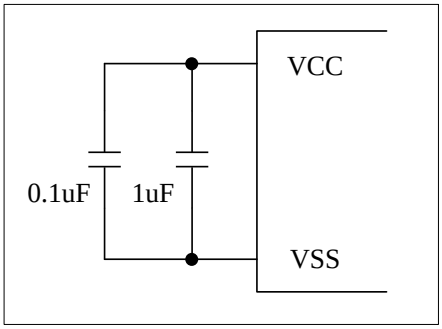
STOP mode can be waken up only by External pin interrupt.

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

```
BANKSEL   CLKCTL
BSF       SLOWSTP      ; Disable Slow-clock.
MOVLW     x000x000B    ; Disable WKT counting
MOVWF     INTIE
BCF       RP0          ; SLEEP  instruction can only be executed when RP0 = 0
SLEEP                  ; Enter STOP mode.
```

3.3 System Clock Oscillator

In the Fast Internal RC (FIRC) mode, the on-chip oscillator generates 16 MHz system clock. Since power noise degrades the performance of Internal Clock Oscillator, placing power supply bypass capacitors 1 uF and 0.1 uF very close to VCC/VSS pins improves the stability of clock and the overall system.



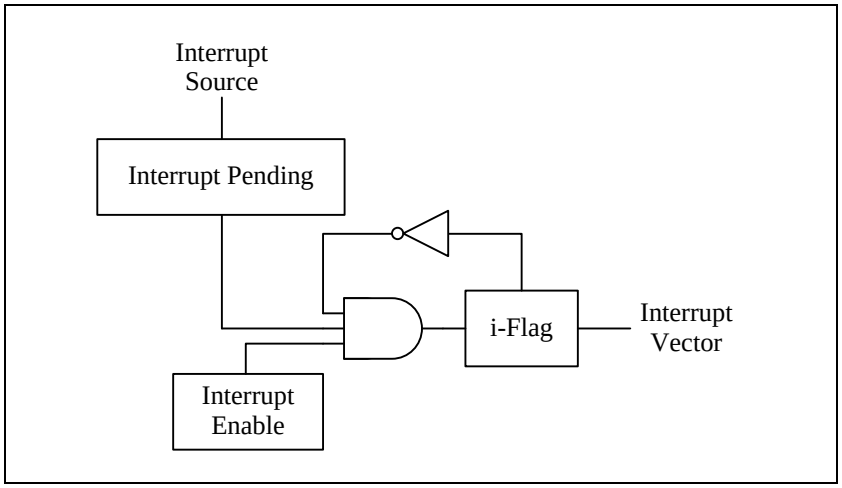
Internal RC Mode

4. Interrupt

TM55M8228 has 1 level, 1 vector and 8 interrupt sources (without Touch Key interrupt). TM55M8428 has 1 level, 1 vector and 9 interrupt sources. Each interrupt source has its own enable control bit. An interrupt event will set its individual pending flag, no matter its enable control bit is 0 or 1.

If the corresponding interrupt enable bit has been set (INTIE), it would trigger CPU to service the interrupt. CPU accepts interrupt in the end of current executed instruction cycle. In the mean while, a "CALL 004" 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.



0Bh/8Bh/10Bh/18Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	T2IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

INTIE.7 **ADCIE:** ADC interrupt enable
0: disable
1: enable

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

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

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

INTIE.3 **WKTIE:** Wakeup Timer interrupt enable
0: disable
1: enable

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

INTIE.1 **INT1IE:** INT1 (PA1) interrupt enable
0: disable
1: enable

INTIE.0 **INT0IE:** INT0 (PA6) interrupt enable
0: disable
1: enable

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	T2IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Ch.7 **ADCIF:** ADC interrupt event pending flag
This bit is set by H/W after end of ADC conversion , write 0 to this bit will clear this flag

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

0Ch.5 **TM1IF:** Timer1 interrupt event pending flag
This bit is set by H/W while Timer1 overflows, write 0 to this bit will clear this flag

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

0Ch.3 **WKTIF:** Wakeup Timer interrupt event pending flag
This bit is set by H/W while Wakeup Timer is timeout, write 0 to this bit will clear this flag

0Ch.2 **INT2IF:** INT2 (PA7) pin falling interrupt pending flag
This bit is set by H/W at INT2 pin's falling edge, write 0 to this bit will clear this flag

0Ch.1 **INT1IF:** INT1 (PA1) pin falling interrupt pending flag
This bit is set by H/W at INT1 pin's falling/rising edge, write 0 to this bit will clear this flag

0Ch.0 **INT0IF:** INT0 (PA6) pin falling/rising interrupt pending flag
This bit is set by H/W at INT0 pin's falling/rising edge, write 0 to this bit will clear this flag

1Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCON1	TKIF	TKIE	TKSOC	TKEOC	TKCHS			
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	0	0	1	1	1	1	1

1Fh.7 **TKIF:** Touch Key interrupt event pending flag,
set by H/W after end of TK conversion , write 0 to this bit or write 1 to TKSOC will clear this flag

1Fh.6 **TKIE:** Touch Key interrupt enable
0: disable
1: enable

5. I/O Port

5.1 PA0-6, PB0-1, PD0-7

These pins can be used as Schmitt-trigger input, CMOS push-pull 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 I/O pin to Mode0 or Mode1 and PxD=1. Reading the pin data (PxD) 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.

These pins can operate in four different modes as below.

Mode	PA0~PA6, PB0~PB1, PD0~PD7 pin function	PxD SFR data	Pin State	Resistor Pull-up	Digital Input
Mode 0	Open Drain	0	Drive Low	N	N
	Input	1	Pull-up	Y	Y
Mode 1	Open Drain	0	Drive Low	N	N
		1	Hi-Z	N	Y
Mode 2	CMOS Output	0	Drive Low	N	N
		1	Drive High	N	N
	Touch Key (when TKCHS)	0	TK	N	N
Mode 3	ADC	0	–	N	N
	-	1	–	Y	Y

I/O Pin Function Table

Beside I/O port function, each pin has one or more alternative functions, such as ADC and Touch Key.

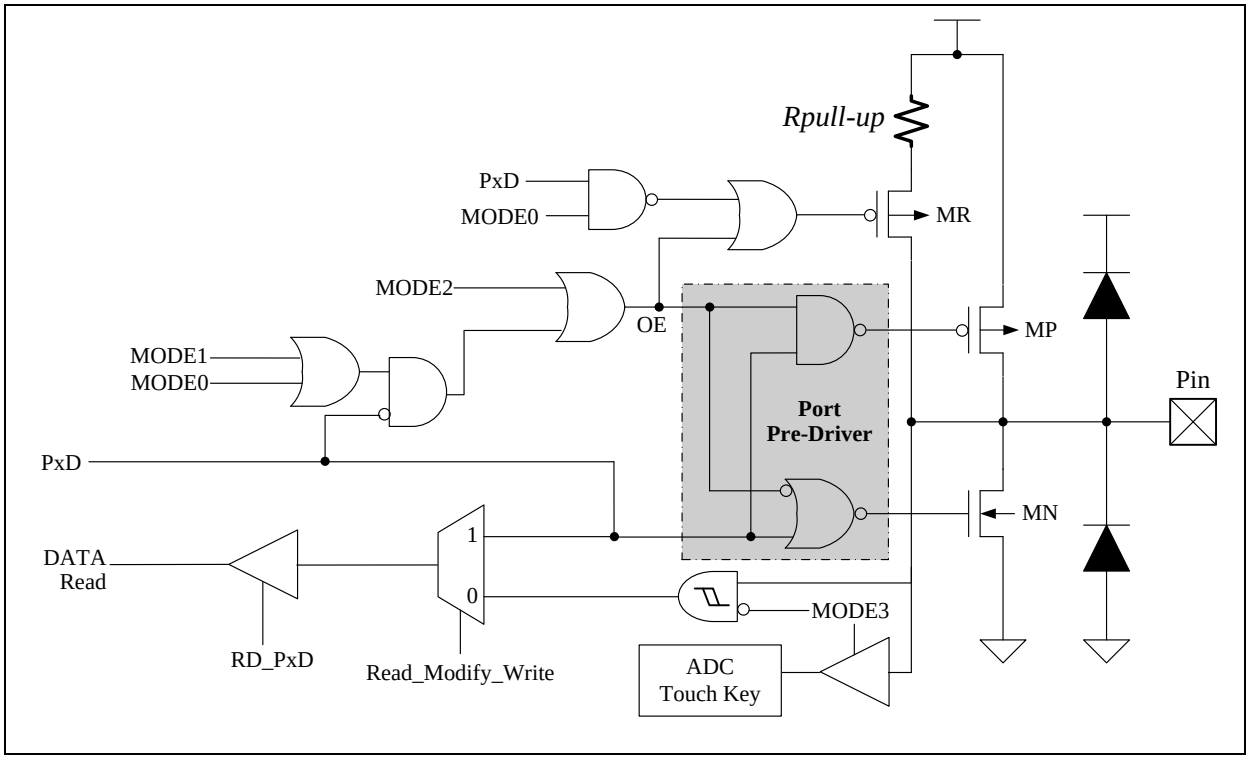
Pin Name	Wake-up	CKO	ADC/TK	others	Mode3
PA0			ADC8/TK6	PWM0N/PWM1A	ADC8
PA1	INT1		ADC1/TK1		ADC1
PA2	INT0		ADC2/TK2		ADC2
PA3				PWM0P/PWM1C	
PA4				PWM0N/PWM1B	
PA5			ADC5/TK5	PWM1C	ADC5
PA6	INT0		ADC0/TK0		ADC0
PA7	INT2				
PB0			ADC7/TK7	PWM0P/PWM1C	ADC7
PB1			ADC3/TK3	PWM1A	ADC3
PD0		TM1OUT			
PD1			ADC11		ADC11
PD2			ADC12	PWM1B	ADC12
PD3			ADC13	PWM1C	ADC13
PD4			ADC10		ADC10
PD5			ADC9		ADC9
PD6		TCOUT	ADC6		ADC6
PD7			ADC4/TK4	PWM1B	ADC4

PortA/B/D multi-function Table

The necessary SFR setting for pin's alternative function is list below.

Alternative Function	Mode	PxD SFR data	Pin State	Other necessary SFR setting
INT0, INT1	**0**	1	Input with Pull-up	INTxIE
	1	1	Input	INTxIE
TK0~TK7	**2**	0	Touch Key Idling, CMOS output Low	TKCHS
			Touch Key Scanning	TKCHS
AD0~AD14	**3**	0	ADC Channel	
PWM0P, PWM0N, PWM1A, PWM1B, PWM1C	**1**	X	PWM Output (Open Drain)	PWM0POEx PWM0NOEx PWM1AOEx PWM1BOEx PWM1COEx
	2	X	PWM Output (COMS Output)	

Mode Setting for Port Alternative Function



5.2 PA7

PA7 (VPP) can be only used as Schmitt-trigger input or open-drain output, with pull-up resistor. PA7 pin is shared with RSTn, INT2 and VPP function. When PA7 is set high, the IO port can be pulled to approximately $0.6V_{CC}$ and the current consumption is approximately 2uA. When PA7 is set low, PA7 will not have additional current consumption.

V_{CC}	PA7 pull-up approximately voltage
5V	3.1V
4V	2.3V
3V	1.5V

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

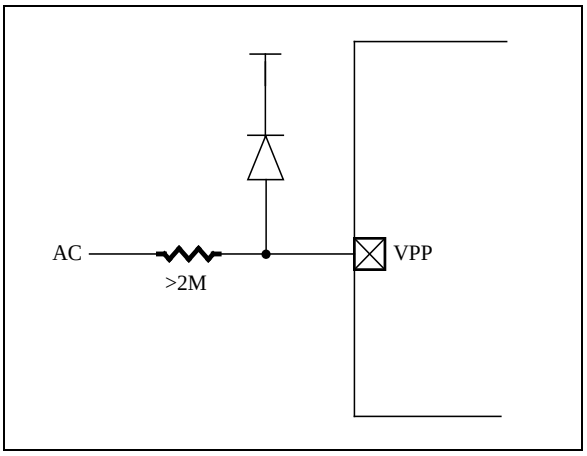
CFGWH[12]	PAD7	PN STATE	Pull-up	MODE
0	0	Low	No	open-drain output
0	1	High	Yes	input with pull-high
1	X	High	Yes	reset input with pull-high

◇ Example: Read state from PA7.

Condition: CFGWH[12] is set to “0”. If CFGWH[12] = “1”, then PA7 pin is external reset pin function.

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

VPP (PA7) has no high voltage protection diode, need an external diode and resistor to achiever AC zero crossing detection.



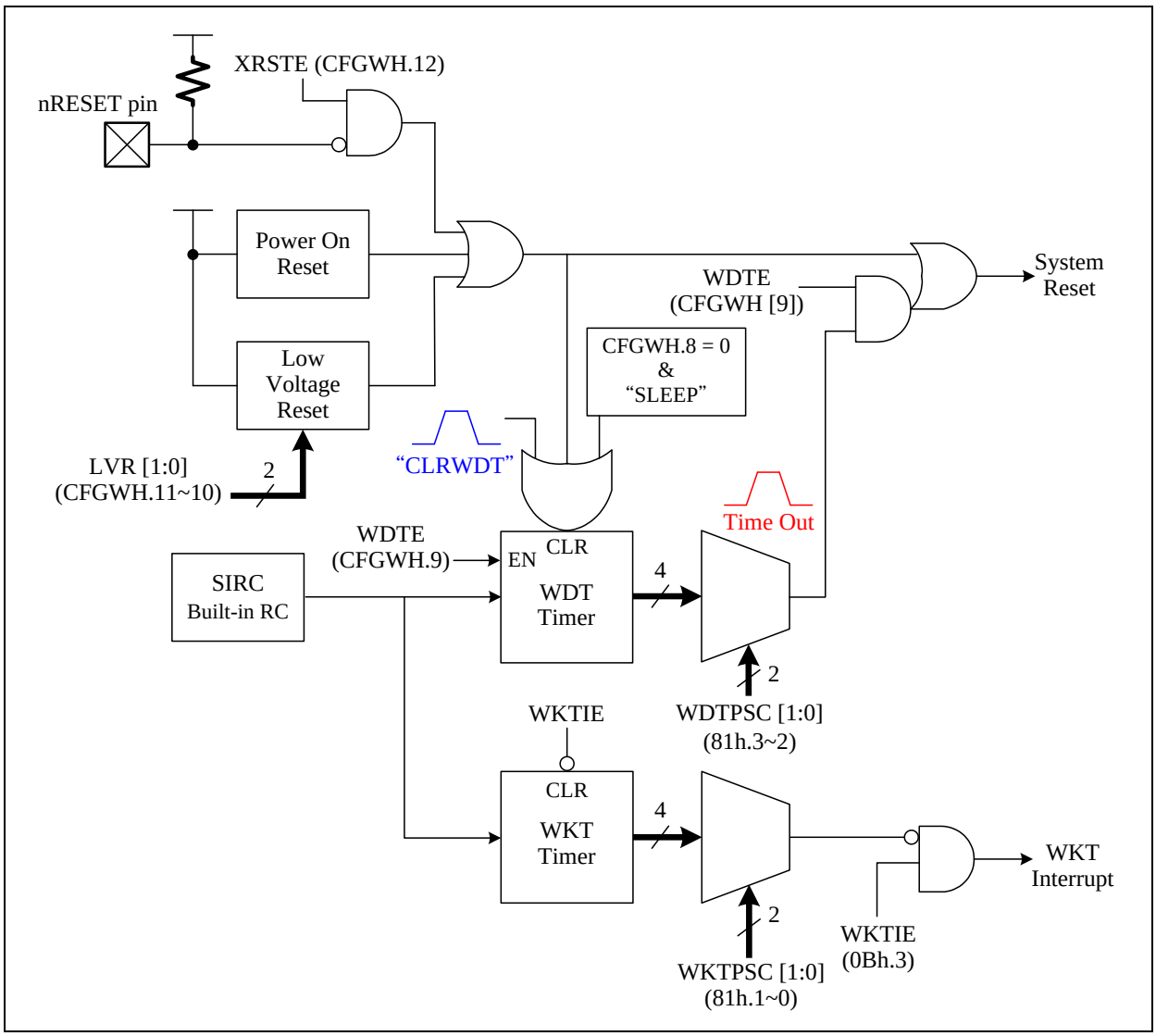
Zero crossing detector circuit for VPP pin

6. Peripheral Functional Block

6.1 Watchdog (WDT) /Wakeup (WKT) Timer

The WDT and WKT share the same built-in internal RC Oscillator and have individual own counters. The overflow period of WDT, WKT can be selected by individual prescaler (WDTPSC [1:0] , WKTPSC [1:0]) . The WDT timer is cleared by the CLRWDT instruction. If the Watchdog is enabled (CFGWH.9=WDTE=1) , the WDT generates the chip reset signal. Set CFGWH.8 to '0' can let WDT timer stop counting after executing SLEEP instruction, i.e. CFGWH.8=1 WDT timer is always keep counting even if the SLEEP instruction is executed.

The WKT timer is an interval timer, WKT time out 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 to clear watchdog timer.

◇ Example: Clear watchdog timer by CLRWDT instruction.

```
MAIN:
        …                       ; Execute program.
        BCF     RP0             ; CLRWDT  instruction can only be executed when RP0 = 0
        CLRWDT                  ; Execute CLRWDT instruction.
        …
        GOTO    MAIN
```

◇ Example: Clear watchdog timer by write 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.

```
        BANKSEL OPTION
        MOVLW   0000 01 11B
        MOVWF   OPTION          ; Select WDT Time out=256 ms @5V

        BCF     RP0             ; SLEEP  instruction can only be executed when RP0 = 0
        SLEEP
```

◇ Example: Set WKT period and interrupt function.

```
        BANKSEL OPTION
        MOVLW   00000 10 B
        MOVWF   OPTION          ; Select WKT period=64 ms @5V.

        BANKSEL INTIF
        MOVLW   1111 0 111B     ; Clear WKT interrupt request flag by using byte operation
                                ; Don’t use bit operation “BCF WKTIF” clear interrupt flag
        MOVWF   INTIF           ;

        MOVLW   0000 1 000B     ; Enable WKT interrupt function
        MOVWF   INTIE
```

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	T2IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Ch.3 **WKTIF:** Wakeup Timer interrupt event pending flag
This bit is set by H/W while Wakeup Timer is timeout, write 0 to this bit will clear this flag

0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	T2IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Bh.3 **WKTIE:** Wakeup Timer interrupt enable
0: disable
1: enable

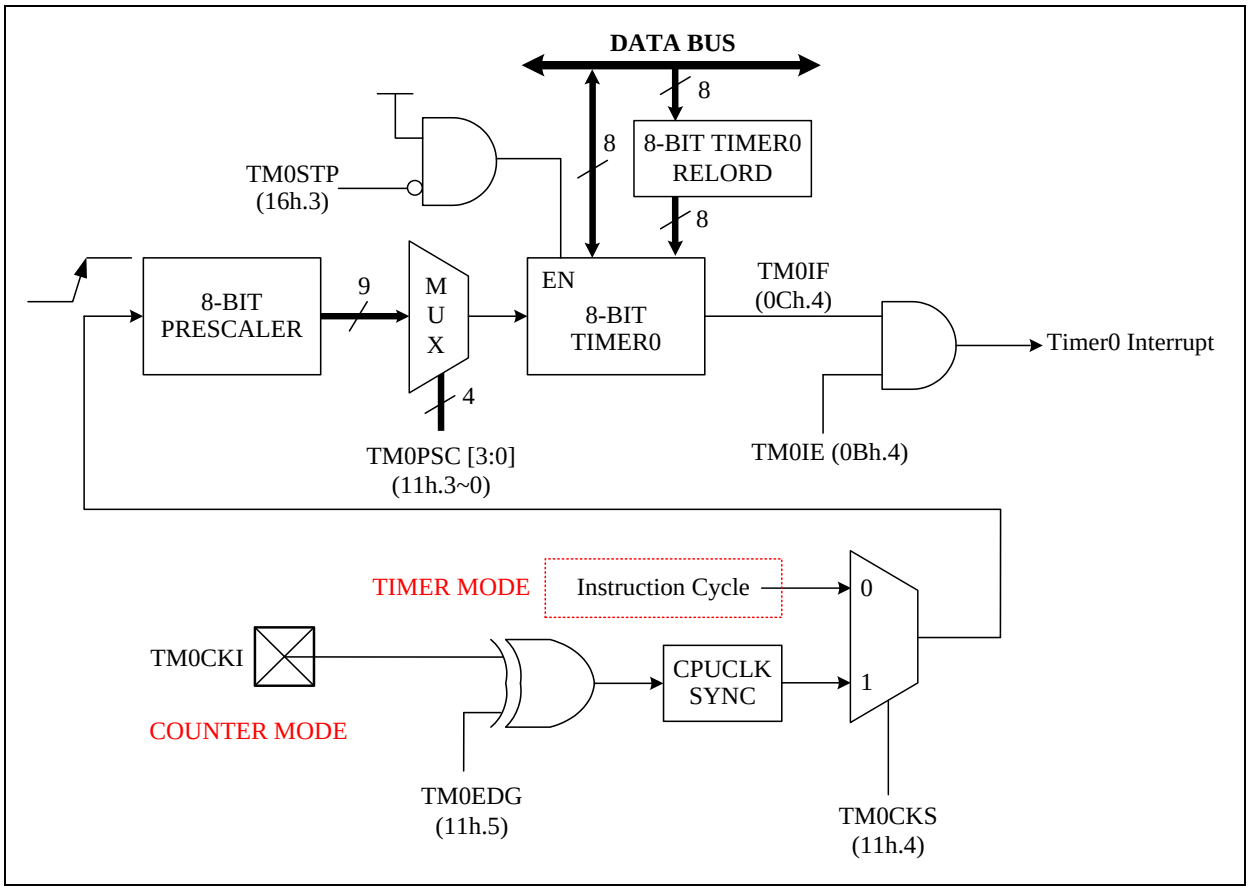
81h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTON	HWAUTO	INT0EDG	INT1EGE	INT0SEL	WDTPSC		WKTPSC	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	1	1

81h.3~2 **WDTPSC:** WDT period (@VCC=5V)
00: 128 ms 01: 256 ms 10: 1024 ms 11: 2048 ms

81h.1~0 **WKTPSC:** WKT period (@VCC=5V)
00: 16 ms 01: 32 ms 10: 64 ms 11: 128 ms

6.2 Timer0

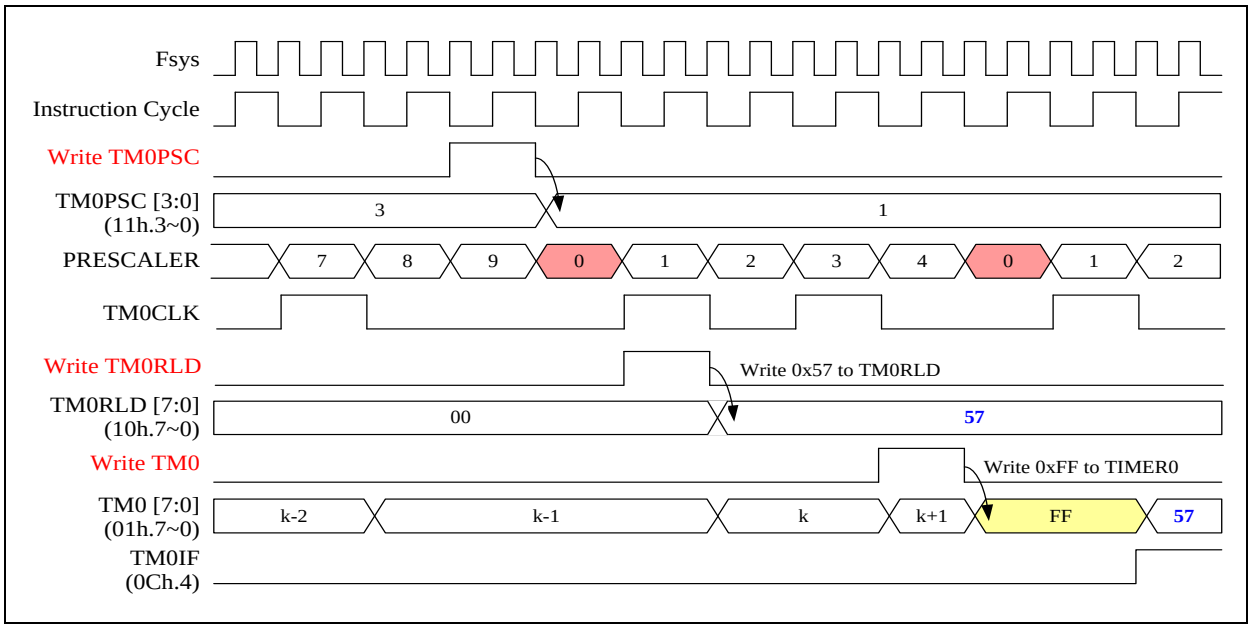
The Timer0 is an 8-bit wide register of F-Plane 01h (TM0) . It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatically rolls over a new "offset value" (TM1RLD) while it rolls over based on the pre-scaled clock source, which can be Fsys/2 or TM0CKI (PA2) rising/falling input. The Timer0 increase rate is determined by “Timer0 Pre-Scale” (TM0PSC) register. The Timer0 always generates TM0IF when its count rolls over. It generates Timer0 Interrupt if (TM0IE) is set. Timer0 can be stopped counting if the TM0STP bit is set.



Timer0 Block Diagram

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

When the Timer0 prescaler (TM0PSC) is written, the internal 8-bit prescaler will be cleared to 0 to make the counting period correct at the first Timer0 count. TM0CLK is the internal signal that causes the Timer0 to increase by 1 at the end of TM0CLK. TM0WR is also the internal signal that indicates the Timer0 is directly written by instruction; meanwhile, the internal 8-bit prescaler will be cleared. When Timer0 counts from FFh to TM0RLD, TM0IF (Timer0 Interrupt Flag) will be set to 1 and generate interrupt if TM0IE (Timer0 Interrupt Enable) is set.



Timer0 works in Timer mode (TM0CKS=0)

The equation of TM0 interrupt time value is as following:

TM0 interrupt interval cycle time=Fsys/2/TM0PSC/ (256-TM0)

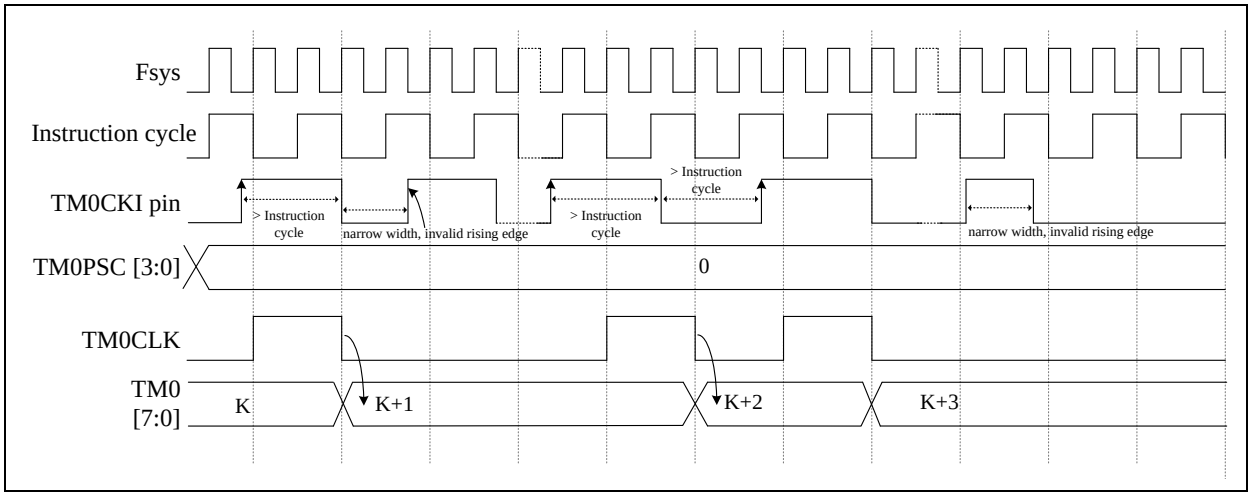
◇ Example: Setup TM0 work in Timer mode

```
; Setup TM0 clock source and divider
        BANKSEL   TM0CTL
        MOVLW     0000 0101B      ; TM0CKS=0, Setup TM0 clock= Fsys/2
        MOVWF     TM0CTL          ; TM0PSC=5, TM0PSC= Fsys/64
; Set TM0 timer.
        BSF       TM0STP          ; Disable TM0 counting (Default "0").
        MOVLW     156
        MOVWF     TM0             ; Write 156 into TM0 register
        MOVLW     124
        MOVWF     TM0RLD          ; Write 156 into TM0RLD register

; Enable TM0 timer and interrupt function.
        MOVLW     1110 1111B      ; Clear TM0 request interrupt flag by byte operation
        MOVWF     INTIF           ; F-Plane 0Ch
        MOVLW     0001 0000B      ; Enable TM0 interrupt function
        MOVWF     INTIE           ; F-Plane 0Bh
        BCF       TM0STP          ; Enable TM0 counting (Default “0”).
```

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

If TM0CKS=1 then Timer0 counter source clock is from TM0CKI pin. TM0CKI signal is synchronized by instruction cycle (Fsys/2) that means the high/low time durations of TM0CKI must be longer than one instruction cycle time (Fsys/2) to guarantee each TM0CKI's change will be detected correctly by the synchronizer.



Timer0 works in Counter mode for TM0CKI (TM0EDG=0) , TM0CKS=1

◇ Example: Setup TM0 work in Counter mode and clock source from TM0CKI pin (PA2)

```
; Setup TM0 clock source from TM0CKI pin (PA2) and divider.
        BANKSEL   TM0CTL
        MOVLW     00110000B
        MOVWF     TM0CTL        ; TM0EDG=1
                                ; Select TM0 prescaler counting edge=falling edge.
                                ; TM0CKS=1, Setup TM0 clock=TM0CKI pin (PA2)
                                ; TM0PSC=0
                                ; TM0 clock prescaler= TM0CKI 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 01H.
; Start TM0 count and read TM0 counter.
        BCF       TM0STP        ; Enable TM0 counting.
        NOP
        NOP
        NOP
        BSF       TM0STP        ; Disable TM0 counting (Default "0")

        MOVFW     TM0
```

01h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0	TM0							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

01h **TM0:** Timer0 content

0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	T2IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	T2IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

10h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0RLD	TM0RLD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

10h **TM0RLD:** Timer0 Reload Data

11h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0CTL	–	–	TM0EDG	TM0CKS	TM0PSC			
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	0	0	0	0

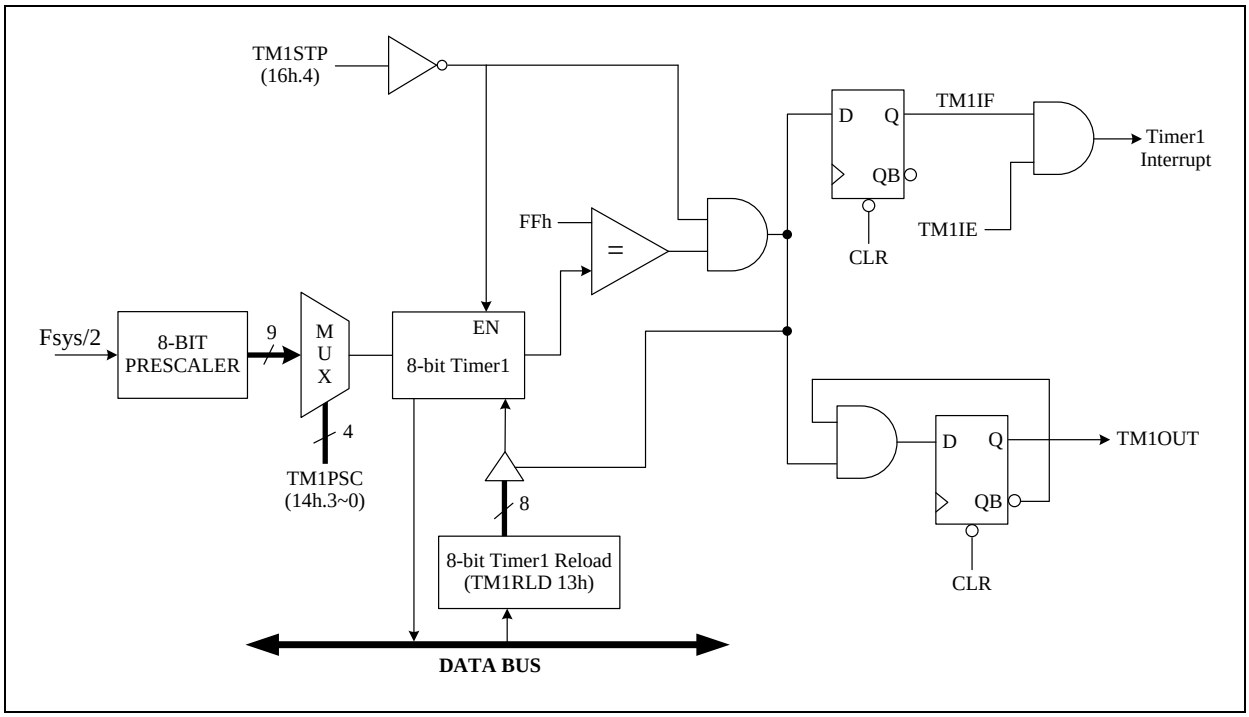
11h.5 **TM0EDG:** Timer0 prescaler counting edge for TM0CKI pin
0: rising edge 1: falling edge
11h.4 **TM0CKS:** Timer0 prescaler clock source
0:Fsys/2 1: TM0CKI pin (PA2 pin)
11h.3~0 **TM0PSC:** Timer0 prescaler. Timer0 prescaler clock source divided by
0000: /1 0001: /2 0010: /4 0011: /8 0100: /16
0101: /32 0110: /64 0111: /128 1xxx: /256

16h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF016	LVDF	LVDEN	T2CLR	TM1STP	TM0STP	LVRSAV	LVDS	
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	0	1	0	0	1	0	1

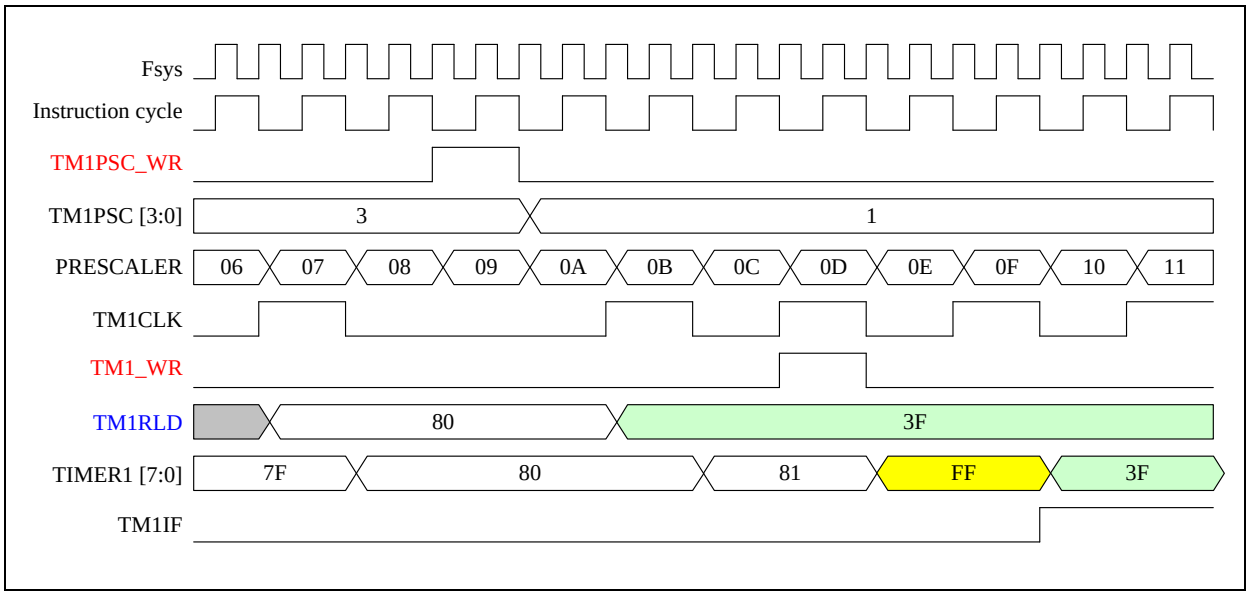
16h.3 **TM0STP:** Timer0 counter stop
0: Release 1: Stop counting

6.3 Timer1

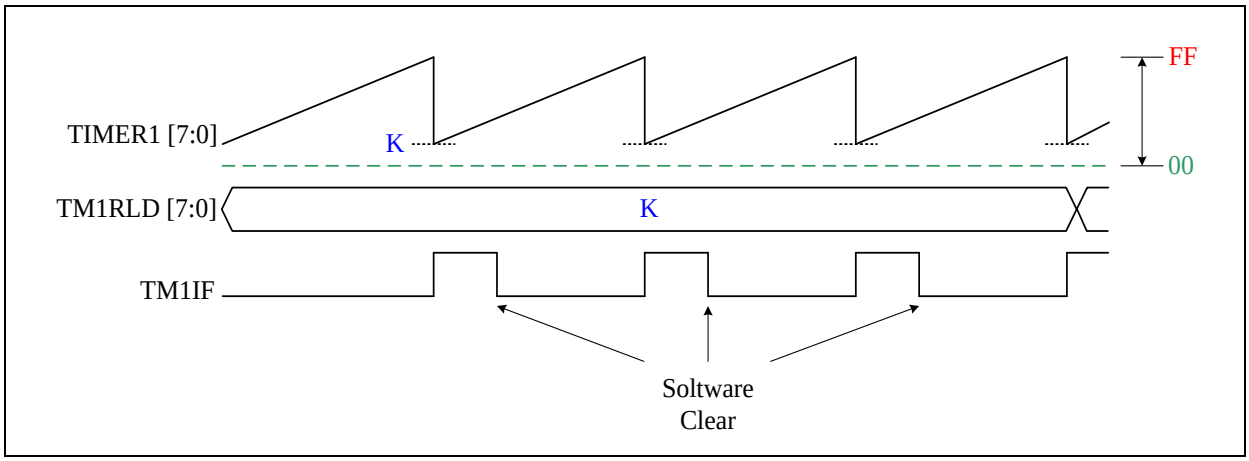
The Timer1 is an 8-bit wide register of F-Plane. It can be read or written as any other register. Besides, Timer1 increases itself periodically and automatically reloads a new "offset value" (TM1RLD) while it rolls over based on the pre-scaled instruction clock(Fsys/2). The Timer1 increase rate is determined by TM1PSC register in R-Plane. Set the TM1STP bit will stop Timer1 counting. TM1OUT is an output signal that toggles when Timer1 overflow.



Timer1 Block Diagram



Timer1 Timing Diagram



Timer1 Reload Diagram

◇ Example: Setup TM0 work in Timer mode and counting overflow toggle out to TM1OUT (PD0) configuration.

```
; Setup TM1 clock source, divider and enable TM1OUT
        BANKSEL   TM1CTL
        MOVLW     0000 0101B
        MOVWF     TM1CTL        ; TM1PSC=5 , Select TM1 clock=Fsys/64.
        BANKSEL   TM1OE
        BSF       TM1OE         ; Enable TM1OUT function pin (PD0).

; Set TM1 timer offset and stops TM1 counting
        BANKSEL   TM1STP
        BSF       TM1STP        ; Stop TM1 counting (Default "0").
        MOVLW     F0H
        MOVWF     TM1           ; Write F0H into TM1 counter

; Enable TM1 timer and interrupt function.
        MOVLW     11011111B     ; Clear TM1 request interrupt flag by byte operation
        MOVWF     INTIF         ; F-Plane 09H

        MOVLW     00100000B     ; Enable TM1 interrupt function.
        MOVWF     INTIE         ;

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

Example:

Fsys=4 MHz, TM1PSC=1, TM1 clock source=Fsys/4=1 MHz

TM1RLD=0xF0,

TM1 interrupt time= (1/1 MHz) * (0xFF – 0xF0) =1 us*16=16 us

TM1OUT output time period=16 us *2=32 us.

TM1OUT output frequency=1/32 us=31.250 KHz.

0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	T2IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	T2IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

0Ch.5 **TM1IF:** Timer1 interrupt event pending flag
This bit is set by H/W while Timer1 overflows, write 0 to this bit will clear this flag

12h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1	TM1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

12h **TM1:** Timer1 content

13h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1RLD	TM1RLD							
R/W	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

13h.7~0 **TM1RLD:** Timer1 reload offset value while it rolls over

14h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1CTL	–	–	–	–	TM1PSC			
R/W	–	–	–	–	W	W	W	W
Reset	–	–	–	–	0	0	0	0

14h.3~0 **TM1PSC:** Timer1 prescaler. Timer1 clock source divided by

0000: Fsys/2	0101: Fsys/64
0001: Fsys/4	0110: Fsys/128
0010: Fsys/8	0111: Fsys/256
0011: Fsys/16	1xxx: Fsys/512
0100: Fsys/32	

16h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF016	LVDF	LVDEN	T2CLR	TM1STP	TM0STP	LVRSAV	LVDS	
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	0	1	0	0	1	0	1

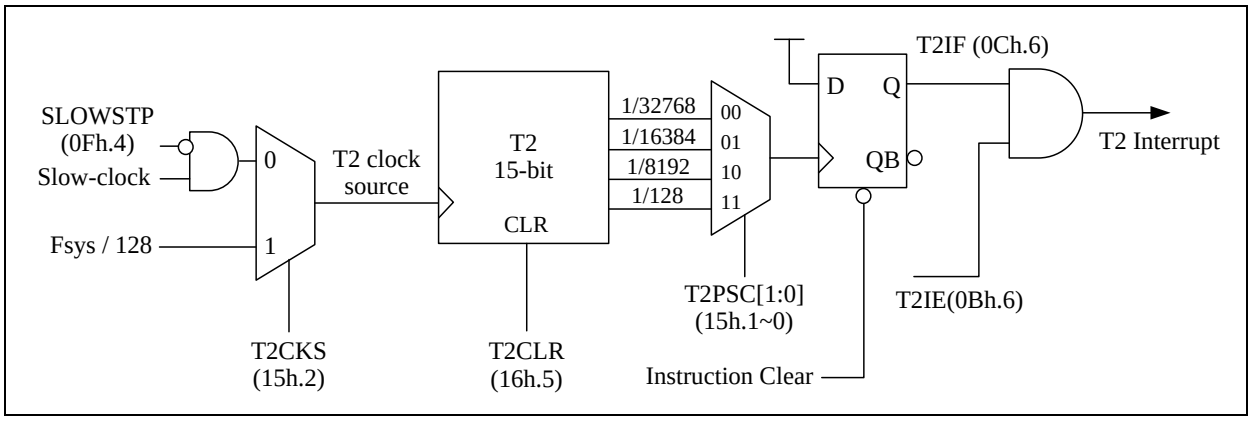
16h.4 **TM1STP:** Timer1 counter stop
0: Release
1: Stop counting

9Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF09C	TCOE	TM1OE	PWM1COE3	PWM1COE2	PWM0NOE1	PWM0NOE0	PWM0POE1	PWM0POE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F1B.5 **TM1OE:** Enable Timer1 overflow toggle output to PD0 pin (TM1OUT)

6.4 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 (0Ch.6) with the clock divided by 32768/16384/8192/128 depends on T2PSC[1:0] (15h.1~0) register bits. The following figure shows the block diagram of T2.



T2 Block Diagram

Example:

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

◇ Example:

```
; Setup T2 clock source and divider .
        BANKSEL   T2CTL
        MOVLW     00000101B      ;15h.2 (T2CKS) = 1, T2 clock source = Fsys/128
        MOVWF     T2CTL          ;15h.1~0 (T2PSC) =1, Divided by 16384

        BSF       T2CLR          ;16h.5 (T2CLR)=1, Stop T2 counting.

; Enable T2 timer and interrupt function.
        MOVLW     10111111B      ; Clear T2 request interrupt flag by byte operation
        MOVWF     INTIF          ;

        MOVLW     01000000B      ; Enable T2 interrupt function.
        MOVWF     INTIE          ;

        BCF       T2CLR          ; Enable T2 counting (Default “0”).
```

T2 clock source is Fsys/128 = 4 MHz/128 = 31250 Hz, T2PSC = /16384

T2 frequency = 31250 Hz / 16384 = 1.907 Hz

Example:

[CPU running at SLOW mode, Fsys = Slow-clock = SIRC 80Hz]

◇ Example:

```
; Setup T2 clock source and divider
        BANKSEL   T2CTL
        MOVLW     00000000B      ; 15h.2 (T2CKS) = 0, T2 clock source = Slow-clock
        MOVWF     T2CTL          ; 15.1~0 (T2PSC) =0, Divided by 32768

        BSF       T2CLR          ; Stop T2 counting.

; Enable T2 timer and interrupt function.
        MOVLW     10111111B      ; Clear T2 request interrupt flag
        MOVWF     INTIF

        MOVLW     01000000B      ; Enable T2 interrupt function.
        MOVWR     INTIE

        BCF       T2CLR          ; Enable T2 counting (Default “0”).
```

T2 clock source is Slow-clock = 80KHz, T2PSC = /32768,

T2 frequency = 80000Hz / 32768 ≒ 2.44Hz

0Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADCIE	T2IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

0Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADCIF	T2IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

0Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCTL	–	–	–	SLOWSTP	FASTSTP	CPUCKS	CPUPSC	
R/W	–	–	–	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	0	1	0	1	1

0Fh.4 **SLOWSTP:** Stop Slow-clock in Stop Mode
0: no Stop 1: Stop

15h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CTL	–	–	–	–	–	T2CKS	T2PSC	
R/W	–	–	–	–	–	R/W	R/W	R/W
Reset	–	–	–	–	–	0	0	0

15h.2 **T2CKS:** "T2 clock source" selection.
1: Fsys/128 0: Slow-clock

15h.1~0 **T2PSC:** T2 prescaler. "T2 clock source" divided by -
00: 32768 01: 16384 10: 8192 11: 128

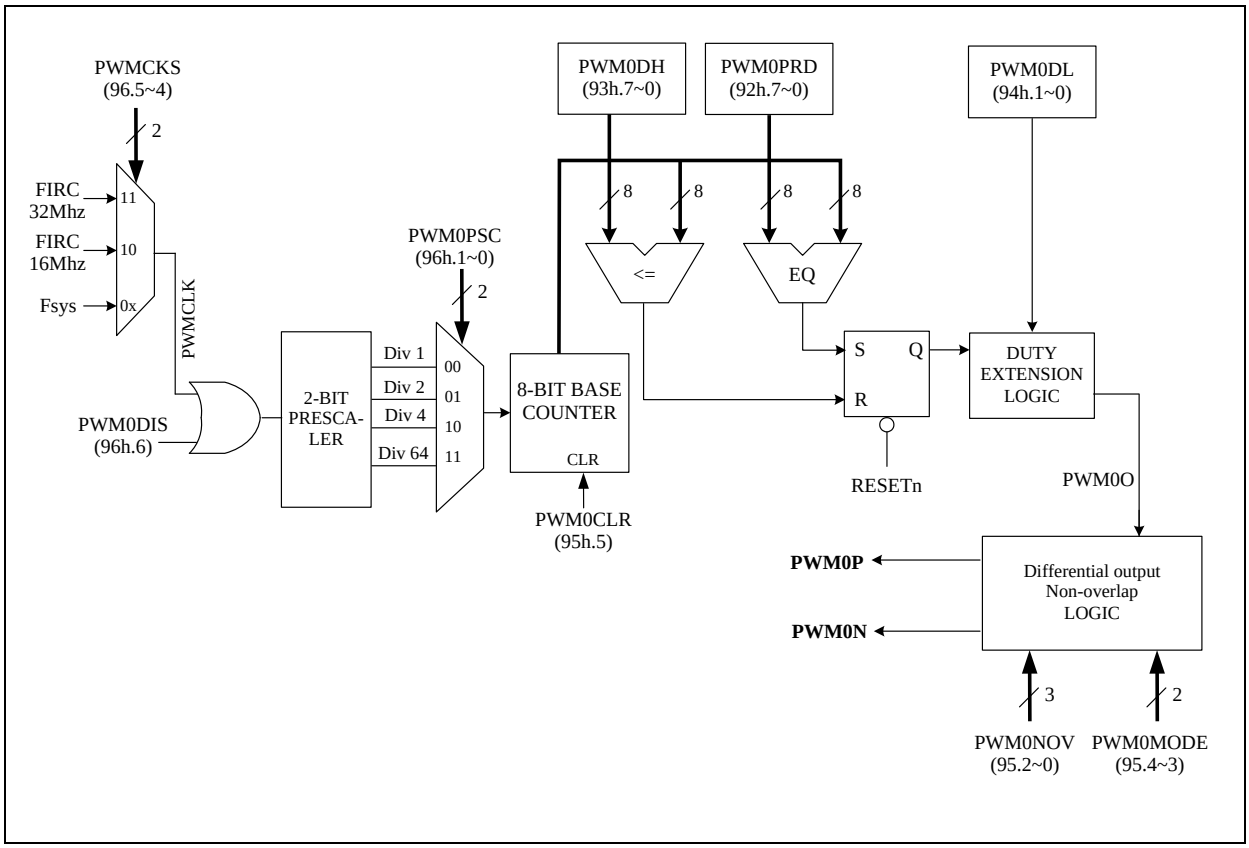
16h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF016	LVDF	LVDEN	T2CLR	TM1STP	TM0STP	LVRSAV	LVDS	
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	0	1	0	0	1	0	1

16h.5 **T2CLR:** T2 counter clear
0: Release 1: Stop counting

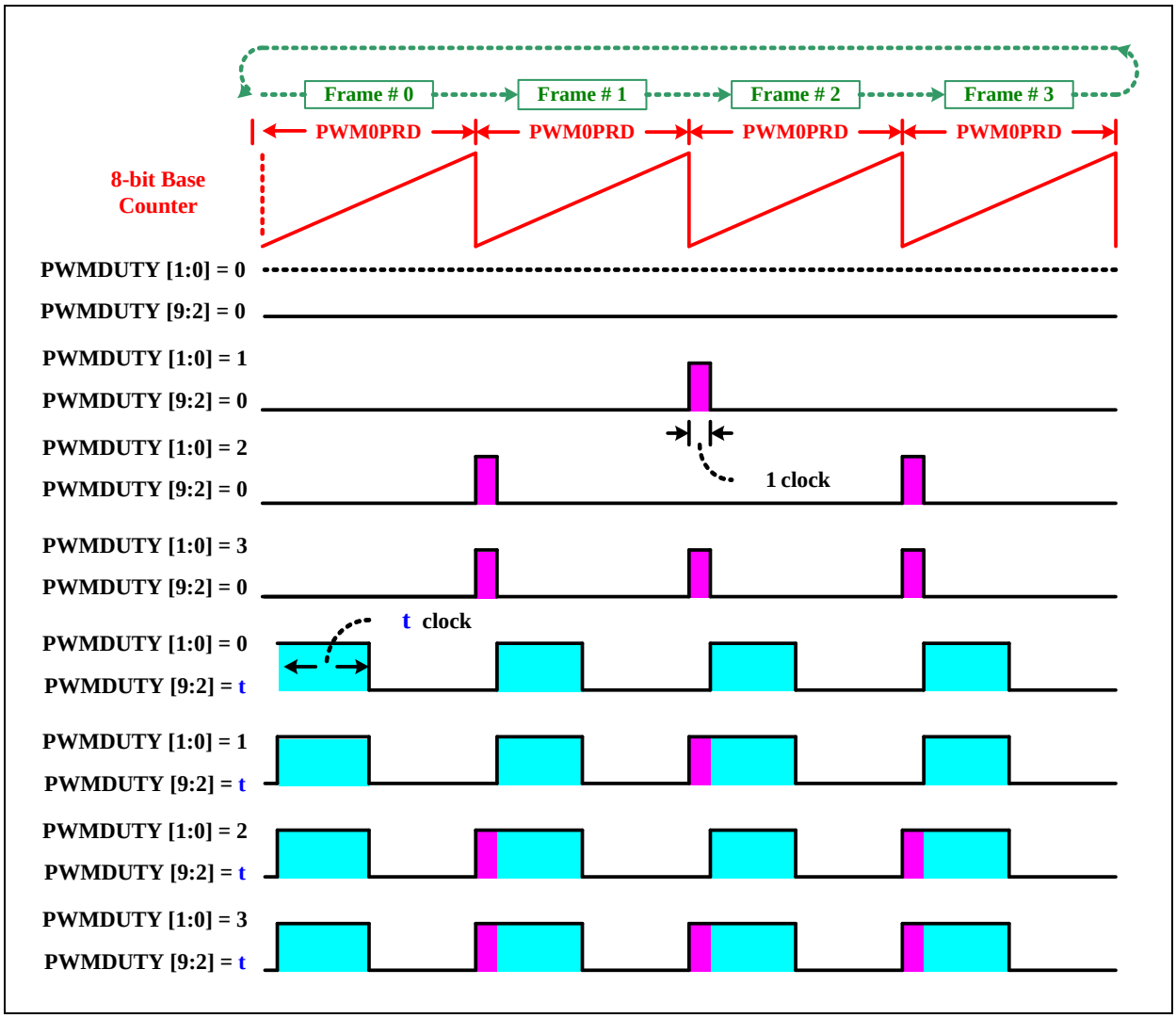
6.5 PWM0: (8+2) bits PWM

The PWM can generate various frequency waveform with 1024 duty resolution based on PWM0CLK, which can select Fsys or FIRC 16 MHz, decided by PWM0CKS (F17.4) . A spread LSB technique allows PWM0 to run its frequency at “PWM0CLK divided by 256” instead of “PWM0CLK divided by 1024”, which means the PWM is 4 times faster than normal. The advantage of higher PWM frequency is that the post RC filter can transform the PWM signal to more stable DC voltage level. The PWM output signal reset to low level whenever the 8-bit base counter matches the 8-bit MSB of PWM duty register PWM0DH (R21.7~0) . When the base counter rolls over, the 2-bit LSB of PWM duty register PWM0DL (R22.1~0) decides whether to set the PWM output signal high immediately or set it high after one clock cycle delay.

The PWM0 period can be set by writing period value to PWM0PRD register (R20) . Note that changing the PWM0PRD will immediately change the PWM0PRD values, which are different from PWM0DH/PWM0DL 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 PWM0 counter and PWM0RD, if PWM0 counter is larger than PWM0PRD after setting the PWM0PRD, a fault long PWM cycle will be generated because PWM0 counter must count to overflow then keep counting to PWM0PRD to finish the cycle.

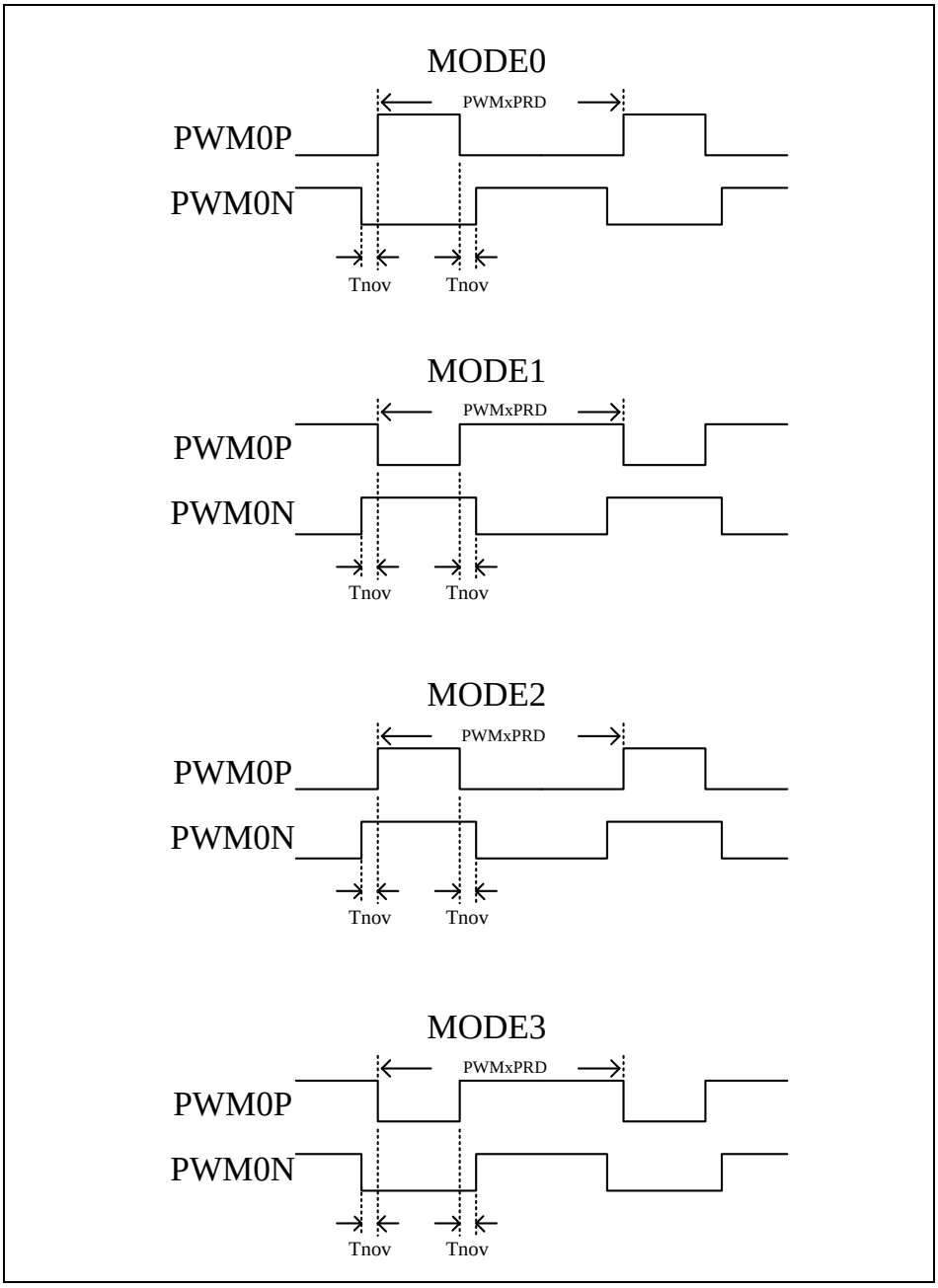


PWM0 Block Diagram



PWM0 8+2 Timing Diagram

PWM0 can be output via PWM0P and PWM0N with four different modes. The edges of the PWM pulse can be separated with 6 different time non-overlap clocks intervals (Tnov), 0s, 4 PWMCLKs, 5 PWMCLKs, 6 PWMCLKs, 7 PWMCLKs, and 8 PWMCLKs which are selected by PWM0CTL (95h). The default output form is MODE0. The waveforms of the four output modes are shown below.



PWM0 Waveform Modes

92h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0PRD	PWM0PRD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

92h.7~0 **PWM0PRD:** PWM0 period data

93h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DH	PWM0DH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

93h.7~0 **PWM0DH:** PWM0 duty 8-bit MSB

94h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DL	–	–	–	–	–	–	PWM0DL	
R/W	–	–	–	–	–	–	R/W	R/W
Reset	–	–	–	–	–	–	0	0

94h.1~0 **PWM0DL:** PWM0 duty 2-bit LSB

95h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0CTL	–	–	PWM0CLR	PWM0MODE		PWM0NOV		
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	0	0	0	0

95h.5 **PWM0CLR:** PWM0 clear and hold
0:PWM0 enable 1:PWM0 clear and hold

95h.4~3 **PWM0MODE:** PWM0 differential output mode
00 : Mode 0,
01 : Mode 1,
10 : Mode 2,
11 : Mode 3

95h2~0 **PWM0NOV:** PWM0 non-overlap control
000 : original PWM0
001 : non-overlap 4 PWM0CLKs
010 : non-overlap 5 PWM0CLKs
011 : non-overlap 6 PWM0CLKs
100 : non-overlap 7 PWM0CLKs
101 : non-overlap 8 PWM0CLKs

96h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMCTL	PWM1DIS	PWM0DIS	PWMCKS		PWM1PSC		PWM0PSC	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

96h.6 **PWM0DIS:** PWM0 Clock Disable
0:Clock Enable 1:Clock Disable

96h.5~4 **PWM0CKS:** PWM0 Clock Source
0x: Fsys
10:FIRC16M
11:FIRC32M

96h.1~0 **PWM0PSC:** PWM0 Clock Source Prescaler
00: DIV1 01: DIV2 10:DIV4 11:DIV64

9Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF09C	TCOE	TM1OE	PWM1COE3	PWM1COE2	PWM0NOE1	PWM0NOE0	PWM0POE1	PWM0POE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

9Ch.3 **PWM0NOE1:** PWM0N Output Enable 1 (priority > PWM1A Output Enable 1)
0: Disable 1:Enable, PWM0N output to PA0

9Ch.2 **PWM0NOE0:** PWM0N Output Enable 0 (priority > PWM1B Output Enable 1)
0: Disable 1:Enable, PWM0N output to PA4

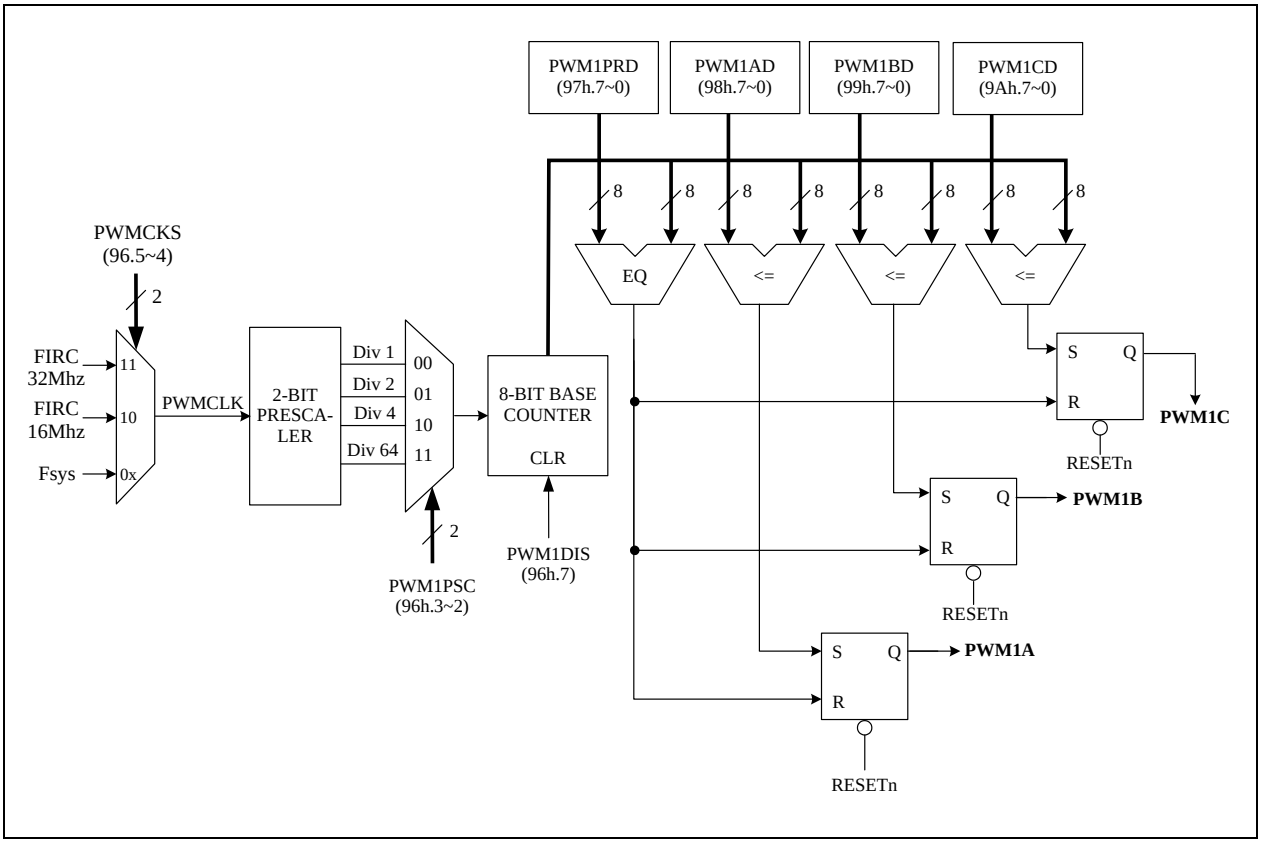
9Ch.1 **PWM0POE1:** PWM0P Output Enable 1 (priority > PWM1C Output Enable 1)
0: Disable 1:Enable, PWM0P output to PB0

9Ch.0 **PWM0POE1:** PWM0N Output Enable 0 (priority > PWM1C Output Enable 2)
0: Disable 1:Enable, PWM0P output to PA3

6.6 PWM1A/PWM1B/PWM1C: 8 bits PWMs

PWM1A/PWM1B/PWM1C are 3 PWMs which have independent duty and common period. The PWMs can generate various frequency waveform with 256 duty resolution based on PWMCLK, which can select Fsys, FIRC 16 MHz or 32Mhz, decided by PWMCKS (96h.5~4). 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 of PWM duty register PWM1AD/PWM1BD/PWM1CD.

The PWM1A/1B/1C common period can be set by writing period value to PWM1PRD register (97h). Note that changing the PWM0PRD will immediately change the PWM1PRD values. The Programmer must pay attention to the current time to change PWM1PRD by observing the following figure. There is a digital comparator that compares the PWM1A/1B/1C counter and PWM1RD, if PWM1A/1B/C counter is larger than PWM1PRD after setting the PWM1PRD, a fault long PWM cycle will be generated because PWM1A/1B/1C counter must count to overflow then keep counting to PWM1PRD to finish the cycle.



PWM1 Block Diagram

97h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1PRD	PWM1PRD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

97h.7~0 **PWM1PRD:** PWM1 period data

98h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1AD	PWM1AD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

98h.7~0 **PWM1A:** PWM1A duty 8-bit

99h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1BD	PWM1BD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

99h.7~0 **PWM1B:** PWM1B duty 8-bit

9Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CD	PWM1CD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

9Ah.7~0 **PWM1C:** PWM1C duty 8-bit

96h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMCTL	PWM1DIS	PWM0DIS	PWMCKS		PWM1PSC		PWM0PSC	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

96h.7 **PWM1DIS:** PWM1 Clock Disable and PWM1 clear
0:Clock Enable 1:Clock Disable

96h.5~4 **PWMCKS:** PWM0 Clock Source
0x: Fsys
10:FIRC16M
11:FIRC32M

96h.3~2 **PWM1PSC:** PWM1 Clock Source Prescaler
00: DIV1 01: DIV2 10:DIV4 11:DIV64

9Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF09B	PWM1COE1	PWM1COE0	PWM1BOE2	PWM1BOE1	PWM1BOE0	PWM1AOE2	PWM1AOE1	PWM1AOE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

9Bh.7 **PWM1COE1:** PWM1C Output Enable 1 (priority < PWM0P Output Enable 1)
0: Disable 1:Enable, PWM1C output to PB0

9Bh.6 **PWM1COE0:** PWM1C Output Enable 0
0: Disable 1:Enable, PWM1C output to PA5

9Bh.5 **PWM1BOE2:** PWM1B Output Enable 2
0: Disable 1:Enable, PWM1B output to PD2

9Bh.4 **PWM1BOE1:** PWM1B Output Enable 1 (priority < PWM0N Output Enable 0)
0: Disable 1:Enable, PWM1B output to PA4

9Bh.3 **PWM1BOE0:** PWM1B Output Enable 0
0: Disable 1:Enable, PWM1B output to PD7

9Bh.2 **PWM1AOE2:** PWM1B Output Enable 2
0: Disable 1:Enable, PWM1A output to PD1

9Bh.1 **PWM1AOE1:** PWM1B Output Enable 1 (priority < PWM0N Output Enable 1)
0: Disable 1:Enable, PWM1A output to PA0

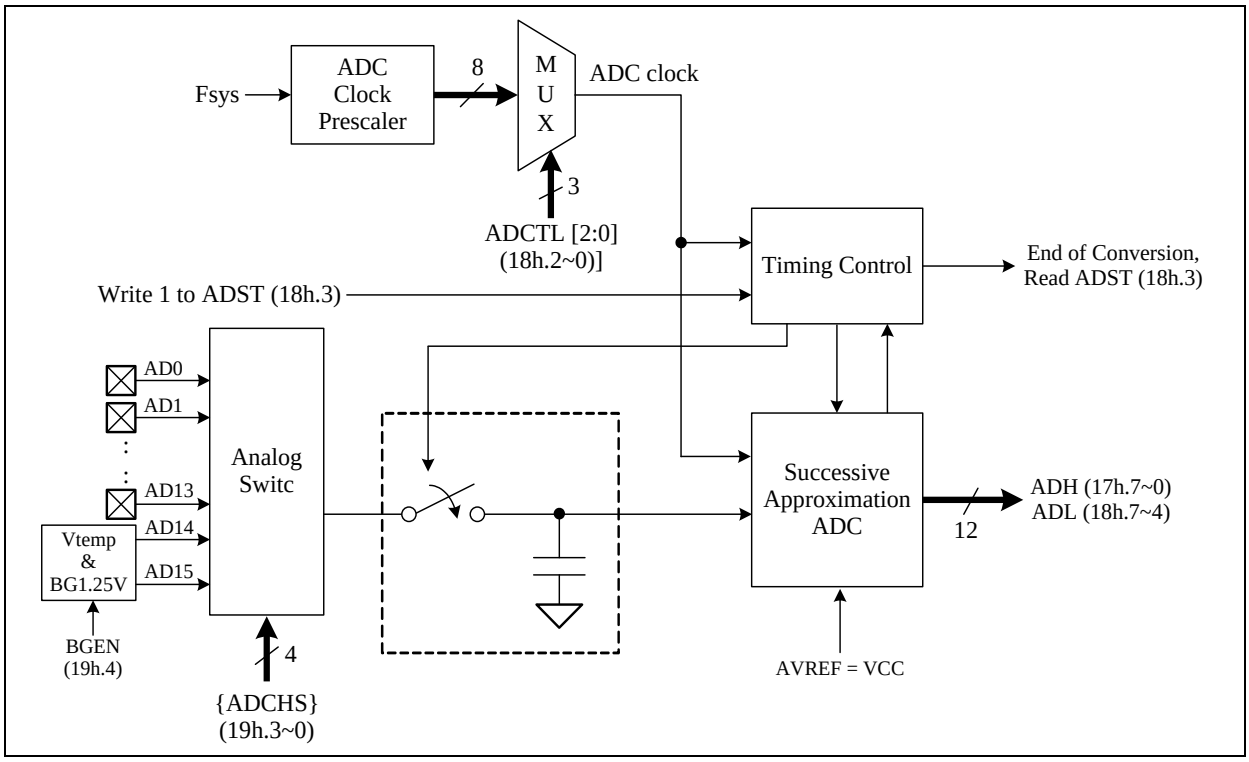
9Bh.0 **PWM1AOE0:** PWM1B Output Enable 0
0: Disable 1:Enable, PWM1A output to PB1

9Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF09C	TCOE	TM1OE	PWM1COE3	PWM1COE2	PWM0NOE1	PWM0NOE0	PWM0POE1	PWM0POE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

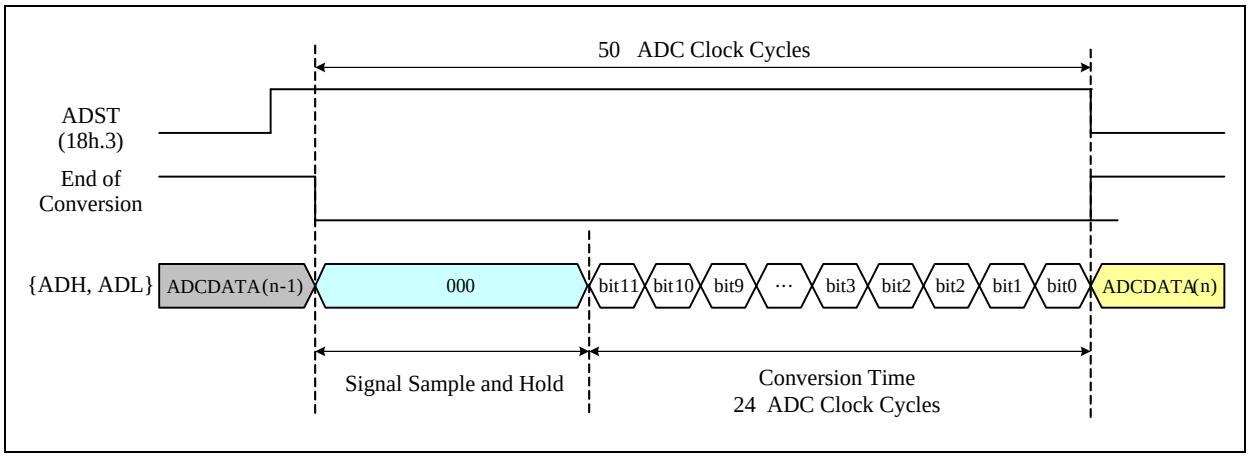
9Ch.5 **PWM1COE3:** PWM1C Output Enable 3
0: Disable 1:Enable, PWM1C output to PD3

9Ch.4 **PWM1COE2:** PWM1C Output Enable 2 (priority < PWM0P Output Enable 0)
0: Disable 1:Enable, PWM1C output to PA3

6.7 Analog-to-Digital Converter



The 12-bit ADC (Analog to Digital Converter) consists of a 16-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 (18h.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 (18h.3) control bit. After end of conversion, H/W automatic clears the ADST (18h.3) bit. User can poll this bit to know the conversion status. The PxMODE control registers are used for ADC pin configuration, user must set the Pin Mode=3 when the pin is used as an ADC input. The setting can disable the pin logical input path to save power consumption. User needs to set {ADCHS} (19h.3~0) to choose the input channel of ADC. One of them, AD15 is VBG1.25V input for ADC. Besides, AD14 is Vtemp for ADC. Both are controlled by BGEN(19h.4). ADC reference voltage is VCC.



Example:

[CPU running at FAST mode , Fsys=FIRC 16 MHz]

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

◇ Example:

```
        BANKSEL   ADCTL
        MOVLW     00000011B      ; Fsys=16 MHz
        MOVWF     ADCTL          ; 18h.2~0 (ADCKS) =ADC clock=Fsys/16=1 MHz

        BANKSEL   PAMODL
        MOVLW     0111 0101B     ; ADC2 (PA2) Pin Mode=3=ADC input
        MOVWF     PAMODL;

        BANKSEL   ADCTL
        MOVLW     00000011B      ; Fsys=16 MHz
        MOVWF     ADCTL          ; 18h.2~0 (ADCKS) =ADC clock=Fsys/32=500KHz

        MOVLW     0110 0010B
        MOVWF     MF019          ; 19h.3~0 (ADCHS [3:0]) =2, ADC select ADC2 (PA2 pin).

        BSF       ADST           ; 18h.3 (ADST) , ADC start conversion.

WAIT_ADC:
        BTFSC     ADST           ; Wait ADC conversion finish.
        GOTO      WAIT_ADC

        MOVFW     ADH            ; 17h.7~0, Read ADC result [11:4] into W
        MOVFW     ADCTL          ; 18h.7~4, Read ADC result [3:0] into W

        :
        :
```

17h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADH	ADH							
R/W	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

17h.7~0 **ADH:** ADC output data MSB, ADQ [11:4]

18h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCTL	ADL				ADST	ADCKS		
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset	–	–	–	–	0	0	0	0

18h.7~4 **ADL:** ADC output data LSB, ADQ [3:0]

18h.3 **ADST:** ADC start bit.
0: H/W clear after end of conversion
1: ADC start conversion

18h.2~0 **ADCKS:** ADC clock frequency selection:
000: Fsys/256 100: Fsys/16
001: Fsys/128 101: Fsys/8
010: Fsys/64 110: Fsys/4
011: Fsys/32 111: Fsys/2

19h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF019	–	–	TEST0	BGEN	ADCHS			
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	1	0	0	0	0

F19.5 **TEST0:** Test bit, keep this bit = 0;

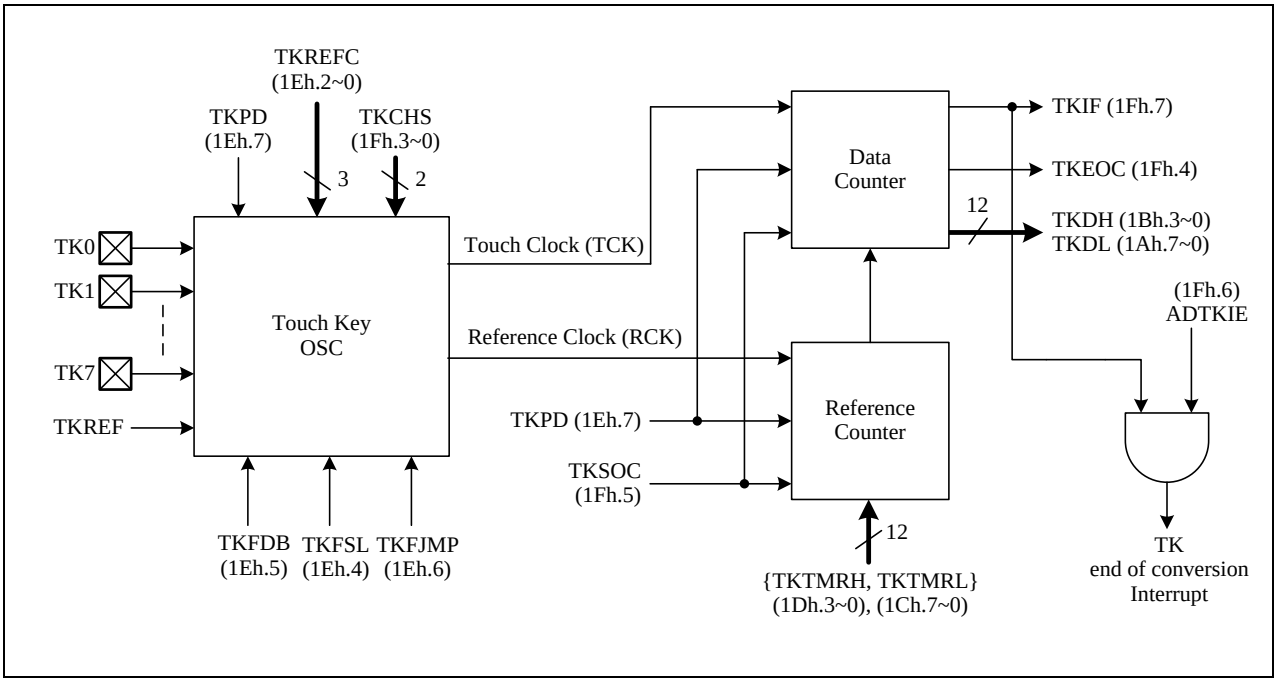
F19.4 **BGEN:** Band Gap BG1.25V & Vtemp function enable
0: Disable
1: Enable and Auto disable in STOP/IDLE mode

F19.3~0 **ADCHS:** ADC channel select

0000: ADC0 (PA6)	0110: ADC6 (PD6)	1100: ADC12(PD2)
0001: ADC1 (PA1)	0111: ADC7 (PB0)	1101: ADC13(PD3)
0010: ADC2 (PA2)	1000: ADC8(PA0)	1110: Vtemp
0011: ADC3 (PB1)	1001: ADC9(PD5)	1111: VBG 1.25V
0100: ADC4 (PD7)	1010: ADC10(PD4)	
0101: ADC5 (PA5)	1011: ADC11(PD1)	

6.8 Touch Key

The Touch Key offers an easy, simple and reliable method to implement finger touch detection. In most applications, it doesn't require any external component. The device support 8 channels touch key detection.



Touch Key Structure

To use the Touch Key, user must setup the Pin Mode (***see Section 6***) correctly as below table. Setting Mode2 for an Idling Touch Key pin can CMOS output Low and reduce the mutual interference between the adjacent keys.

PxMODx setting for Touch Key	TK0~TK7
Pin is Touch Key, Idling	CMOS output Low (Mode2)
Pin is Touch Key, Scanning	

In the Touch Key Module, there are two oscillators: Reference Clock (RCK) and Touch Clock (TCK). They are connected to the Reference Counter and Data Counter respectively. The frequency of RCK can be adjusted by setting TKREFC. Reference Counter is used to control conversion time. From starting touch key conversion to end, it will take 0 to 4096 RCK oscillation cycles by setting TKTMR. After end of conversion, user can get TKDATA (TKDH, TKDL) from Data counter. TKDATA is affected by finger touching. As finger touching TCK is getting slower, the value of TKDATA is smaller than the no finger touching. According to the difference of TKDATA, user can check if it is touched of not. A suitable TKTMR and TKREFC setting can adjust TKDATA to adapt the system board circumstances. To get the best TKREFC setting, user can try different TKREFC value (with TKFDB=0), then find the one which makes the TKDATA and TKTMR as close as possible. TKDATA can be doubled by setting TKFDB=1. In the other hand, user can adjust the overall operating frequency of the TK system (including TCK&RCK) by setting TKFSL (frequency select).

To start the Scanning, user assigns TKPD=0, then set the TKSOC bit to start touch key conversion, the TKSOC bit can be automatically cleared while end of conversion. However, if the SYSCLK is too slow, H/W might fail to clear TKSOC due to clock sampling rate. TKEOC=0 means conversion is in process. TKEOC=1 means the conversion is finish, and the touch key counting result is stored into the 12 bits TK Data Counter TKDH and TKDL.

TKIF will active at the first time enable Touch Key function (TKPD=0), user should clear TKIF after TKPD cleared.

◇ Example:

```
        BANKSEL   TKCON0
        MOVLW     00011100B     ; TKPD=0, TKFSL=TKFDB=0
        MOVWF     TKCON0        ; TKFSEL=3,TKREFC=4h
        MOVLW     04h
        MOVWF     TKTMRH
        MOVLW     00H
        MOVWF     TKTMRL        ; TKTMR=400h
        MOVLW     00000000B     ;
        MOVWF     TKCON1        ; clear TKIF , Select TK0
        BSF       TKIE          ; Enable TK interrupt
        BSF       TKSOC
WAIT_TK:
        BTFSC     TKSOC         ; Wait TK conversion finish.
        GOTO      WAIT_TK
        MOVFW     TKDH          ; 1Bh.3~0, Read TK result [11:8] into W
        MOVFW     TKDL          ; 1Ah.7~0, Read TK result [7:0] into W
```

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

1Ah.7~0 **TKDL:** Touch Key counter data bit 7~0

1Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKDH	–	–	–	–	TKDH			
R/W	–	–	–	–	R	R	R	R
Reset	–	–	–	–	–	–	–	–

1Bh.3~0 **TKDH:** Touch Key counter data bit 10~8

1Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKTMRL	TKTMRL							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

1Ch.7~0 **TKTMRL**: Touch Key reference counter LSB[7~0]

1Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKTMRH	–	–	–	–	TKTMRH			
R/W	–	–	–	–	R/W			
Reset	–	–	–	–	0	0	0	0

1Dh.3~0 **TKTMRH:** Touch Key reference counter MSB[11~8]

1Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCON0	TKPD	TKFJMP	TKFDB	TKFSEL		TKREFC		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	1	0	0

1Eh.7 **TKPD:** Touch Key power down
0: Touch Key enable 1: Touch Key disable

1Eh.6 **TKFJMP:** TCK Frequency auto adjust option
0: Disable 1: Enable

1Eh.5 **TKFDB:** Touch Key counter data double enable
0: select normal counter data 1: select counter data double

1Eh.4~3 **TKFSL:** Touch Key clock (RCK/TCK) frequency select
00: Lowest frequency
11: Highest frequency

1Eh.2~0 **TKREFC:** Touch Key reference clock capacitor select
000: smallest (RCK frequency fastest, conversion time shortest)
…
111: biggest (RCK frequency slowest, conversion time longest)

1Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCON1	TKIF	TKIE	TKSOC	TKEOC	TKCHS			
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	0	0	1	1	1	1	1

1F.7 **TKIF**: Touch Key interrupt event pending flag, set by H/W at the end of Touch Key conversion
S/W writes 0 to TKIF or sets the TKSOC bit to clear this flag.

1F.6 **TKIE:** Touch Key interrupt enable
0: Disable Touch Key interrupt
1: Enable Touch Key interrupt

1F.5 **TKSOC:** Touch Key Start of Conversion
Set 1 to start Touch Key conversion. If SYSCLK is fast enough, this bit will be cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag.

1F.4 **TKEOC:** Touch Key end of conversion flag, TKEOC may have 3uS delay after TKSOC=1, so F/W must wait enough time before polling this Flag.
0: Indicates conversion is in progress
1: Indicates conversion is finished

1F.3~0 **TKCHS:** Touch Key channel select
0000: TK0 (PA6)
0001: TK1 (PA1)
0010: TK2 (PA2)
0011: TK3 (PB1)
0100: TK4 (PD7)
0101: TK5 (PA5)
0110: TK6 (PA0)
0111: TK7 (PB0)
1xxx: TKREF

MEMORY MAP

F-Plane

Name	Address	R/W	Rst	Description
INDF (00h/80h/100h/180h)				**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
TM0 (01h/101h)				**Function related to: Timer0**
TM0	01.7~0	R/W	0	Timer0 content
PCL (02h/82h/105h/182h)				**Function related to: PROGRAM COUNT**
PCL	02.7~0	R/W	0	Programming Counter LSB [7~0]
STATUS (03h/83h/103h/183h)				**Function related to: STATUS**
IRP	03.7	R/W	0	Register Bank Select bit (used for indirect addressing)
RP1	03.6	R/W	0	Register Bank Select bit 1 (used for direct addressing)
RP0	03.5	R/W	0	Register Bank Select bit 0 (used for direct addressing)
TO	03.4	R	0	WDT timeout flag, cleared by PWRST, ‘SLEEP’ or ‘CLRWDT’ instruction
PD	03.3	R	0	Power down 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
FSR (04h/84h/104h/184h)				**Function related to: RAM W/R**
FSR	04.7~0	R/W	-	File Select Register, indirect address mode pointer
PAD (05h)				**Function related to: Port A**
PAD	05.7~0	R	-	Port A pin or “data register” state
		W	FF	Port A output data register
PBD (06h)				**Function related to: Port B**
PBD	06.7~0	R	-	Port B pin or “data register” state
		W	FF	Port B output data register
PDD (07h)				**Function related to: Port D**
PDD	07.2~0	R	-	Port D pin or “data register” state
		W	FF	Port D output data register
PCLATH (0Ah/8Ah/10Ah/18Ah)				**Function related to: PROGRAM COUNT**
PCLATH	0A.2~0	R/W	0	Write Buffer for the upper 3 bits of the Program Counter
INTIE (0Bh/8Bh/10Bh/18Bh)				**Function related to: Interrupt Enable**
ADCIE	0B.7	R/W	0	ADC interrupt enable, 1=enable, 0=disable
T2IE	0B.6	R/W	0	T2 interrupt enable, 1=enable, 0=disable
TM1IE	0B.5	R/W	0	Timer1 interrupt enable, 1=enable, 0=disable
TM0IE	0B.4	R/W	0	Timer0 interrupt enable, 1=enable, 0=disable
WKTIE	0B.3	R/W	0	Wakeup Timer interrupt enable, 1=enable, 0=disable Set 0 to clear & disable WKT timer
INT2IE	0B.2	R/W	0	INT2 pin (PA7) interrupt enable, 1=enable, 0=disable
INT1IE	0B.1	R/W	0	INT1 pin (PA1) interrupt enable, 1=enable, 0=disable
INT0IE	0B.0	R/W	0	INT0 pin (PA6 or PA2) interrupt enable, 1=enable, 0=disable

Name	Address	R/W	Rst	Description
INTIF (0Ch)				**Function related to: Interrupt Flag**
ADCIF	0C.7	R	-	ADC interrupt flag, set by H/W after end of ADC conversion
		W	0	write 0: clear this flag; write 1: no action
T2IF	0C.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	0C.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	0C.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	0C.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	0C.2	R	-	INT2 (PA7) interrupt event pending flag, set by H/W at INT2 pin's falling edge
		W	0	write 0: clear this flag; write 1: no action
INT1IF	0C.1	R	-	INT1 (PA1) interrupt event pending flag, set by H/W at INT1 pin's falling/rising edge
		W	0	write 0: clear this flag; write 1: no action
INT0IF	0C.0	R	-	INT0 (PA6) interrupt event pending flag, set by H/W at INT0 pin's falling/rising edge
		W	0	write 0: clear this flag; write 1: no action
CLKCTL (0Fh)				**Function related to: Fsys**
-	0F.7~5	-	-	Reserved
SLOWSTP	0F.4	R/W	0	Stop Slow-clock in Stop Mode 0: no Stop 1: Stop
FASTSTP	0F.3	R/W	0	Stop Fast-clock 0:no Stop 1:Stop
CPUCKS	0F.2	R/W	0	Select Fast-clock 0: Fsys=Slow-clock 1: Fsys=Fast-clock
CPUPSC	0F.1~0	R/W	11	Fsys Prescaler, 0: div 8, 1: div 4, 2: div 2, 3: div 1
TM0RLD (10h)				**Function related to: TM0**
TM0RLD	10.7~0	R/W	0	Timer0 reload Data
TM0CTL (11h)				**Function related to: TM0**
	11.7~6			
TM0EDG	11.5	R/W	0	Timer0 prescaler counting edge for TM0CKI pin 0: rising edge 1: falling edge
TM0CKS	11.4	R/W	0	Timer0 prescaler clock source 0: Fsys/2 1: TM0CKI pin (PA2 pin)
TM0PSC	11.3~0	R/W	0	Timer0 prescaler. Timer0 prescaler clock source divided by 0000: /1 0101: /32 0001: /2 0110: /64 0010: /4 0111: /128 0011: /8 1xxx: /256 0100: /16

Name	Address	R/W	Rst	Description
TM1 (12h)				**Function related to: Timer1**
TM1	12.7~0	R/W	0	Timer1 content
TM1RLD (13h)				**Function related to: Timer1**
TM1RLD	13.7~0	R/W	0	Timer1 reload Data
TM1CTL (14h)				**Function related to: Timer1**
TM1PSC	14.3~0	R/W	0	Timer1 prescaler. Timer1 clock source 0000: Fsys/2 0001: Fsys/4 0010: Fsys/8 0011: Fsys/16 0100: Fsys/32 0101: Fsys/64 0110: Fsys/128 0111: Fsys/256 1xxx: Fsys/512
T2CTL (15h)				**Function related to: T2**
T2CKS	15.2	R/W	0	T2 clock source selection. 1: Fsys/128 0: Slow-clock
T2PSC	15.1~0	R/W	0	T2 prescaler. T2 clock source divided by - 00: 32768 01: 16384 10: 8192 11: 128
MF016 (16h)				**Function related to: T2/TM1/TM0/LVR/LVD**
LVDF	16.7	R	-	Low voltage detection flag, set by H/W while Vcc ≦ LVD
LVDEN	16.6	R/W	0	Low voltage detection function enable, (When LVR=2.3V) 1=enable, 0=disable
T2CLR	16.5	R/W	1	T2 counter clear 0: Release 1: Stop counting
TM1STP	16.4	R/W	0	Timer1 counter stop 0: Release 1: Stop counting
TM0STP	16.3	R/W	0	Timer0 counter stop 0: Release 1: Stop counting
LVRSAV	16.2	R/W	1	LVR/LVD power save 1: LVR/LVD auto power off in STOP/IDLE mode 0: LVR/LVD enable in in STOP/IDLE mode
LVDS	16.1~0	R/W	01	LVD select 00: 3.6V 01: 2.8V 1x: 4.17V (When LVR=2.3V)
ADH (17h)				**Function related to: ADC**
ADH	17.7~0	R	-	ADC output data MSB, ADQ [11:4]
ADCTL (18h)				**Function related to: ADC**
ADL	18.7~4	R	-	ADC output data LSB, ADQ [3:0]
ADST	18.3	R/W	0	ADC start bit. 0: H/W clear after end of conversion 1: ADC start conversion
ADCKS	18.2~0	R/W	0	ADC clock frequency selection: 000: Fsys/256 100: Fsys/16 001: Fsys/128 101: Fsys/8 010: Fsys/64 110: Fsys/4 011: Fsys/32 111: Fsys/2

Name	Address	R/W	Rst	Description
MF019 (19h)				**Function related to: ADC**
-	19.7~6	-	-	Reserved
TEST0	19.5	R/W	0	Test bit, Keep 0
BGEN	19.4	R/W	1	Band Gap BG1.25V & Vtemp function enable 0: Disable 1: Enable and Auto disable in STOP/IDLE mode
ADCHS	19.3~0	R/W	0	ADC channel select 0000: ADC0 (PA6) 0110: ADC6 (PD6) 1100: ADC12(PD2) 0001: ADC1 (PA1) 0111: ADC7 (PB0) 1101: ADC13(PD3) 0010: ADC2 (PA2) 1000: ADC8(PA0) 1110: Vtemp 0011: ADC3 (PB1) 1001: ADC9(PD5) 1111: VBG 1.25V 0100: ADC4 (PD7) 1010: ADC10(PD4) 0101: ADC5 (PA5) 1011: ADC11(PD1)
TKDL (1Ah)				**Function related to: Touch Key**
TKDL	1A.7~0	R	-	Touch key data LSB [7~0]
TKDH (1Bh)				**Function related to: Touch Key**
TKDH	1B.3~0	R	-	Touch key data MSB [11~8]
TKTMRL (1Ch)				**Function related to: Touch Key**
TKTMRL	1C.7~0	R/W	FF	Touch Key reference counter LSB[7~0]
TKTMRH (1Dh)				**Function related to: Touch Key**
TKTMRH	1D.3~0	R/W	0	Touch Key reference counter MSB[11~8]
TKCON0 (1Eh)				**Function related to: Touch Key**
TKPD	1E.7	R/W	1	Touch Key power down 0: Touch Key enable 1: Touch Key disable
TKFJMP	1E.6	R/W	0	TCK Frequency auto adjust option 0: Disable 1: Enable
TKFDB	1E.5	R/W	0	Touch Key reference clock (RCK) double frequency enable 0: select normal RCK 1: select double RCK
TKFSEL	1E.4~3	R/W	0	Touch Key reference clock (RCK) frequency select 00: Lowest frequency 11: Highest frequency
TKREFC	1E.2~0	R/W	100	Touch Key reference clock capacitor select 000: smallest (RCK frequency fastest, conversion time shortest) … 111: biggest (RCK frequency slowest, conversion time longest)

Name	Address	R/W	Rst	Description
TKCON1 (1Fh)				**Function related to: Touch Key**
TKIF	1F.7	R/W	0	Touch Key interrupt event pending flag, set by H/W at the end of Touch Key conversion S/W writes 0 to TKIF or sets the TKSOC bit to clear this flag.
TKIE	1F.6	R/W	0	Touch Key interrupt enable 0: Disable Touch Key interrupt 1: Enable Touch Key interrupt
TKSOC	1F.5	R/W	0	Touch Key Start of Conversion Set 1 to start Touch Key conversion. If SYSCLK is fast enough, this bit will be cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag.
TKEOC	1F.4	R	1	Touch Key end of conversion flag, TKEOC may have 3uS delay after TKSOC=1, so F/W must wait enough time before polling this Flag. 0: Indicates conversion is in progress 1: Indicates conversion is finished
TKCHS	1F.3~0	R/W	0	Touch Key channel select 0000: TK0 (PA6) 0001: TK1 (PA1) 0010: TK2 (PA2) 0011: TK3 (PB1) 0100: TK4 (PD7) 0101: TK5 (PA5) 0110: TK6 (PA0) 0111: TK7 (PB0) 1xxx: TKREF
User Data Memory				
RAM	20~6F	R/W	-	RAM Bank0 area (80 Bytes)
RAM	70~7F	R/W	-	RAM common area (16 Bytes)

Name	Address	R/W	Rst	Description
OPTION (81h/181h)				**Function related to: STATUS/INT0/INT1/WDT/WKT**
HWAUTO	81.7	R/W	0	Save/Restore STATUS w/o TO, PD 0:disable 1: Enable
INT0EDG	81.6	R/W	0	INT0 pin edge interrupt event 0: falling edge to trigger 1: rising edge to trigger
INT1EDG	81.5	R/W	0	INT1 pin edge interrupt event 0: falling edge to trigger 1: rising edge to trigger
INT0SEL	81.4	R/W	0	INT0 pin select, 0: PA6 1: PA2
WDTPSC	81.3~2	R/W	11	WDT pre-scale selections: 00: 128mS 01: 256mS 10: 1024mS 11: 2048mS
WKTPSC	81.1~0	R/W	11	WKT pre-scale selections: 00: 16mS 01: 32mS 10: 64mS 11: 128mS
PAMODH (8Ch)				**Function related to: Port A**
-	8C.7~6	-	-	Reserved
PA6MOD	8C.5~4	R/W	01	PA6~PA4 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PA5MOD	8C.3~2	R/W	01	
PA4MOD	8C.1~0	R/W	01	
PAMODL (8Dh)				**Function related to: Port A**
PA3MOD	8D.7~6	R/W	01	PA3~PA0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PA2MOD	8D.5~4	R/W	01	
PA1MOD	8D.3~2	R/W	01	
PA0MOD	8D.1~0	R/W	01	
PBMODL (8Fh)				**Function related to: Port B**
-	8F.7~6	-	-	PB1~PB0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3Reserved
-	8F.5~4	-	-	
PB1MOD	8F.3~2	R/W	01	
PB0MOD	8F.1~0	R/W	01	
PDMODH (90h)				**Function related to: Port D**
PD7MOD	90.7~6	R/W	01	PD7~PD4 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PD6MOD	90.5~4	R/W	01	
PD5MOD	90.3~2	R/W	01	
PD4MOD	90.1~0	R/W	01	
PDMODL (91h)				**Function related to: Port D**
PD3MOD	91.7~6	R/W	01	PD3~PD0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PD2MOD	91.5~4	R/W	01	
PD1MOD	91.3~2	R/W	01	
PD0MOD	91.1~0	R/W	01	

Name	Address	R/W	Rst	Description
PWM0PRD (92h)				**Function related to: PWM0**
PWM0PRD	92.7~0	R/W	FF	PWM0 period data
PWM0DH (93h)				**Function related to: PWM0**
PWM0DH	93.7~0	R/W	0	PWM0 Duty MSB 8bit
PWM0DL (94h)				**Function related to: PWM0**
PWM0DL	94.1~0	R/W	0	PWM0 Duty LSB 2bit
PWM0CTL (95h)				**Function related to: PWM0**
PWM0CLR	95.5	R/W	0	PWM0 clear and hold 0:PWM0 enable 1:PWM0 clear and hold
PWM0MODE	95.4~3	R/W	0	PWM0 differential output mode 00 : Mode 0, 01 : Mode 1, 10 : Mode 2, 11 : Mode 3
PWM0NOV	95.2~0	R/W	0	PWM0 non-overlap control 000 : original PWM0 001 : non-overlap 4 PWM0CLKs 010 : non-overlap 5 PWM0CLKs 011 : non-overlap 6 PWM0CLKs 100 : non-overlap 7 PWM0CLKs 101 : non-overlap 8 PWM0CLKs
PWMCTL (96h)				**Function related to: PWM0/PWM1**
PWM1DIS	96.7	R/W	0	PWM1 Clock Disable and PWM1 clear 0:Clock Enable 1:Clock Disable
PWM0DIS	96.6	R/W	0	PWM0 Clock Disable 0:Clock Enable 1:Clock Disable
PWMCKS	96.5~4	R/W	0	PWM0 Clock Source 0x: Fsys 10:FIRC16M 11:FIRC32M
PWM1PSC	96.3~2	R/W	0	PWM1 Clock Source Prescaler 00: DIV1 01: DIV2 10:DIV4 11:DIV64
PWM0PSC	96.1~0	R/W	0	PWM0 Clock Source Prescaler 00: DIV1 01: DIV2 10:DIV4 11:DIV64
PWM1PRD (97h)				**Function related to: PWM1**
PWM1PRD	97.7~0	R/W	FF	PWM1 (PWM1A/PWM1B/PWM1C) period data
PWM1AD (98h)				**Function related to: PWM1A**
PWM1AD	98.7~0	R/W	0	PWM1A Duty
PWM1BD (99h)				**Function related to: PWM1B**
PWM1BD	99.7~0	R/W	0	PWM1B Duty
PWM1CD (9Ah)				**Function related to: PWM1C**
PWM1CD	9A.7~0	R/W	0	PWM1C Duty

Name	Address	R/W	Rst	Description
MF09B (9Bh)				**Function related to: PWM1A/PWM1B/PWM1C**
PWM1COE1	9B.7	R/W	0	PWM1C Output Enable 1 (priority < PWM0P Output Enable 1) 0: Disable 1:Enable, PWM1C output to PB0
PWM1COE0	9B.6	R/W	0	PWM1C Output Enable 0 0: Disable 1:Enable, PWM1C output to PA5
PWM1BOE2	9B.5	R/W	0	PWM1B Output Enable 2 0: Disable 1:Enable, PWM1B output to PD2
PWM1BOE1	9B.4	R/W	0	PWM1B Output Enable 1 (priority < PWM0N Output Enable 0) 0: Disable 1:Enable, PWM1B output to PA4
PWM1BOE0	9B.3	R/W	0	PWM1B Output Enable 0 0: Disable 1:Enable, PWM1B output to PD7
PWM1AOE2	9B.2	R/W	0	PWM1A Output Enable 2 0: Disable 1:Enable, PWM1A output to PD1
PWM1AOE1	9B.1	R/W	0	PWM1A Output Enable 1 (priority < PWM0N Output Enable 1) 0: Disable 1:Enable, PWM1A output to PA0
PWM1AOE0	9B.0	R/W	0	PWM1A Output Enable 0 0: Disable 1:Enable, PWM1A output to PB1
MF09C (9Ch)				**Function related to: PWM0/PWM1/TM1/TC**
TCOE	9C.7	R/W	0	TCOUT Output Enable 0: Disable 1: Enable, output to PD6
TM1OE	9C.6	R/W	0	Timer1 overflow toggle Output Enable 0: Disable 1:Enable, output to PD0
PWM1COE3	9C.5	R/W	0	PWM1C Output Enable 3 0: Disable 1:Enable, PWM1C output to PD3
PWM1COE2	9C.4	R/W	0	PWM1C Output Enable 2 (priority < PWM0P Output Enable 0) 0: Disable 1:Enable, PWM1C output to PA3
PWM0NOE1	9C.3	R/W	0	PWM0N Output Enable 1 (priority > PWM1A Output Enable 1) 0: Disable 1:Enable, PWM0N output to PA0
PWM0NOE0	9C.2	R/W	0	PWM0N Output Enable 0 (priority > PWM1B Output Enable 1) 0: Disable 1:Enable, PWM0N output to PA4
PWM0POE1	9C.1	R/W	0	PWM0P Output Enable 1 (priority > PWM1C Output Enable 1) 0: Disable 1:Enable, PWM0P output to PB0
PWM0POE0	9C.0	R/W	0	PWM0P Output Enable 0 (priority > PWM1C Output Enable 2) 0: Disable 1:Enable, PWM0P output to PA3
User Data Memory				
RAM	A0~EF	R/W	-	RAM Bank1 area (80 Bytes)
TESTREG (105h)				**Function related to: TEST**
TESTREG	105.1~0	R/W	11	Test bits, Keep 11
LVRPD (109h)				**Function related to: LVR**
LVRPD	109	W	-	Write 0x37 to force LVR disable
User Data Memory				
RAM	120~16F	R/W	-	RAM Bank2area (80 Bytes)

Name	Address	R/W	Rst	Description
DPL				**Function related to: Table Read**
DPL	185.7~0	R/W	0	Table read low address, data ROM pointer (DPTR) low byte
DPH				**Function related to: Table Read**
DPH	186.3~0	R/W	0	Table read high address, data ROM pointer (DPTR) high byte
(F18F) IRCF				**Function related to: Internal RC**
IRCF	18f.6~0	R/W		FIRC frequency adjustment: 00h: Lowest frequency 7Fh: Highest frequency

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” 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
b	Bit address
k	Literal. Constant data or label
d	Destination selection field, 0: Working register, 1: Register file
W	Working Register
Z	Zero Flag
C	Carry Flag or/Borrow Flag
DC	Decimal Carry Flag or Decimal/Borrow Flag
PC	Program Counter
TOS	Top Of Stack
GIE	Global Interrupt Enable Flag (i-Flag)
[]	Option Field
( )	Contents
.	Bit Field
B	Before
A	After
←	Assign direction

Mnemonic		Op Code	Cycle	Flag Affect	Description
Byte-Oriented File Register Instruction					
ADDWF	f, d	00 0111 dfff ffff	1	C, DC, Z	Add W and "f"
ANDWF	f, d	00 0101 dfff ffff	1	Z	AND W with "f"
CLRF	F	00 0001 1fff ffff	1	Z	Clear "f"
CLRW		00 0001 0100 0000	1	Z	Clear W
COMF	f, d	00 1001 dfff ffff	1	Z	Complement "f"
DECF	f, d	00 0011 dfff ffff	1	Z	Decrement "f"
DECFSZ	f, d	00 1011 dfff ffff	1 or 2	-	Decrement "f", skip if zero
INCF	f, d	00 1010 dfff ffff	1	Z	Increment "f"
INCFSZ	f, d	00 1111 dfff ffff	1 or 2	-	Increment "f", skip if zero
IORWF	f, d	00 0100 dfff ffff	1	Z	OR W with "f"
MOVF	f,d	00 1000 dfff ffff	1	Z	Move "f"
MOVFW	f	00 1000 0fff ffff	1	Z	Move "f" to W
MOVWF	f	00 0000 1fff ffff	1	-	Move W to "f"
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"
TSTF	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	11 00bb bfff ffff	1	-	Clear "b" bit of "f"
BSF	f, b	11 01bb bfff ffff	1	-	Set "b" bit of "f"
BTFSC	f, b	11 10bb bfff ffff	1 or 2	-	Test "b" bit of "f", skip if clear
BTFSS	f, b	11 11bb bfff 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 0kkk kkkk kkkk	2	-	Call subroutine "k"
CLRWDT		01 1110 0000 0100	1	TO, PD	Clear Watch Dog Timer
GOTO	k	10 1kkk 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		01 1110 0000 0011	1	TO, PD	Go into Power-down mode, Clock oscillation stops
SUBLW	k	01 1111 kkkk kkkk	1	C, DC, Z	Subtract W from literal
TABRH		00 0000 0101 1000	2	-	Lookup ROM high data to W
TABRL		00 0000 0101 0000	2	-	Lookup ROM low data to W
XORLW	k	01 1101 kkkk kkkk	1	Z	XOR Literal "k" with W

Note: SLEEP / CLRWDT instructions can only be executed when RP0 = 0 .

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 ~ 7Fh, b : 0 ~ 7
Operation	(f.b) ← 0
Status Affected	-
OP-Code	11 00bb bfff 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 ~ 7Fh, b : 0 ~ 7
Operation	(f.b) ← 1
Status Affected	-
OP-Code	11 01bb bfff 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 ~ 7Fh, b : 0 ~ 7
Operation	Skip next instruction if (f.b) =0
Status Affected	-
OP-Code	11 10bb bfff 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 — B : PC =LABEL1 TRUE GOTO SUB1 — A : if FLAG.1 =0, PC =FALSE FALSE ... — if FLAG.1 =1, PC =TRUE

BTFSS — Test "b" bit of "f", skip if set(1)

Syntax	BTFSS f [,b]
Operands	f : 00h ~ 7Fh, b : 0 ~ 7
Operation	Skip next instruction if (f.b) =1
Status Affected	-
OP-Code	11 11bb bfff 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 — B : PC =LABEL1 TRUE GOTO SUB1 — A : if FLAG.1 =0, PC =TRUE FALSE ... — if FLAG.1 =1, PC =FALSE

CALL	Call subroutine "k"	
Syntax	CALL k	
Operands	k : 000h ~ 7FFh	
Operation	Operation: TOS ← (PC) + 1, PC.10~0 ← k	
Status Affected	-	
OP-Code	10 0kkk kkkk kkkk	
Description	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The 11-bit immediate address is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH. CALL is a two-cycle instruction.	
Cycle	2	
Example	LABEL1 CALL SUB1	B : PC =LABEL1 A : PC =SUB1, TOS =LABEL1 + 1

CLRF	Clear "f"	
Syntax	CLRF f	
Operands	f : 00h ~ 7Fh	
Operation	(f) ← 00h, Z ← 1	
Status Affected	Z	
OP-Code	00 0001 1fff ffff	
Description	The contents of register 'f' are cleared and the Z bit is set.	
Cycle	1	
Example	CLRF FLAG_REG	B : FLAG_REG =0x5A A : FLAG_REG =0x00, Z =1

CLRW	Clear W	
Syntax	CLRW	
Operands	-	
Operation	(W) ← 00h, Z ← 1	
Status Affected	Z	
OP-Code	00 0001 0100 0000	
Description	W register is cleared and Z bit is set.	
Cycle	1	
Example	CLRW	B : W =0x5A A : W =0x00, Z =1

CLRWDT	Clear Watchdog Timer	
Syntax	CLRWDT	
Operands	-	
Operation	WDT/WKT Timer ← 00h	
Status Affected	TO, PD	
OP-Code	01 1110 0000 0100	
Description	CLRWDT instruction clears the Watchdog/Wakeup Timer	
Cycle	1	
Example	CLRWDT	B : WDT counter =? A : WDT counter =0x00

Note: CLRWDT instruction can only be executed when RP0 = 0

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 ~ 7FFh
Operation	PC.10~0 ← k
Status Affected	-
OP-Code	10 1kkk kkkk kkkk
Description	GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>.The upper bits of PC are loaded from PCLATH. 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

MOVF	Move f	
Syntax	MOVF f,d	
Operands	f : 00h ~ 7Fh	
Operation	(destination) ← (f)	
Status Affected	Z	
OP-Code	00 1000 dfff ffff	
Description	The contents of register 'f' are moved to a destination dependent upon the status of d. If d=0, destination is W register. If d =1, the destination is file register f itself. d=1 is useful to test a file register, since status flag Z is affected.	
Cycle	1	
Example	MOVF FSR,0	B : FSR =0xC2, W =? A : FSR =0xC2, W 0xC2

MOVFW	Move "f" to W	
Syntax	MOVFW f	
Operands	f : 00h ~ 7Fh	
Operation	(W) ← (f)	
Status Affected	Z	
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

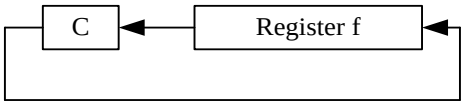
NOP	No Operation	
Syntax	NOP	
Operands	-	
Operation	No Operation	
Status Affected	-	
OP-Code	00 0000 0000 0000	
Description	No Operation	
Cycle	1	
Example	NOP	-

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

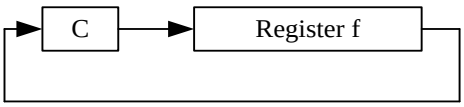
RETI	Return from Interrupt	
Syntax	RETI	
Operands	-	
Operation	PC ← TOS, GIE ← 1	
Status Affected	-	
OP-Code	00 0000 0110 0000	
Description	Return from Interrupt. Stack is POPed and Top-of-Stack (TOS) is loaded in to the PC. Interrupts are enabled. This is a two-cycle instruction.	
Cycle	2	
Example	RETI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

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

RLF — Rotate Left "f" through Carry

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

RRF — Rotate Right "f" through Carry

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

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

Syntax	SLEEP
Operands	-
Operation	-
Status Affected	TO, PD
OP-Code	01 1110 0000 0011
Description	Go into Power-down mode with the oscillator stops.
Cycle	1
Example	SLEEP — -

Note: SLEEP instruction can only be executed when RP0 = 0 .

SUBLW — Subtract W from Literal

Syntax	SUBLW k
Operands	k : 00h ~ FFh
Operation	(W) ← (W) + k
Status Affected	C, DC, Z
OP-Code	01 1111 kkkk kkkk
Description	The W register is subtracted (2's complement method) from the eight-bit literal "k". The result is placed in the W register.
Cycle	1
Example	SUBLW 0x15 — B : W =0x25 A : W =0x10

SUBWF — Subtract W from "f"

Syntax	SUBWF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	(destination) ← (f) – (W)
Status Affected	C, DC, Z
OP-Code	00 0010 dfff ffff
Description	Subtract (2's complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Cycle	1
Example	SUBWF REG1, 1 — B : REG1 =0x03, W =0x02, C =?, Z =? A : REG1 =0x01, W =0x02, C =1, Z =0
	SUBWF REG1, 1 — B : REG1 =0x02, W =0x02, C =?, Z =? A : REG1 =0x00, W =0x02, C =1, Z =1
	SUBWF REG1, 1 — B : REG1 =0x01, W =0x02, C =?, Z =? A : REG1 =0xFF, W =0x02, C =0, Z =0

SWAPF — Swap Nibbles in "f"

Syntax	SWAPF f [,d]
Operands	f : 00h ~ 7Fh, d : 0, 1
Operation	(destination,7~4) ← (f.3~0), (destination.3~0) ← (f.7~4)
Status Affected	-
OP-Code	00 1110 dfff ffff
Description	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.
Cycle	1
Example	SWAPF REG, 0 — B : REG1 =0xA5 A : REG1 =0xA5, W =0x5A

TABRH Return DPTR high byte to W

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

```
MOVLW   (TAB1&0xFF)
MOVWF   FSR                 ;Where FSR is F-Plane register
MOVLW   (TBA1>>8)&0xFF
MOVWF   DPH                 ;Where DPH is F-Plane register

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

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

TABRL Return DPTR low byte to W

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

```
MOVLW   (TAB1&0xFF)
MOVWF   FSR                 ;Where FSR is F-Plane register
MOVLW   (TBA1>>8)&0xFF
MOVWF   DPH                 ;Where DPH is F-Plane register

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

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

TSTF	Test if "f" is zero	
Syntax	TSTF 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	TSTF 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 1101 kkkk kkkk	
Description	The contents of the W register are XOR'ed with the eight-bit literal 'k'. The result is placed in the W register.	
Cycle	1	
Example	XORLW 0xAF	B : W =0xB5 A : W =0x1A

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

ELECTRICAL CHARACTERISTICS

1. Absolute Maximum Ratings (T_A =25°C)

Parameter	Rating	Unit
Supply voltage	V_{SS} -0.3 to V_{SS} +5.5	V
Input voltage	V_{SS} -0.3 to V_{CC} +0.3	
Output voltage	V_{SS} -0.3 to V_{CC} +0.3	
Output current high per 1 PIN	-25	mA
Output current high per all PIN	-80	
Output current low per 1 PIN	+30	
Output current low per all PIN	+150	
Maximum operating voltage	5.5	V
Operating temperature	-40 to +85	°C
Storage temperature	-65 to +150	

2. DC Characteristics (T_A =25°C, V_{DD} =5.0V, unless otherwise specified)

Parameter	Sym	Conditions		Min	Typ	Max	Unit
Operating Voltage	V_{cc}	Fsys=16Mhz		2.8		5.5	
		Fsys= 8Mhz		LVR		5.5	
Input High Voltage	V_{IH}	All Input, except PA7	V_{CC} = 3~5V	0.6Vcc	–	Vcc	V
		PA7	V_{CC} = 3~5V	0.6Vcc	–	Vcc	V
Input Low Voltage	V_{IL}	All Input, except PA7	V_{CC} = 3~5V	Vss	–	0.2Vcc	V
		PA7	V_{CC} = 3~5V	Vss	–	0.2Vcc	V
Output High Current	I_{OH}	All Output	V_{CC} = 5V,V_{OH} = 4.5V	5	10	–	mA
			V_{CC} = 3V,V_{OH} = 2.7V	2.5	5	–	
Output Low Current	I_{OL}	All Output	V_{CC} = 5V,V_{OL} = 0.5V	20	40	–	mA
			V_{CC} = 3V,V_{OL} = 0.3V	10	20	–	
Input Leakage Current (pin high)	I_{ILH}	All Input	V_{IN} = V_{CC}	–	–	1	uA
Input Leakage Current (pin low)	I_{ILL}	All Input	V_{IN} = 0V	–	–	–1	uA
Power Supply Current (No Load)	I_{CC}	FAST mode FIRC 16 MHz	V_{CC} = 5V	–	4.6	–	mA
		FAST mode FIRC 8 MHz	V_{CC} = 5V		3		
		FAST mode FIRC 4 MHz	V_{CC} = 5V	–	2.3	–	
		FAST mode FIRC 2 MHz	V_{CC} = 5V		1.9		
		FAST mode FIRC 2 MHz	V_{CC} = 3V		1.3		
		SLOW mode SIRC 65 KHz FIRC STOP BGEN = 1	V_{CC} = 5.0V	–	1250	–	uA
		SLOW mode SIRC 65 KHz FIRC STOP BGEN = 1	V_{CC} = 3.0V	–	700	–	uA

Parameter	Sym	Conditions		Min	Typ	Max	Unit
Power Supply Current (No Load)	I_{CC}	IDLE mode SIRC 65 KHz LVRSAV= 1	V_{CC} = 5.0V	–	6	–	uA
			V_{CC} = 3.0V	–	1.8	–	
		STOP mode LVRSAV = 1 PA7 = 1	V_{CC} = 5.0V	–	2	–	uA
			V_{CC} = 3.0V	–	0.5	–	
		STOP mode LVRSAV = 1 PA7= 0	V_{CC} = 5.0V	–	0.1	1	uA
			V_{CC} = 3.0V	–	0.1	–	
Pull-up Resistor	R_{UP}	V_{IN} = 0 V Ports A/B/D/E	V_{CC} = 5.0V	–	39	–	KΩ
			V_{CC} = 3.0V	–	70	–	
		V_{IN} = 0 V PA7	V_{CC} = 5.0V	–	194	–	KΩ
			V_{CC} = 3.0V	–	196	–	

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

Parameter	Condition	Min	Typ	Max	Unit
FIRC Frequency (*)	-40°C ~ 85°C, V_{CC} = 3.0 ~ 5.0V	-6.5%	16	+1.5%	MHz
	-40°C ~ 85°C, V_{CC} = 4.0 V	-5%	16	+1.5%	
	0°C ~ 70°C, V_{CC} = 4.0 V	-2%	16	+1.5%	
	25°C, V_{CC} = 3.0 ~ 5.0 V	-1.0%	16	+1.2%	
	25°C, V_{CC} =4.0 V	-0.5%	16	+0.5%	

(*) FIRC frequency can be divided by 1/2/4/8.

4. Reset Timing Characteristics (T_A = -40°C to +85°C)

Parameter	Conditions	Min	Typ	Max	Unit
RESET Input Low width	Input V_{CC} = 5 V ±10 %	30	–	–	μs
WDT time	V_{CC} = 3 V, WDTPSC = 11	–	2120	–	ms
	V_{CC} = 5 V, WDTPSC = 11		1930		
WKT time	V_{CC} = 3 V, WKTPSC = 11	–	133	–	ms
	V_{CC} = 5 V, WKTPSC = 11		123		
CPU start up time	V_{CC} = 5 V	–	26	–	ms

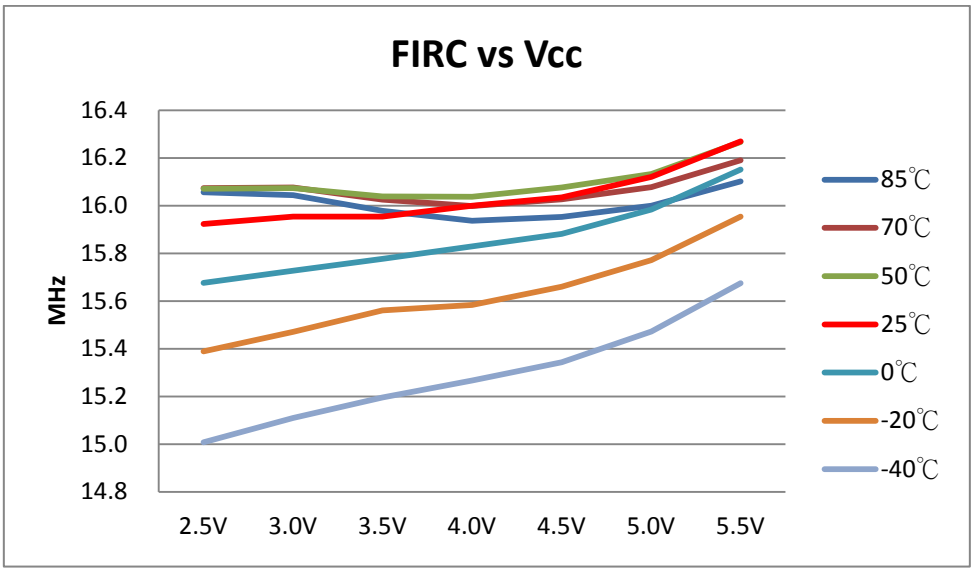
5. LVR Circuit Characteristics (T_A = 25°C)

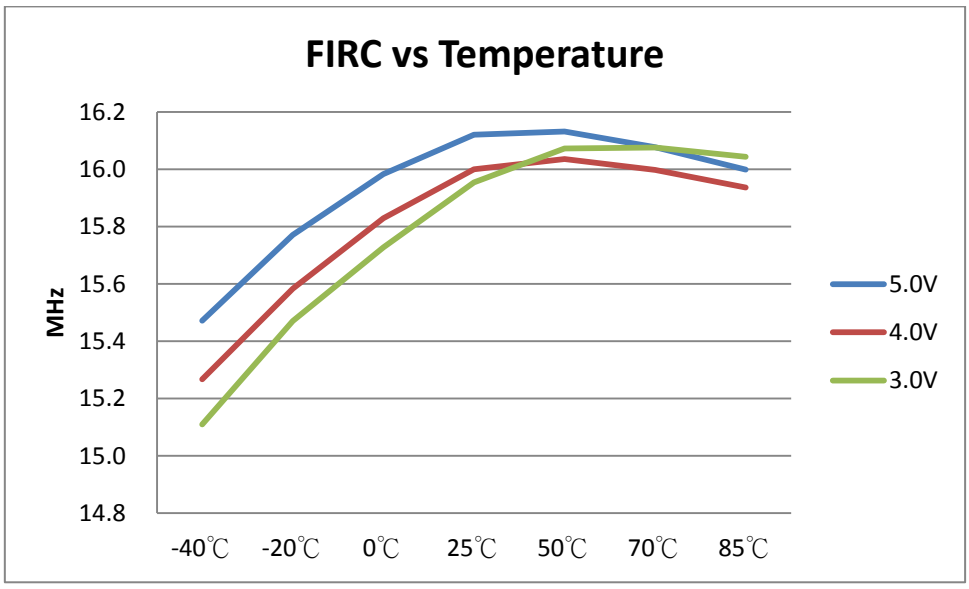
Parameter	Symbol	Min	Typ	Max	Unit
LVR Reference Voltage	LVR_{th}	–	2.3	–	V
		–	2.8	–	
		–	3.6	–	
		–	4.2	–	
LVR Hysteresis Voltage	V_{HYST}	–	±0.1	–	V
Low Voltage Detection time	t_{LVR}	100	–	–	μs

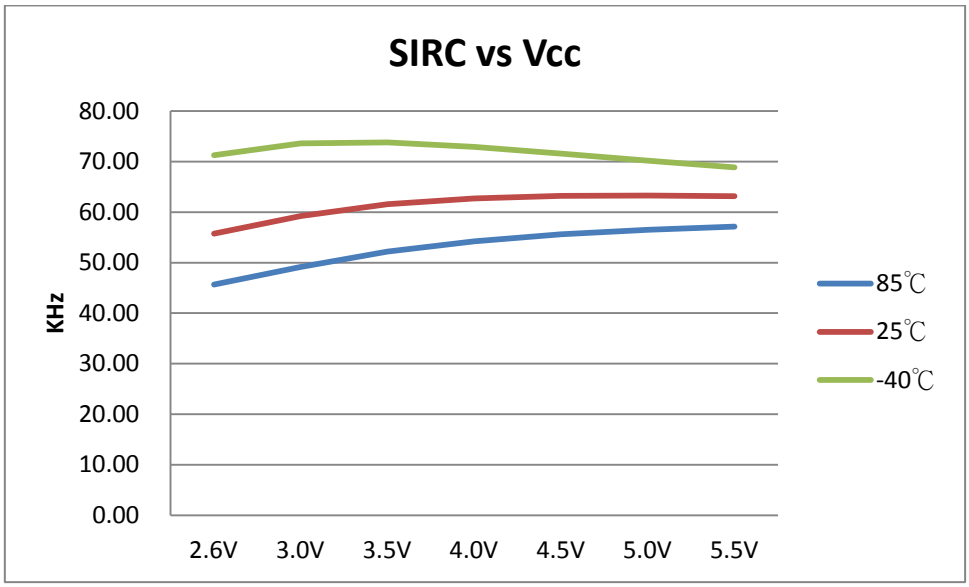
6. ADC Electrical Characteristics (T_A = 25°C, VCC = 3.0V to 5.5V, VSS = 0V)

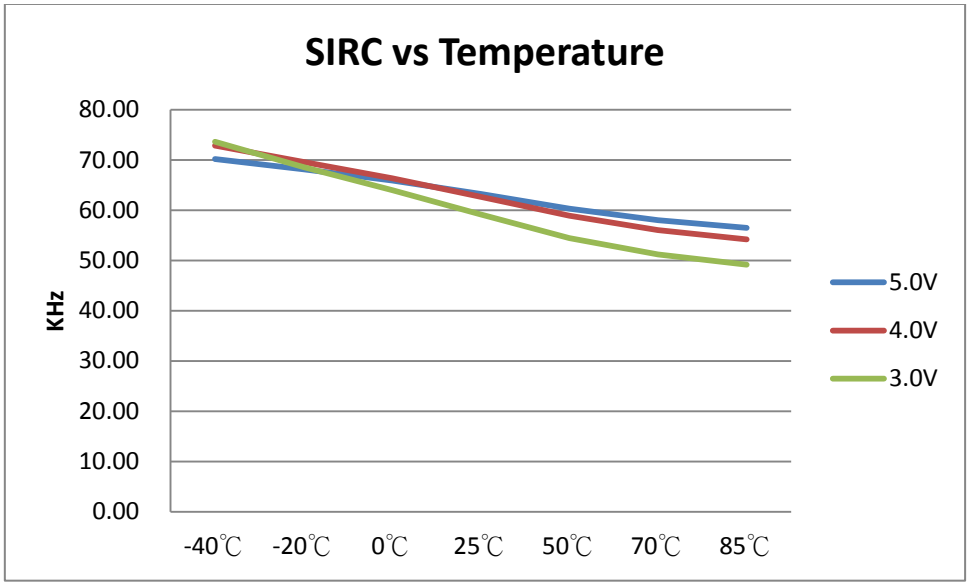
Parameter	Conditions	Min	Typ	Max	Units
Total Accuracy	V_{CC} = 5V, V_{SS} = 0V, f_{ADC} = 1 MHz	–	±2.5	±13	LSB
Integral Non-Linearity		–	±3.2	±15	
Differential Non-linearity		–	±1	±4	
Max Input Clock freq. (f_{ADC})	Source impedance (Rs<10K omh)	–	–	2	MHz
	Source impedance (Rs<20K omh)	–	–	1	
	Source impedance (Rs<50K omh)	–	–	0.5	
	Source is VBG (ADCHS=1111)	–	–	2	
Conversion Time	f_{ADC} = 1 MHz	–	50	–	μs
Input Voltage	–	V_{SS}	–	**Vcc**	V

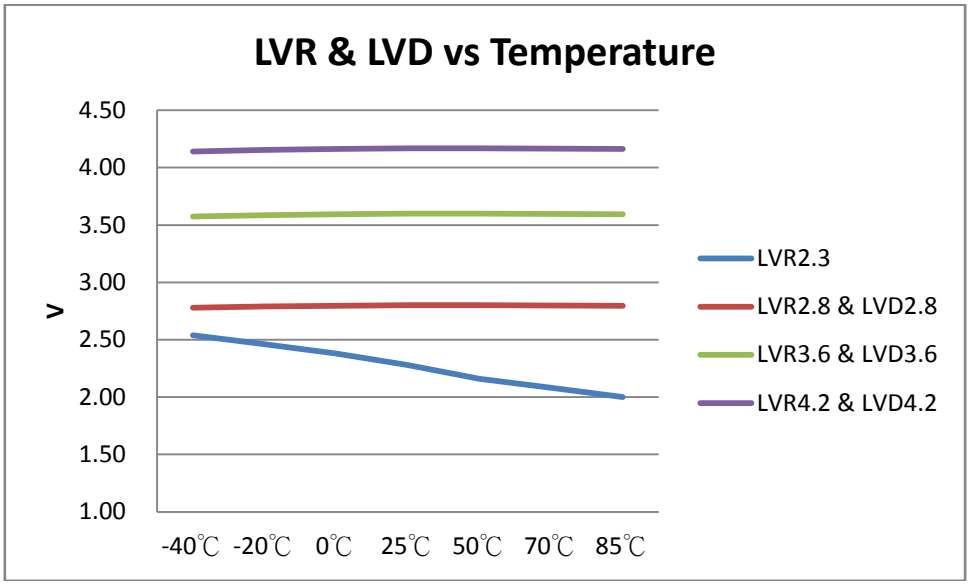
7. Characteristic Graphs






SIRC vs Vcc
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
KHz
2.6V
3.0V
3.5V
4.0V
4.5V
5.0V
5.5V
85℃
25℃
-40℃


SIRC vs Temperature
80.00
70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00
KHz
-40℃
-20℃
0℃
25℃
50℃
70℃
85℃
5.0V
4.0V
3.0V


LVR & LVD vs Temperature
4.50
4.00
3.50
3.00
2.50
2.00
1.50
1.00
V
-40℃ -20℃ 0℃ 25℃ 50℃ 70℃ 85℃
LVR2.3
LVR2.8 & LVD2.8
LVR3.6 & LVD3.6
LVR4.2 & LVD4.2

PACKAGING INFORMATION

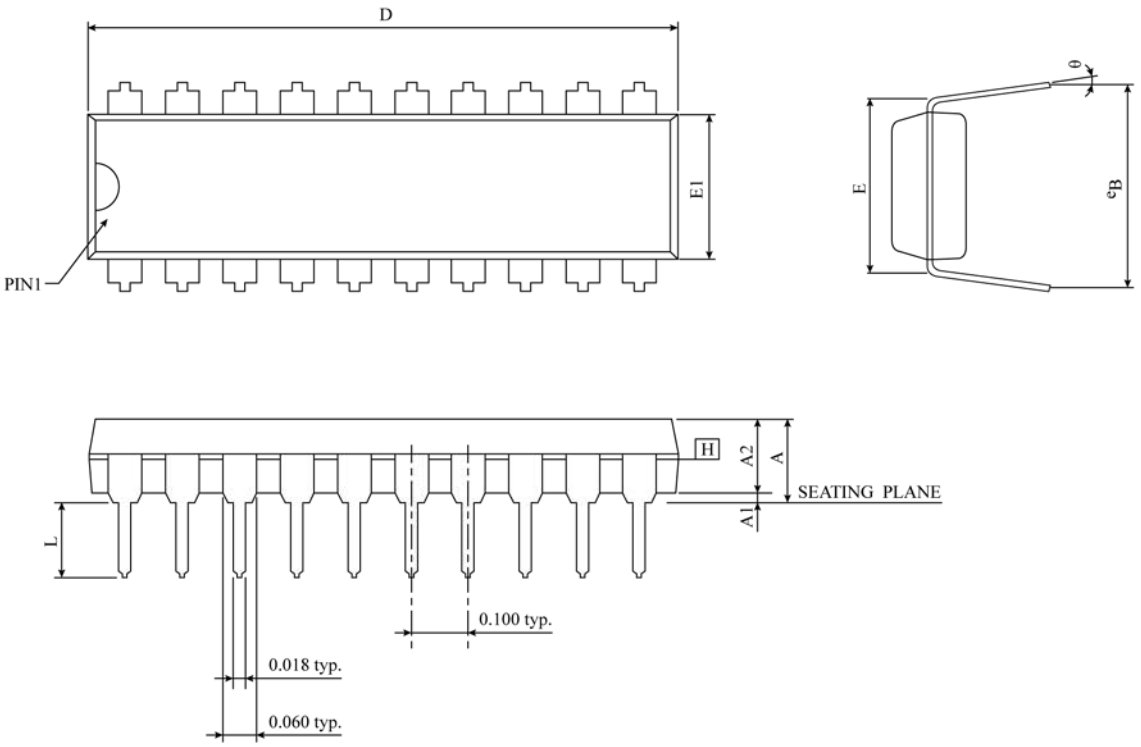
The ordering information:

Ordering number	Package
TM55M8228-MTP	Wafer/Dice blank chip
TM55M8228-COD	Wafer/Dice with code
TM55M8228-MTP-21	SOP 20-pin (300 mil)
TM55M8228-MTP-05	DIP 20-pin (300 mil)
TM55M8228-MTP-16	SOP 16-pin (150 mil)

Ordering number	Package
TM55M8428-MTP	Wafer/Dice blank chip
TM55M8428-COD	Wafer/Dice with code
TM55M8428-MTP-21	SOP 20-pin (300 mil)
TM55M8428-MTP-05	DIP 20-pin (300 mil)
TM55M8428-MTP-16	SOP 16-pin (150 mil)

Ordering number	Package
TM55M8428T-MTP-21	SOP 20-pin (300 mil)
TM55M8428T-MTP-05	DIP 20-pin (300 mil)
TM55M8428T-MTP-16	SOP 16-pin (150 mil)

- **DIP-20 (300mil) Package Dimension**

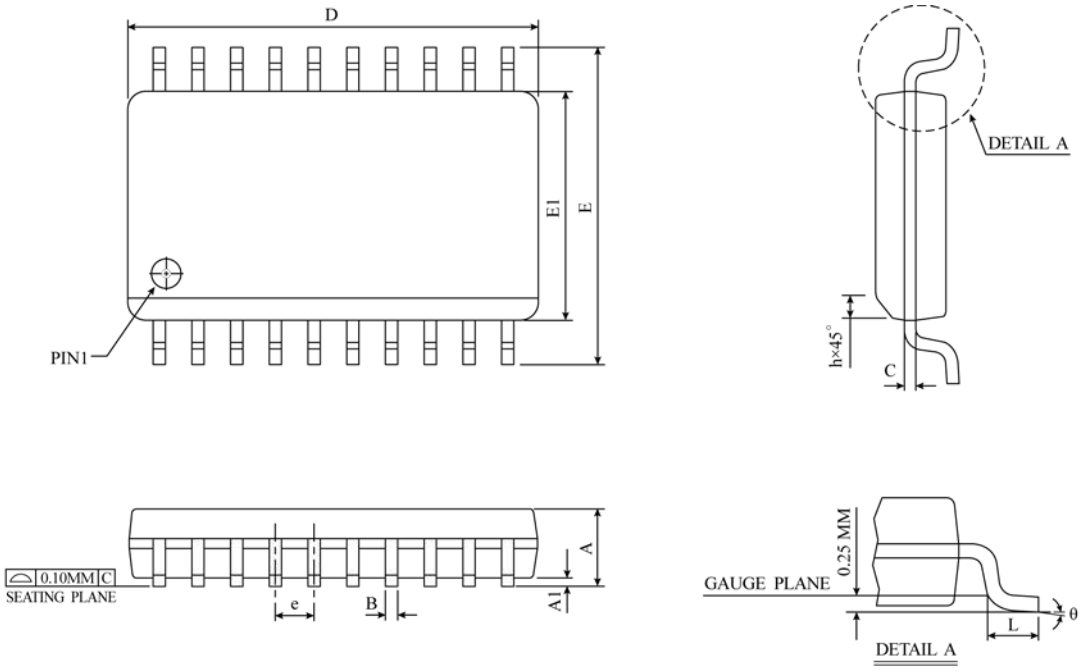


SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	4.445	-	-	0.175
A1	0.381	-	-	0.015	-	-
A2	3.175	3.302	3.429	0.125	0.130	0.135
D	25.705	26.061	26.416	1.012	1.026	1.040
E	7.620	7.747	7.874	0.300	0.305	0.310
E1	6.223	6.350	6.477	0.245	0.250	0.255
L	3.048	3.302	3.556	0.120	0.130	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.

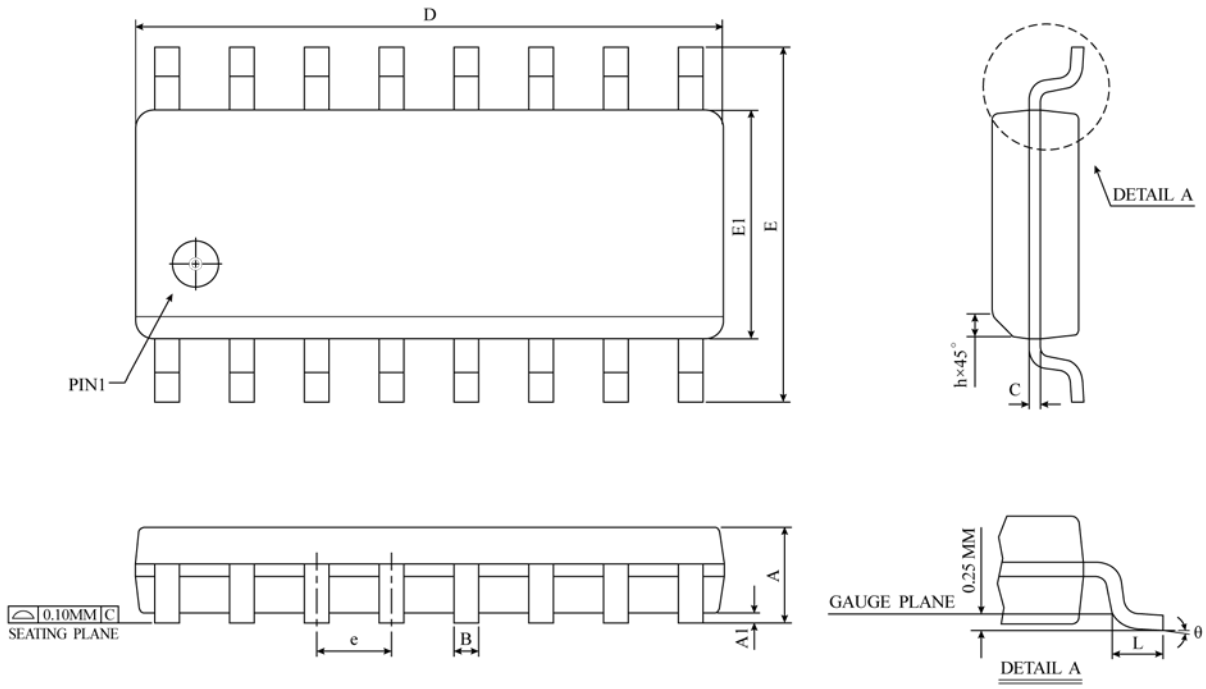
- **SOP-20 (300mil) Package Dimension**



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

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

- **SOP-16 (150mil) 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.