十速

TM57M5526C/36C

DATA SHEET

Rev 0.92

tenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **tenx** does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.

AMENDMENT HISTORY

Version	Date	Description
V0.90	May, 2017	New release.
V0.91	May, 2017	Error correction (P51. P64. P65. P67. P69)
V0.92	May, 2017	1. Modify TM57M5526/36 into TM57M5526C/36C 2. Add DIP/SOP8 package info (P7, P9, P89, P90, P91) 3. Modify the error of LVR value (P24)

CONTENTS

FEATURES

1. **ROM: 2K x 14 bits MTP ROM with Page Locker function**
2. **RAM: 176 x 8 bits**
3. **STACK: 6 Levels**
4. **I/O port: One Bit-programmable I/O port (Max. 6 pins)**
5. **2 Independent Timers**
   - Timer0
     - 8-bit timer with divided by 1~256 pre-scale option/auto-reload/counter/interrupt/stop function
   - Timer1
     - 8-bit timer with divided by 1~256 pre-scale option/auto-reload/interrupt/stop function
6. **PWM: One 8-bit PWM0 with pre-scale/period-adjustment function**
7. **4-channel Touch Key (TM57M5526C only)**
   - 1~2 Key H/W auto scan, upper/lower boundary adjustable for each key
   - Interrupt/Wake-up CPU while key is pressed (H/W auto mode)
   - Interrupt while Touch Key is end of conversion (S/W manual mode)
   - ATK with 4 adjustable scan interval time

     37.5 ms/75 ms/150 ms/300 ms @V_{DD}=3V, 26 ms/52 ms/104 ms/208 ms @V_{DD}=5V
   - Internal built-in reference capacitor
8. **System Oscillation Sources (Fsys)**
   - Fast-clock
     - FIRC (Fast Internal RC): 0.5/2/4 MHz (can be trimmed)
   - Slow-clock
     - SIRC (Slow Internal RC): 5/20/40/80 KHz @V_{DD}=5V
9. **Dual System Clock**
   - FIRC + SIRC
10. **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 or Wake-up Timer keep running
    - STOP Mode: All Clocks stop, Wake-up Timer stop

11. Reset

- Power On Reset
- Watchdog Reset
- Low Voltage Reset
- External pin Reset

12. 2-Level Low Voltage Reset: 1.7V/2.4V/3.7V (Can be disabled)

LVR Freq	1.7V	2.4V	3.7V	Disable
2 MHz	☑	☑	☑	☑
4 MHz	☒	☑	☑	☑

If LVR=Disable, VDD power-on must exceed the lowest LVR (~1.7V@25°C)

13. Operation Voltage: Low Voltage Reset Level to 5.5V

- Fsys=2 MHz, 1.5V ~ 5.5V, V_{DD} power-on must exceed the lowest LVR (~1.7V@25°C)
- Fsys=4 MHz, 1.8V ~ 5.5V

14. Operating Temperature Range:

- Fsys=2 MHz, -40°C to +85°C
- Fsys=4 MHz, -40°C to +75°C

15. Interrupt

- Two External Interrupt pins:
  - One pin is falling edge triggered
  - One pin is rising or falling edge triggered
- TM0, TM1, Wake-up Timer, Touch Key (TM57M5526C only) Interrupt

16. Wake-up Timer (WKT)

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

  37.5 ms/75 ms/150 ms/300 ms @V_{DD}=3V, 26 ms/52 ms/104 ms/208 ms @V_{DD}=5V

17. Watchdog Timer (WDT)

- Clocked by built-in RC oscillator with 4 adjustable reset times
- 75 ms/150 ms/300 ms/600 ms @V_{DD}=3V, 52 ms / 104 ms/208 ms/416 ms @V_{DD}=5V
- Watchdog timer can be disabled/enabled in Power-down mode

18. I/O Port

- CMOS Output
- Open-Drain Output
- Schmitt Trigger Input with/without pull-up resistor

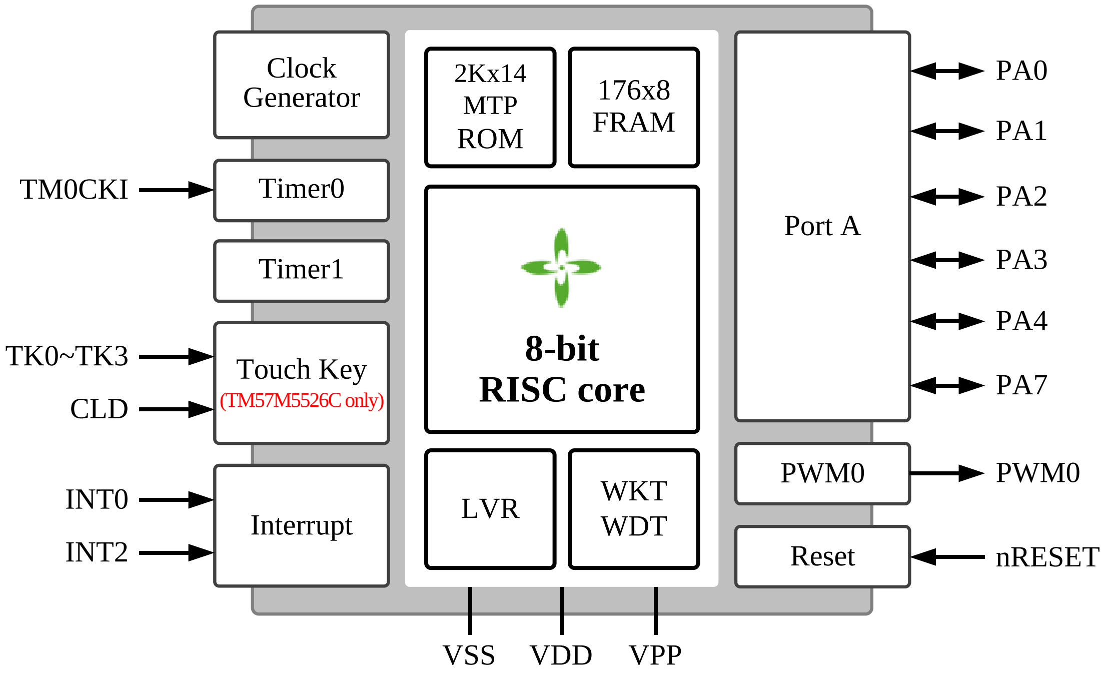
19. PA0 and PA2 individual pin change wake up

20. PA3 and PA4 with High Sink and Drive ability

21. **Page Locker Size: 512W/640W/768W …./1792W by 128 words step**
22. **Support 5-wire program**
23. **Instruction set: 39 Instructions**
24. **Package Types:**
    - 8-pin DIP (300 mil)
    - 8-pin SOP (150 mil)
    - SOT23-6
25. **Supported EV Board on ICE**

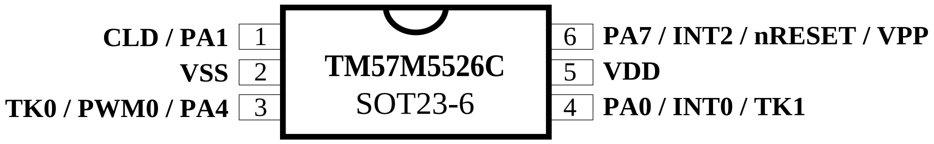
    EV Board: EV8209

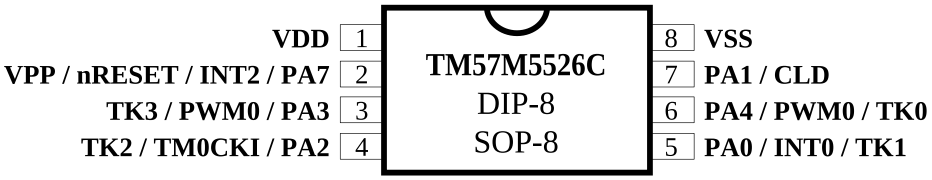
BLOCK DIAGRAM

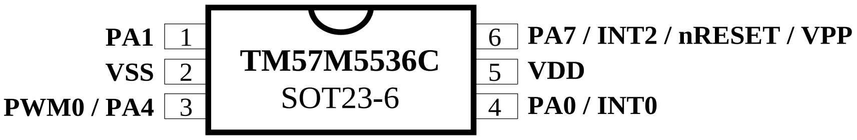


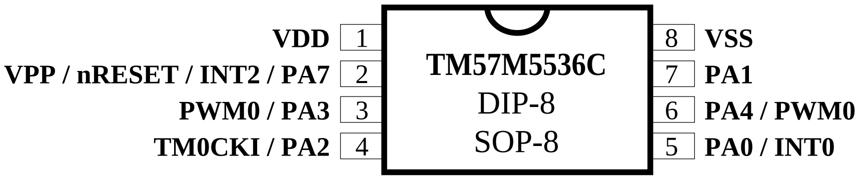
TM57M5526C/36C Block Diagram

PIN ASSIGNMENT









PIN DESCRIPTION

Name	In/Out	Pin Description
PA0–PA4	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or open-drain output. Pull-up resistors are assignable by software.
PA7	I/O	Bit-programmable I/O port for Schmitt-trigger input or open-drain output. Pull-up resistors are assignable by software.
nRESET	I	External active low reset
VDD, 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
NC	-	Not connected
PWM0	O	PWM0 output
TK0–TK3	I	Touch key input (TM57M5526C only)
CLD	I	Touch key capacitor input (TM57M5526C only)

PIN SUMMARY

Pin Number		Pin Name	Type	GPIO				Function After Reset	Alternate Function		
				Input		Output					
10-SOP/DIP	SOT23-6			Weak Pull-up	Ext. Interrupt	O.D	P.P		PWM	Touch Key (TM57M5526C only)	MISC
1	-	NC									
2	5	VDD	P								
3	6	PA7/INT2/nRESET/VPP	I/O	○	○	○		PA7			nRESET
4	-	PA3/PWM0/**TK3**	I/O	○		○	○	PA3	○	○	
5	-	PA2/TM0CKI/**TK2**	I/O	○		○	○	PA2		○	TM0CKI
6	4	PA0/INT0/**TK1**	I/O	○	○	○	○	PA0		○	
7	3	PA4/PWM0/**TK0**	I/O	○		○	○	PA4	○	○	
8	1	PA1/**CLD**	I/O	○		○	○	PA1		○	
9	2	VSS	P								
10	-	NC									

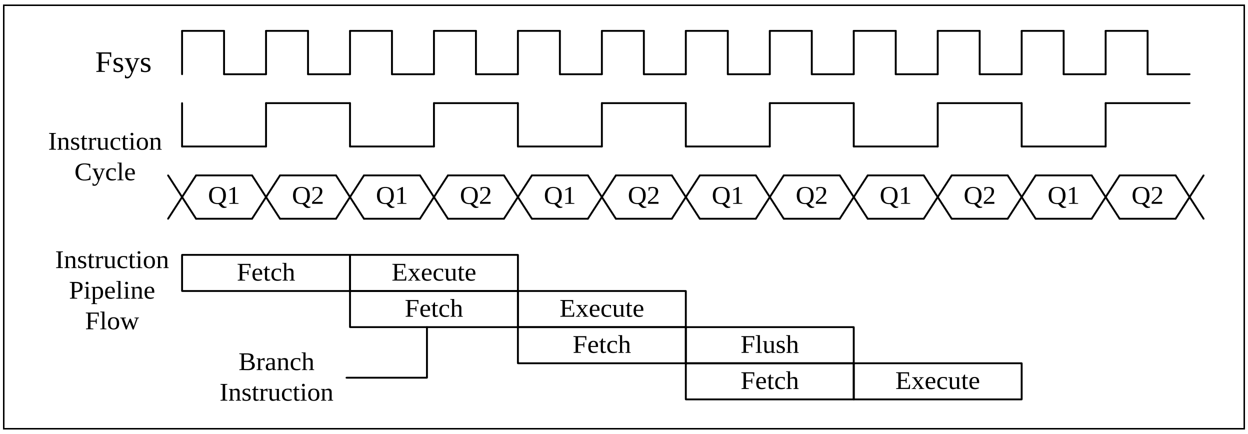
Symbol：P.P. = COM Push-Pull Output
O.D. = Open Drain Output

FUNCTIONAL DESCRIPTION

1. CPU Core

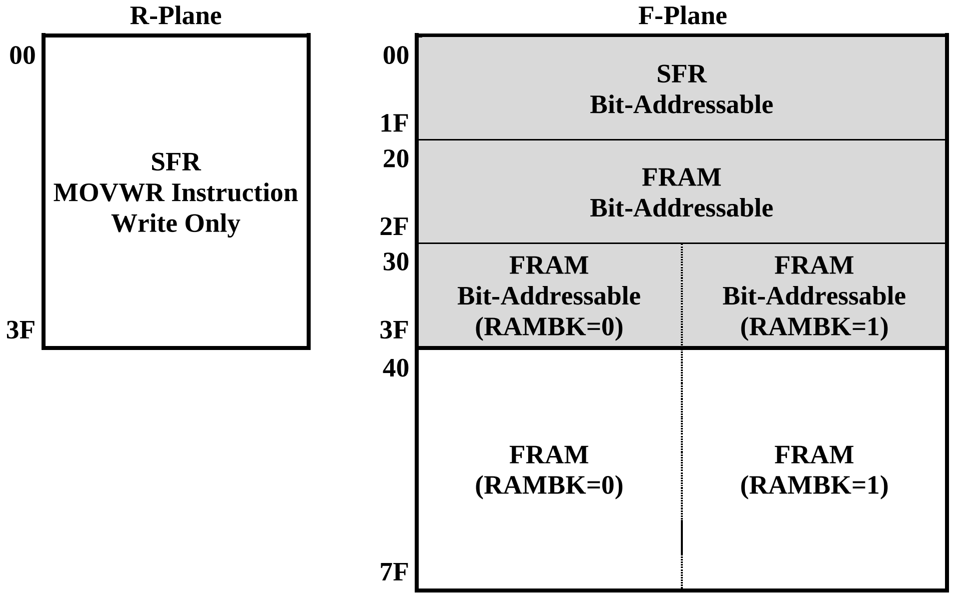
1.1 Clock Scheme and Instruction Cycle

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



1.2 Addressing Mode

There are two Data Memory Planes in CPU, R-Plane and F-Plane. The registers in R-Plane are write-only. The “MOVWR” instruction copies the W-register’s content to R-Plane registers by direct addressing mode. The lower locations of F-Plane are reserved for the SFR. Above the SFR is General Purpose Data Memory, implemented as static RAM. F-Plane can be addressed directly or indirectly. Indirect Addressing is made by INDF register. The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a pointer). The first half of F-Plane is bit-addressable, while the second half of F-Plane is not bit-addressable. And there are two RAM banks can be selected by RAMBK (F03.5).



◇ Example: Write immediate data into R-Plane register

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

◇ Example: Write immediate data into F-Plane register

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

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

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

◇ Example: Clear FRAM Bank0 data by indirect addressing mode

```
            MOVLW    20H                ; W=20H (FRAM start address)
            MOVWF    FSR                ; Set start address of user FRAM into FSR register
            BCF      STATUS, 5          ; Set RAMBK=0
  LOOP:
            MOVLW    00H
            MOVWF    INDF               ; Clear user FRAM data
            INCF     FSR, 1             ; Increment the FSR for next address
            MOVLW    80H                ; W=80H (FRAM end address)
            XORWF    FSR, 0             ; Check the FSR is end address of user FRAM?
            BTFSS    STATUS, 2          ; Check the Z flag
            GOTO     LOOP               ; If Z=0, goto LOOP label
            …                           ; If Z=1, exit LOOP
```

1.3 Programming Counter (PC) and Stack

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

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.
            …
            MOVLW     (TABLE2>>8)&0xff
            MOVWF     DPH                    ; DPH register (F1E.1~0)
            MOVLW     (TABLE2)&0xff
            MOVWF     DPL                    ; DPL register (F1D.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
```

1.4 ALU and Working (W) Register

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

Note: /Borrow represents inverted of Borrow register.
/Digit Borrow represents inverted of Digit Borrow register.

1.5 STATUS Register (F-Plane 03H)

This register contains the arithmetic status of ALU 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. The RAMBK bit is used to the FRAM Bank selection.

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

<table>
<tr><th>Bit</th><th colspan="2">Description</th></tr>
<tr><td>7</td><td colspan="2"><b>GB1</b>: General Purpose Bit 1</td></tr>
<tr><td>6</td><td colspan="2"><b>GB0</b>: General Purpose Bit 0</td></tr>
<tr><td>5</td><td colspan="2"><b>RAMBK</b>: FRAM Bank Selection
0: FRAM Bank0
1: FRAM Bank1</td></tr>
<tr><td>4</td><td colspan="2"><b>TO</b>: 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="2"><b>PD</b>: STOP Flag
0: after Power On Reset, LVR Reset, or CLRWDT instruction
1: after SLEEP instruction</td></tr>
<tr><td>2</td><td colspan="2"><b>Z</b>: 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="2"><b>DC</b>: Decimal Carry Flag or Decimal /Borrow Flag</td></tr>
<tr><td>ADD instruction</td><td>SUB instruction</td></tr>
<tr><td>0: no carry
1: a carry from the low nibble bits of the result occurs</td><td>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="2"><b>C</b>: Carry Flag or /Borrow Flag</td></tr>
<tr><td>ADD instruction</td><td>SUB instruction</td></tr>
<tr><td>0: no carry
1: a carry occurs from the MSB</td><td>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
BCF       STATUS, 0             ; Clear C=0
```

◇ Example: Determine the C flag by BTFSS instruction

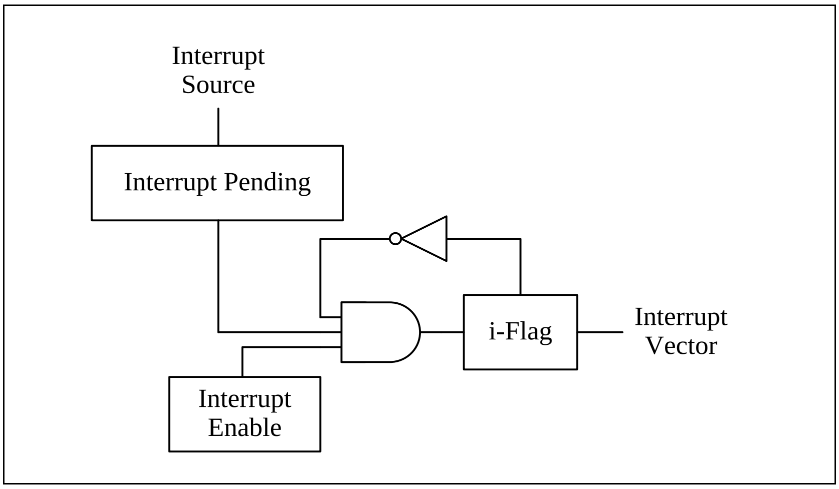
```
BTFSS     STATUS, 0             ; Check the C flag
GOTO      LABEL_1               ; If C=0, goto LABEL_1 label
GOTO      LABEL_2               ; If C=1, goto LABEL_2 label
```

1.6 Interrupt

The TM57M5536C has 1 level, 1 vector and 5 interrupt sources. TM57M5526C has 1 level, 1 vector and 6 interrupt sources. Each interrupt source has its own enable control bit. An interrupt event will set its individual pending flag; no matter its interrupt enable control bit is 0 or 1. Because TM57M5526C/36C has only 1 vector, there is not an interrupt priority register. The interrupt priority is determined by F/W.

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

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



◇ Example: Setup INT0 (PA0) interrupt request with rising edge trigger

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

          ORG       001H                ; All interrupt vector
          GOTO      INT                 ; If INT0 (PA0) input occurred rising edge

          ORG       002H
START:
          MOVLW     xxxxxx00B
          MOVWR     PAMODL              ; Select INT0 Pin Mode as Mode0
                                        ; Open drain output low or input with Pull-up
          MOVLW     xxxxxxx1B
          MOVWF     PAD                 ; Release INT0, it becomes Schmitt-trigger
                                        ; input with input pull-up resistor
          MOVLW     x1xxxxxxB
          MOVWR     MR0B                ; Set INT0 interrupt trigger as rising edge
          MOVLW     11111110B
          MOVWF     INTIF               ; Clear INT0 interrupt request flag
          MOVLW     00000001B
          MOVWF     INTIE               ; Enable INT0 interrupt
MAIN:
          …
          GOTO      MAIN

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

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

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	–	TM1IE	TM0IE	WKTIE	INT2IE	–	INT0IE
R/W	R/W	–	R/W	R/W	R/W	R/W	–	R/W
Reset	0	–	0	0	0	0	–	0

F08.7 **TKIE:** Touch Key interrupt enable (TM57M5526C only)
0: disable
1: enable

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

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

F08.3 **WKTIE:** Wakeup Timer interrupt enable
0: disable
1: enable

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

F08.0 **INT0IE:** INT0 (PA0) interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	TKIF	–	TM1IF	TM0IF	WKTIF	INT2IF	–	INT0IF
R/W	R/W	–	R/W	R/W	R/W	R/W	–	R/W
Reset	0	–	0	0	0	0	–	0

F09.7 **TKIF:** Touch Key interrupt event pending flag (TM57M5526C only)
This bit is set by H/W while Key's TK Data Count is over the pre-set compare threshold range (H/W auto mode) or TK is end of conversion (S/W manual mode), write 0 to this bit will clear this flag

F09.5 **TM1IF:** Timer1 interrupt event pending flag
This bit is set by H/W while Timer1 overflows, write 0 to this bit will clear this flag

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

F09.3 **WKTIF:** 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

F09.2 **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

F09.0 **INT0IF:** INT0 (PA0) 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

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

R0B.6 **INT0EDG:** INT0 (PA0) trigger edge select
0: INT0 (PA0) pin falling edge to trigger interrupt event
1: INT0 (PA0) pin rising edge to trigger interrupt event

2. Chip Operation Mode

2.1 Reset

The TM57M5526C/36C can be RESET in four ways.

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

Resets can be caused by Power on Reset (POR), External Pin Reset (XRST), Low Voltage Reset (LVR), or Watchdog Timer Reset (WDTR). 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.

2.1.1 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.

2.1.2 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:

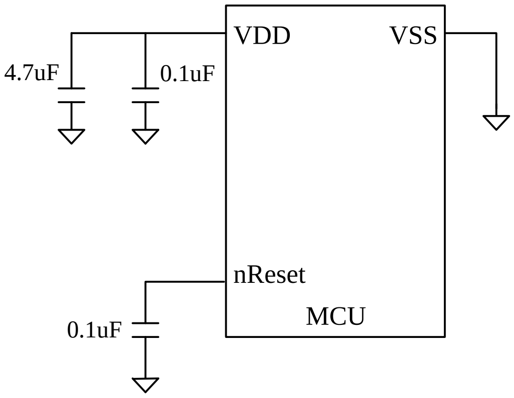
LVR Threshold Level	Operating voltage
LVR1.7	5.5V > V_{DD} > 1.8V
LVR2.4	5.5V > V_{DD} > 2.5V
LVR3.7	5.5V > V_{DD} > 3.8V or V_{DD} =5.0V

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

2.1.3 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 flag is not affected by these resets.

External reset pin is low level active. The system is running when reset pin is high level voltage input. The reset pin receives the low voltage and the system is reset. The external reset can reset the system during power on duration, and good external reset circuit can protect the system to avoid working at unusual power condition.



2.1.4 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                ; Reset Vector
            GOTO        START               ; Jump to user program address.

            ORG         010H                ; All interrupt vector
   START:
            …                               ; 010H, The head of user program
            …
            GOTO        START
```

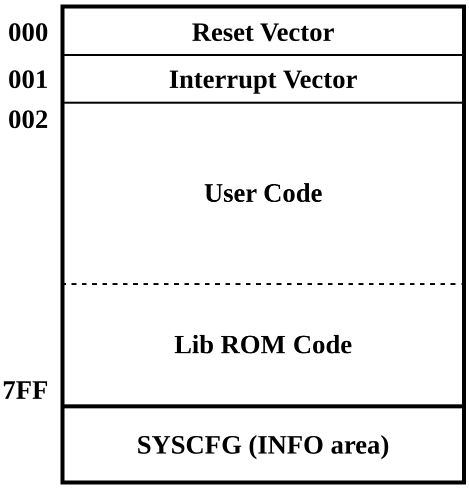
2.2 System Configuration Register (SYSCFG)

The System Configuration Register (SYSCFG) is located at MTP INFO area, and it contains two 14bits registers (CFGWL/CFGWH). The SYSCFG determines the option for initial condition of MCU. It is written by PROM Writer only. User can select clock source, LVR threshold voltage and chip operation mode by SYSCFG register. The 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		**CFGL: 00_000x_xxxx_xxxx** **CFGH: 00_0000_xxxx_xxxx**	
Bit		Description	
CFGWL	13	**LPROT:** Lib Code protection selection	
		1	Enable
		0	Disable
	12-9	**LSIZE**: Lib Size selection	
		1111	Invalid
		…	Invalid
		1100	Invalid
		1011	1792W
		…	… (128W/step)
		0001	512W
		0000	No use page locker function
	8~0	**Reserved**	
CFGWH	13	**PROTECT**: Code protection selection	
		1	Enable
		0	Disable
	12	**XRSTE**: External pin Reset Enable	
		1	Enable
		0	Disable (PA7 as input I/O pin)
	11-10	**LVR**: Low Voltage Reset Mode	
		11	LVR=1.7V
		10	LVR Disable
		01	LVR=2.4V
		00	LVR=3.7V
	9-8	**WDTE**: WDT Reset Enable	
		11	Always Enable
		10	Enable in FAST/SLOW mode ,Disable in IDLE/STOP mode
		0X	Disable
	7-0	**Reserved**	

2.3 MTP Program ROM

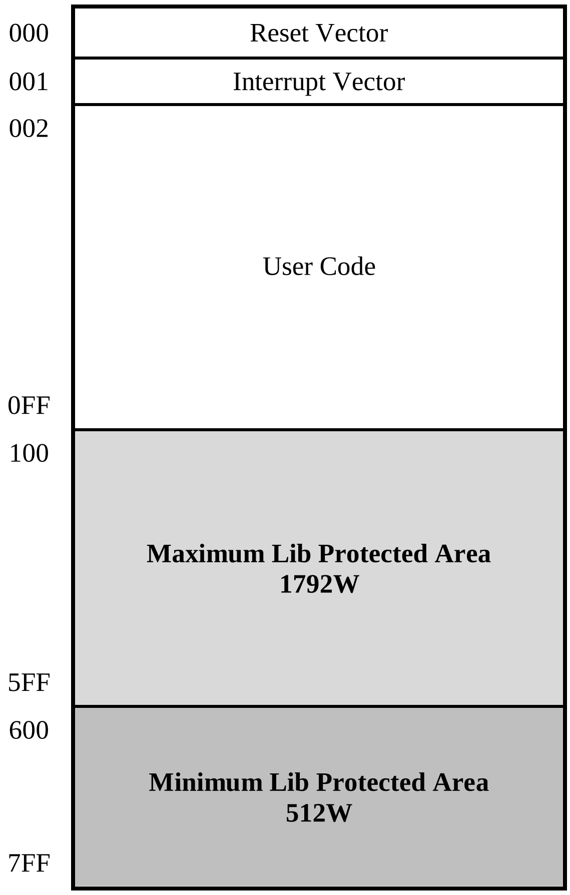
The MTP Program ROM of this device is 2K words, with an extra INFO area to store the SYSCFG. The ROM can be written multi-times and can be read as long as the PROTECT and LPROT bits of SYSCFG are not set. The SYSCFG can be read no matter PROTECT or LPROT is set, but PROTECT bit can be cleared only when 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.



2.4 Page Locker Function

TM57M5526C/36C 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 (512~1792W) 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.

2K Program ROM



LSIZE	Lib Protected Area
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)

2.5 Power-Down Mode

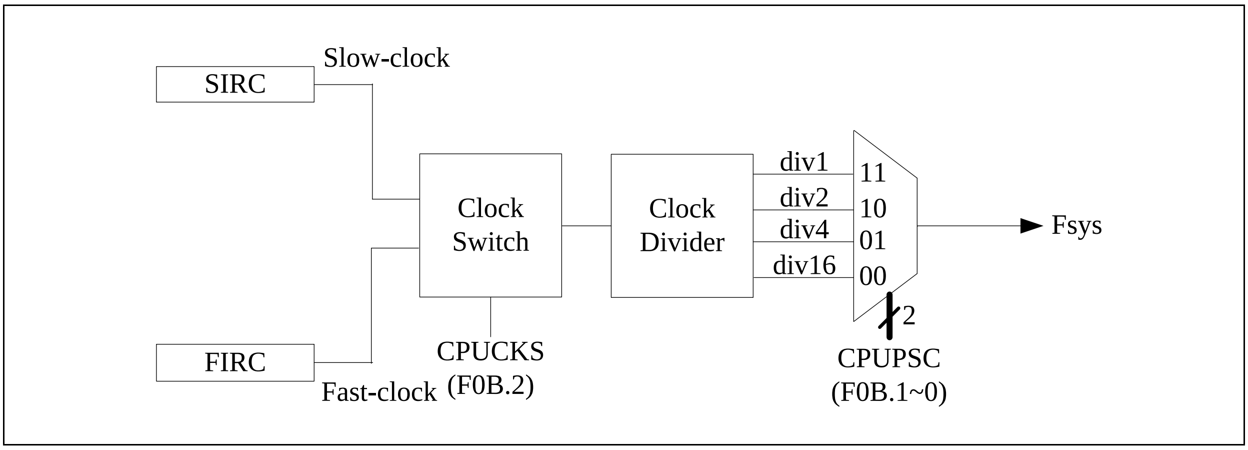
The Power-down mode includes IDLE Mode and STOP Mode. It is activated by SLEEP instruction. During the Power-down mode, the system clock and peripherals stop to minimize power consumption, whether the WDT/WKT Timer are working or not depend on F/W setting. The Power-down mode can be terminated by Reset, or enabled Interrupts (External pins, WKT and ATK interrupts) or PA0, PA2 pin change wake up.

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

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

2.6 Dual System Clock

TM57M5526C/36C is designed with dual-clock system. There are two kinds of clock source, SIRC (Slow Internal RC) Clock and FIRC (Fast Internal RC) Clock. Both of them can be applied to CPU kernel as system clock source. Refer to the figure as below.



FAST Mode:

TM57M5526C/36C enters FAST mode by setting the CPUCKS (F0B.2). In this mode, the system clock source is FIRC. The Timer0, Timer1 and PWM0 blocks are driven by Fast-clock.

SLOW Mode:

After power on or reset, TM57M5526C/36C enters SLOW mode, the default system clock source is SIRC. In this mode, the Fast-clock can be stopped by setting FASTSTP=1 (F0B.3) for power saving or run by setting FASTSTP=0, and Slow-clock is enabled. All peripheral blocks (Timer0, Timer1 and PWM0, etc…) are driven by Slow-clock.

IDLE Mode:

When SLOWSTP (F0B.4) is cleared, the TM57M5526C/36C will enter the “IDLE Mode” after executing the SLEEP instruction. In this mode, the Slow-clock will keep running to provide clock to WKT/WDT or Auto Touch Key block. CPU stops fetching code and all blocks are stop.

Another way to keep clock oscillation in IDLE mode is setting WKTIE=1 (F08.3) to keeping WKT running before executing the SLEEP instruction or WDTE=11B (CFGWH.9~8) to keeping WDT running. In such condition, the Slow-clock will also keep running no matter SLOWSTP is set or cleared. It is possible to keep WKT working for wake-up CPU periodically in the IDLE mode, which is useful for low power Touch Key detection.

The third way to keep clock oscillation in IDLE mode is setting ATKEN=1 (F13.2) before executing the SLEEP instruction. In such condition, The Auto Touch Key will keep working. (TM57M5526C only)

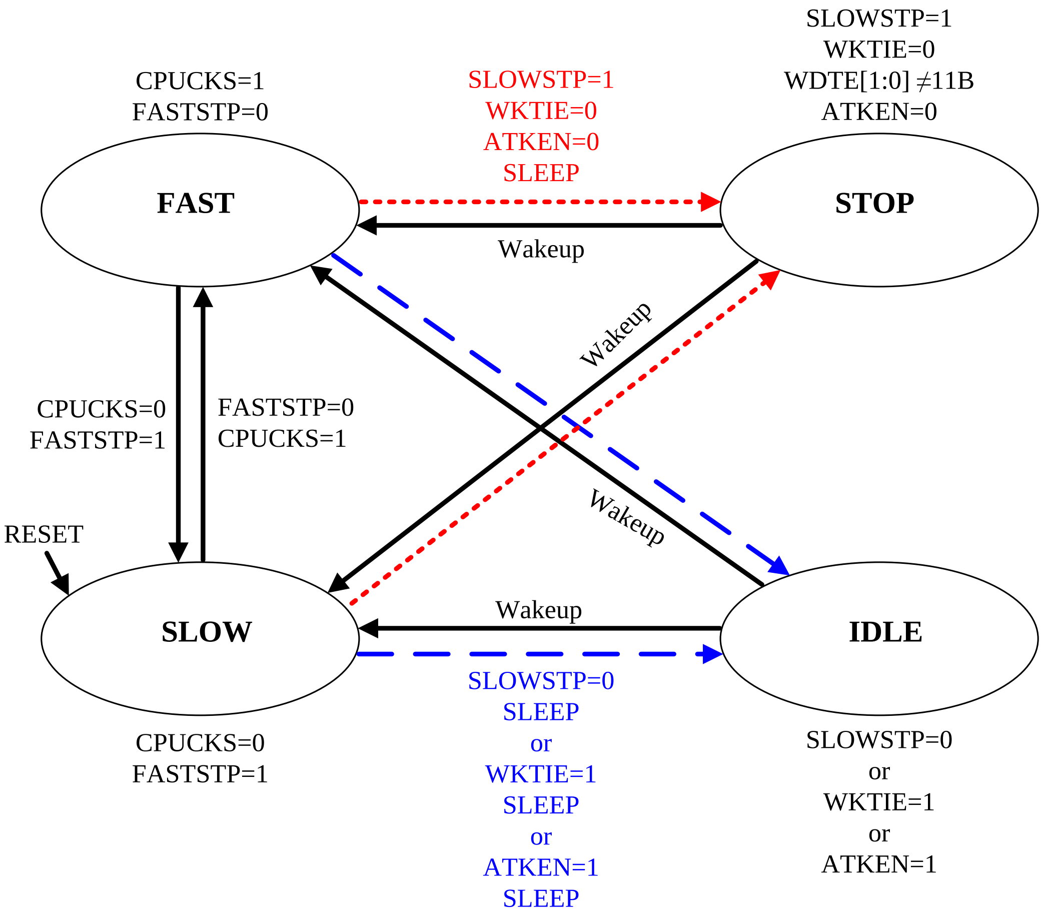
STOP Mode:

If Slow-clock, WKT, WDT and ATK are disabled before executing the SLEEP instruction, every block is turned off and the TM57M5526C/36C enters the STOP mode. STOP mode is similar to IDLE mode. The difference is all clock oscillators either Fast-clock or Slow-clock are stopped and no clocks are generated.

2.7 Dual System Clock Modes Transition

TM57M5526C/36C is operated in one of four modes: FAST Mode, SLOW Mode, IDLE Mode, and STOP Mode.

Modes Transition Diagram:



Note:

- SLEEP denotes SLEEP instruction
- Wakeup denotes wake-up events, such as External pin, WKT, ATK, interrupts or pin wake-up
- CPUCKS (F0B.2), FASTSTP (F0B.3), SLOWSTP (F0B.4), WKTIE (F08.3), WDTE (CFGWH.9~8)
- ATKEN (F13.2) (TM57M5526C only)

CPU Mode & Clock Functions Table:

Mode	Oscillator	Fsys	Fast-clock	Slow-clock	TM0/TM1/ TK/PWM0	ATK	Wakeup event
FAST	FIRC	Fast-clock	Run	Run	Run	Run	–
SLOW	SIRC	Slow-clock	Set by FASTSTP bit	Run	Run	Run	–
IDLE	SIRC	Stop	Stop	Run	Stop	Run	WKT/ATK/IO
STOP	Stop	Stop	Stop	Stop	Stop	Stop	IO

FAST Mode transits to SLOW Mode:

The source clock of Slow-clock is SIRC. The following steps are suggested to be executed by order when FAST mode transits to SLOW mode:

(1) Switch system clock source to Slow-clock (CPUCKS=0)

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

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

```
BCF       CPUCKS          ; Switch system clock source to Slow-clock
BSF       FASTSTP         ; Stop Fast-clock
```

SLOW Mode transits to FAST Mode:

The source clock of Fast-clock is FIRC. The following steps are suggested to be executed by order when SLOW mode transits to FAST mode:

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

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

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

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

IDLE Mode Setting:

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

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

(2) Execute SLEEP instruction

IDLE mode can be woken up by interrupts (External pins, WKT or ATK) or pins (PA0 and PA2) change wake up.

◇ Example: Switch operating mode to IDLE mode

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

STOP Mode Setting:

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

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

(2) Disable WKT/WDT (WKTIE=0)

(3) Disable ATK (ATKEN=0)

(4) Execute SLEEP instruction

STOP mode can be woken up by External pins interrupt or pins (PA0 and PA2) change wake up.

◇ Example: Switch operating mode to STOP mode

```
BSF       SLOWSTP       ; Stop Slow-clock
BCF       WKTIE         ; Disable WDT/WKT
BCF       ATKEN         ; Disable ATK
SLEEP                   ; Enter STOP mode
```

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	–	TM1IE	TM0IE	WKTIE	INT2IE	–	INT0IE
R/W	R/W	–	R/W	R/W	R/W	R/W	–	R/W
Reset	0	–	0	0	0	0	–	0

F08.3 **WKTIE:** Wakeup Timer interrupt enable
0: disable
1: enable

F0B	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	
Reset	–	–	–	0	0	0	1	1

F0B.4 **SLOWSTP**: Slow-clock stop
0: Slow-clock is running
1: Slow-clock stops running in Power-down mode

F0B.3 **FASTSTP**: Fast-clock stop
0: Fast-clock is running
1: Fast-clock stops running

F0B.2 **CPUCKS**: System clock source select
0: Slow-clock
1: Fast-clock (forbid using, when CPUPSC=3)

F13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 3	Bit 0
ATKCTL	–	–	ATKSIT		–	ATKEN	–	ATKNUM
R/W	–	–	R/W		–	R/W	–	R/W
Reset	–	–	0	0	–	0	–	0

F13.2 **ATKEN**: Touch key auto scan mode enable (TM57M5526C only)
0: disable H/W Auto Mode
1: enable H/W Auto Mode

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

R03.7~0 **PWRDN:** Write this register to enter Power Down Mode

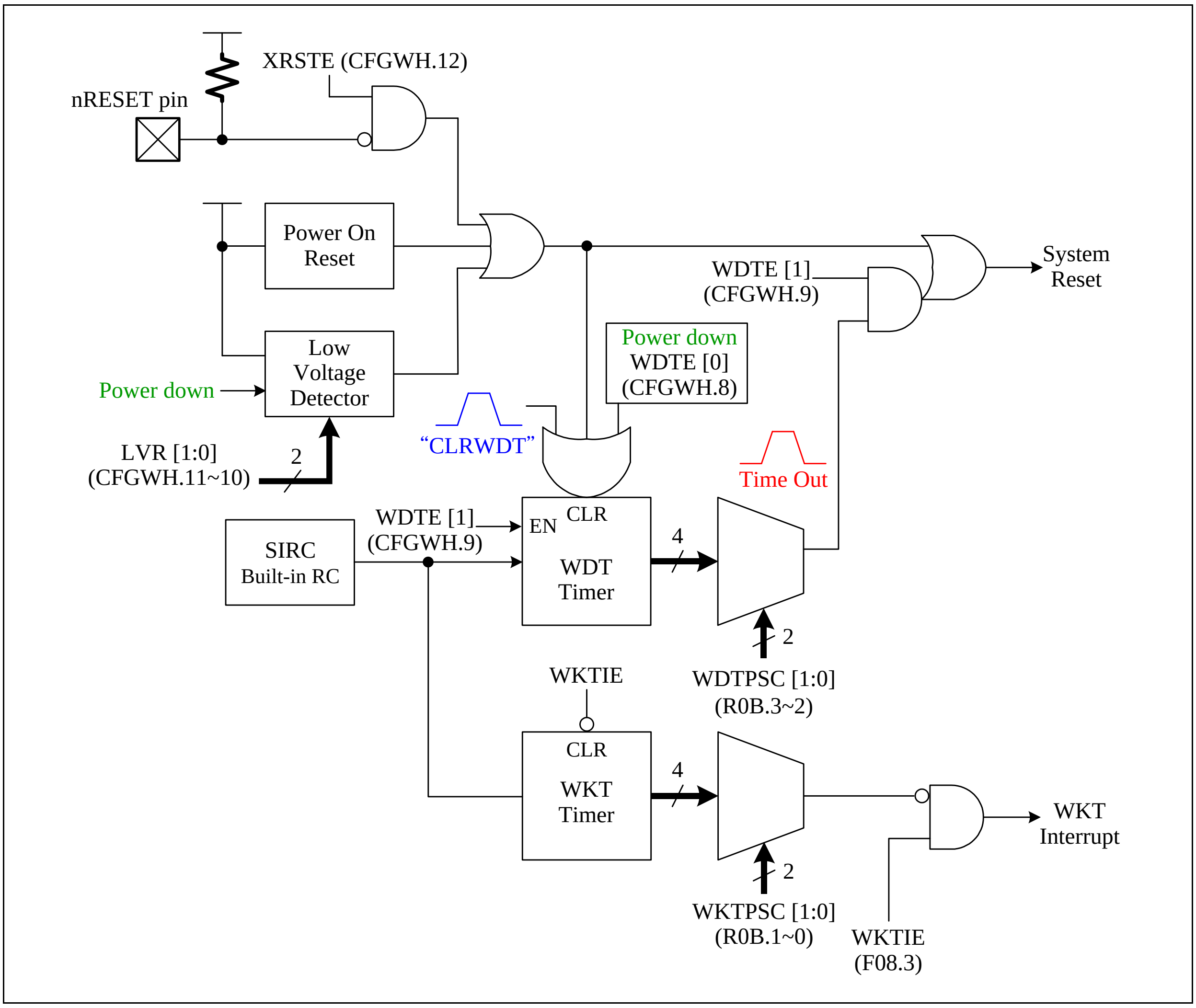
3. Peripheral Functional Block

3.1 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]=1), the WDT generates the chip reset signal. In IDLE mode, the WDT is only enabled when WDTE[1:0]=11B. Otherwise it will be disabled and stopped for power saving.

The WKT timer is an interval timer. When WKT overflow time out, it will generate overflow time out flag "WKTIF" (F09.3). The WKT timer is cleared/stopped by WKTIE=0. Set WKTIE=1, the WKT timer will generate WKT overflow time out interrupt and always count regardless at any CPU operating mode.

Watchdog clear is controlled by CLRWDT instruction and moving any value into WDTCLR (R04) is to clear watchdog timer.



WDT/WKT Block Diagram

If the user program needs the MCU totally shut down for power conservation in STOP mode, the below setting of control bits should be followed.

Mode		WDTE		WKTIE	Watchdog RC Oscillator
Normal Mode		0	x	0	Stop
		1	x	0	Run
		x	x	1	
Power Down Mode	IDLE Mode	0	x	0	Stop
		1	0		
		1	1	0	Run
		x	x	1	
	STOP Mode	0	x	0	Stop

◇ Example: Clear watchdog timer by CLRWDT instruction

```
MAIN:
            …                                     ; Execute program
            CLRWDT      WKTIE                     ; Execute CLRWDT instruction
            …
            GOTO        MAIN
```

◇ Example: Clear watchdog timer by WDTCLR register

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

◇ Example: Set WKT period and interrupt function

```
MAIN:
            MOVLW       xxxxxx10B                 ; WKTPSC=2, WKT period=64 ms @5V
            MOVWR       MR0B

            MOVLW       11110111B                 ; Clear WKT interrupt flag by using byte operation
                                                  ; Don’t use bit operation “BCF WKTIF” to clear flag
            MOVWF       INTIF                     ; F-Plane 09H

            MOVLW       00001000B                 ; Enable WKT interrupt function
            MOVWF       INTIE
```

F03	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	GB1	GB0	RAMBK	TO	PD	Z	DC	C
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F03.4 **TO:** WDT time out flag, read-only
0: after Power On Reset, LVR Reset, or CLRWDT/SLEEP instructions
1: WDT time out occurs

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	–	TM1IE	TM0IE	WKTIE	INT2IE	–	INT0IE
R/W	R/W	–	R/W	R/W	R/W	R/W	–	R/W
Reset	0	–	0	0	0	0	–	0

F08.3 **WKTIE:** Wakeup Timer interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	TKIF	–	TM1IF	TM0IF	WKTIF	INT2IF	–	INT0IF
R/W	R/W	–	R/W	R/W	R/W	R/W	–	R/W
Reset	0	–	0	0	0	0	–	0

F09.3 **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

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

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

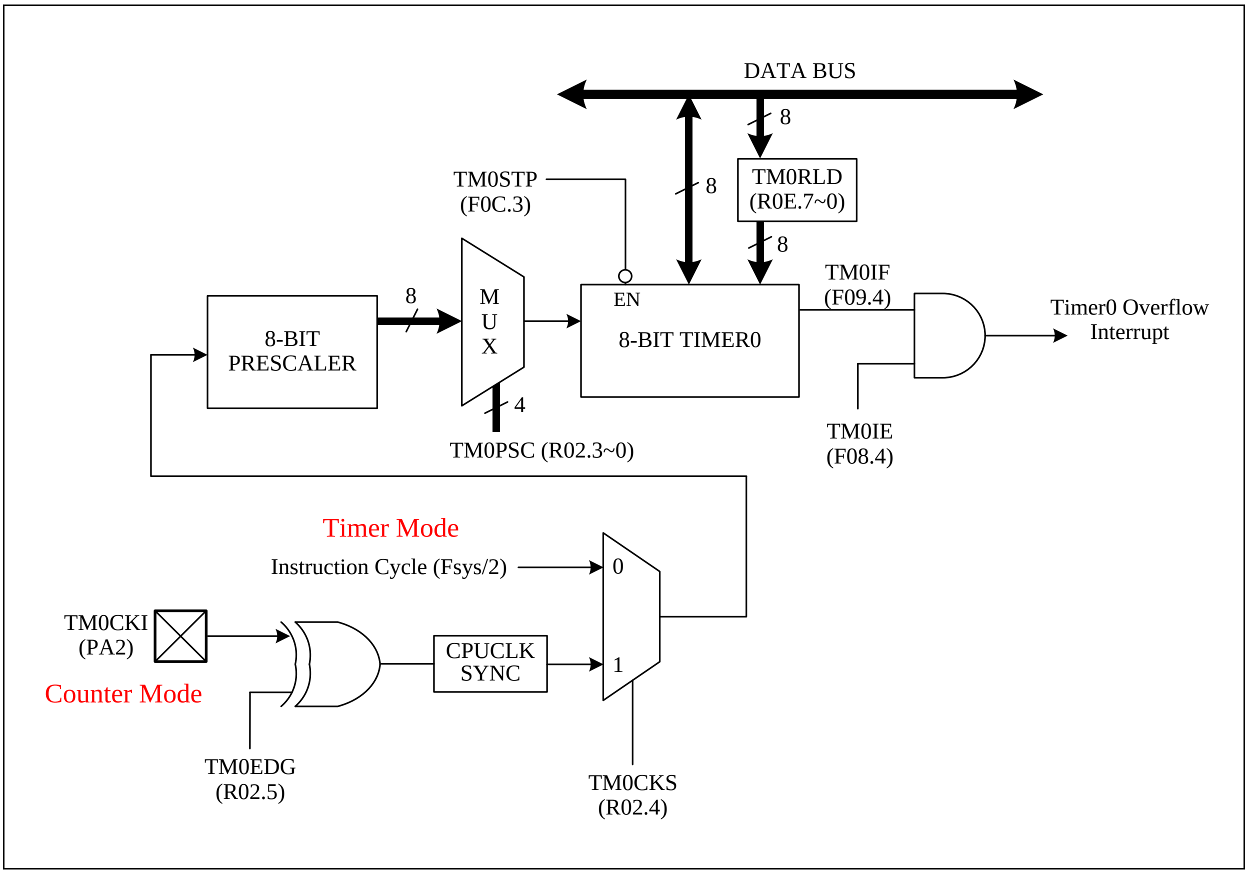
R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	HWAUTO	INT0EDG	–	–	WDTPSC		WKTPSC	
R/W	W	W	–	–	W		W	
Reset	0	0	–	–	1	1	1	1

R0B.3~2 **WDTPSC:** WDT pre-scale option select
00: WDT period is 52 ms, @5V; 75 ms, @3V
01: WDT period is 104 ms, @5V; 150 ms, @3V
10: WDT period is 208 ms, @5V; 300 ms, @3V
11: WDT period is 416 ms, @5V; 600 ms, @3V

R0B.1~0 **WKTPSC:** WKT pre-scale option select
00: WKT period is 26 ms, @5V; 37.5 ms, @3V
01: WKT period is 52ms, @5V; 75 ms, @3V
10: WKT period is 104 ms, @5V; 150 ms, @3V
11: WKT period is 208 ms, @5V; 300 ms, @3V

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

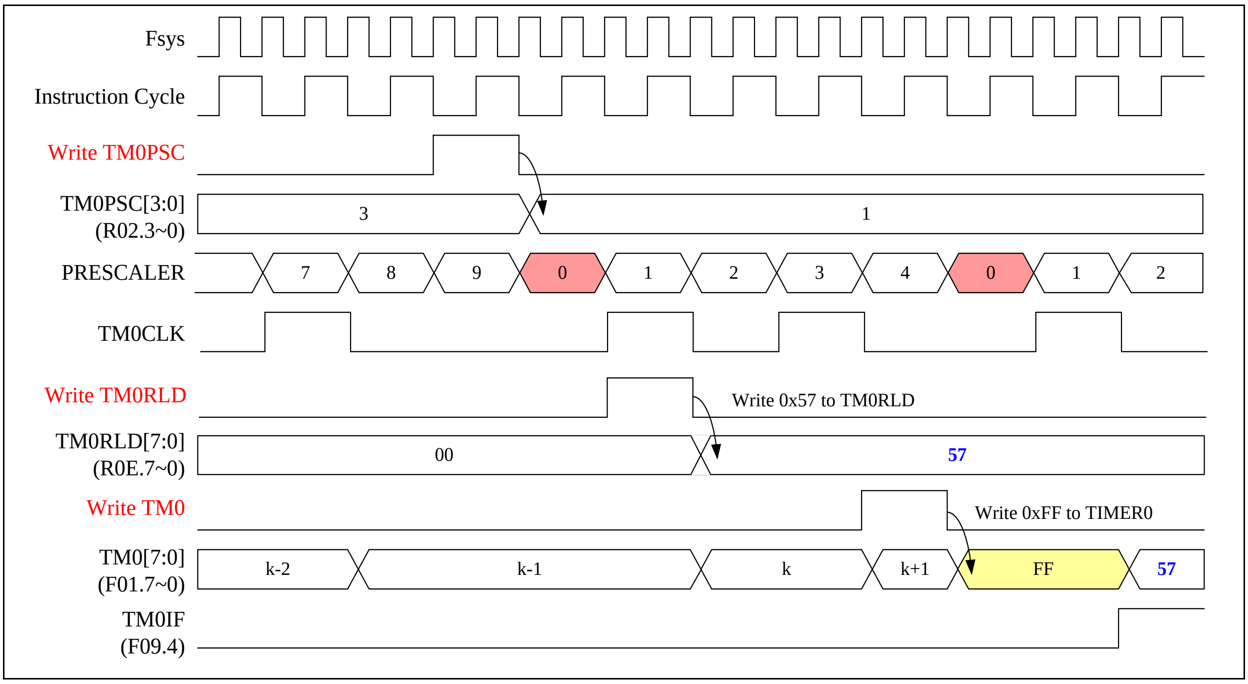
The Timer0 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatically rolls over based on the pre-scaled clock source, which can be the instruction cycle or TM0CKI (PA2) rising/falling input. The Timer0's increasing rate is determined by the TM0PSC (R02.3~0). The Timer0 can generate interrupt flag TM0IF (F09.4) and also reload the new data from TM0RLD (R0E.7~0) when it rolls over. It generates Timer0 interrupt if the TM0IE (F08.4) bit is set. Timer0 can be stopped counting if the TM0STP (F0C.3) bit is set.



Timer0 Block Diagram

Timer Mode:

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



Timer0 works in Timer mode (TM0CKS=0)

The equation of Timer0 interrupt timer value is as following:

Timer0 interrupt frequency=Instruction cycle time/TM0PSC/(256-TM0RLD)

◇ Example: Setup Timer0 work in Timer mode, Fsys=Fast-clock/CPUPSC=FIRC 8 MHz/2=4 MHz

```
; Setup Timer0 clock source and divider
            MOVLW     000000 10B
            MOVWF     CLKCTL            ; CPUPSC=10b, divided by 2
            BSF       CPUCKS            ; Set Fast-clock as system clock
            MOVLW     00x0 0100B        ; TM0CKS=0, Timer0 clock is instruction cycle
            MOVWR     TM0CTL            ; TM0PSC=0100b, divided by 16

; Setup Timer0 reload data
            MOVLW     80H
            MOVWR     TM0RLD            ; Set Timer0 reload data=128
```

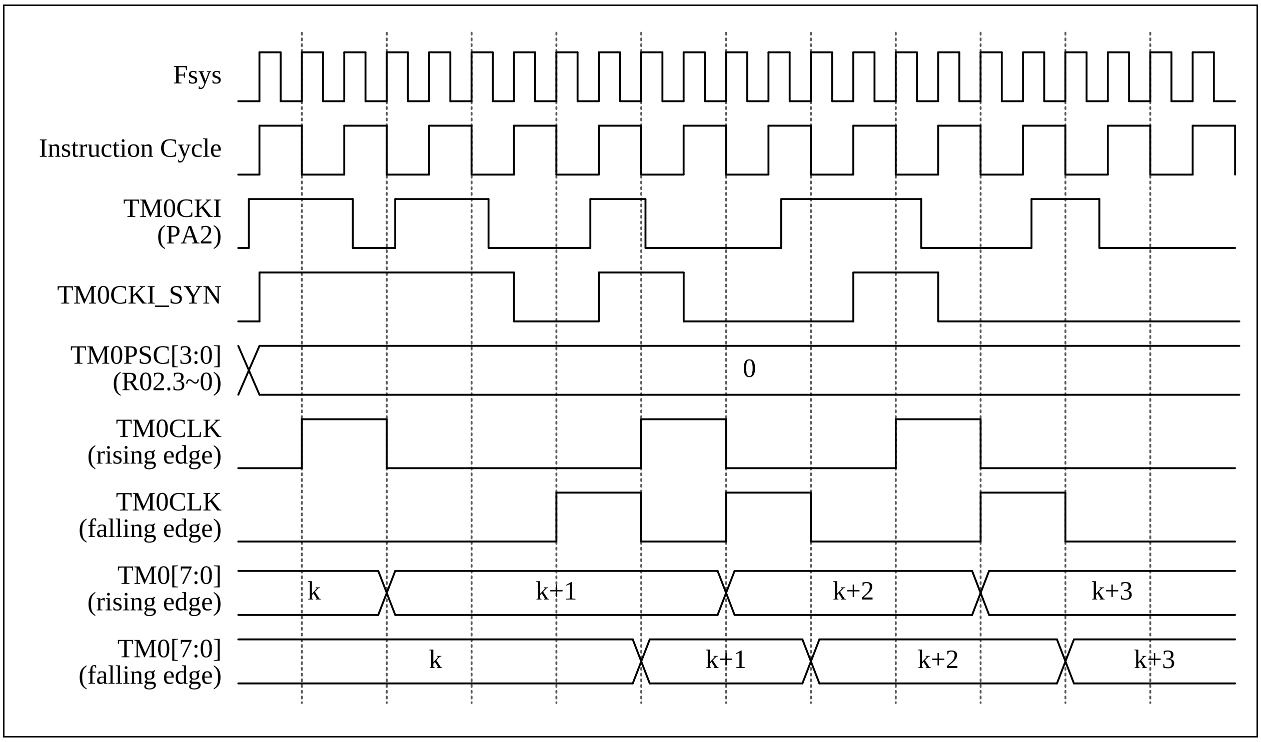
```
; Setup Timer0
            BSF         TM0STP                 ; Timer0 stops counting
            CLRF        TM0                    ; Clear Timer0 content

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

Timer0 clock source is Fsys/2=4 MHz/2=2 MHz, Timer0 divided by 16

Timer0 interrupt frequency=2 MHz/16/(256-128) =976.56 Hz

Counter Mode:

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



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

◇ Example: Setup Timer0 works in Counter mode

```
; Setup Timer0 clock source and divider
          MOVLW     001 1 0000B        ; TM0EDG=1, counting edge is falling edge
          MOVWR     TM0CTL             ; TM0CKS=1, Timer0 clock is TM0CKI (PA2)
                                       ; TM0PSC=0000b, divided by 1
; Setup Timer0
          BSF       TM0STP             ; Timer0 stops counting
          CLRF      TM0                ; Clear Timer0 content

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

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

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

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	–	TM1IE	TM0IE	WKTIE	INT2IE	–	INT0IE
R/W	R/W	–	R/W	R/W	R/W	R/W	–	R/W
Reset	0	–	0	0	0	0	–	0

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

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	TKIF	–	TM1IF	TM0IF	WKTIF	INT2IF	–	INT0IF
R/W	R/W	–	R/W	R/W	R/W	R/W	–	R/W
Reset	0	–	0	0	0	0	–	0

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

F0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0C	CLKFLT	VCCFLT	–	TM1STP	TM0STP	PWM0CLR	PWM0PSC	
R/W	R/W	R/W	–	R/W	R/W	R/W	R/W	
Reset	0	0	–	0	0	0	0	0

F0C.3 **TM0STP**: Timer0 counter stop
0: Timer0 is counting
1: Timer0 stops counting

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

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

R02.4 **TM0CKS:** Timer0 Prescaler clock select
0: Instruction Cycle as Timer0 Prescaler clock
1: TM0CKI (PA2) as Timer0 Prescaler clock

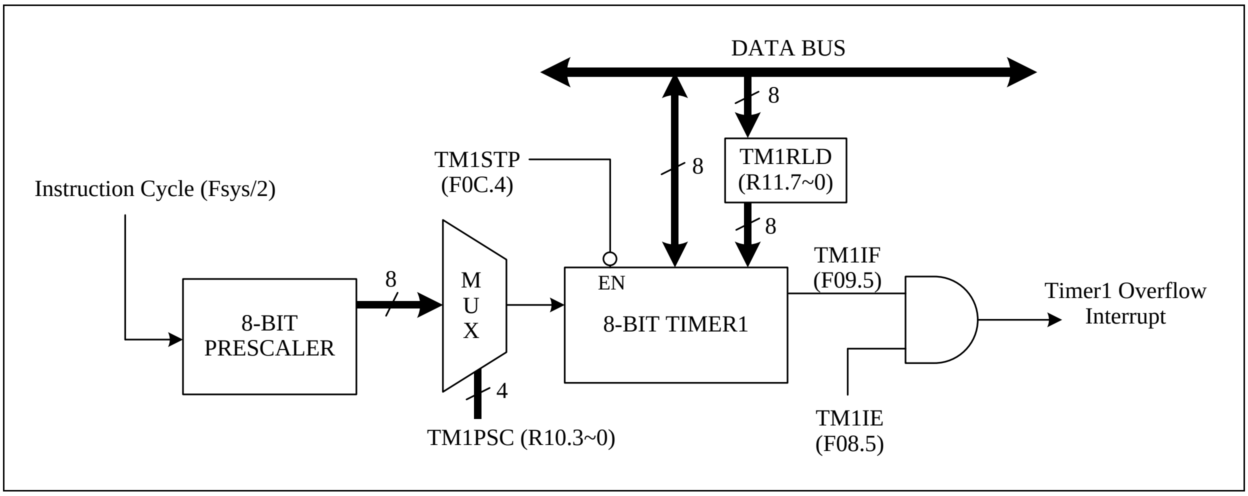
R02.3~0 **TM0PSC:** Timer0 prescaler. Timer0 clock source
0000: divided by 1
0001: divided by 2
0010: divided by 4
0011: divided by 8
0100: divided by 16
0101: divided by 32
0110: divided by 64
0111: divided by 128
1xxx: divided by 256

R0E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0RLD	TM0RLD							
R/W	W							
Reset	0	0	0	0	0	0	0	0

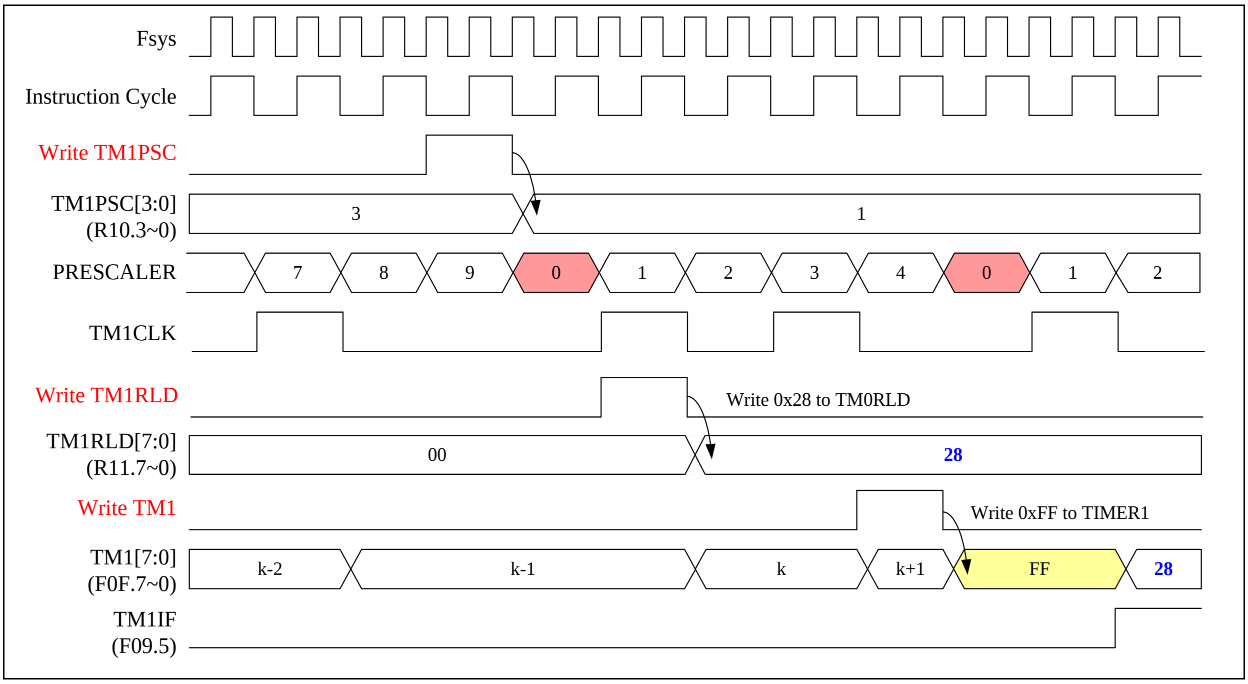
R0E.7~0 **TM0RLD:** Timer0 reload data

3.3 Timer1: 8-bit Timer with Pre-scale (PSC)

The Time1 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. It is almost the same as Timer0, except Timer1 doesn't have Counter Mode. Timer1 increases itself periodically and automatically rolls over based on the pre-scaled instruction cycle. The Timer1's increasing rate is determined by the TM1PSC (R10.3~0). The Timer1 can generate interrupt flag TM1IF (F09.5) and also reload the new data from TM1RLD (R11.7~0) when it rolls over. It generates Timer1 interrupt if the TM1IE (F08.5) bit is set. Timer1 can be stopped counting if the TM1STP (F0C.4) bit is set.



Timer1 Block Diagram



Timer1 works in Timer mode

When the Timer1 prescaler (TM1PSC) is written, the internal 8-bit prescaler will be cleared to 0 to make the counting period correct at the first Timer1 count. TM1CLK is the internal signal that causes the Timer1 to increase by 1 at the end of TM1CLK. TM1WR is also the internal signal that indicates the Timer1 is directly written by instruction; meanwhile, the internal 8-bit prescaler will be cleared. When Timer1 counts from FFh to TM1RLD data, TM1IF (Timer1 Interrupt Flag) will be set to 1 and generate interrupt if TM1IE (Timer1 Interrupt Enable) is set. The timing diagram describes the Timer1 works in pure Timer mode is shown in above.

The equation of Timer1 interrupt frequency is as following:

Timer1 interrupt frequency=Instruction cycle frequency/TM1PSC/(256-TM1RLD)

◇ Example: CPU is running in SLOW mode, Fsys = Slow-clock/CPUPSC=SIRC 80 KHz/2=40 KHz

```
; Setup Timer1 clock source and divider
            MOVLW     000000 10B          ; Set Slow-clock as system clock
            MOVWF     CLKCTL              ; CPUPSC=10b, divided by 2
            MOVLW     00000111B
            MOVWR     TM1CTL              ; TM1PSC=0111b, divided by 128

; Setup Timer1 reload data
            MOVLW     64H
            MOVWR     TM1RLD              ; Set Timer1 reload data=100

; Setup Timer1
            BSF       TM1STP              ; Timer1 stops counting
            CLRF      TM1                 ; Clear Timer1 content

; Enable Timer1 and interrupt function
            MOVLW     11011111B
            MOVWF     INTIF               ; Clear Timer1 request interrupt flag
            BSF       TM1IE               ; Enable Timer1 interrupt function
            BCF       TM1STP              ; Enable Timer1 counting
```

Timer1 clock source is Fsys/2=40 KHz/2=20 KHz, Timer1 divided by 128

Timer1 interrupt frequency=20 KHz/128/(256-100) =1.002 Hz

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	–	TM1IE	TM0IE	WKTIE	INT2IE	–	INT0IE
R/W	R/W	–	R/W	R/W	R/W	R/W	–	R/W
Reset	0	–	0	0	0	0	–	0

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

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	TKIF	–	TM1IF	TM0IF	WKTIF	INT2IF	–	INT0IF
R/W	R/W	–	R/W	R/W	R/W	R/W	–	R/W
Reset	0	–	0	0	0	0	–	0

F09.5 **TM1IF:** Timer1 interrupt event pending flag
This bit is set by H/W while Timer1 overflows, write 0 to this bit will clear this flag

F0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0C	CLKFLT	VCCFLT	–	TM1STP	TM0STP	PWM0CLR	PWM0PSC	
R/W	R/W	R/W	–	R/W	R/W	R/W	R/W	
Reset	0	0	–	0	0	0	0	0

F0C.4 **TM1STP**: Timer1 counter stop
0: Timer1 is counting
1: Timer1 stops counting

F0F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1	TM1							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F0F.7~0 **TM1:** Timer1 content

R10	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1CTL	–	–	–	–	TM1PSC			
R/W	–	–	–	–	W			
Reset	–	–	–	–	0	0	0	0

R10.3~0 **TM1PSC:** Timer1 prescaler. Timer1 clock source
0000: divided by 1 (Fsys/2)
0001: divided by 2 (Fsys/4)
0010: divided by 4 (Fsys/8)
0011: divided by 8 (Fsys/16)
0100: divided by 16 (Fsys/32)
0101: divided by 32 (Fsys/64)
0110: divided by 64 (Fsys/128)
0111: divided by 128 (Fsys/256)
1xxx: divided by 256 (Fsys/512)

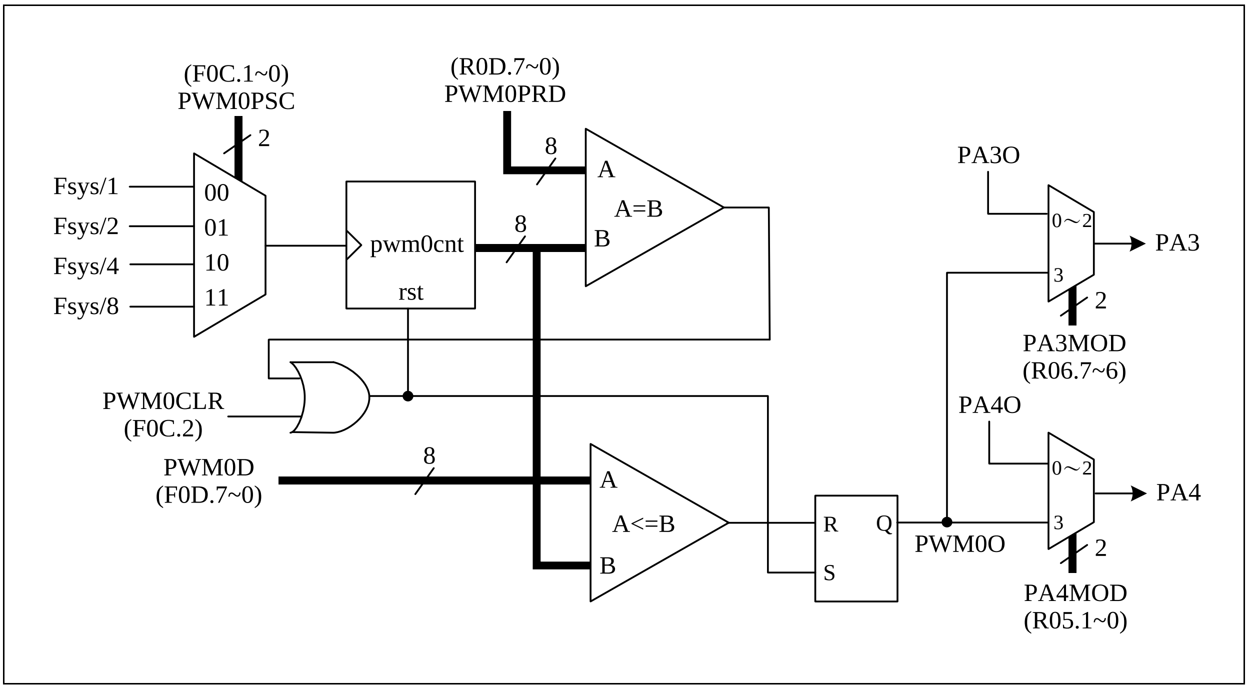
R11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1RLD	TM1RLD							
R/W	W							
Reset	0	0	0	0	0	0	0	0

R11.7~0 **TM1RLD:** Timer1 reload data

3.4 PWM0: 8-bit PWM

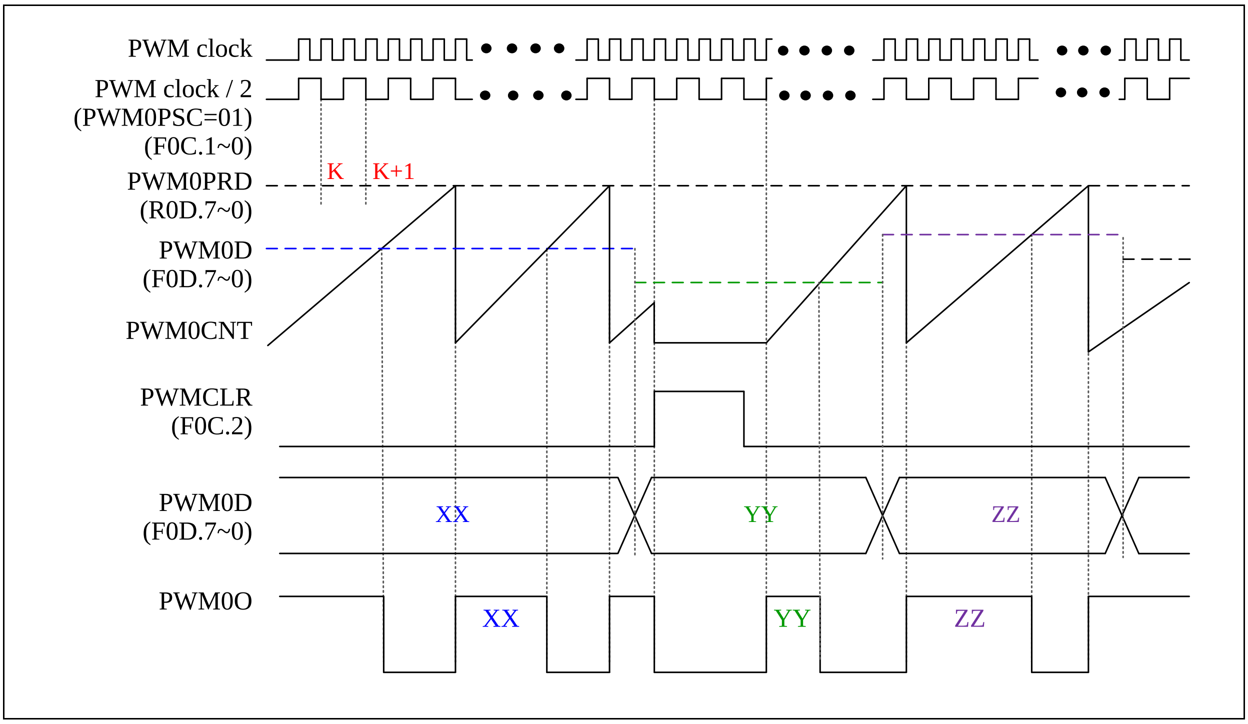
The chip has a built-in 8-bit PWM generator. The source clock comes from Fsys divided by 1, 2, 4, and 8 according to PWM0PSC (F0C.1~0). The PWM0 duty cycle can be changed with writing to PWM0D (F0D.7~0).

The PWM0 can output to PA3 or PA4 by Pin Mode setting. If PA3's Pin Mode "PA3MOD" (R06.7~6) is set as Mode3, the PWM0 will output to PA3. If PA4's Pin Mode "PA4MOD" (R05.1~0) is set as Mode3, the PWM0 will output to PA4. Setting the PWM0CLR (F0C.2) bit will clear the PWM0 counter and disable PWM0 output. Figure shows the block diagram of PWM0.



PWM0 Block Diagram

The next figure shows the PWM0 waveforms. When PWM0CLR bit is set or PWM0D equals to zero, the PWM0 output is cleared to '0' no matter what its current status is. Once the PWM0CLR bit is cleared and PWM0D is not zero, the PWM0 output is set to '1' to begin a new PWM cycle. PWM0 output will be '0' when PWM0CNT is greater than or equals to PWM0D. PWM0CNT keeps counting up when equals to PWM0PRD (R0D.7~0), the PWM0 output is set to '1' again.



PWM0 Block Diagram

◇ Example: CPU is running in FAST mode, Fsys=Fast-clock/CPUPSC=FIRC 8 MHz/2=4 MHz

```
; CLK_setting
          MOVLW     00000 1 10B          ; CPUCKS=1, select Fast-clock as system clock
          MOVWF     CLKCTL               ; CPUPSC=10B,System clock source divided by 2

; PINMODE _setting
          MOVLW     xxxxxx11B            ; PA4MOD=11B
          MOVWR     PAMODH               ; set PA4’s Pin Mode as Mode3 (PWM0 output)
          MOVLW     11xxxxxxB            ; PA3MOD=11B
          MOVWR     PAMODL               ; set PA3’s Pin Mode as Mode3 (PWM0 output)

; Setup PWM0 period and duty
          BSF       PWM0CLR              ; PWM0CLR=1, PWM0 clear and hold
          MOVLW     FFH
          MOVWR     PWM0PRD              ; Set PWM0 period=FFH+1=256
          MOVLW     80H
          MOVWF     PWM0D                ; Set PWM0 duty=80H=128

; Setup PWM0 prescaler and output enable
          MOVLW     xxxxx0 01B           ; PWM0CLR=0, PWM0 start working
          MOVWF     MF0C                 ; PWM0PSC=01B, divided by 2
```

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

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

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

F0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0C	CLKFLT	VCCFLT	–	TM1STP	TM0STP	PWM0CLR	PWM0PSC	
R/W	R/W	R/W	–	R/W	R/W	R/W	R/W	
Reset	0	0	–	0	0	0	0	0

F0C.2 **PWM0CLR**: PWM0 clear and hold
0: PWM0 is running
1: PWM0 is clear and hold

F0C.1~0 **PWM0PSC**: PWM0 prescaler, PWM0 clock source
00: divided by 1
01: divided by 2
10: divided by 4
11: divided by 8

F0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0D	PWM0D							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F0D.7~0 **PWM0D:** PWM0 duty

R05	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAMODH	–	PA7MOD	–	–	–	–	PA4MOD	
R/W	–	W	–	–	–	–	W	
Reset	–	0	–	–	–	–	0	1

PA4MOD: PA4 I/O mode control
00: Mode0
R05.1~0 01: Mode1
10: Mode2
11: Mode3, used as PWM0 output with high sink

R06	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAMODL	PA3MOD		PA2MOD		PA1MOD		PA0MOD	
R/W	W		W		W		W	
Reset	0	1	0	1	0	1	0	1

PA3MOD: PA3 I/O mode control
00: Mode0
R06.7~6 01: Mode1
10: Mode2
11: Mode3, used as PWM0 output with high sink

R0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0PRD	PWM0PRD							
R/W	W							
Reset	1	1	1	1	1	1	1	1

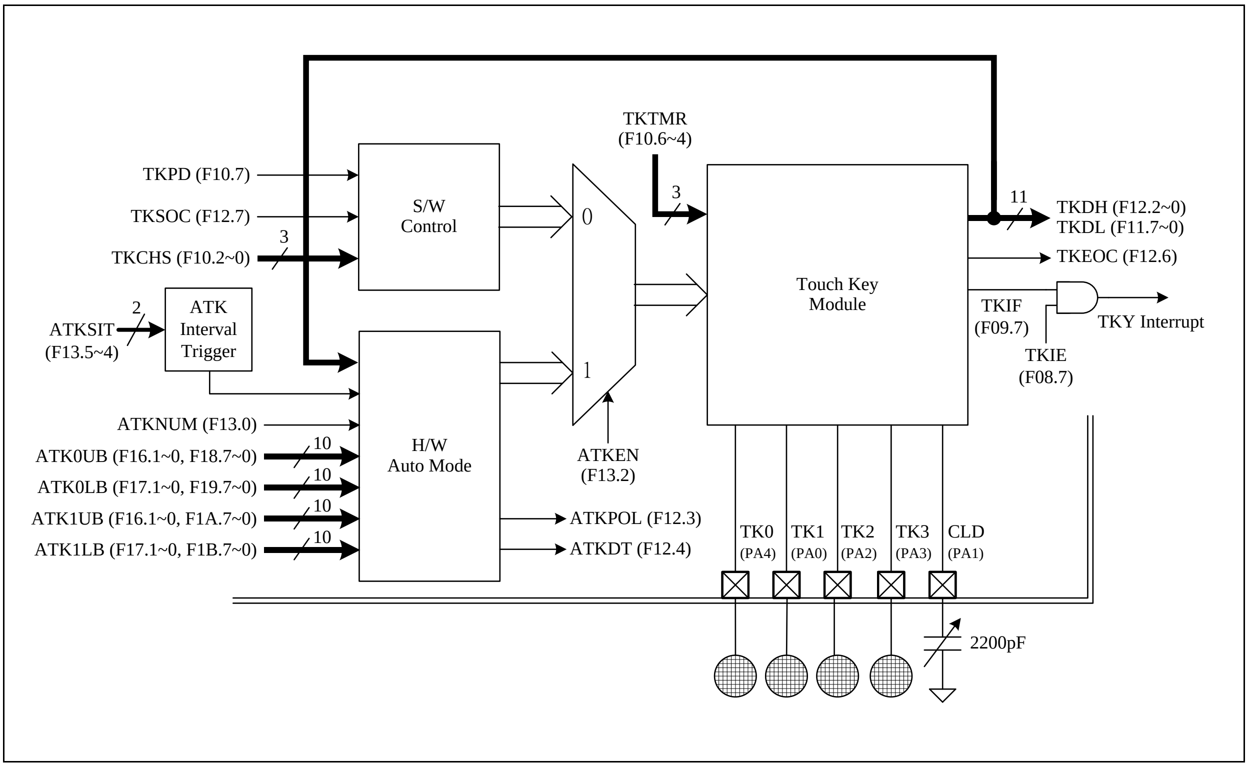
R0D.7~0 **PWM0PRD**: PWM0 period data

3.5 Touch Key (TM57M5526C only)

Only TM57M5526C supported the Touch Key function. The Touch Key offers an easy, simple and reliable method to implement finger touch detection. For most applications, it only requires an external capacitor component on CLD pin. The device support 4 channels touch key detection with S/W manual mode and 1~2 channel(s) touch key detection with H/W auto (ATK) mode. Only one mode can be active at a time.

To use the Touch Key, user must setup the Pin Mode (***see section 4***) correctly as below table. Setting Mode0 for Touch Key pin can pull up the pin and reduce the Key's mutual interference. While a TK pin is under scanning, TM57M5526C will disable the pull up resistor automatically.

Pin Mode Setting for Touch Key	TK0~1	TK2~3
Pin is not Touch Key	Mode0, 1, 2, 3	Mode0, 1, 2, 3
Pin is Touch Key, Idling	Mode0	Mode0
Pin is Touch Key, S/W Scanning	Mode0	Mode0
Pin is Touch Key, H/W Auto Scan	Mode0	N/A



Touch Key Block Diagram

S/W Manual Mode Touch Key Detection

All touch key (TK0~TK3) can be used for S/W mode, it can be select by TKCHS (F10.2~0) bits. To start the S/W mode, ATKEN (F13.2) and TKPD (F10.7) have to be cleared. After setting the TKSOC (F12.7) bit, the touch key starts conversion. The TKSOC bit can be automatically cleared while end of conversion. However, if the system clock is too slow, H/W might lose the auto clear TKSOC capability. “TKEOC=0” means conversion is in process, while “TKEOC=1” means the conversion is finish. When conversion is finish, TKIF (F09.7) will be set to 1 and generate interrupt if TKIE (F08.6) is set. After TKEOC’s (F12.6) edge rising, user must wait at least 10 us for next conversion. The touch key counting values is stored into the 11 bits touch key data count “TKDH (F12.2~0), TKDL (F11.7~0)”. Reduce/Increase TKTMR (F10.6~4), can reduce/increase touch key data count to adapt the system board circumstances.

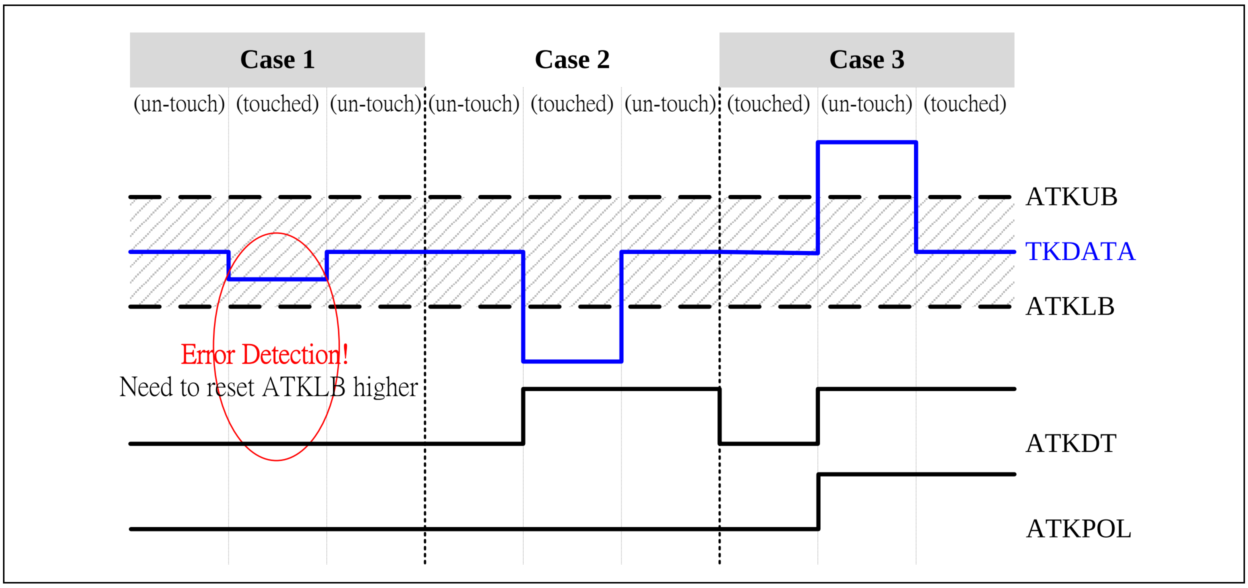
The Touch Key unit has an internal built-in reference capacitor to simulate the Key behavior. Set TKCHS=7 and start the Touch Key can get the TKDATA of this reference capacitor. Since the internal capacitor never affected by water or mobile phone, it is useful for comparing the environment background noise.

◇ Example: S/W Mode, Touch Key channel=TK3 (PA3).

```
            MOVLW     00xx11xxB           ; PA3MOD=0, PA1MOD=3
            MOVWR     PAMODH              ; Set PA3MOD as Mode0 for touch key input
                                          ; Set PA1MOD as Mode3 for touch key CLD
            BSF       PAD, 3              ; Set PA3 is input with pull-up

            MOVLW     1 100x011B          ; TKPD=1
            MOVWF     TKCTL1              ; TKTMR=4, TKCHS=3 (TK3)

            BCF       TKPD                ; TKPD=0
            :
            BSF       TKSOC               ; TKSOC=1, touch key start conversion
            NOP

  WAIT_TK:
            BTFSS     TKEOC               ; Polling TKEOC
            GOTO      WAIT_TK             ; Waiting touch key conversion finish

            MOVFW     TKDH                ; Read TKDH[2:0]
            MOVFW     TKDL                ; Read TKDL[7:0]
```

H/W Auto Mode Touch Key (ATK) Detection

Only TK0~TK1 are eligible for H/W auto mode by setting ATKNUM (F13.0). This function can work in normal/IDLE mode and save the S/W effort as well as minimize the chip current consumption. To use this function, user need to set ATKEN=1 and TKPD=1 to enable H/W fully control the TK unit. If ATKEN is set to "1", the ATK Interval Timer will generate an overflow flag after time out to trigger the touch key H/W auto mode starting. That can enable H/W control the touch key module fully. H/W then automatically detects the TK0~TK1's TK Data Count at every 28/56/112/224 ms (@V_{DD}=5V) rate by ATKSIT (F13.5~4). If a keys' TK Data Count is less than the pre-set compare lower boundary ATKnLB (n=0~1) or more than the pre-set compare upper boundary ATKnUB (n=0~1), H/W will generate interrupt flag TKIF (F9.7). It generates auto touch key interrupt and wake up CPU if the TKIE (F8.7) bit is set. At the same time, H/W will also record the compare result in the ATKDT (F14.4) and the scan polarity in the ATKPOL (F14.3). User can switch the TK module to S/W Manual Mode after the TK interrupt and identify/confirm the Key touch event.



◇ Example: H/W Auto Mode, Touch key auto scan number is 2 keys (TK0~TK1)

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

            ORG         001H                ; All interrupt vector
            GOTO        INT

            ORG         002H
START:
            MOVLW       xxx1xx11B           ; Set PAD[0]=1, PAD[1]=1, PAD[4]=1
            MOVWF       PAD

            MOVLW       xxxxxx00B           ; Set PA4 Pin Mode as Mode0
            MOVWR       PAMODH

            MOVLW       xxxx11 00B          ; Set PA0 Pin Mode as Mode0
            MOVWR       PAMODL              ; Set PA1 Pin Mode as Mode3
            MOVLW     1 100xxxxB          ; Make sure TKPD=1
            MOVWF     TKCTL               ; Set TKTMR= Level 4

            MOVLW     xx00x0x1B           ; ATKSIT=00B (ATK scan interval time=28ms)
            MOVWF     ATKCTL              ; ATKEN=0 (ATK scan disable)
                                          ; ATKNUM=1 (TK auto scan number=2)

            MOVLW     00H                 ; Set ATK compare boundary
            MOVWF     ATKUBH
            MOVLW     00H
            MOVWF     ATKLBH

            MOVLW     DCH
            MOVWF     ATK0UBL             ;Set ATK0UB=220
            MOVLW     C8H
            MOVWF     ATK0LBL             ;Set ATK0LB=200

            MOVLW     D7H
            MOVWF     ATK0UBL             ;Set ATK1UB=215
            MOVLW     C5H
            MOVWF     ATK0LBL             ;Set ATK1LB=197

            MOVLW     01111111B
            MOVWF     INTIF               ; Clear ATK interrupt request flag

            MOVLW     10000000B
            MOVWF     INTIE               ; Enable ATK interrupt

            BSF       ATKEN               ; ATKEN=1, enable TK H/W auto Mode
MAIN:
            (SLEEP)                       ; Set system into IDLE mode
            :                             ; to reduce power consumption (dispensable)
            :
            GOTO      MAIN

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

            BTFSC     TKIF                ; Check TKIF bit
            CALL      INT_ATK             ; TKIF=0, exit interrupt subroutine
                                          ; TKIF interrupt service routine
            …
            GOTO      EXIT_INT


INT_ATK:
            MOVLW     01111111B
            MOVWF     INTIF               ; Clear TK interrupt request flag
```

```
                MOVFW       TKDH
                MOVWF       22H                         ; Store ATK scan result to FRAM 22H
                RET

    EXIT_INT:
                MOVFW       21H                         ; Get FRAM 21H data
                MOVWF       STATUS                      ; Restore STATUS data
                MOVFW       20H                         ; Restore W data
                RETI                                    ; Return from interrupt
```

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	–	TM1IE	TM0IE	WKTIE	INT2IE	–	INT0IE
R/W	R/W	–	R/W	R/W	R/W	R/W	–	R/W
Reset	0	–	0	0	0	0	–	0

F08.7 **TKIE:** Auto touch key interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	TKIF	–	TM1IF	TM0IF	WKTIF	INT2IF	–	INT0IF
R/W	R/W	–	R/W	R/W	R/W	R/W	–	R/W
Reset	0	–	0	0	0	0	–	0

F09.7 **TKIF:** Touch Key interrupt event pending flag
This bit is set by H/W while Key's TK Data Count is over the pre-set compare threshold range (H/W auto mode) or TK is end of conversion (S/W manual mode), write 0 to this bit will clear this flag

F10	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCTL	TKPD	TKTMR			–	TKCHS		
R/W	R/W	R/W			–	R/W		
Reset	1	0	0	0	–	0	0	0

F10.7 **TKPD**: Touch key power down
0: Touch key running
1: Touch key power down

F10.6~4 **TKTMR**: Touch key conversion time select
000: Level 0 (shortest)
001: Level 1
010: Level 2
011: Level 3
100: Level 4
101: Level 5
110: Level 6
111: Level 7 (longest)

F10.2~0 **TKCHS**: Touch key channel select
000: TK0 (PA4)   100: reserved
001: TK1 (PA0)   100: reserved
100: TK2 (PA2)   100: reserved
011: TK3 (PA3)   111: internal reference capacitor

F11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKDL	TKDL							
R/W	R							
Reset	0	0	0	0	0	0	0	0

F11.7~0 **TKDL**: Touch key data LSB[7:0]

F12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKDH	TKSOC	TKEOC	–	ATKDT	ATKPOL	TKDH		
R/W	R/W	R	–	R	R	R		
Reset	0	0	–	0	0	0	0	0

F12.7 **TKSOC**: Touch key start of conversion, rising edge to start.
H/W auto cleared while end of conversion

F12.6 **TKEOC**: Touch key end of conversion
0: conversion is in process
1: end of conversion

F12.4 **ATKDT**: Touch key auto scan result
0: TK0 has a touch event
1: TK1 has a touch event

F12.3 **ATKPOL**: Touch key auto scan polarity
0: Touch key counter data is lower than lower boundary
1: Touch key counter data is higher than upper boundary

F12.2~0 **TKDH**: Touch key data MSB[10:8]

F13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATKCTL	–	–	ATKSIT		–	ATKEN	–	ATKNUM
R/W	–	–	R/W		–	R/W	–	R/W
Reset	–	–	0	0	–	0	–	1

F13.5~4 **ATKSIT**: Touch Key auto scan interval time
00: 26 ms @V_{DD}=5V, 37.5 ms @V_{DD}=3V
01: 52 ms @V_{DD}=5V, 75 ms @V_{DD}=3V
10: 104 ms @V_{DD}=5V, 150 ms @V_{DD}=3V
11: 208 ms @V_{DD}=5V, 300 ms @V_{DD}=3V

F13.2 **ATKEN**: Touch key auto scan mode enable
0: disable H/W Auto Mode
1: enable H/W Auto Mode

F13 0 **ATKNUM**: Touch key auto scan channel number
0: only detect TK0
1: detect TK0 and TK1

F16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATKUBH	–	–	–	–	ATK1UBH		ATK0UBH	
R/W	–	–	–	–	R/W		R/W	
Reset	–	–	–	–	1	1	1	1

F16.3~2 **ATK1UBH**: Touch key auto scan TK1 upper boundary MSB[9:8]

F16.1~0 **ATK0UBH**: Touch key auto scan TK0 upper boundary MSB[9:8]

F17	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATKLBH	–	–	–	–	ATK1LBH		ATK0LBH	
R/W	–	–	–	–	R/W		R/W	
Reset	–	–	–	–	0	0	0	0

F17.3~2 **ATK1LBH**: Touch key auto scan TK1 lower boundary MSB[9:8]

F17.1~0 **ATK0LBH**: Touch key auto scan TK0 lower boundary MSB[9:8]

F18	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATK0UBL	ATK0UBL							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

F18.7~0 **ATK0UBL**: Touch key auto scan TK0 upper boundary LSB[7:0]

F19	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATK0LBL	ATK0LBL							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F19.7~0 **ATK0LBL**: Touch key auto scan TK0 lower boundary LSB[7:0]

F1A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATK1UBL	ATK1UBL							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

F1A.7~0 **ATK1UBL**: Touch key auto scan TK1 upper boundary LSB[7:0]

F1B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATK1LBL	ATK1LBL							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F1B.7~0 **ATK1LBL**: Touch key auto scan TK1 lower boundary LSB[7:0]

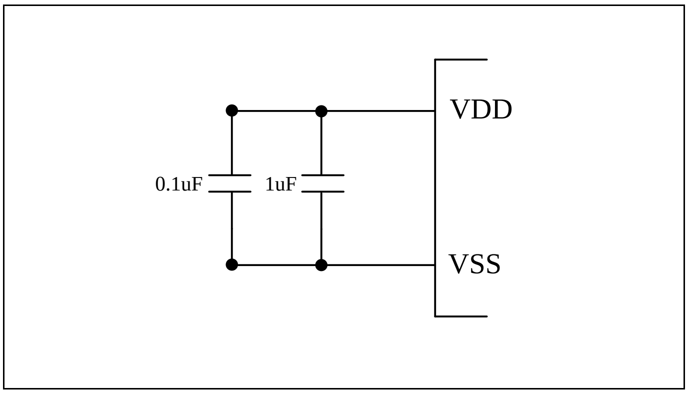
R06	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAMODL	PA3MOD		PA2MOD		PA1MOD		PA0MOD	
R/W	W		W		W		W	
Reset	0	1	0	1	0	1	0	1

R06.3~2 **PA1MOD**: PA1 I/O mode control

00: Mode0
01: Mode1
10: Mode2
11: Mode3, used as Touch Key CLD

3.6 System Clock Oscillator

System clock can be operated in two different oscillation modes. The two oscillation modes are FIRC and SIRC. In the Fast Internal RC mode (FIRC), the on-chip oscillator generates 8 MHz system clock. For the operation voltage can be protected by LVR, we suggest setting the FIRC/2=4 MHz as system clock. In the Slow Internal RC mode (SIRC), the on-chip oscillator generates 80 KHz system clock. Since power noise degrades the performance of Internal Clock Oscillator, placing power supply bypass capacitors 1 µF and 0.1 µF very close to VDD/VSS pins to improve the stability of clock and the overall system.



Internal RC Mode

4. I/O Port

4.1 PA0-4

These pins can be used as Schmitt-trigger input, CMOS push-pull output or Open-drain output. The pull-up resistor is assignable to each pin by S/W setting. User can set each pin by their Pin Mode register. There are 4 kinds of pin modes Mode0, Mode1, Mode2 and Mode3 for each pin can be selected.

Mode	PA0-4 pin function		PXn SFR data	Pin State	Resistor Pull-up	Digital Input
Mode 0	Open Drain		0	Drive Low	N	N
			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
Mode 3	Alternative Function	Wake-up	0	Drive Low	N	N
			1	Pull-up	Y	Y
		Touch Key CLD (TM57M5526C only)	x	–	N	N
		PWM output	0	PWM0	N	N
			1			

PA0-4 I/O Pin Function Table

If a PA0-4 pin is used for Schmitt-trigger input, S/W must set the I/O pin to Mode0 or Mode1 and set the corresponding Port Data SFR to 1 to disable the pin's output driving circuitry. Beside I/O port function, each PA0-4 pin has one or more alternative functions. Most of the functions are activated by setting the individual pin mode control SFR to Mode3. Reading the pin data (PA0-4) has different meaning. In "Read-Modify-Write" instruction, CPU actually reads the output data register. In the other instructions, CPU reads the pin state. The so-called "Read-Modify-Write" instruction includes BSF, BCF and all instructions using F-Plane as destination.

Pin Name	Function	INT	WAKEUP	TKY	Mode3
PA0	–	INT0	WAKEUP	TK1	WAKEUP
PA1	–	–	–	CLD	CLD
PA2	TM0CKI	–	WAKEUP	TK2	WAKEUP
PA3	PWM0	–	–	TK3	PWM0
PA4	PWM0	–	–	TK0	PWM0

PA0-4 multi-function Table

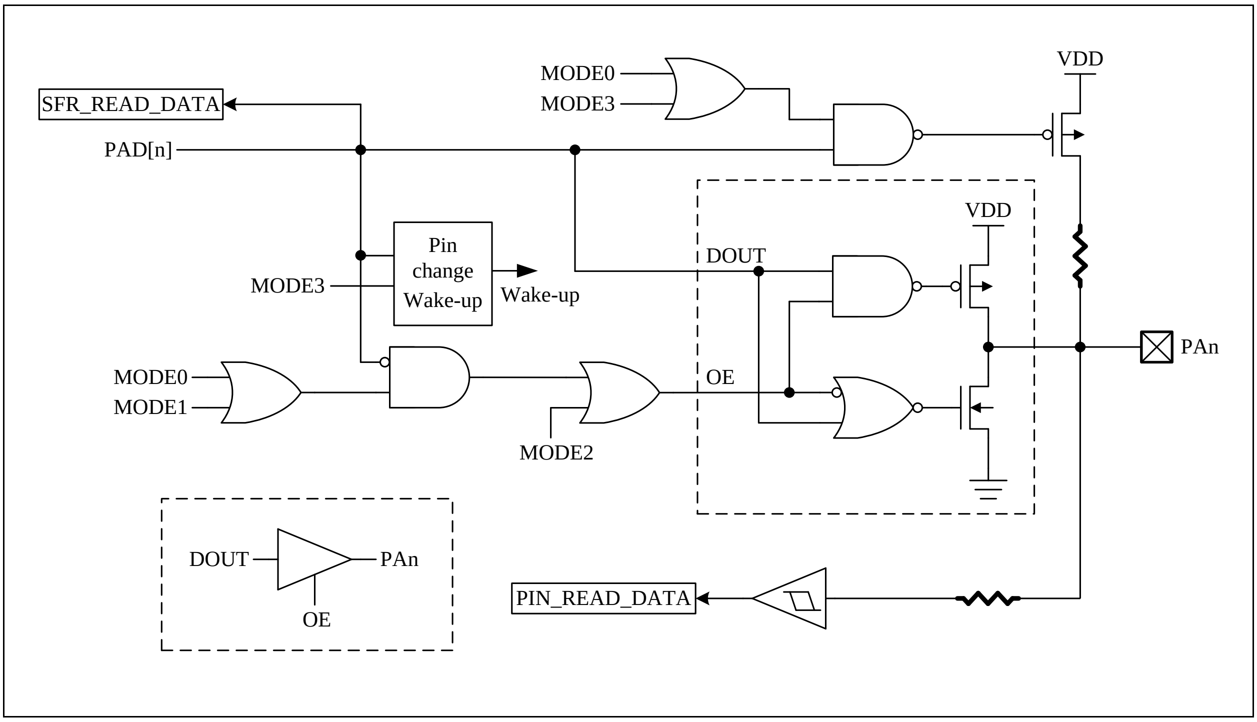
The necessary SFR setting for PA0-4 pin's alternative function is list below.

Alternative Function	Mode	PXn SFR data	Pin State	Other necessary SFR setting
TM0CKI, INT0	**0**	1	Input with Pull-up	TM0CTL, INTIE
	1	1	Input	
TK0~TK3 (TM57M5526C only)	**0**	1	Touch Key Idling with Pull-up	TKCTL, ATKCTL
			Touch Key Scanning without Pull-up (automatically)	
CLD (TM57M5526C only)	**3**	x	Touch Key Idling with Pull-up	–
Wake-up	**3**	0	Input	–
		1	Input with Pull-up	
PWM0	**3**	x	PWM CMOS Push-Pull Output	MF0C

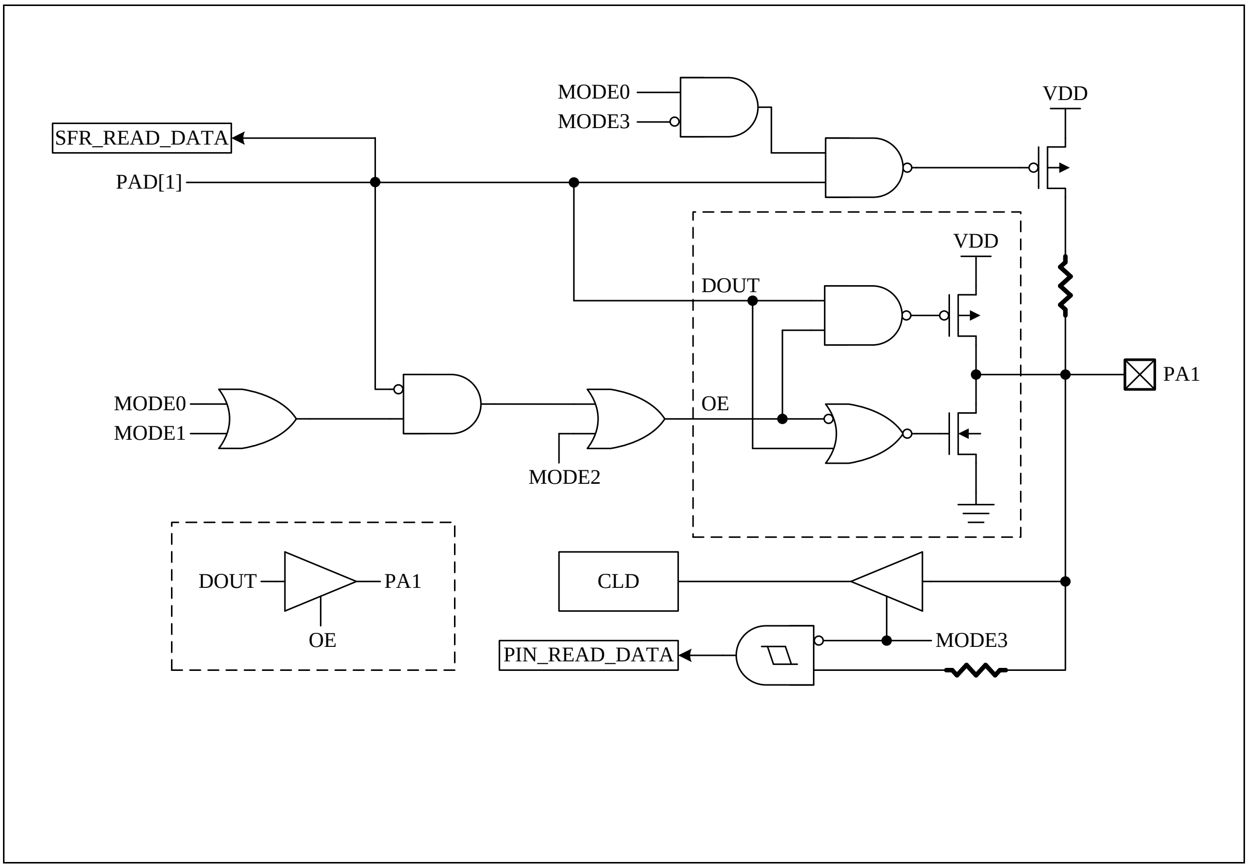
Mode Setting for PA0-4 Alternative Function

For tables above, a "CMOS Output" pin means it can sink and drive at least 4mA current. It is not recommended to use such pin as input function.

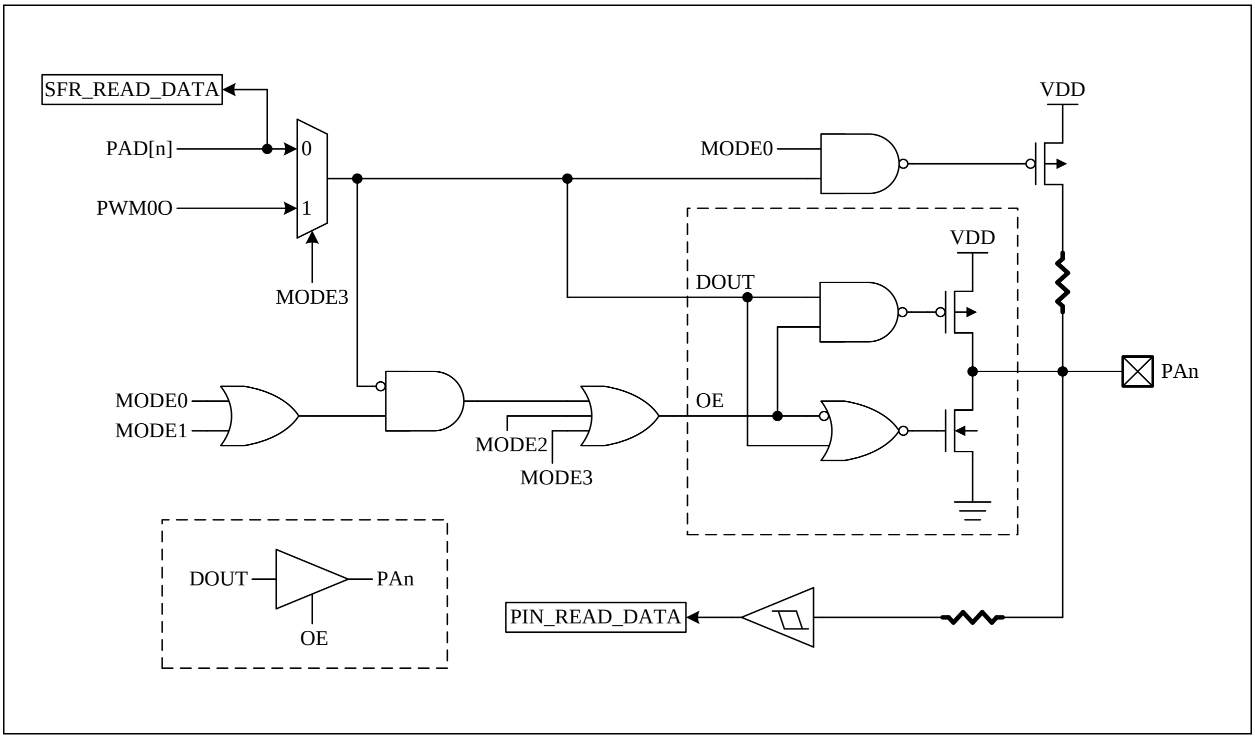
An "Open Drain" pin means it can sink at least 4mA current but only drive a small current (<20µA). It can be used as input or output function and typically needs an external pull up resistor.



PA0, PA2 Pin Structure



PA1 Pin Structure



PA3, PA4 Pin Structure

◇ Example: Set PA0 as Schmitt-trigger input with pull-up (Mode0)

```
MOVLW     xxxxxxx1B
MOVWF     PAD
MOVLW     xxxxxx00B
MOVWR     PAMODL              ; Set PA0 as Schmitt-trigger input with pull-up
```

◇ Example: Set PA0 as Schmitt-trigger input without pull-up (Mode1)

```
MOVLW     xxxxxxx1B
MOVWF     PAD
MOVLW     xxxxxx01B
MOVWR     PAMODL              ; Set PA0 as Schmitt-trigger input without pull-up
```

◇ Example: Set PA0 as CMOS push-pull output mode and drive Low (Mode2)

```
MOVLW     xxxxxxx0B           ; PAD[0] =0
MOVWF     PAD                 ; Set PA0 as CMOS push-pull output Low
MOVLW     xxxxxx10B
MOVWR     PAMODL
```

◇ Example: Set PA0 as Schmitt-trigger input and pin change wake up control (Mode3)

```
MOVLW     xxxxxxx1B
MOVWF     PAD
MOVLW     xxxxxx11B           ; Set PA0 as Schmitt-trigger input with pull-high
MOVWR     PAMODL              ; and enable low level wake up
```

◇ Example: Set PA1 as TKY CLD (Mode3) (TM57M5526C only)

```
MOVLW     xxxx11xxB
MOVWR     PAMODL              ; Set PA1 as TKY CLD
```

◇ Example: Set PA3 and PA4 as PWM0 output (Mode3)

```
MOVLW     xxxxxx11B
MOVWR     PAMODH              ; Set PA4 as PWM0 output
MOVLW     11xxxxxxB
MOVWR     PAMODL              ; Set PA3 as PWM0 output
```

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

F05.4~0 **PAD:** PA4~PA0 data
0: output low
1: output high or Schmitt-trigger input mode

R05	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAMODH	–	PA7MOD	–	–	–	–	PA4MOD	
R/W	–	W	–	–	–	–	W	
Reset	–	0	–	–	–	–	0	1

R05.1~0 **PA4MOD**: PA4 I/O mode control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, used as PWM0 output with high sink

R06	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAMODL	PA3MOD		PA2MOD		PA1MOD		PA0MOD	
R/W	W		W		W		W	
Reset	0	1	0	1	0	1	0	1

R06.7~6 **PA3MOD**: PA3 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, used as PWM0 output with high sink

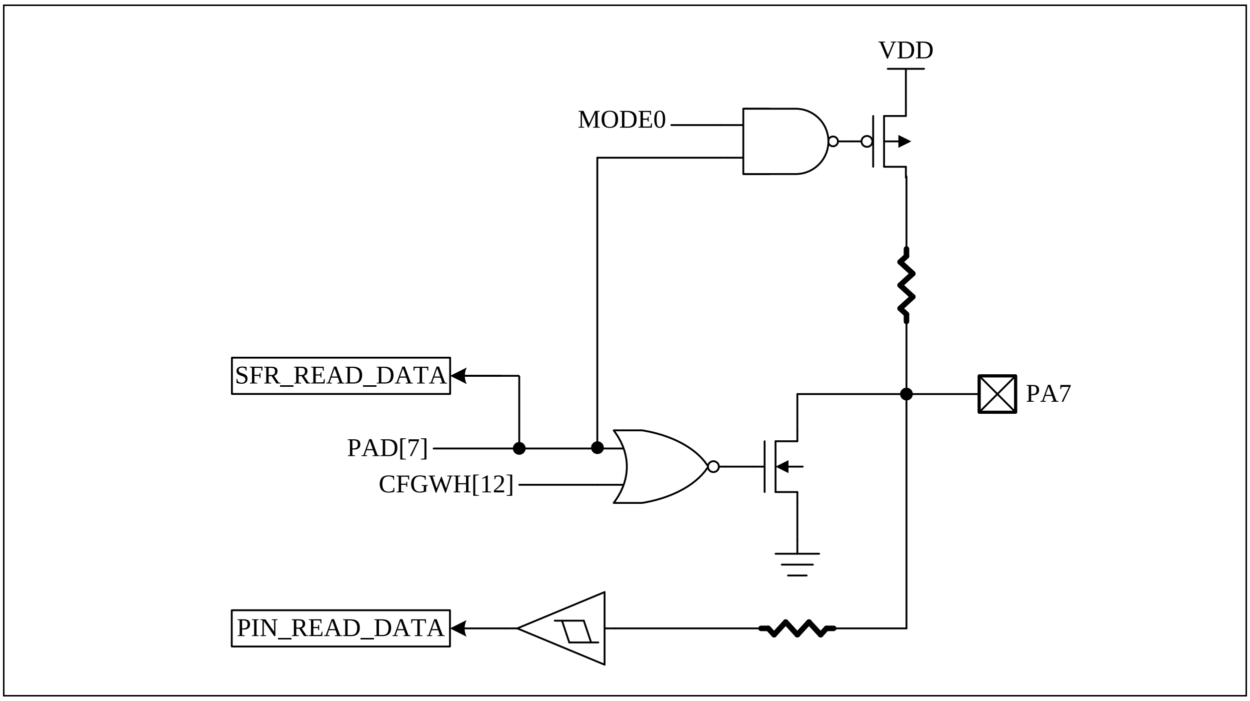
R06.5~4 **PA2MOD**: PA2 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, used as pin change wake up

R06.3~2 **PA1MOD**: PA1 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, used as TKY CLD (TM57M5526C only)

R06.1~0 **PA0MOD**: PA0 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, used as pin change wake up

4.2 PA7

PA7 can be used in Schmitt-trigger input or open-drain output which is setting by the PAD[7] (F05.7) bit. When the PAD[7] bit is set, PA7 is assigned as Schmitt-trigger input mode, otherwise is assigned as open-drain output mode and output low. The pull-up resistor is connected or not to this pin by pin mode setting. If PA7's pin mode is set as Mode0, PA7 can be used in Schmitt-trigger input with pull-up or open-drain output. Otherwise, the pull-up resistor is disabled if PA7's pin mode is set as Mode1. When CFGWH[12] is set, PA7 is only used in Schmitt-trigger input for external active low reset.



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

Mode	PA7 pin function	PAD7 SFR data	Pin State	Resistor Pull-up	Digital Input
Mode 0	Open Drain	0	Drive Low	N	N
		1	Pull-up	Y	Y
Mode 1	Open Drain	0	Drive Low	N	N
		1	Hi-Z	N	Y

◇ 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
```

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

F05.7 **PAD7:** PA7 data or pin mode control

0: PA7 is open-drain output mode and output low

1: PA7 is Schmitt-trigger input mode

R05	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAMODH	–	PA7MOD	–	–	–	–	PA4MOD	
R/W	–	W	–	–	–	–	W	
Reset	–	0	–	–	–	–	0	1

R05.6 **PA7MOD**: PA7 I/O mode control

0: Mode0

1: Mode1

MEMORY MAP

F-Plane

Name	Address	R/W	Rst	Description
(F00) INDF				**Function related to: RAM W/R**
INDF	00.7~0	R/W	-	Not a physical register, addressing INDF actually point to the register whose address is contained in the FSR register
(F01) TM0				**Function related to: Timer0**
TM0	01.7~0	R/W	0	Timer0 content
(F02) PCL				**Function related to: Programming Counter**
PCL	02.7~0	R/W	0	Programming counter LSB[7:0]
(F03) STATUS				**Function related to: Status**
GB1	03.7	R/W	0	General purpose bit 1
GB0	03.6	R/W	0	General purpose bit 0
RAMBK	03.5	R/W	0	FRAM Bank selection, 0: Bank0, 1: Bank1
TO	03.4	R	0	WDT time out flag
PD	03.3	R	0	STOP mode flag
Z	03.2	R/W	0	Zero flag
DC	03.1	R/W	0	Decimal Carry flag or Decimal/Borrow flag
C	03.0	R/W	0	Carry flag or /Borrow flag
(F04) FSR				**Function related to: RAM W/R**
FSR	04.6~0	R/W	-	File select register, indirect address mode pointer
(F05) PAD				**Function related to: Port A**
PAD7	05.7	R	-	PA7 pin or "data register" state
		W	1	0: PA7 is open-drain output mode 1: PA7 is Schmitt-trigger input mode
PAD	05.4~0	R	-	Port A pin or "data register" state
		W	1F	Port A output data register
(F08) INTIE				**Function related to: Interrupt Enable**
TKIE **(TM57M5526C only)**	08.7	R/W	0	Touch key interrupt enable 0: disable 1: enable
TM1IE	08.5	R/W	0	Timer1 interrupt enable 0: disable 1: enable
TM0IE	08.4	R/W	0	Timer0 interrupt enable 0: disable 1: enable
WKTIE	08.3	R/W	0	Wakeup timer interrupt enable 0: disable 1: enable
INT2IE	08.2	R/W	0	INT2 (PA7) falling interrupt enable 0: disable 1: enable
INT0IE	08.0	R/W	0	INT0 (PA0) falling/rising interrupt enable 0: disable 1: enable

Name	Address	R/W	Rst	Description
(F09) INTIF				**Function related to: Interrupt Flag**
TKIF (TM57M5526C only)	09.7	R	-	Touch Key interrupt event pending flag, set by H/W while Key's TK Data Count is over the pre-set compare threshold range (H/W auto mode) or TK is end of conversion (S/W manual mode)
		W	0	0: clear this flag 1: no action
TM1IF	09.5	R	-	Timer1 interrupt event pending flag, set by H/W while Timer1 overflows
		W	0	0: clear this flag 1: no action
TM0IF	09.4	R	-	Timer0 interrupt event pending flag, set by H/W while Timer0 overflows
		W	0	0: clear this flag 1: no action
WKTIF	09.3	R	-	WKT interrupt event pending flag, set by H/W while WKT is timeout
		W	0	0: clear this flag 1: no action
INT2IF	09.2	R	-	INT2 (PA7) pin falling interrupt pending flag, set by H/W at INT2 pin's falling edge
		W	0	0: clear this flag 1: no action
INT0IF	09.0	R	-	INT0 (PA0) pin falling/rising interrupt pending flag, set by H/W at INT0 pin's falling/rising edge
		W	0	0: clear this flag 1: no action
(F0A) PCH				**Function related to: Programming Counter**
PCH	0A.2~0	R/W	0	Programming counter MSB[10:8]
(F0B) CLKCTL				**Function related to: Fsys**
SLOWSTP	0B.4	R/W	0	Slow-clock stop 0: Slow-clock is running 1: Slow-clock stops running in Power-down mode
FASTSTP	0B.3	R/W	1	Fast-clock stop 0: Fast-clock is running 1: Fast-clock stops running
CPUCKS	0B.2	R/W	0	System clock source select 0: Slow-clock 1: Fast-clock (forbid using when CPUPSC=3)
CPUPSC	0B.1~0	R/W	3	System clock source prescaler. System clock source 00: divided by 16 01: divided by 4 10: divided by 2 11: divided by 1 (forbid using when CPUCKS=1)

Name	Address	R/W	Rst	Description
(F0C) MF0C				**Function related to: Fsys/EFT/Timer0/Timer1/PWM0**
CLKFLT	0C.7	R/W	0	Fsys clock filter for noise defending 0: Higher Fsys 1: Lower Fsys
VCCFLT	0C.6	R/W	0	VCC filter, enhance the chip's power noise immunity 0: disable 1: enable
TM1STP	0C.4	R/W	0	Timer1 counter stop 0: Release 1: Stop counting
TM0STP	0C.3	R/W	0	Timer0 counter stop 0: Release 1: Stop counting
PWM0CLR	0C.2	R/W	0	PWM0 clear and hold 0: PWM0 is running 1: PWM0 is clear and hold
PWM0PSC	0C.1~0	R/W	0	PWM0 prescaler, PWM0 clock source 00: divided by 1 01: divided by 2 10: divided by 4 11: divided by 8
(F0D) PWM0D				**Function related to: PWM0**
PWM0D	0D.7~0	R/W	0	PWM0 duty
(F0F) TM1				**Function related to: Timer1**
TM1	0F.7~0	R/W	0	Timer1 content
(F10) TKCTL (TM57M5526C only)				**Function related to: Touch Key**
TKPD	10.7	R/W	1	Touch key power down 0: Touch key running 1: Touch key power down
TKTMR	10.6~4	R/W	0	Touch key conversion time select 000: Level 0 (shortest) 001: Level 1 010: Level 2 011: Level 3 100: Level 4 101: Level 5 110: Level 6 111: Level 7 (longest)
TKCHS	10.2~0	R/W	0	Touch key channel select 000: TK0 (PA4) 001: TK1 (PA0) 010: TK2 (PA2) 011: TK3 (PA3) 100: reserved 101: reserved 110: reserved 111: internal reference capacitor
(F11) TKDL (TM57M5526C only)				**Function related to: Touch Key**
TKDL	11.7~0	R	0	Touch key data LSB[7:0]

Name	Address	R/W	Rst	Description
(F12) TKDH (TM57M5526C only)				**Function related to: Touch Key**
TKSOC	12.7	R/W	0	Touch Key start of conversion, rising edge to start H/W auto cleared while end of conversion
TKEOC	12.6	R	0	Touch key end of conversion 0: conversion is in process 1: end of conversion
ATKDT	12.4	R	0	Touch key auto scan result 0: TK0 has a touch event 1: TK1 has a touch event
ATKPOL	12.3	R	0	Touch key auto scan polarity 0: Touch key counter data is lower than lower boundary 1: Touch key counter data is higher than upper boundary
TKDH	12.2~0	R	0	Touch key data MSB[10:8]
(F13) ATKCTL (TM57M5526C only)				**Function related to: Touch Key**
ATKSIT	13.5~4	R/W	0	Touch Key auto scan interval time 00: 26 ms @VDD=5V, 37.5 ms @VDD=3V 01: 52 ms @VDD=5V, 75 ms @VDD=3V 10: 104 ms @VDD=5V, 150 ms @VDD=3V 11: 208 ms @VDD=5V, 300 ms @VDD=3V
ATKEN	13.2	R/W	0	Touch key auto scan mode enable 0: disable H/W auto mode 1: enable H/W auto mode
ATKNUM	13. 0	R/W	1	Touch key auto scan channel number 0: only detect TK0 1: detect TK0 and TK1
(F16) ATKUBH (TM57M5526C only)				**Function related to: Touch Key**
ATK1UBH	16.3~2	R/W	3	Touch key auto scan TK1 upper boundary MSB[9:8]
ATK0UBH	16.1~0	R/W	3	Touch key auto scan TK0 upper boundary MSB[9:8]
(F17) ATKLBH (TM57M5526C only)				**Function related to: Touch Key**
ATK1LBH	17.3~2	R/W	0	Touch key auto scan TK1 lower boundary MSB[9:8]
ATK0LBH	17.1~0	R/W	0	Touch key auto scan TK0 lower boundary MSB[9:8]
(F18) ATK0UBL (TM57M5526C only)				**Function related to: Touch Key**
ATK0UBL	18.7~0	R/W	FF	Touch key auto scan TK0 upper boundary LSB[7:0]
(F19) ATK0LBL (TM57M5526C only)				**Function related to: Touch Key**
ATK0LBL	19.7~0	R/W	0	Touch key auto scan TK0 lower boundary LSB[7:0]
(F1A) ATK1UBL (TM57M5526C only)				**Function related to: Touch Key**
ATK1UBL	1A.7~0	R/W	FF	Touch key auto scan TK1 upper boundary LSB[7:0]
(F1B) ATK1LBL (TM57M5526C only)				**Function related to: Touch Key**
ATK1LBL	1B.7~0	R/W	0	Touch key auto scan TK1 lower boundary LSB[7:0]
(F1C) RSR				**Function related to: RAM W/R**
RSR	1C.7~0	R/W	-	R-Plane file select register, indirect address mode pointer
(F1D) DPL				**Function related to: Table Read**
DPL	1D.7~0	R/W	0	Table read low address, data ROM pointer (DPTR) low byte[7~0]
(F1E) DPH				**Function related to: Table Read**
DPH	1E.1~0	R/W	0	Table read high address, data ROM pointer (DPTR) high byte[9~8]

Name	Address	R/W	Rst	Description
(F1F) IRCF				Function related to : Internal RC
IRCF	1F.4~0	R/W	by SYS.	FIRC frequency adjustment 00H: central frequency . . 0FH: highest frequency 10H: lowest frequency . . 1FH: central frequency
User Data Memory				
FRAM	20~2F	R/W	-	FRAM common area (8 bytes)
	30~7F	R/W	-	FRAM Bank0 area (RAMBK=0, 88 bytes)
	30~7F	R/W	-	FRAM Bank1 area (RAMBK=1, 88 bytes)

R-Plane

Name	Address	R/W	Rst	Description
(R02) TM0CTL				**Function related to: Timer0**
TM0EDG	02.5	W	0	TM0CKI (PA2) edge selection for Timer0 prescaler count 0: TM0CKI (PA2) rising edge for Timer0 Prescaler count 1: TM0CKI (PA2) falling edge for Timer0 Prescaler count
TM0CKS	02.4	W	0	Timer0 clock source select 0: Instruction Cycle as Timer0 Prescaler clock 1: TM0CKI as Timer0 Prescaler clock
TM0PSC	02.3~0	W	0	Timer0 prescaler. Timer0 clock source 0000: divided by 1 0100: divided by 16 1xxx: divided by 256 0001: divided by 2 0101: divided by 32 0010: divided by 4 0110: divided by 64 0011: divided by 8 0111: divided by 128
(R03) PWRDN				**Function related to: Power Down**
PWRDN	03	W	-	Write this register to enter Power Down Mode
(R04) WDTCLR				**Function related to: WDT**
WDTCLR	04	W	-	Write this register to clear WDT/WKT
(R05) PAMODH				**Function related to: Port A**
PA7MOD	05.6	W	0	PA7 I/O mode control 0: Mode0 1: Mode1
PA4MOD	05.1~0	W	1	PA4 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, used as PWM0 output with high sink
(R06) PAMODL				**Function related to: Port A**
PA3MOD	06.7~6	W	1	PA3 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, used as PWM0 output with high sink
PA2MOD	06.5~4	W	1	PA2 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, used as pin change wake up
PA1MOD	06.3~2	W	1	PA1 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, used as TKY CLD (TM57M5526C only)
PA0MOD	06.1~0	W	1	PA0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, used as pin change wake up

Name	Address	R/W	Rst	Description
(R0B) MR0B				**Function related to: STATUS/INT0/WKT/WDT**
HWAUTO	0B.7	W	0	Auto save/restore STATUS w/o TO, PD 0: Disable 1: Enable
INT0EDG	0B.6	W	0	INT0 pin (PA0) edge interrupt event 0: INT0 (PA0) pin falling edge to trigger interrupt event 1: INT0 (PA0) pin rising edge to trigger interrupt event
WDTPSC	0B.3~2	W	3	WDT pre-scale option select 00: WDT period is 52 ms, @5V; 75 ms, @3V 01: WDT period is 104 ms, @5V; 150 ms, @3V 10: WDT period is 208 ms, @5V; 300 ms, @3V 11: WDT period is 416 ms, @5V; 600 ms, @3V
WKTPSC	0B.1~0	W	3	WKT pre-scale option select 00: WKT period is 26 ms, @5V; 37.5 ms, @3V 01: WKT period is 52ms, @5V; 75 ms, @3V 10: WKT period is 104 ms, @5V; 150 ms, @3V 11: WKT period is 208 ms, @5V; 300 ms, @3V
(R0D) PWM0PRD				**Function related to: PWM0**
PWM0PRD	0D.7~0	W	FF	PWM0 period data
(R0E) TM0RLD				**Function related to: Timer0**
TM0RLD	0E.7~0	W	0	Timer0 reload data
(R10) TM1CTL				**Function related to: Timer1**
TM1PSC	10.3~0	W	0	Timer1 prescaler. Timer1 clock source 0000: divided by 1 (Fsys/2) 0001: divided by 2 (Fsys/4) 0010: divided by 4 (Fsys/8) 0011: divided by 8 (Fsys/16) 0100: divided by 16 (Fsys/32) 0101: divided by 32 (Fsys/64) 0110: divided by 64 (Fsys/128) 0111: divided by 128 (Fsys/256) 1xxx: divided by 256 (Fsys/512)
(R11) TM1RLD				**Function related to: Timer1**
TM1RLD	11.7~0	W	0	Timer1 reload data

INSTRUCTION SET

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

For byte-oriented instructions, "f" or "r" represents the address designator and "d" represents the destination designator. The address designator is used to specify which address in Program memory is to be used by the instruction. The destination designator specifies where the result of the operation is to be placed. If "d" is "0", the result is placed in the W register. If "d" is "1", the result is placed in the address specified in the instruction.

For bit-oriented instructions, "b" represents a bit field designator, which selects the number of the bit affected by the operation, while "f" represents the address designator. For literal operations, "k" represents the literal or constant value.

Field/Legend	Description
f	F-Plane Register File Address
r	R-Plane Register File Address
b	Bit address
k	Literal. Constant data or label
d	Destination selection field. 0: Working register 1: Register file
TO	WDT Time Out Flag
PD	Power Down Flag
W	Working Register
Z	Zero Flag
C	Carry Flag
DC	Decimal Carry Flag
PC	Program Counter
TOS	Top Of Stack
GIE	Global Interrupt Enable Flag (i-Flag)
[]	Option Field
( )	Contents
.	Bit Field
B	Before
A	After
←	Assign direction

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

ADDLW Add Literal "k" and W

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

ADDWF Add W and "f"

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

ANDLW Logical AND Literal "k" with W

Syntax	ANDLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) AND k	
Status Affected	Z	
OP-Code	01 1011 kkkk kkkk	
Description	The contents of W register are AND’ed with the eight-bit literal 'k'. The result is placed in the W register.	
Cycle	1	
Example	ANDLW 0x5F	B : W = 0xA3 A : W = 0x03

ANDWF AND W with "f"

Syntax	ANDWF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (W) AND (f)	
Status Affected	Z	
OP-Code	00 0101 dfff ffff	
Description	AND the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	ANDWF FSR, 1	B : W = 0x17, FSR = 0xC2 A : W = 0x17, FSR = 0x02

BCF	**Clear "b" bit of "f"**	
Syntax	BCF f [,b]	
Operands	f : 00h ~ 3Fh, b : 0 ~ 7	
Operation	(f.b) ← 0	
Status Affected	-	
OP-Code	01 000b bbff ffff	
Description	Bit 'b' in register 'f' is cleared.	
Cycle	1	
Example	BCF FLAG_REG, 7	B : FLAG_REG = 0xC7
		A : FLAG_REG = 0x47

BSF	**Set "b" bit of "f"**	
Syntax	BSF f [,b]	
Operands	f : 00h ~ 3Fh, b : 0 ~ 7	
Operation	(f.b) ← 1	
Status Affected	-	
OP-Code	01 001b bbff ffff	
Description	Bit 'b' in register 'f' is set.	
Cycle	1	
Example	BSF FLAG_REG, 7	B : FLAG_REG = 0x0A
		A : FLAG_REG = 0x8A

BTFSC	**Test "b" bit of "f", skip if clear(0)**	
Syntax	BTFSC f [,b]	
Operands	f : 00h ~ 3Fh, b : 0 ~ 7	
Operation	Skip next instruction if (f.b) = 0	
Status Affected	-	
OP-Code	01 010b bbff ffff	
Description	If bit 'b' in register 'f' is 1, then the next instruction is executed. If bit 'b' in register 'f' is 0, then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.	
Cycle	1 or 2	
Example	LABEL1 BTFSC FLAG, 1	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 ~ 3Fh, b : 0 ~ 7	
Operation	Skip next instruction if (f.b) = 1	
Status Affected	-	
OP-Code	01 011b bbff ffff	
Description	If bit 'b' in register 'f' is 0, then the next instruction is executed. If bit 'b' in register 'f' is 1, then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.	
Cycle	1 or 2	
Example	LABEL1 BTFSS FLAG, 1	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 ~ FFFh	
Operation	Operation: TOS ← (PC) + 1, PC.11~0 ← k	
Status Affected	-	
OP-Code	10 kkkk kkkk kkkk	
Description	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The 12-bit immediate address is loaded into PC bits <11:0>. CALL is a two-cycle instruction.	
Cycle	2	
Example	LABEL1 CALL SUB1	B : PC = LABEL1 A : PC = SUB1, TOS = LABEL1 + 1

CLRF Clear "f"

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

CLRW Clear W

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

CLRWDT Clear Watchdog Timer

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

COMF Complement "f"

Syntax	COMF f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← ($\bar{f}$)	
Status Affected	Z	
OP-Code	00 1001 dfff ffff	
Description	The contents of register 'f' are complemented. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f'.	
Cycle	1	
Example	COMF REG1, 0	B : REG1 = 0x13 A : REG1 = 0x13, W = 0xEC

DECF Decrement "f"

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

DECFSZ Decrement "f", Skip if 0

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

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

GOTO Unconditional Branch

Syntax	GOTO k	
Operands	k : 000h ~ FFFh	
Operation	PC.11~0 ← k	
Status Affected	-	
OP-Code	11 kkkk kkkk kkkk	
Description	GOTO is an unconditional branch. The 12-bit immediate value is loaded into PC bits <11:0>. GOTO is a two-cycle instruction.	
Cycle	2	
Example	LABEL1 GOTO SUB1	B : PC = LABEL1 A : PC = SUB1

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

INCFSZ	**Increment "f", Skip if 0**	
Syntax	INCFSZ f [,d]	
Operands	f : 00h ~ 7Fh, d : 0, 1	
Operation	(destination) ← (f) + 1, skip next instruction if result is 0	
Status Affected	-	
OP-Code	00 1111 dfff ffff	
Description	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead, making it a 2 cycle instruction.	
Cycle	1 or 2	

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

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

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

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

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

MOVRW	**Move "r" to W**	
Syntax	MOVRW r	
Operands	r : 00h ~ 3Fh	
Operation	(W) ← (r)	
Status Affected	-	
OP-Code	01 1111 00rr rrrr	
Description	The contents of register 'r' are moved to W register.	
Cycle	1	
Example	MOVRW MR0B	B : MR0B = 0x0F, W = ? A : MR0B = 0x0F, W = 0x0F

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

MOVWR Move W to "r"

Syntax	MOVWR r	
Operands	r : 00h ~ 3Fh	
Operation	(r) ← (W)	
Status Affected	-	
OP-Code	01 1110 00rr rrrr	
Description	Move data from W register to register 'r'.	
Cycle	1	
Example	MOVWR REG1	B : REG1 = 0xFF, W = 0x4F A : REG1 = 0x4F, W = 0x4F

NOP No Operation

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

RET Return from Subroutine

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

RETI Return from Interrupt

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

RETLW	Return with Literal in W
Syntax	RETLW k
Operands	k : 00h ~ FFh
Operation	PC ← TOS, (W) ← k
Status Affected	-
OP-Code	01 1000 kkkk kkkk
Description	The W register is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Cycle	2

Example

```
CALL  TABLE                         B : W = 0x07
          :                         A : W = value of k8
TABLE  ADDWF  PCL, 1
       RETLW  k1
       RETLW  k2
          :
       RETLW  kn
```

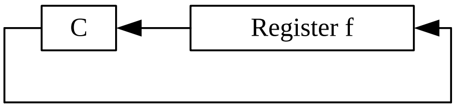
TABRL	Return DPTR low byte to W
Syntax	TABRL
Operands	-
Operation	(W) ← ROM[DPTR] low byte content, Where DPTR={DPH[max:8],DPL[7:0]}
Status Affected	-
OP-Code	00 0000 0101 0000
Description	The W register is loaded with low byte of ROM[DPTR]. This is a two-cycle instruction.
Cycle	2

Example

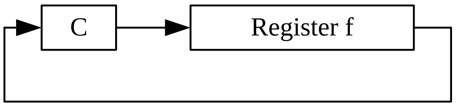
```
:
:
MOVLW    (TAB1&0xFF)
MOVWF    DPL                  ; Where DPL is F-plane register
MOVLW    (TAB1>>8)&0xFF
MOVWF    DPH                  ; Where DPH is F-plane register

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


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

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

RLF Rotate Left "f" through Carry

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

RRF Rotate Right "f" through Carry

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

SLEEP Go into standby mode, Clock oscillation stops

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

SUBWF Subtract W from "f"

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

SWAPF Swap Nibbles in "f"

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

TESTZ Test if "f" is zero

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

XORLW Exclusive OR Literal with W

Syntax	XORLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) XOR k	
Status Affected	Z	
OP-Code	01 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℃)

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

2. DC Characteristics (T_A =25℃, V_{DD} =1.3V to 5.5V)

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Operating Voltage	V_{DD}	FAST mode, 25°C, Fsys =4 MHz		1.8	–	5.5	V
		FAST mode, 25°C, Fsys =2 MHz		1.5	–	5.5	
		FAST mode, 25°C, Fsys =500 KHz		1.3	–	5.5	
		SLOW mode, 25°C, SIRC		1.3	–	5.5	
Input High Voltage	V_{IH}	All Input	V_{DD} =5V	0.6V_{DD}	–	–	V
			V_{DD} =3V	0.6V_{DD}	–	–	
Input Low Voltage	V_{IL}	All Input	V_{DD} =5V	–	–	0.2V_{DD}	V
			V_{DD} =3V	–	–	0.2V_{DD}	
I/O Port Source Current	I_{OH}	PA0 ~ PA2	V_{DD} =5V, V_{OH} =0.9V_{DD}	6	12	–	mA
			V_{DD} =3V, V_{OH} =0.9V_{DD}	3	6	–	
		PA3, PA4	V_{DD} =5V, V_{OH} =0.9V_{DD}	10	20	–	
			V_{DD} =3V, V_{OH} =0.9V_{DD}	4	8	–	
I/O Port Sink Current	I_{OL}	PA0 ~ PA2	V_{DD} =5V, V_{OL} =0.1V_{DD}	24	48	–	mA
			V_{DD} =3V, V_{OL} =0.1V_{DD}	10	20	–	
		PA3, PA4	V_{DD} =5V, V_{OL} =0.1V_{DD}	38	76	–	
			V_{DD} =3V, V_{OL} =0.1V_{DD}	16	32	–	
		PA7	V_{DD} =5V, V_{OL} =0.1V_{DD}	14	28	–	
			V_{DD} =3V, V_{OL} =0.1V_{DD}	6	12	–	
Input Leakage Current (pin high)	I_{ILH}	All Input	V_{IN} =V_{DD}	–	–	1	μA
Input Leakage Current (pin low)	I_{ILL}	All Input	V_{IN} =0V	–	–	-1	

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Supply Current	I_{DD}	FAST mode, LVR enable	V_{DD} =5V, FIRC=4MHz CPUPSC=10 (div2)	–	1.7	–	mA
			V_{DD} =3V, FIRC=4MHz CPUPSC=10 (div2)	–	1.1	–	
			V_{DD} =5V, FIRC=2MHz CPUPSC=01 (div4)	–	1.1	–	
			V_{DD} =3V, FIRC=2MHz CPUPSC=01 (div4)	–	0.7	–	
			V_{DD} =5V, FIRC=0.5MHz CPUPSC=00 (div16)	–	0.6	–	
			V_{DD} =3V, FIRC=0.5MHz CPUPSC=00 (div16)	–	0.4	–	
		SLOW mode, LVR enable	V_{DD} =5 V, SIRC, CPUPSC=11 (div1)	–	33	–	µA
			V_{DD} =3 V, SIRC, CPUPSC=11 (div1)	–	14	–	
			V_{DD} =5 V, SIRC, CPUPSC=10 (div2)	–	20	–	
			V_{DD} =3 V, SIRC, CPUPSC=10 (div2)	–	8	–	
			V_{DD} =5 V, SIRC, CPUPSC=01 (div4)	–	14	–	
			V_{DD} =3 V, SIRC, CPUPSC=01 (div4)	–	5	–	
			V_{DD} =5 V, SIRC, CPUPSC=00 (div16)	–	9	–	
			V_{DD} =3 V, SIRC, CPUPSC=00 (div16)	–	3	–	
		IDLE mode, LVR enable	V_{DD} =5 V, SIRC, CPUPSC=11	–	7.6	–	
			V_{DD} =3 V, SIRC, CPUPSC=11	–	2.3	–	
		IDLE mode, LVR disable	V_{DD} =5 V, SIRC, CPUPSC=11	–	5.4	–	
			V_{DD} =3 V, SIRC, CPUPSC=11	–	1.5	–	
		STOP mode, LVR enable	V_{DD} =5V	–	2.3	–	
			V_{DD} =3V	–	0.7	–	
		STOP mode, LVR disable	V_{DD} =5V	–	–	0.1	
			V_{DD} =3V	–	–	0.1	
System Clock Frequency	Fsys	V_{DD} > LVR_{th}	V_{DD} =3.7V	–	–	4	MHz
			V_{DD} =2.4V	–	–	4	
			V_{DD} =1.7V	–	–	2	
LVR Reference Voltage	V_{LVR}	T_A =25°C		–	3.7	–	V
				–	2.4	–	
				–	1.7	–	
LVR Hysteresis Voltage	V_{HYST}	T_A =25°C		–	±0.1	–	V

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Low Voltage Detection time	t_{LVR}	T_A =25°C		100	–	–	µs
Pull-Up Resistor	R_P	V_{IN} = 0 V Port A, except PA7	V_{DD} =5V	–	58	–	KΩ
			V_{DD} =3V		104		
		V_{IN} = 0 V PA7	V_{DD} =5V	–	56	–	
			V_{DD} =3V		101		

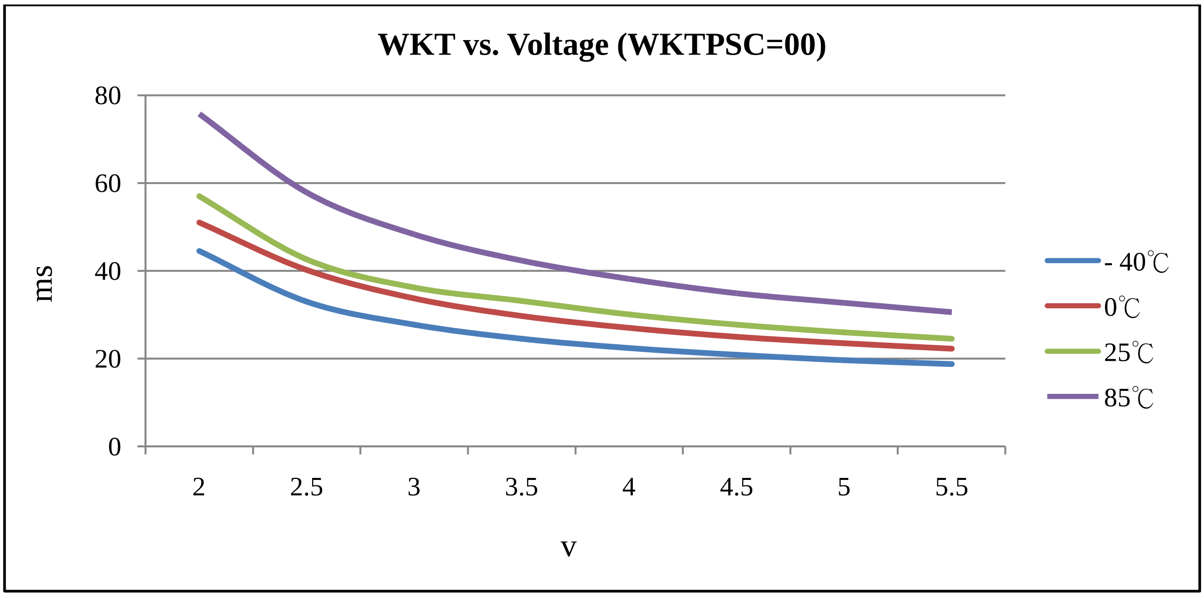
3. Clock Timing (T_A = -40℃ to +85℃)

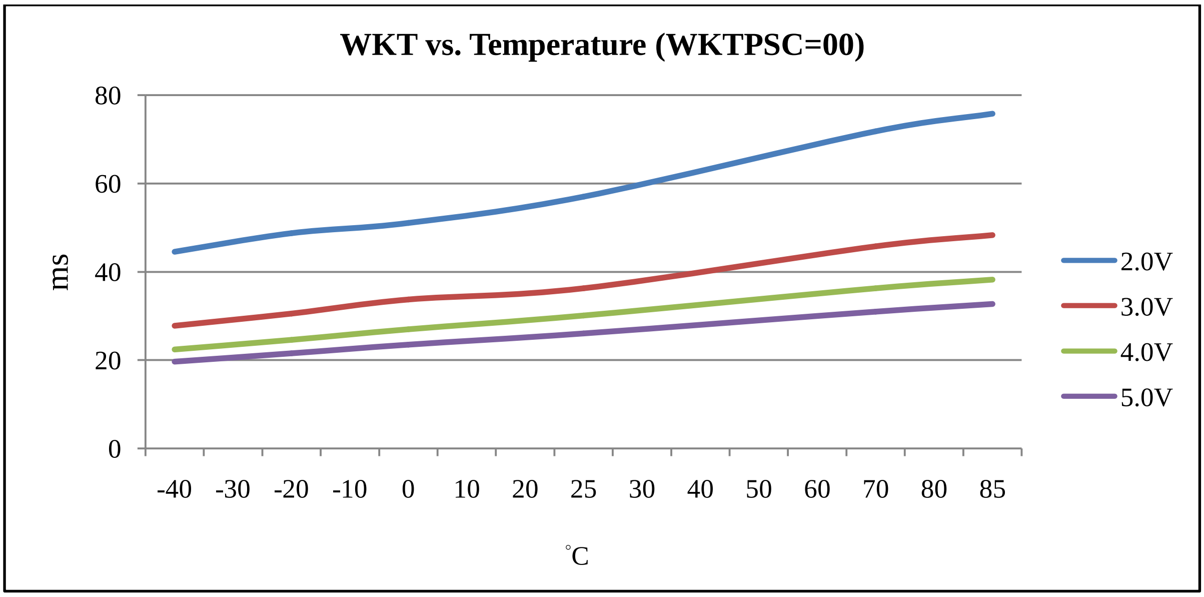
Parameter	Condition	Min	Typ	Max	Unit
Internal RC Frequency	25°C, V_{DD} =3.1 ~ 5.5V	7.75	8	8.25	MHz
	25°C, V_{DD} =2.0 ~ 3.0V	7.6	8	8.4	
	-40°C ~ 85°C, V_{DD} =2.5 ~ 5.5V	7.5	8	8.5	

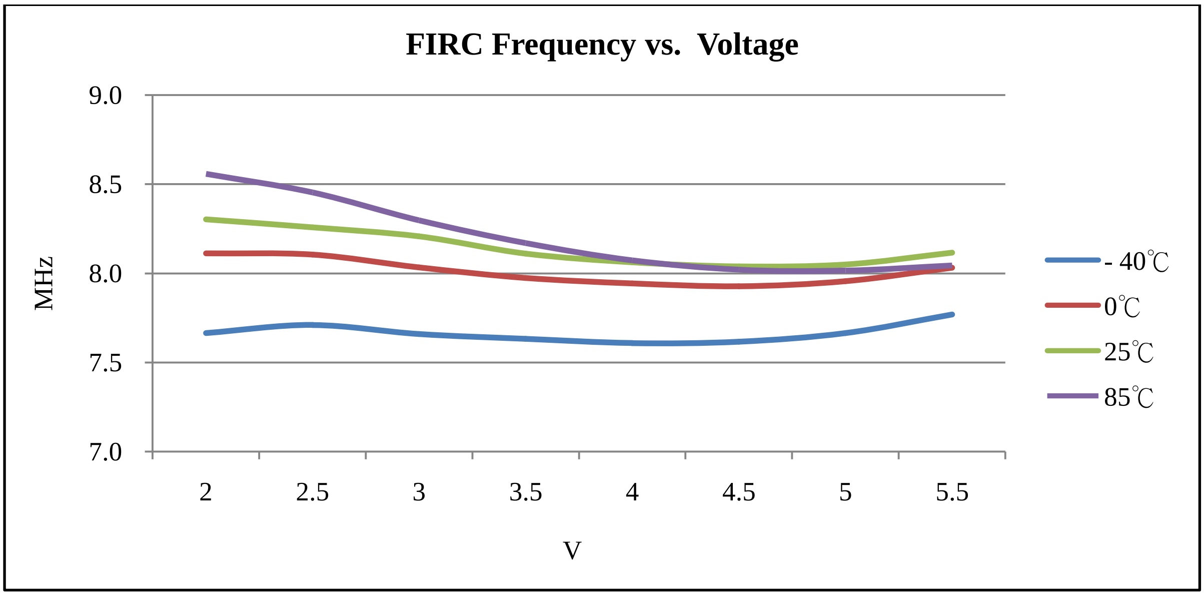
4. Reset Timing Characteristics (T_A = -40℃ to +85℃)

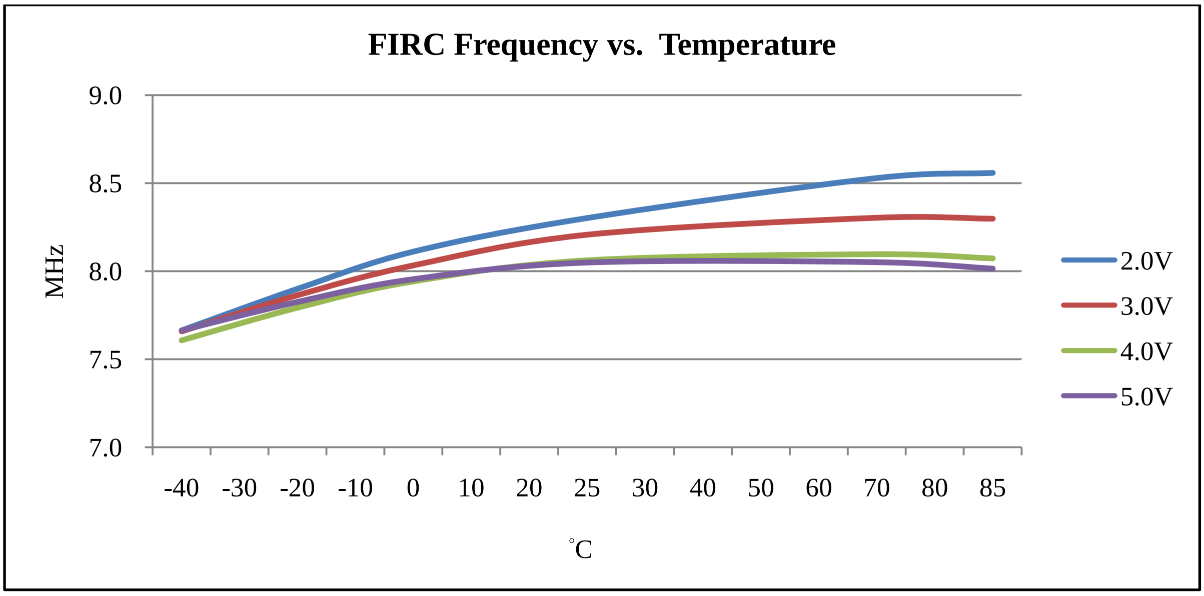
Parameter	Conditions	Min	Typ	Max	Unit
RESET Input Low width	Input V_{DD} =5 V ±10 %	3	–	–	µs
WDT wakeup time	V_{DD} =5V, WDTPSC=00	–	52	–	ms
	V_{DD} =3V, WDTPSC=00	–	75	–	
WKT wakeup time	V_{DD} =5V, WKTPSC=00	–	26	–	ms
	V_{DD} =3V, WKTPSC=00	–	37.5	–	
CPU start up time	V_{DD} =5V	–	24	–	ms
	V_{DD} =3V	–	35	–	

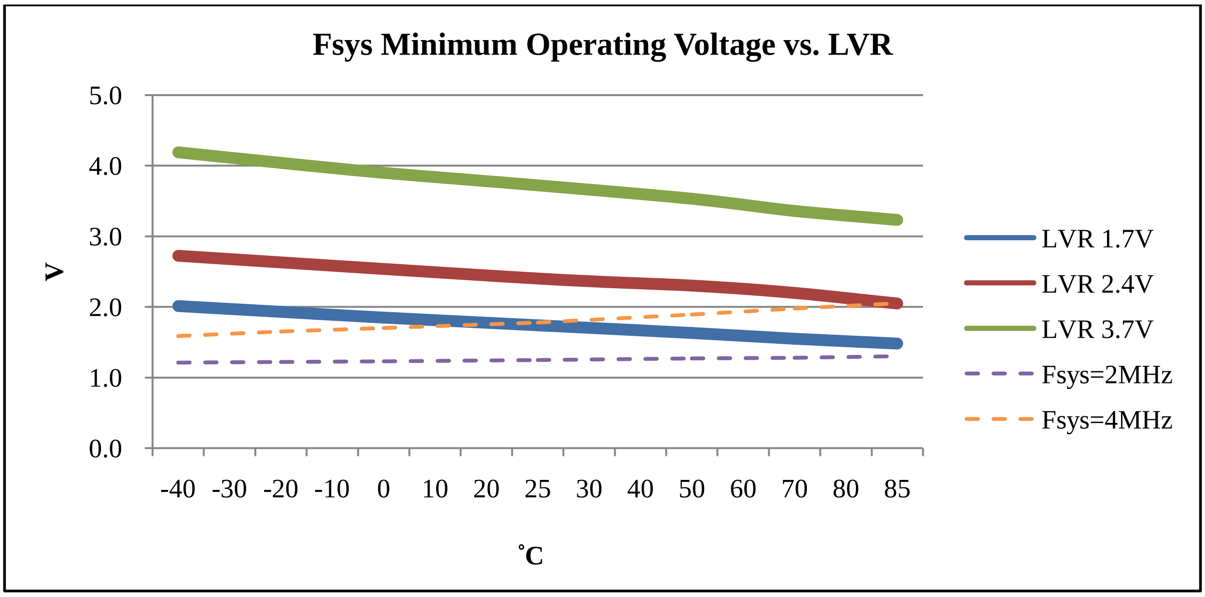
5. Characteristic Graphs






FIRC Frequency vs. Voltage
9.0
8.5
8.0
7.5
7.0
MHz
2
2.5
3
3.5
4
4.5
5
5.5
V
- 40℃
0℃
25℃
85℃


FIRC Frequency vs. Temperature
9.0
8.5
8.0
7.5
7.0
MHz
-40
-30
-20
-10
0
10
20
25
30
40
50
60
70
80
85
˚C
2.0V
3.0V
4.0V
5.0V



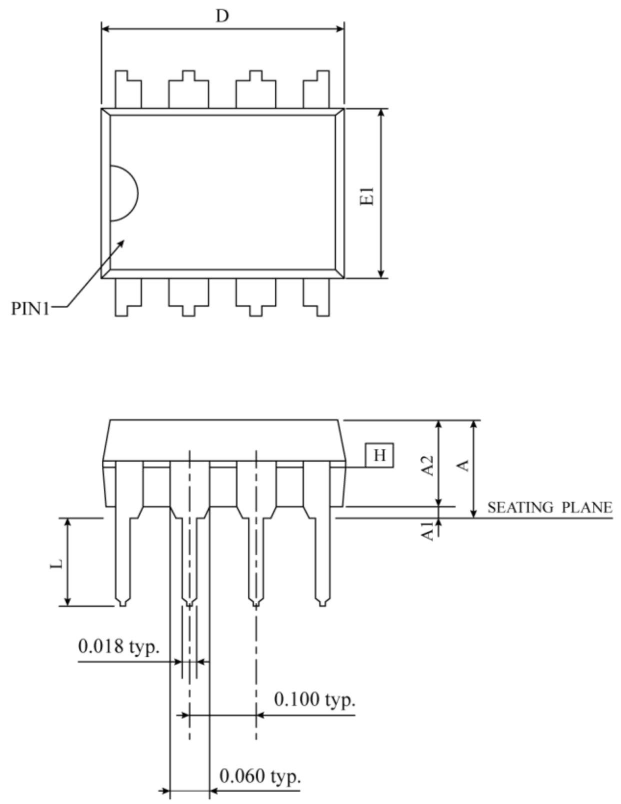
PS. Due to the variation of manufacturing process, this LVR will slightly vary between different chips.

ORDERING INFORMATION

The ordering information:

Ordering number	Package
TM57M5526C-MTP	Wafer/Dice blank chip
TM57M5526C-COD	Wafer/Dice with code
TM57M5526C-MTP-01	DIP-8 (300mil)
TM57M5526C-MTP-14	SOP-8 (150mil)
TM57M5526C-MTP-A8	SOT23-6
TM57M5536C- MTP	Wafer/Dice blank chip
TM57M5536C-COD	Wafer/Dice with code
TM57M5536C-MTP-01	DIP-8 (300mil)
TM57M5536C-MTP-14	SOP-8 (150mil)
TM57M5536C-MTP-A8	SOT23-6

DIP-8 (300mil) Package Dimension

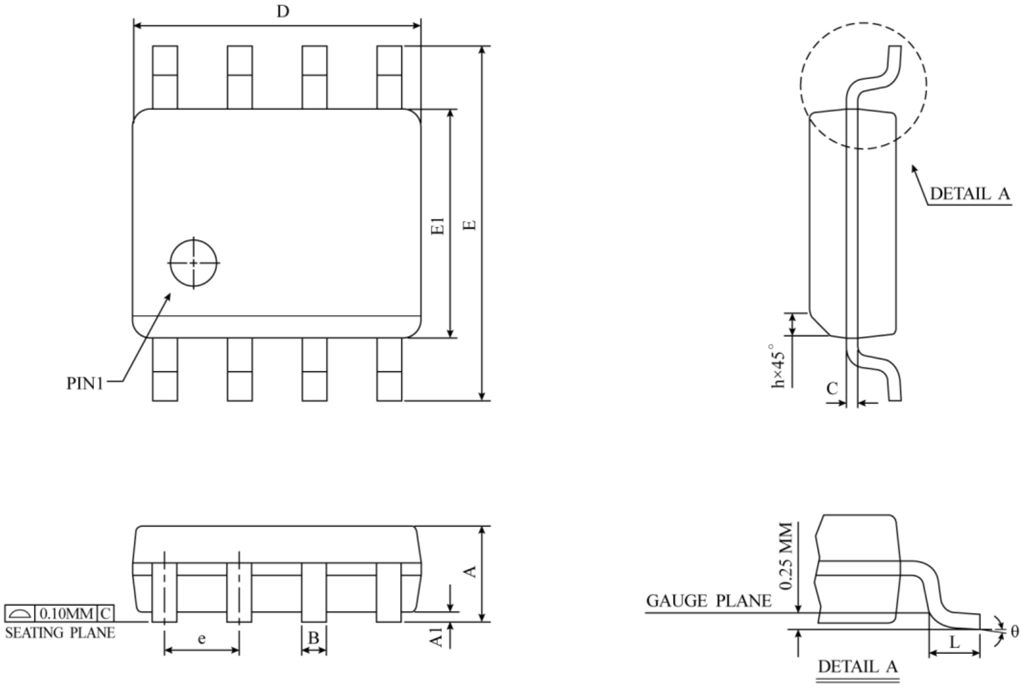


SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	5.334	-	-	0.210
A1	0.381	-	-	0.015	-	-
A2	3.175	3.302	3.429	0.125	0.130	0.135
D	9.017	9.586	10.160	0.355	0.378	0.400
E	7.620 BSC			0.300 BSC		
E1	6.223	6.350	6.477	0.245	0.250	0.255
L	2.921	3.366	3.810	0.115	0.133	0.150
eB	8.509	9.017	9.525	0.335	0.355	0.375
θ	0°	7.5°	15°	0°	7.5°	15°
JEDEC	MS-001 (BA)					

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.

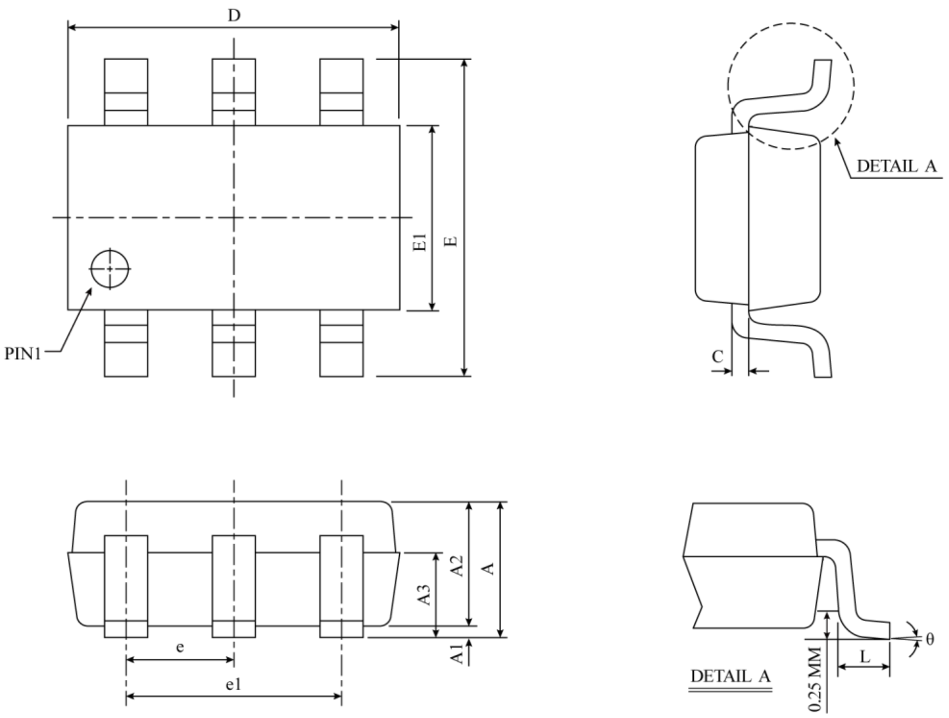
SOP-8 (150mil) Package Dimension



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	4.80	4.90	5.00	0.1890	0.1939	0.1988
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 (AA)					

△ * 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.

SOT23-6 Package Dimension



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	1.45	-	-	0.057
A1	0	0.08	0.15	0	0.003	0.006
A2	0.90	1.10	1.30	0.035	0.043	0.051
A3	0.60	0.65	0.70	0.024	0.026	0.028
c	0.12	0.16	0.19	0.005	0.006	0.007
D	2.82	2.92	3.02	0.111	0.115	0.119
E	2.70	2.90	3.10	0.106	0.114	0.122
E1	1.52	1.62	1.72	0.060	0.064	0.068
e	0.85	0.95	1.05	0.033	0.037	0.041
e1	1.80	1.90	2.00	0.071	0.075	0.079
L	0.35	0.48	0.60	0.014	0.019	0.024
θ	0°	4°	8°	0°	4°	8°
JEDEC	M0-178 (AB)					

△A * NOTES：ALL DIMENSIONS REFER TO JEDEC STANDARD MO-178 AB
DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.