十速

TM57M5541

DATA SHEET

Rev 0.93

AMENDMENT HISTORY

Version	Date	Description
0.90	Apr, 2016	New release.
0.91	Jan, 2017	1. P19, 20, 33, 34,49, 51, 53, 55, 57, 76, 79, 82, 62, 75, 76, 82, 86: Correcting errors. 2. P106, 107: Add Characteristic Graphs. 3. P7: Add HS function description
0.92	Mar, 2017	1. P21, 29, 30: LVR2.0V modified to LVR1.8V 2. P106: Update LVR characteristics graph 3. P67,P69, P80: Modify the description of RCVBF/RCVOVF
0.93	Sep, 2017	1. P15: SYSCFG LVR 2.0V modified to1.8V 2. P7, 10: ISP modified to ICP

CONTENTS

FEATURES

1. **ROM: 4K x 14 bits MTP (Multi Time Programmable ROM) with Page Locker function**
2. **RAM: 368 x 8 bits**
3. **STACK: 8 Levels**
4. **System Oscillation Sources (Fsys) :**
   - Fast-clock
     - FIRC (Fast Internal RC) : 16 MHz
     - FXT (Fast Crystal) : 1~16 MHz
   - Slow-clock
     - SIRC (Slow Internal RC) : 128 KHz @VCC=3V
     - SXT (Slow Crystal) : 32768 Hz
5. **System Clock Prescaler:**
   - System Oscillation Sources can be divided by 1/2/4/16 as System Clock (Fsys)
6. **Dual System Clock:**
   - FIRC+SIRC or FXT+SIRC or FIRC+SXT
7. **Power Saving Operation Mode**
   - FAST Mode: Slow-clock can be disabled or enabled, Fast-clock keeps CPU running
   - SLOW Mode: Fast-clock can be disabled or enabled, Slow-clock keeps CPU running
   - IDLE Mode: Fast-clock and CPU stop. Slow-clock, Timer3, or Wake-up Timer keep running
   - STOP Mode: All clocks stop, T2 and Wake-up Timer stop
8. **3 Independent Timers**
   - Timer0
     - 8-bit timer divided by 1~256 pre-scaler option, Reload/Interrupt/Stop function
   - Timer1
     - 8-bit timer divided by 1~256 pre-scaler option, Reload/Interrupt/Stop function
     - Overflow and Toggle out
   - Timer3
     - 16-bit timer with 1~256 pre-scaler options, Reload/Interrupt/Stop function
     - IDLE mode wake-up timer
     - Clock sources: Slow-clock, Fsys

9. Interrupt

- Five External Interrupt pins
  - 1 pin are falling edge wake-up triggered & interrupts
  - 4 pins is rising or falling edge wake-up triggered & interrupt
- Timer0/Timer1/Timer3/WKT (wake-up) Interrupts
- TK (Touch Key) /ADC/LVD Interrupt
- SPI/I2C Interrupt
- Individual Interrupt Vector

10. Wake-up (WKT) Timer

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

  17 ms/34 ms/68 ms/136 ms @VCC=3V, 16 ms/32 ms/64 ms/128 ms @VCC=5V

11. Watchdog Timer

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

  140 ms/280 ms/1120 ms/2240 ms @VCC=3V, 128 ms/256 ms/1024 ms/2048 ms @VCC=5V
- Watchdog timer can be disabled/enabled in STOP mode

12. PWMx4

- PWM0:
  - 8+2 bits, duty-adjustable, period-adjustable controlled PWM
  - PWM0 clock source: Fast-clock or FIRC 16 MHz, with 1~64 pre-scalers
- PWM1A/PWM1B/PWM1C:
  - 8 bits, duty-adjustable (Independent) , period-adjustable controlled (Shared) PWM x3
  - PWM1A/1B/1C clock source (Shared) : Fast-clock or FIRC 16 MHz, with 1~64 pre-scalers

13. 12-bit ADC Converter with 8 input channels and 1 internal reference voltage

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

14. Reset Sources

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

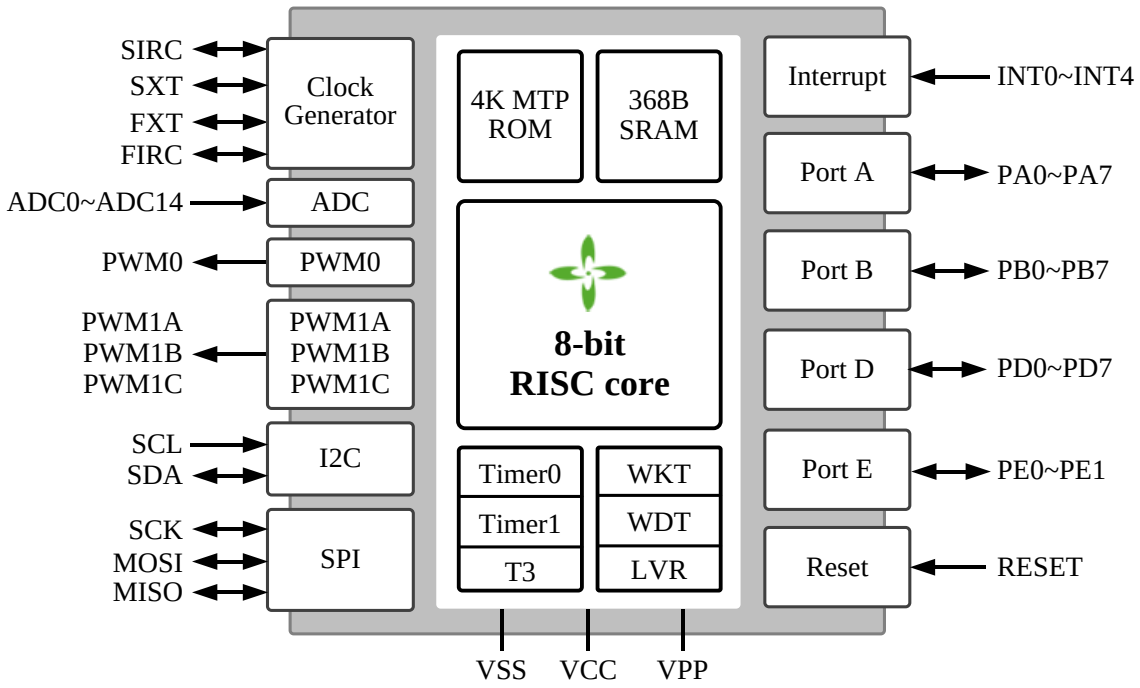
15. Low Voltage Reset (LVR) /Low Voltage Detection Flag (LVD) Option:

- 3-Level Low Voltage Reset: 2.0V/2.3V/2.9V
- 4-Level Low Voltage Detection Flag: 2.2V/2.5V/3.1V/4.5V

16. **SPI Interface:**
    - Master or Slave mode selectable
    - Programmable transmit bit rate
    - Serial clock phase and polarity options
    - MSB-first or LSB-first selectable
17. **I2C Interface:**
    - Specific purpose slave I2C interface with interrupt function
18. **16-Channel Touch Key**
    - 1~4 Key 10-bit H/W auto scan touch key (ATK0~3) with upper and lower boundaries for each key
    - Interrupt/Wake-up CPU while key is pressed
    - ATK scanning intervals (30/60/120/240 mS)
    - TKCLD capacitor can select Internal CLD or External CLD
19. **Operating Voltage:**
    - Fsys=1 MHz, LVR ~5.5V
    - Fsys=16 MHz, 3.0~5.5V
20. **Operating Temperature Range : -40°C to + 85°C**
21. **Table Read Instruction: 14-bit ROM data lookup table**
22. **Instruction set: 39 Instructions**
23. **Instruction Execution Time**
    - 2 system clocks (Fsys) per instruction except branch
24. **I/O ports: Maximum 26 programmable I/O pins**
    - Open-Drain Output
    - CMOS Push-Pull Output
    - Schmitt Trigger Input with pull-up resistor option
    - 7 High Sink Pins (50mA@5V)
25. **Programming connectivity support 5-wire (ICP) or 8-wire program**
26. **Page Locker Size: 512W/640W/768W …./2304W by 128 words step**
27. **Package Types:**
    - SOP-28/SOP-24/SOP-20/DIP-28/DIP-20
28. **Supported EV board on ICE**

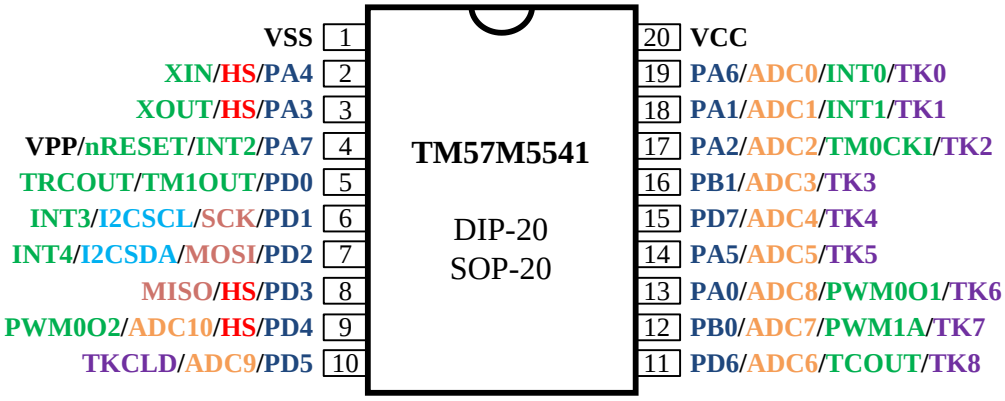
    EV board: EV8212

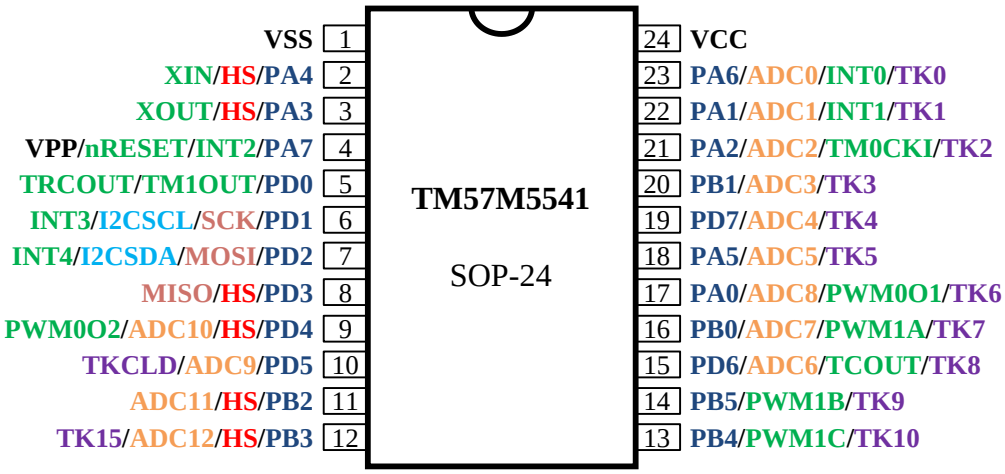
BLOCK DIAGRAM

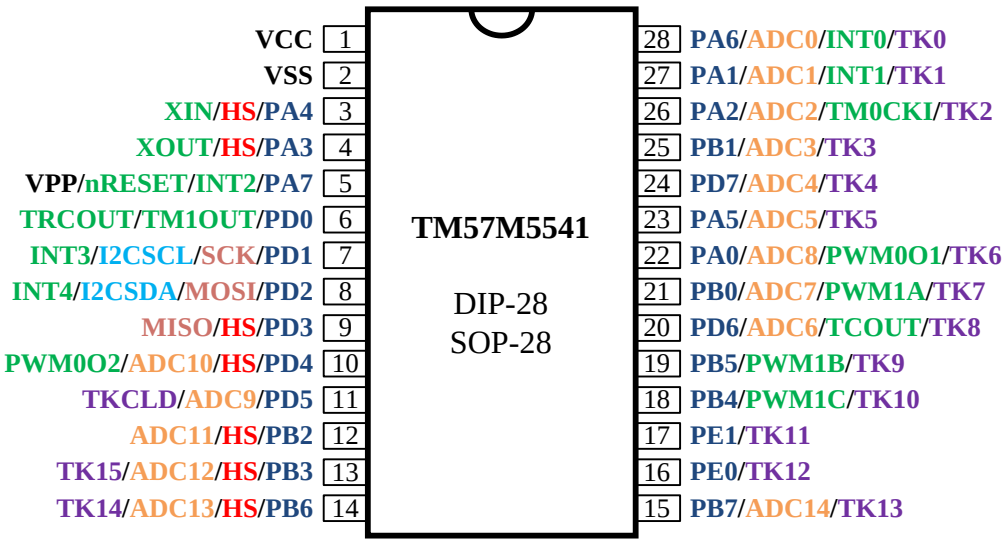


TM57M5541 Block Diagram

PIN ASSIGNMENT







PIN DESCRIPTIONS

Name	In/Out	Pin Description
PA0-PA6 PB0-PB7 PD0-PD7 PE0-PE1	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 Schmitt-trigger input with pull-high
nRESET	I	External active low reset, internal pull-high
VCC, VSS	P	Power Voltage input pin and ground
VPP	I	PROM programming high voltage input
INT0–INT4	I	External interrupt input
XIN, XOUT	-	Crystal / Resonator oscillator connection for system clock.
TM0CKI	I	Timer0's input in counter mode
TM1OUT	O	Timer1 match output, TM1OUT toggles when Timer1 overflow occurs.
TRCOUT	O	Touch Key clock output
PWM0O1	O	(8+2) bit PWM0 output 1
PWM0O2	O	(8+2) bit PWM0 output 2
PWM1A	O	8 bit PWM1A output
PWM1B	O	8 bit PWM1B output
PWM1C	O	8 bit PWM1C output
ADC14~ADC0	I	A/D channels input
IICSDA, IICSCL	-	Inter-Integrated Circuit Pin
MISO, MOSI, SCK	-	Serial Peripheral Interface Pin
TK0-TK13	I	Touch key input
TKCLD	I	Touch key capacitor input
HS	O	High sink Pin
TCOUT	O	Post-prescaler Instruction Cycle (Fsys/2) output

Programming pins:

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

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

PIN SUMMARY

28-SOP/DIP	24-SOP	20-SOP/DIP	Pin Name	Type	GPIO				Function After Reset	Alternate Function				
					Input		Output							
					Wake up	Ext. Interrupt	O.D	P.P		PWM	High Sink	TK	ADC	MISC
1	24	20	VCC	P										
2	1	1	VSS	P										
3	2	2	**PA4/XIN/HS**	I/O			○	○	PA4		○			XIN
4	3	3	**PA3/XOUT/HS**	I/O			○	○	PA3		○			XOUT
5	4	4	**VPP/nRESET/INT2/PA7**	I/O	○	○	○		PA7					nRESET
6	5	5	**TRCOUT/TM1OUT/PD0**	I/O			○	○	PD0					TM1OUT
7	6	6	**INT3/IICSCL/SCK/PD1**	I/O	○	○	○	○	PD1					I2C/SPI
8	7	7	**INT4/IICSDA/MOSI/PD2**	I/O	○	○	○	○	PD2					I2C/SPI
9	8	8	**MISO/PD3/HS**	I/O			○	○	PD3		○			SPI
10	9	9	**PWM0O2/ADC10/PD4/HS**	I/O			○	○	PD4	○	○		○	
11	10	10	**TKCLD/ADC9/PD5**	O			○	○	PD5			○	○	TKCLD
12	11		**ADC11/PB2/HS**	I/O			○	○	PB2		○		○	
13			**TK15/ADC12/PB3/HS**	I/O			○	○	PB3		○	○	○	
14	12		**TK14/ADC13/PB6/HS**	I/O			○	○	PB6		○	○	○	
15			**PB7/ADC14/TK13**	I/O			○	○	PB7			○	○	
16			**PE0/TK12**	I/O	○		○	○	PE0			○		
17			**PE1/TK11**	I/O	○		○	○	PE1			○		
18	13		**PB4/PWM1C/TK10**	I/O			○	○	PB4	○		○		
19	14		**PB5/PWM1B/TK9**	I/O			○	○	PB5	○		○		
20	15	11	**PD6/ADC6/TCOUT/TK8**	I/O			○	○	PD6			○	○	TCOUT
21	16	12	**PB0/ADC7/PWM1A/TK7**	I/O			○	○	PB0	○		○	○	
22	17	13	**PA0/ADC8/PWM0O1/TK6**	I/O			○	○	PA0	○		○	○	

28-SOP/DIP	24-SOP	20-SOP/DIP	Pin Name	Type	GPIO				Function After Reset	Alternate Function				
					Input		Output							
					Wake up	Ext. Interrupt	O.D	P.P		PWM	High Sink	TK	ADC	MISC
23	18	14	**PA5/ADC5/TK5**	I/O			○	○	PA5			○	○	
24	19	15	**PD7/ADC4/TK4**	I/O			○	○	PD7			○	○	
25	20	16	**PB1/ADC3/TK3**	I/O			○	○	PB1			○	○	
26	21	17	**PA2/ADC2/TM0CKI/TK2**	I/O			○	○	PA2			○	○	TM0CKI
27	22	18	**PA1/ADC1/INT1/TK1**	I/O	○	○	○	○	PA1			○	○	
28	23	19	**PA6/ADC0/INT0/TK0**	I/O	○	○	○	○	PA6			○	○	

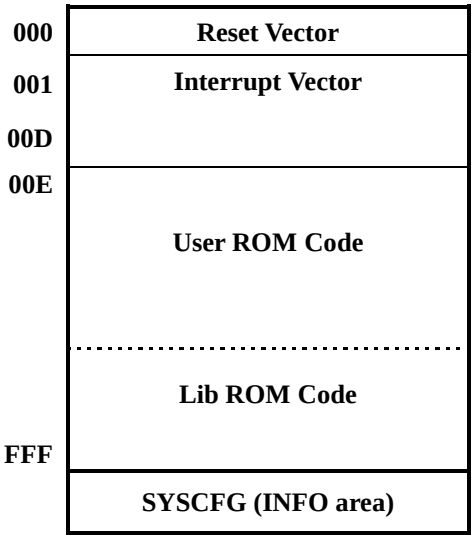
Symbol：P.P. = Push-Pull Output
O.D. = Open Drain
SYS = by SYSCFG bit
HS = High Sink

FUNCTIONAL DESCRIPTION

1. CPU Core

1.1 Program ROM (PROM)

The MTP Program ROM of this device is 4K words, with an extra INFO area to store the SYSCFG. The ROM can be written multi-times and can be read as long as the PROTECT and LPROT bits of SYSCFG are not set. The SYSCFG can be read no matter PROTECT or LPROT is set, but PROTECT bit can be cleared only when the User ROM Code area is erased, and LPROT bit can be cleared only when the Lib ROM Code area is erased. That is, unprotect the PROTECT or LPROT bit needs to erase the corresponding ROM area. If LPROT bit is set, The ROM can still be written multi-times in the User ROM Code area to update user ROM code again by writer, but the Lib ROM Code area will not be read or written again by writer until the LPROT bit is cleared. On the other hand, if PORTECT bit is set, the user ROM code area will not be read by writer, and the user ROM code can't be updated until the PORTECT bit is cleared.



1.1.1 Reset Vector (000H)

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

1.1.2 Interrupt Vector (001H~00DH)

When an interrupt occurs, the program counter (PC) will be pushed onto the stack and jumps to address 001H~00DH with corresponding interrupt.

Address	Source	Description
001	LVD	Low Voltage detection interrupt
002	Timer3	Timer3 Counter Overflow
003	TK	Touch Key interrupt
004	I2C/SPI	I2C interrupt or SPI interrupt
005	Timer0	Timer0 Counter Overflow
006	Timer1	Timer1 Counter Overflow
007	XINT4	PD2 rising/falling interrupt (rising/falling selectable)
008	XINT3	PD1 rising/falling interrupt (rising/falling selectable)
009	XINT2	PA7 falling interrupt
00a	XINT1	PA1 rising/falling interrupt (rising/falling selectable)
00b	XINT0	PA6 rising/falling interrupt (rising/falling selectable)
00c	ADC	ADC interrupt
00d	WKT	Wakeup Timer Match

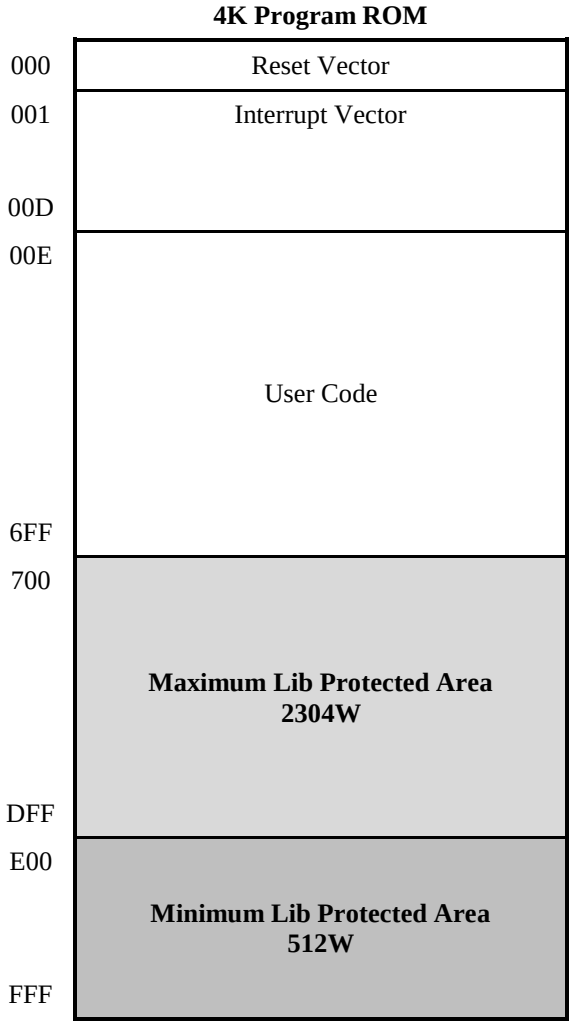
1.2 System Configuration Register (SYSCFG)

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

Bit		**13~0**	
Default Value		**11111111111111**	
Bit		Description	
CFGWL	13	**LPROT**: Lib Code protection selection	
		1	Enable
		0	Disable
	12~9	**LSIZE**: Lib Size selection	
		1111	2304W
		0001	512W
		0000	No use Page locker function
	8~0	Tenx Reserved	
CFGWH	13	**PROTECT**: Code protection selection	
		1	Enable
		0	Disable
	12	**XRSTE**: External Pin (PA7) Reset Enable	
		1	Enable
		0	Disable (PA7 as input I/O pin)
	11-10	**LVR**: Low Voltage Reset Mode	
		11	1.8V
		10	Disable
		01	2.3V
		00	2.9V
	9-8	**WDTE**: WDT Reset Enable	
		11	Always Enable
		10	Enable in FAST/SLOW mode, Disable in IDLE/STOP mode
		0X	Disable
	7-0	Tenx Reserved	

1.3 Page Locker Function

TM57M5541 support Page Locker function. By setting LPROT (CFGWL.13) , user can choose whether to turn it on. If the user A (library code provider) turns this function on, the user A (library code provider) can select different size (512~2304W) of lib protected area by LSIZE (CFGWL.12~9) . In lib protected area, the user B (firmware developer) can't read ROM code by TABRL/TABRH instruction or in any other way. By using the TICE99IDE tool, the user A can provide a protected lib code for the user B to use, but the user B does not know its details, and the user B still can continue to complete the main code in the unprotected area.



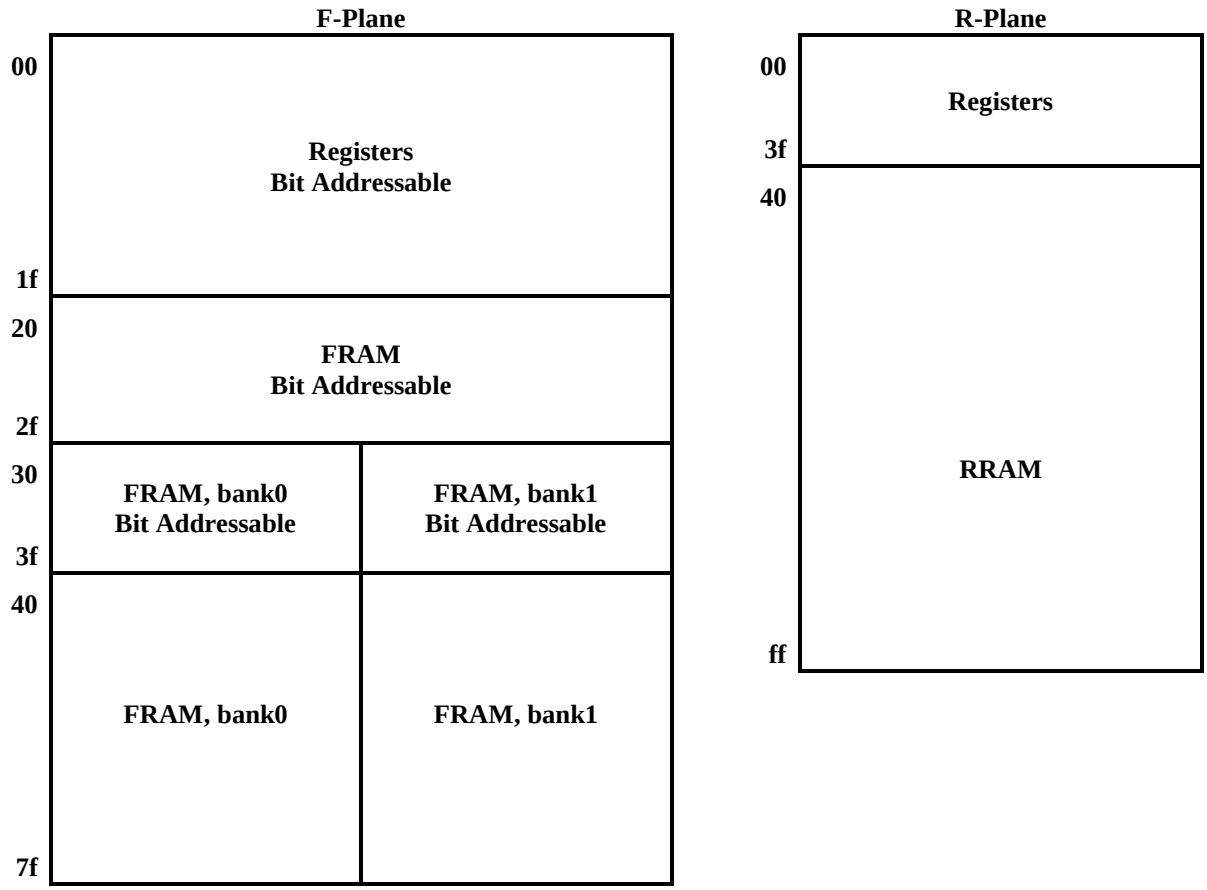
LSIZE	Lib Protected Area
2304	(700H~FFFH)
2176	(780H~FFFH)
2048	(800H~FFFH)
1920	(880H~FFFH)
1792	(900H~FFFH)
1664	(980H~FFFH)
1536	(A00H~FFFH)
1408	(A80H~FFFH)
1280	(B00H~FFFH)
1152	(B80H~FFFH)
1024	(C00H~FFFH)
896	(C80H~FFFH)
768	(D00H~FFFH)
640	(D80H~FFFH)
512	(E00H~FFFH)

1.4 RAM Addressing Mode

There are two Data Memory Planes in CPU, F-Plane and R-Plane.

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.

R-Plane can also be addressed directly or indirectly. Indirect Addressing is made by INDR register. The INDR register is not a physical register. Addressing INDR actually addresses the register whose address is contained in the RSR register (RSR is a pointer) . The R-Plane is not bit-addressable and only support two MOVWR, MOVRW byte operating instructions.



1.5 Programming Counter (PC) and Stack

The Programming Counter is 12-bit wide capable of addressing a 4Kx14 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~00dh) are provided for PC initialization and Interrupt. For CALL/GOTO instructions, PC loads 10 bits address from instruction word. For RET/RETI/RETLW instructions, PC retrieves its content from the top level STACK. For the other instructions updating PC [7:0] , the PC [11:8] keeps unchanged. The STACK is 12-bit wide and 8-level in depth. The CALL instruction and hardware interrupt will push STACK level in order. While the RET/RETI/RETLW instruction pops the STACK level in order.

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

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

```
            ORG     000H                ; Reset Vector
            GOTO    START

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

LOOP:
            MOVFW   INDEX               ; Move index value to W register.
            CALL    TABLE               ; To lookup data, W=55H.
            …..
            INCF    INDEX, 1            ; Increment the index address for next address
            …..
            GOTO    LOOP                ; Go to LOOP label.
            …..
            MOVLW   (TABLE2 >>8) & 0xff
            MOVWF   DPH                 ; DPH register (F0F.2~0)
            MOVLW   (TABLE2) & 0xff
            MOVWF   DPL                 ; DPL register (F13.7~0)
            TABRL                       ; W=86H
            TABRH                       ; W=19H
            …..

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

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

1.5.1 ALU and Working (W) Register

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

Note: /Borrow represents inverted of Borrow register.

/Digit Borrow represents inverted of Digit Borrow register.

1.5.2 STATUS Register (F-Plane 03H)

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

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

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

◇ Example: Write immediate data into STATUS register.

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

◇ Example: Bit addressing set and clear STATUS register.

```
BSF     STATUS, 0   ; Set C=1.
BSF     03H, 5      ; Selection RAM Bank1
BCF     STATUS, 0   ; Clear C=0.
BCF     03H, 5      ; Selection RAM Bank0
```

◇ Example: Determine the C flag by BTFSS instruction.

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

2. Reset

This device can be RESET in four ways.

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

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

2.1 Power on Reset

After Power-On-Reset, all system and peripheral control registers are then set to their default hardware Reset values. The clock source, LVR level and chip operation mode are selected by the SYSCFG register value.

2.2 Low Voltage Reset

The Low Voltage Reset features static reset when supply voltage is below a threshold level. There are three threshold levels can be selected. The LVR's operation mode is defined by the SYSCFG register. See the following LVR Selection Table; user must also consider the lowest operating voltage of operating frequency.

LVR Selection Table:

LVR level	Operating voltage
LVR1.8	5.5V >VCC >2.2V
LVR2.3	5.5V >VCC >2.4V
LVR2.9	5.5V >VCC >3.1V or V_{CC} =5.0V

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

2.3 External Pin Reset

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

2.4 Watchdog Timer Reset

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

◇ Example: Defining Reset Vector

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

        ORG     010H

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

3. Clock Circuitry and Operation Mode

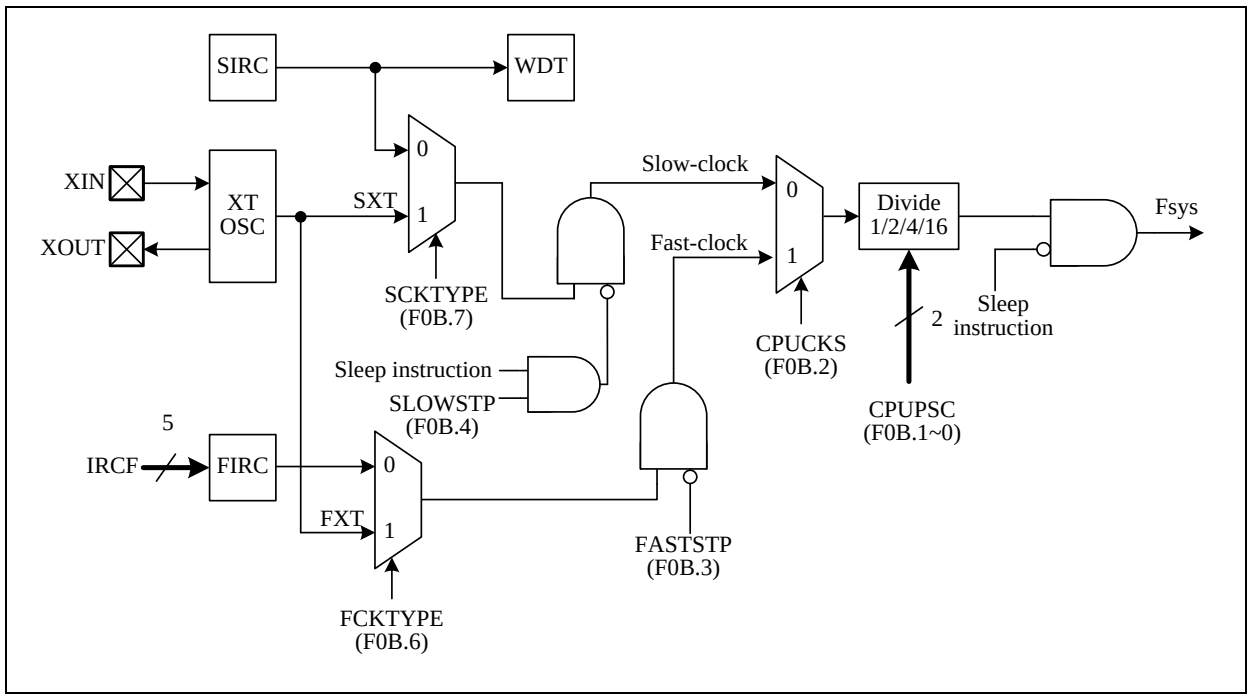
3.1 System Clock

The device is designed with dual-clock system. There are four kinds of clock source, i.e. SIRC (Slow Internal RC) , SXT (Slow Crystal, 32 KHz) , FXT (Fast Crystal, 1~16 MHz) and FIRC (Fast Internal RC) . Each clock source can be applied to CPU kernel as system clock. When in IDLE mode, only Slow-clock can be configured to keep oscillating to provide clock source to TM3 block. Refer to the figure below.

After Reset, the device is running at Slow mode with 128 KHz SIRC. S/W should select the proper clock rate for chip operation safety. The higher V_{CC} allows the chip to run at a higher System clock frequency. In a typical condition, an 16 MHz System clock rate requires V_{CC} > 2.5V.

The **TM57M5441** has an external oscillators connected to the XIN/XOUT pins. It relies on external circuitry for the clock signal and frequency stabilization, such as a stand-alone oscillator, quartz crystal, or ceramic resonator. In Fast mode, the fast oscillator can be used in the range from 1~16 MHz. In Slow mode, the slow oscillator can only use a clock frequency of 32.768 KHz.

The CLKCTL (F0B) SFR controls the System clock operating. H/W automatically blocks the S/W abnormally setting for this register. S/W can only change the Slow-clock type in Fast mode and change the Fast-clock type in Slow mode. Never to write both FASTSTP=1 & CPUCKS=1. It is recommended to write this SFR bit by bit.



Clock Scheme Block Diagram

The frequency of FIRC (Fast Internal RC) can be adjusted by IRCF (F1F) . When IRCF=00h, frequency is the lowest. When IRCF=Fh, frequency is the highest. With this function, we can adjust the frequency of FIRC after power on. Each IC may have different default value of IRCF, to make sure the frequency of FIRC=16 MHz after Power on Reset.

FAST Mode:

In this mode, the program is executed using Fast-clock as CPU clock (Fsys) . The Timer0, Timer1 blocks are also driven by Fast-clock, The PWM0 block can driven by FIRC16M or Fsys. TM3 can also be driven by Fast-clock by setting TM3CKS=0 and CPUCKS=1.

SLOW Mode:

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

IDLE Mode:

If Slow-clock is enabled and TM3CKS=1 before executing the SLEEP instruction, the CPU enters the IDLE mode. In this mode, the Slow-clock will continue running to provide clock to TM3 block. CPU stop fetching code and all blocks are stop except TM3 related circuits.

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

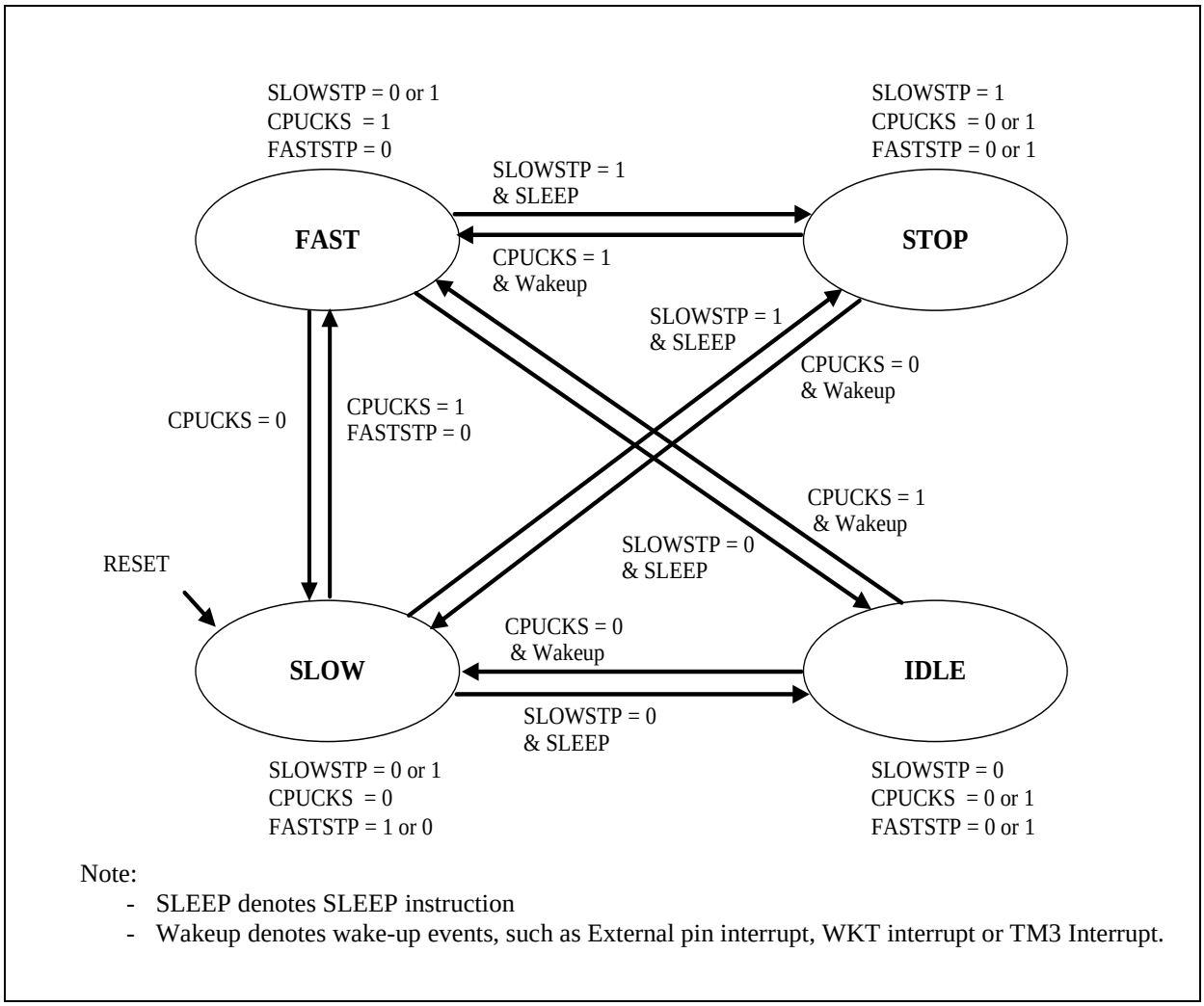
TM3 and WKT/WDT are independent and have their own control registers. It is possible to keep both TM3 and WKT working and wake-up in the IDLE mode.

STOP Mode:

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

3.2 Dual System Clock Modes Transition

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



Note:

- SLEEP denotes SLEEP instruction
- Wakeup denotes wake-up events, such as External pin interrupt, WKT interrupt or TM3 Interrupt.

CPU Operation Block Diagram

CPU Mode & Clock Functions Table:

Mode	Oscillator	Fsys	Fast-clock	Slow-clock	TM0/TM1	TM3	Wakeup event
FAST	FIRC	Fast-clock	Run	Set by SLOWSTP	Run	Run	X
SLOW	SIRC	Slow-clock	Set by FASTSTP	Run	Run	Run	X
IDLE	SIRC	Stop	Stop	Run	Stop	Run	WKT/IO/TM3
STOP	Stop	Stop	Stop	Stop	Stop	Stop	IO

- **FAST mode switches to SLOW mode**

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

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

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

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

◇ Example: Switch FAST mode to SLOW mode.

```
MOVLW   0x101xxB
MOVWF   F0B         ; Slow-clock type=SIRC
BCF     SLOWSTP     ; Enable Slow-clock.
NOP
BCF     CPUCKS      ; Fsys=Slow-clock.
BSF     FASTSTP     ; Disable Fast-clock.
```

- **SLOW mode switches to FAST mode**

SLOW mode can be enabled by CPUCKS=0 in F0B register of F-plane. The following steps are suggested to be executed by order when SLOW mode switches to FAST mode:

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

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

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

```
MOVLW   001001000B
MOVWF   F0B         ; Fast-clock=FIRC
BCF     FASTSTP     ; Enable Fast-clock.
NOP
BSF     CPUCKS      ; Fsys=Fast-clock
```

- **IDLE mode Setting**

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

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

(2) Switch TM3 clock source to Slow-clock (TM3CKS=1)

(3) Execute SLEEP instruction

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

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

```
BCF     SLOWSTP     ; Enable Slow-clock.
MOVLW   00000000B
MOVWR   R2F         ; TM3 Clock source=Slow-clock. TM3PSC=div 1
SLEEP               ; Enter IDLE mode.
```

- **STOP Mode Setting**

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

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

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

(3) Execute SLEEP instruction

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

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

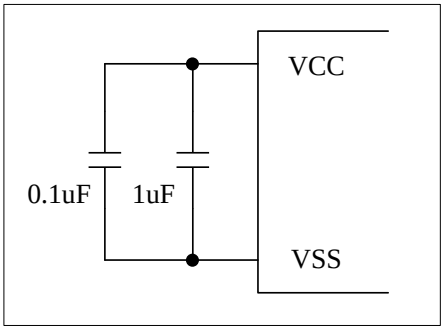
```
BSF     SLOWSTP     ; Disable Slow-clock.
MOVLW   x0000000B   ; Disable WKT counting
MOVWR   INTIE
SLEEP               ; Enter STOP mode.
```

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

3.3 System Clock Oscillator

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

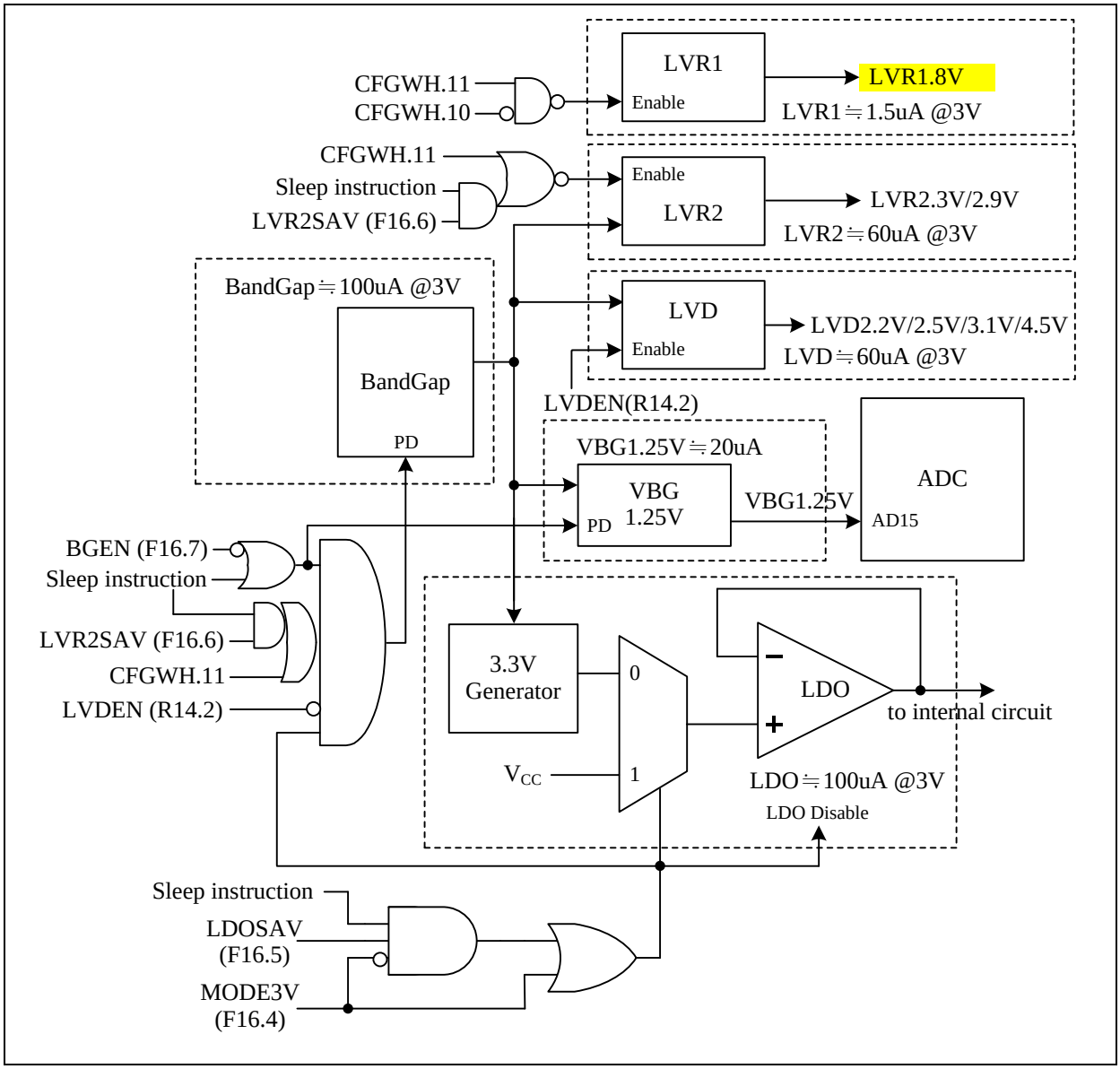


Internal RC Mode

4. Power

The **TM57M5541** has a built-in internal low dropout regulator (LDO) . When MODE3V=0, the voltage regulator outputs 3.3V power to the internal chip circuit. When MODE3V=1, the LDO is turned off, and the internal circuit receives a power supply directly from the VCC pin. Because the LDO consumes 200μA (Including LDO+BandGap) for operation, turning off LDO by setting MODE3V=1 can reduce the chip current consumption. However, setting MODE3V=1 is only valid for an operating condition of V_{CC} < 3.6V. The LDOSAV also control the LDO. When MODE3V=0 and LDOSAV=1, the LDO is turned off in STOP mode for saving power consumption.

If we enable anyone of LDO or LVR2 or LVD or VBG1.25V, BandGap (≒100μA) will be also enabled. To further save power consumption in STOP/IDLE mode, we must disable LDO/LVR2/LVD/VBG1.25V at the same time.



MODE3V=0

Operation Mode	LDOSAV (F16.5)	LVR CFGWH.11~10	LVR2SAV (F16.6)	LVDEN (R14.2)	BGEN (F16.7)	BandGap (100uA)	LDO (100uA)	LVR1 (2uA)	LVR2 (60uA)	LVD (60uA)	VBG 1.25V	Function
Fast / Slow	X	00	X	X	X	ON	ON	ON	ON	X	X	LVR2.9V
	X	01	X	X	X	ON	ON	ON	ON	X	X	LVR2.3V
	X	10	X	X	X	ON	ON	OFF	OFF	X	X	LVR Disable
	X	11	X	X	X	ON	ON	ON	OFF	X	X	LVR1.8V
	X	XX	X	1	X	ON	ON	X	X	ON	X	LVD Enable
	X	XX	X	X	1	ON	ON	X	X	X	ON	VBG1.25V
Idle / Stop	0	00	0	X	X	ON	ON	ON	ON	X	OFF	LVR2.9V
	0	01	0	X	X	ON	ON	ON	ON	X	OFF	LVR2.3V
	0	10	X	X	X	ON	ON	OFF	OFF	X	OFF	LVR Disable
	0	11	X	X	X	ON	ON	ON	OFF	X	OFF	LVR1.8V
	0	00	1	X	X	ON	ON	ON	OFF	X	OFF	LVR1.8V
	0	01	1	X	X	ON	ON	ON	OFF	X	OFF	LVR1.8V
	1	00	0	X	X	ON	OFF	ON	ON	X	OFF	LVR2.9V
	1	01	0	X	X	ON	OFF	ON	ON	X	OFF	LVR2.3V
	1	10	X	0	X	OFF	OFF	OFF	OFF	OFF	OFF	LVR Disable
	1	11	0	0	X	OFF	OFF	ON	OFF	OFF	OFF	LVR1.8V
	1	00	1	0	X	OFF	OFF	ON	OFF	OFF	OFF	LVR1.8V
	1	01	1	0	X	OFF	OFF	ON	OFF	OFF	OFF	LVR1.8V
	X	XX	X	1	X	ON	X	X	X	ON	OFF	LVD Enable

MODE3V=1

Operation Mode	LDOSAV (F16.5)	LVR CFGWH.11~10	LVR2SAV (F16.6)	LVDEN (R14.2)	BGEN (F16.7)	BandGap (100uA)	LDO (100uA)	LVR1 (2uA)	LVR2 (60uA)	LVD (60uA)	VBG 1.25V	Function
Fast / Slow	X	00	X	X	X	ON	OFF	ON	ON	X	X	LVR2.9V
	X	01	X	X	X	ON	OFF	ON	ON	X	X	LVR2.3V
	X	10	X	0	0	OFF	OFF	OFF	OFF	OFF	OFF	LVR Disable
	X	11	X	0	0	OFF	OFF	ON	OFF	OFF	OFF	LVR1.8V
	X	XX	X	1	X	ON	OFF	X	X	ON	X	LVD Enable
	X	XX	X	X	1	ON	OFF	X	X	X	ON	VBG1.25V
Idle / Stop	X	00	0	X	X	ON	OFF	ON	ON	X	OFF	LVR2.9V
	X	01	0	X	X	ON	OFF	ON	ON	X	OFF	LVR2.3V
	X	10	X	0	X	OFF	OFF	OFF	OFF	OFF	OFF	LVR Disable
	X	11	X	0	X	OFF	OFF	ON	OFF	OFF	OFF	LVR1.8V
	X	00	1	0	X	OFF	OFF	ON	OFF	OFF	OFF	LVR1.8V
	X	01	1		X	OFF	OFF	ON	OFF	OFF	OFF	LVR1.8V
	X	XX	X	1	X	ON	X	X	X	ON	OFF	LVD Enable

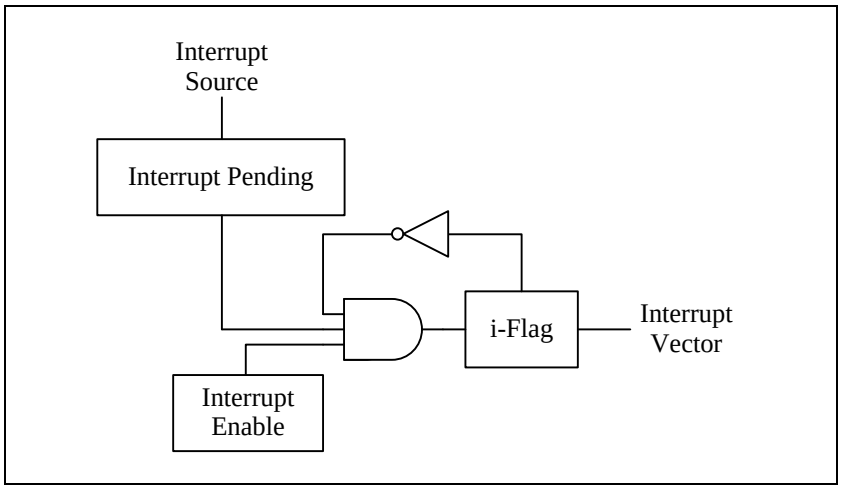
5. Interrupt

This device has 1 level, 13 vector and 14 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 device has 8 vectors, there is not an interrupt priority register. Priority of each interrupt is equal and the device does not support nested interrupt. Another interrupts can be executed only if the current interrupt is exited (that is, RETI instruction is executed). Although the interrupts do not have priority, however, when exit from current interrupt, if there are more than 2 interrupts happened, the priority of the interrupt is LVD >TM3 >TK >I2C/SPI >TM0 >TM1 >XINT4 >XINT3 >XINT2 >XINT1 >XINT0 >ADC >WKT.

No	Address	Source	Description	WakeUp
1	001	LVD	Low Voltage detection interrupt	Yes
2	002	Timer3	Timer3 Counter Overflow	Yes
3	003	TK	Touch Key interrupt	Yes
4	004	I2C/SPI	I2C interrupt or SPI interrupt	Yes/No
5	005	Timer0	Timer0 Counter Overflow	No
6	006	Timer1	Timer1 Counter Overflow	No
7	007	XINT4	PD2 rising/falling interrupt (rising/falling selectable)	Yes
8	008	XINT3	PD1 rising/falling interrupt (rising/falling selectable)	Yes
9	009	XINT2	PA7 falling interrupt	Yes
10	00a	XINT1	PA1 rising/falling interrupt (rising/falling selectable)	Yes
11	00b	XINT0	PA6 rising/falling interrupt (rising/falling selectable)	Yes
12	00c	ADC	ADC interrupt	No
13	00d	WKT	Wakeup Timer Match	Yes

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 00n” (n ranges from 1 to d) instruction is inserted to CPU, and i-flag is set to prevent recursive interrupt nesting.

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



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

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

        ORG     0aH                 ; INT1 interrupt vector.
        GOTO    INT_SUBROUTINE      ; If INT1 (PA1) input occurred rising edge.

        ORG     0bH

START:
        MOVLW   0101 00 01B
        MOVWR   PAMODL              ; Enable INT1 (PA1) input pull up resistor.
        MOVLW   01 1 001111B
        MOVWR   R0B                 ; Set INT1 interrupt trigger as rising edge.
        MOVLW   111111 0 1B
        MOVWF   INTIF               ; Clear INT1 interrupt request flag
        MOVLW   000000 1 0B
        MOVWR   INTIE               ; Enable INT1 interrupt.
MAIN:
        …
        GOTO    MAIN

INT_SUBROUTINE:
        MOVWF   GPR0                ; Push routine to Save W and STATUS data to buffers.
        MOVFW   STATUS              ; F-Plane 03H
        MOVWF   GPR1

        BTFSS   INT1IF              ; Check INT1IF bit.
        GOTO    EXIT_INT            ; INT1IF=0, exit interrupt vector.
        …                           ; INT1 interrupt service routine.
        MOVLW   111111 0 1B
        MOVWF   INTIF               ; Clear INT1 interrupt request flag
        GOTO    EXIT_INT

EXIT_INT:
        MOVFW   GPR1                ; POP Routine W and STATUS data from buffers.
        MOVWF   STATUS
        MOVFW   GPR0
        RETI
```

R08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	TM3IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F08.7 **TKIE:** Touch Key interrupt enable
0: disable
1: enable

F08.6 **TM3IE:** Timer3 interrupt enable
0: disable
1: enable

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

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

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

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

F08.1 **INT1IE:** INT1 (PA1) interrupt enable
0: disable
1: enable

F08.0 **INT0IE:** INT0 (PA6) interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	TKIF	TM3IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F09.7 **TKIF:** Touch Key interrupt event pending flag, set by H/W
ATKEN=1: Key's TK Data Count is over the pre-set compare threshold range
ATKEN=0: end of TK conversion

F09.6 **TM3IF:** Timer3 interrupt event pending flag
This bit is set by H/W while Timer3 overflows, 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

F09.1 **INT1IF:** INT1 (PA1) pin falling interrupt pending flag
This bit is set by H/W at INT1 pin's falling/rising edge, write 0 to this bit will clear this flag

F09.0 **INT0IF:** INT0 (PA6) pin falling/rising interrupt pending flag
This bit is set by H/W at INT0 pin's falling/rising edge, write 0 to this bit will clear this flag

R0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	–	–	LVDIE	ADCIE	I2CIE	SPIE	INT4IE	INT3IE
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	0	0	0	0

F0D.5 **LVDIE:** LVD interrupt enable
0: disable
1: enable

F0D.4 **ADCIE:** ADC interrupt enable
0: disable
1: enable

F0D.3 **I2CIE:** I2C interrupt enable
0: disable
1: enable

F0D.2 **SPIE:** SPI interrupt enable
0: disable
1: enable

F0D.1 **INT4IE:** INT1 (PD2) interrupt enable
0: disable
1: enable

F0D.0 **INT3IE:** INT0 (PD1) interrupt enable
0: disable
1: enable

F0E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	–	–	LVDIF	ADCIF	I2CIF	–	INT4IF	INT3IF
R/W	–	–	R/W	R/W	R/W	–	R/W	R/W
Reset	–	–	0	0	0	–	0	0

F0E.5 **LVDIF:** Low voltage detection interrupt flag,
This bit is set by H/W while Vcc≦ LVD

F0E.4 **ADCIF:** ADC interrupt flag
This bit is set by H/W after end of ADC conversion, write 0 to this bit will clear this flag

F0E.3 **I2CIF:** I2C interrupt event pending flag,
This bit is set by H/W while
1. I2CRCD0 or I2CRCD1 receive new data finished
2. I2CRDC0 or I2CRCD1 data overflow occurred
3. I2CTXD0 or I2CTXD1 data transmit finished
, write 0 to this bit will clear this flag

F0E.1 **INT4IF:** INT4 (PD2) pin falling interrupt pending flag
This bit is set by H/W at INT4 pin's falling/rising edge, write 0 to this bit will clear this flag

F0E.0 **INT3IF:** INT0 (PD1) pin falling/rising interrupt pending flag
This bit is set by H/W at INT3 pin's falling/rising edge, write 0 to this bit will clear this flag

F19	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPSTA	SPIF	WCOL	–	RCVOVF	RCVBF	SPBSY	–	–
R/W	R/W	R/W	–	R/W	R/W	R	–	–
Reset	0	0	–	0	0	–	–	–

F19.7 **SPIF:** SPI Interrupt Flag
Set by H/W at the end of a data transfer. Cleared by H/W when interrupt is vectored into. Write 0 to this bit will clear this flag.

6. I/O Port

6.1 PA0-6, PB0-7, PD0-7, PE0-1

These pins can be used as Schmitt-trigger input, CMOS push-pull output. The pull-up resistor is assignable to each pin by S/W setting. To use the pin in Schmitt-trigger input mode, S/W needs to set the I/O pin to Mode0 or Mode1 and PxD=1. Reading the pin data (PxD) has different meaning. In "Read-Modify-Write" instruction, CPU actually reads the output data register. In the others instructions, CPU reads the pin state. The so-called "Read-Modify-Write" instruction includes BSF, BCF and all instructions using F-Plane as destination.

These pins can operate in four different modes as below.

Mode	PA0~PA6, PB0~PB7, PD0~PD7, PE0~1 pin function	PxD SFR data	Pin State	Resistor Pull-up	Digital Input
Mode 0	Open Drain	0	Drive Low	N	N
	Input	1	Pull-up	Y	Y
	Touch Key (when TKCHS)	1	TK	N	N
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	ADC/CLD/Wakeup	0	–	N	N/N/Y
	Wakeup	1	–	Y	Y

I/O Pin Function Table

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

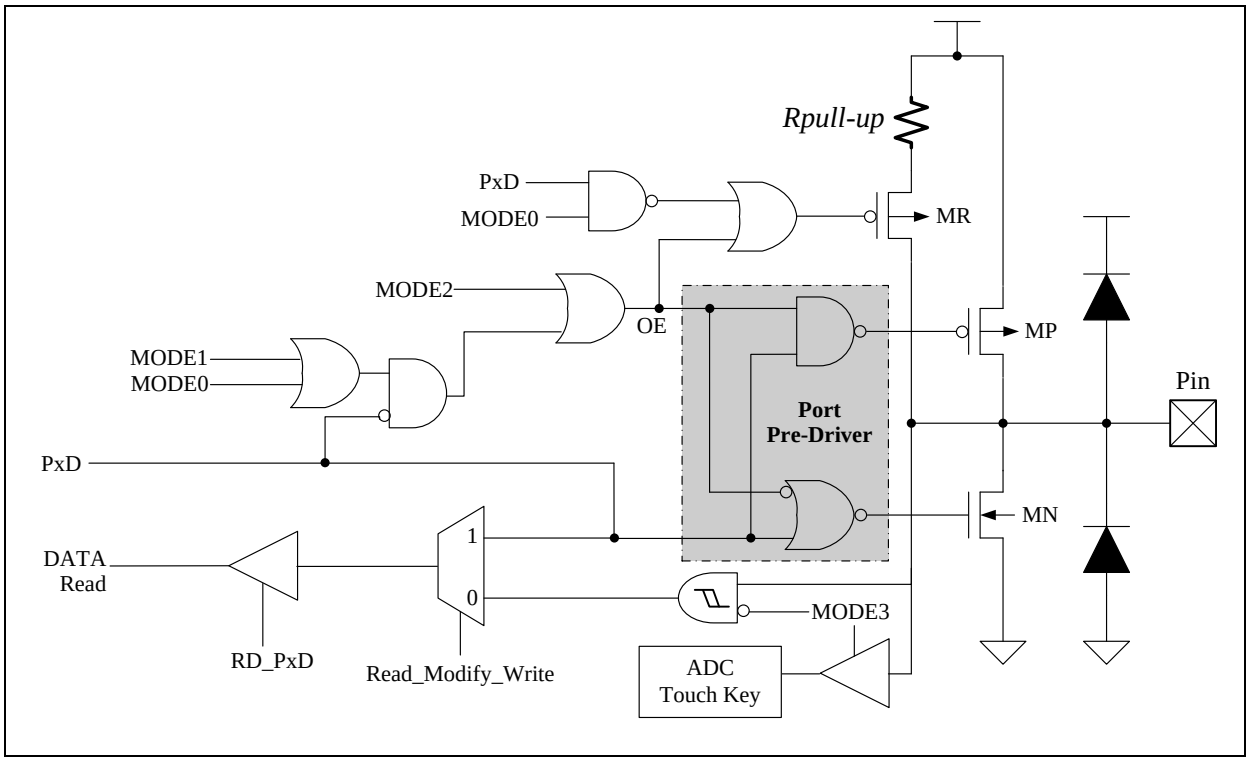
Pin Name	Wake-up	CKO	ADC/TK	others	Mode3
PA0			ADC8/TK6	PWM0O1	ADC8
PA1	INT1		ADC1/TK1		ADC1
PA2			ADC2/TK2		ADC2
PA3				XOUT/HS	
PA4				XIN/HS	
PA5			ADC5/TK5		ADC5
PA6	INT0		ADC0/TK0		ADC0
PA7	INT2				
PB0			ADC7/TK7	PWM1A	ADC7
PB1			ADC3/TK3		ADC3
PB2			ADC11	HS	ADC11
PB3				HS	
PB4			TK10	PWM1C	
PB5			TK9	PWM1B	
PB6			ADC13/TK14	HS	ADC13
PB7			ADC14/TK13		ADC14
PD0		TRCOUT/ TM1OUT			
PD1	INT3			SCK/IICSCL	
PD2	INT4			MOSI/IICSDA	
PD3				MISO/HS	
PD4			ADC10	PWM0O2/HS	ADC10
PD5			ADC9	TKCLD	ADC9/TKCLD
PD6		TCOUT	ADC6/TK8		ADC6
PD7			ADC4/TK4		ADC4
PE0	Wakeup		TK12		Wakeup
PE1	Wakeup		TK11		Wakeup

PortA/B/D/E multi-function Table

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

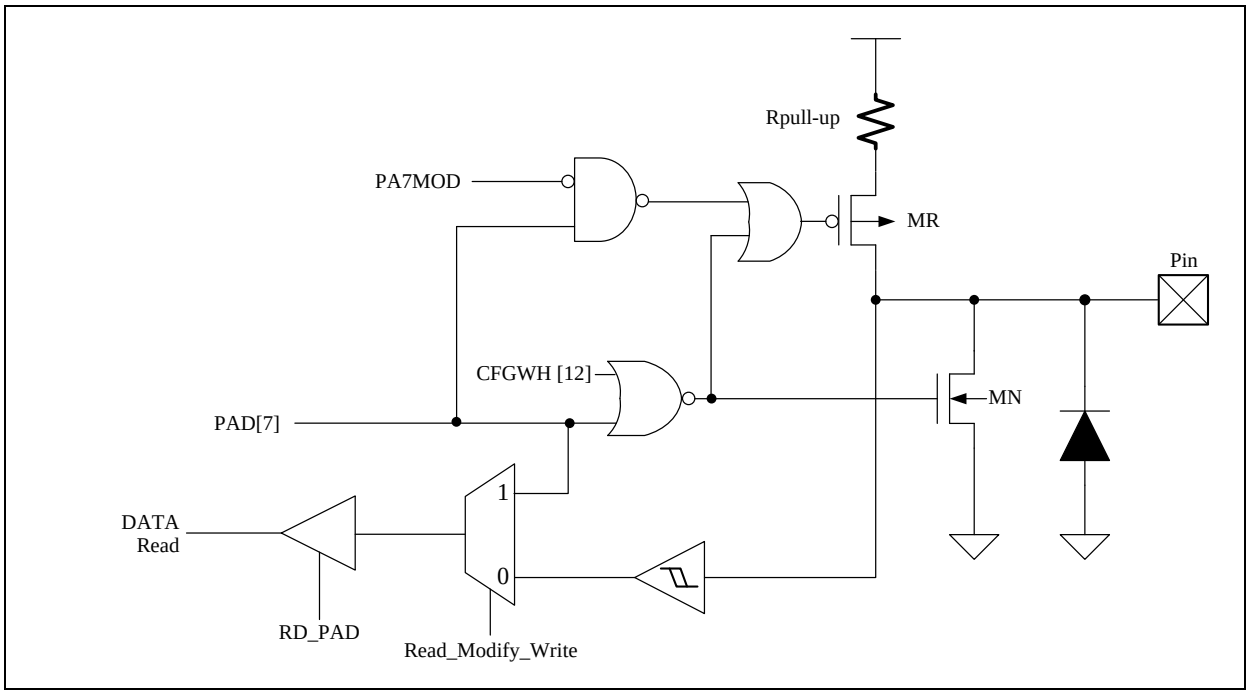
Alternative Function	Mode	PxD SFR data	Pin State	Other necessary SFR setting
INT0, INT1, INT3, INT4	**0**	1	Input with Pull-up	INTxIE
	1	1	Input	INTxIE
TK0~TK15	**0**	1	Touch Key Idling, Pull-up	TKCHS
			Touch Key Scanning	TKCHS
TKCLD	**3**	X	Touch Key Capacitor Connection	TKCLDXS
AD0~AD14	**3**	X	ADC Channel	
PWM0O1, PWM0O2, PWM1A, PWM1B, PWM1C	**X**	X	PWM Output (CMOS Push-Pull)	PWM0O1OE PWM0O2OE PWM1AOE PWM1BOE PWM1COE
SPI Master Mode MISO	**1**	1	SPI Data Input	SPCON
SPI Master Mode SCK, MOSI	**2**	X	SPI Clock/Data Output (CMOS Push-Pull)	SPCON
SPI Slave Mode MISO	**2**	X	SPI Data Output (CMOS Push-Pull)	SPCON
SPI Slave Mode SCK, MOSI	**1**	1	SPI Clock/Data Input	SPCON
XIN, XOUT	**0**	1	Crystal oscillation	CLKCON

Mode Setting for Port Alternative Function



6.2 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 controlled by PA7MOD (R05.6) bit and the default value is enabled (i.e. PA7MOD=0) after system reset.



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

CFGWH.12	PAD[7]	PA7MOD	Pin State	Pull-up	MODE
0	0	0	Low	No	open-drain output without pull-high
0	1	0	High	Yes	input with pull-high
0	1	1	Hi-Z	No	input without pull-high
1	1	0	High	Yes	reset input with pull-high

◇ Example: Read state from PA7.

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

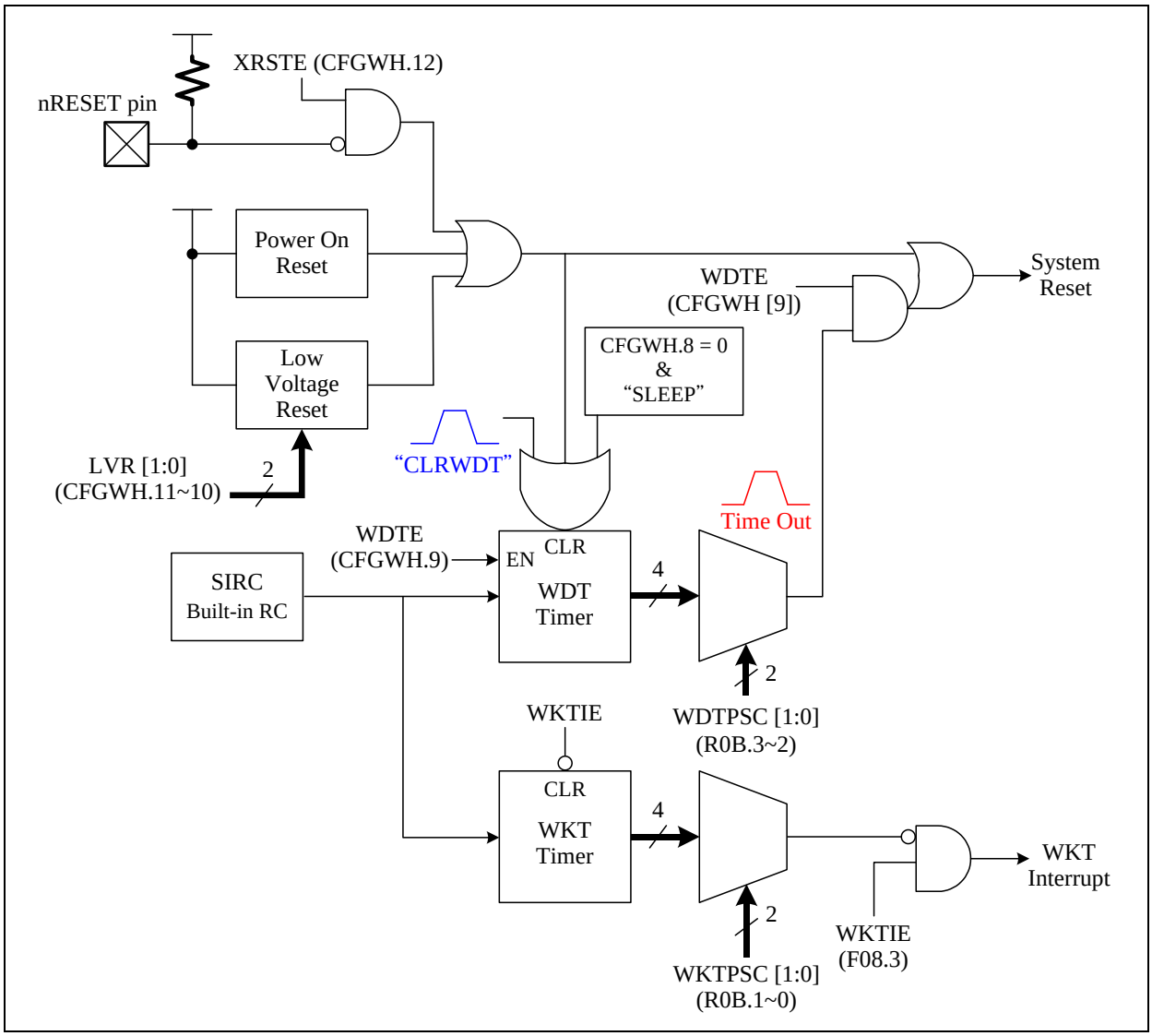
```
BTFSS    PAD,7
GOTO     LOOP_A      ; If PA7=0.
GOTO     LOOP_B      ; If PA7=1.
```

7. Peripheral Functional Block

7.1 Watchdog (WDT) /Wakeup (WKT) Timer

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

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



WDT/WKT Block Diagram

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

◇ Example: Clear watchdog timer by CLRWDT instruction.

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

◇ Example: Clear watchdog timer by write WDTCLR register.

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

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

```
        MOVLW       0000011B
        MOVWR       R0B         ; Select WDT Time out=256 ms @5V

        SLEEP
```

◇ Example: Set WKT period and interrupt function.

```
        MOVLW       0000010B
        MOVWR       R0B         ; Select WKT period=64 ms @5V.

        MOVLW       1111 0111B  ; Clear WKT interrupt request flag by using byte operation
                                ; Don't use bit operation "BCF WKTIF" clear interrupt flag
        MOVWF       INTIF       ; F-Plane 09H

        MOVLW       0000 1000B  ; Enable WKT interrupt function
        MOVWF       F08
```

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	TKIF	TM3IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	TM3IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

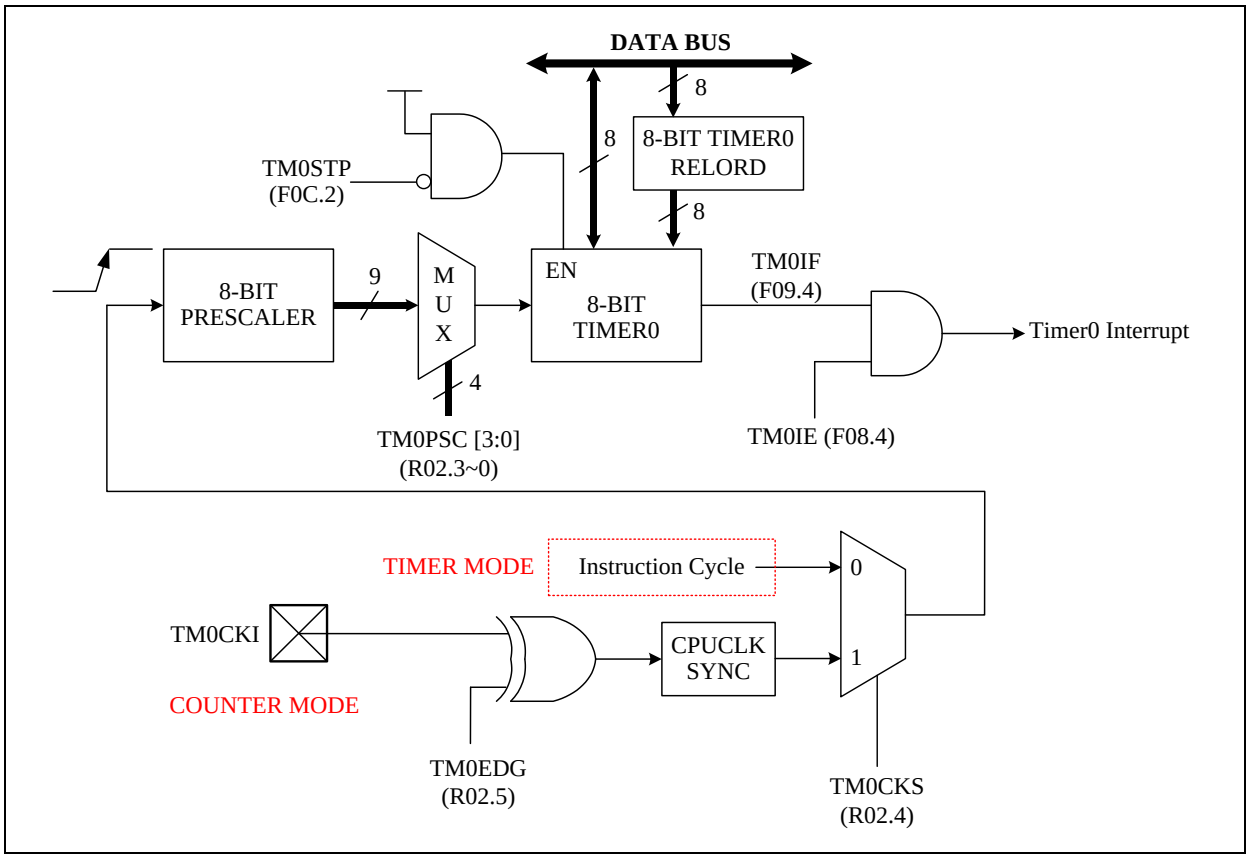
R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	HWAUTO	INT0EDG	–	–	WDTPSC		WKTPSC	
R/W	R/W	R/W	–	–	R/W	R/W	R/W	R/W
Reset	0	0	–	–	1	1	1	1

R0B.3~2 **WDTPSC:** WDT period (@VCC=5V)
00: 128 ms 01: 256 ms 10: 1024 ms 11: 2048 ms

R0B.1~0 **WKTPSC:** WKT period (@VCC=5V)
00: 16 ms 01: 32 ms 10: 64 ms 11: 128 ms

7.2 Timer0

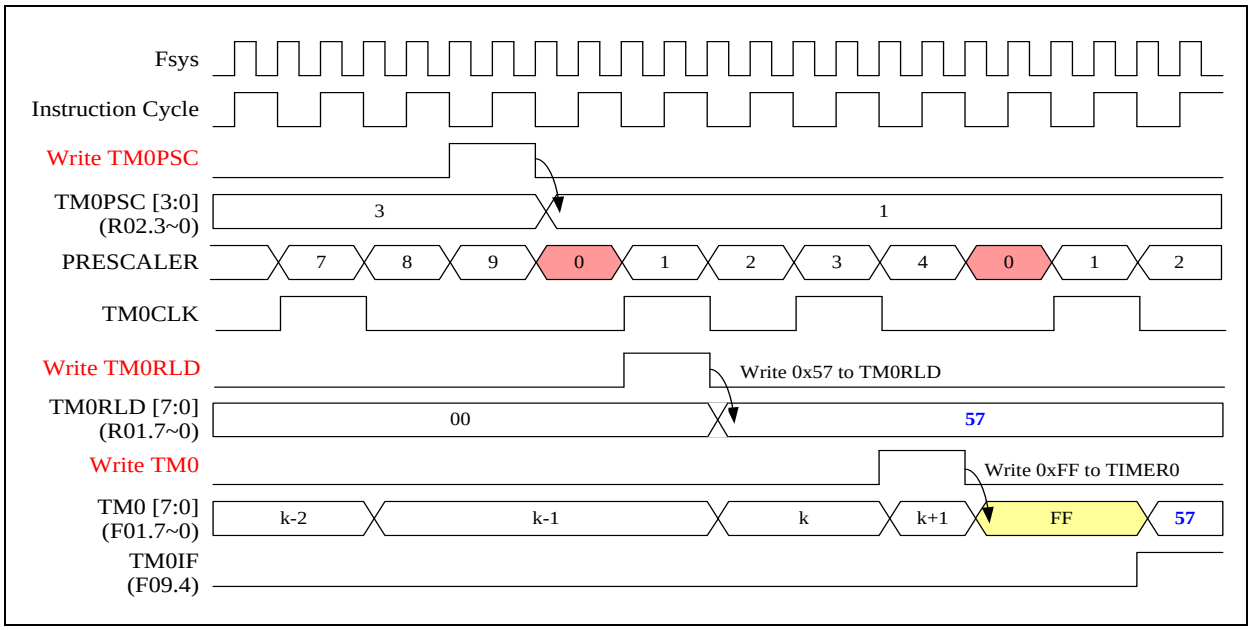
The Timer0 is an 8-bit wide register of F-Plane 01h (TM0) . It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatically rolls over a new "offset value" (TM1RLD) while it rolls over based on the pre-scaled clock source, which can be the instruction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



Timer0 Block Diagram

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

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



Timer0 works in Timer mode (TM0CKS=0)

The equation of TM0 interrupt time value is as following:

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

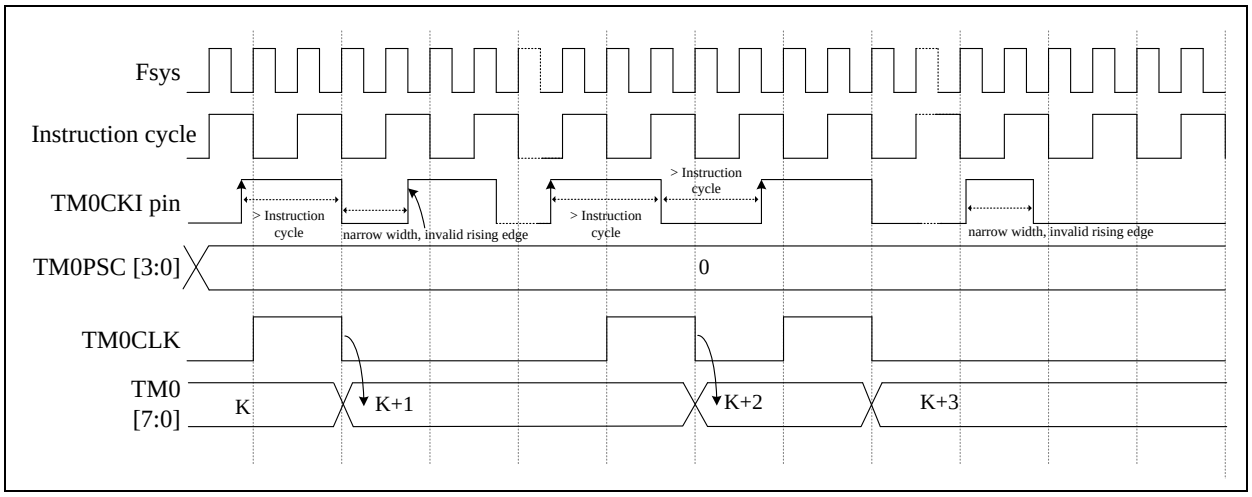
◇ Example: Setup TM0 work in Timer mode

```
; Setup TM0 clock source and divider
        MOVLW   0000 0101B      ; R02.4=0, Setup TM0 clock=Instruction cycle
        MOVWR   R02             ; R02.3~0=5 (TM0PSC)
                                ; TM0 clock prescaler=Instruction cycle divided by 32
; Set TM0 timer.
        BSF     TM0STP          ; Disable TM0 counting (Default "0").
        MOVLW   156
        MOVWF   TM0             ; Write 156 into TM0 register of F-Plane. (F01)
        MOVLW   124
        MOVWR   TM0RLD          ; Write 156 into TM0RLD register of R-Plane. (R01)

; Enable TM0 timer and interrupt function.
        MOVLW   1110 1111B      ; Clear TM0 request interrupt flag by byte operation
        MOVWF   INTIF           ; F-Plane 09H
        MOVLW   0001 0000B      ; Enable TM0 interrupt function
        MOVWR   INTIE           ; F-Plane 08H
        BCF     TM0STP          ; Enable TM0 counting (Default “0”).
```

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

If TM0CKS=1 then Timer0 counter source clock is from TM0CKI pin. TM0CKI signal is synchronized by instruction cycle 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.



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

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

```
; Setup TM0 clock source from TM0CKI pin (PA2) and divider.
        MOVLW   00110000B
        MOVWR   R02             ; R02.5=1, Select TM0 prescaler counting edge=falling
                                  edge.
                                ; R02.4=1, Setup TM0 clock=TM0CKI pin (PA2)
                                ; R02.3~0=0 (TM0PSC)
                                ; TM0 clock prescaler=Instruction cycle divided by 1

; Set TM0 timer and stop TM0 counting.
        BSF     TM0STP          ; Disable TM0 counting (Default "0").
        MOVLW   00H
        MOVWF   TM0             ; Write 0 into TM0 register of F-Plane 01H.

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

        MOVFW   TM0
```

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

F01 **TM0:** Timer0 content

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	TM3IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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	TM3IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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	TM3SET	TM3STP	TM1STP	TM0STP	PED	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	1

F0C.2 **TM0STP:** Timer0 counter stop
0: Release 1: Stop counting

R01	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0RLD	TM0RLD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

R01 **TM0RLD:** Timer0 Reload Data

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

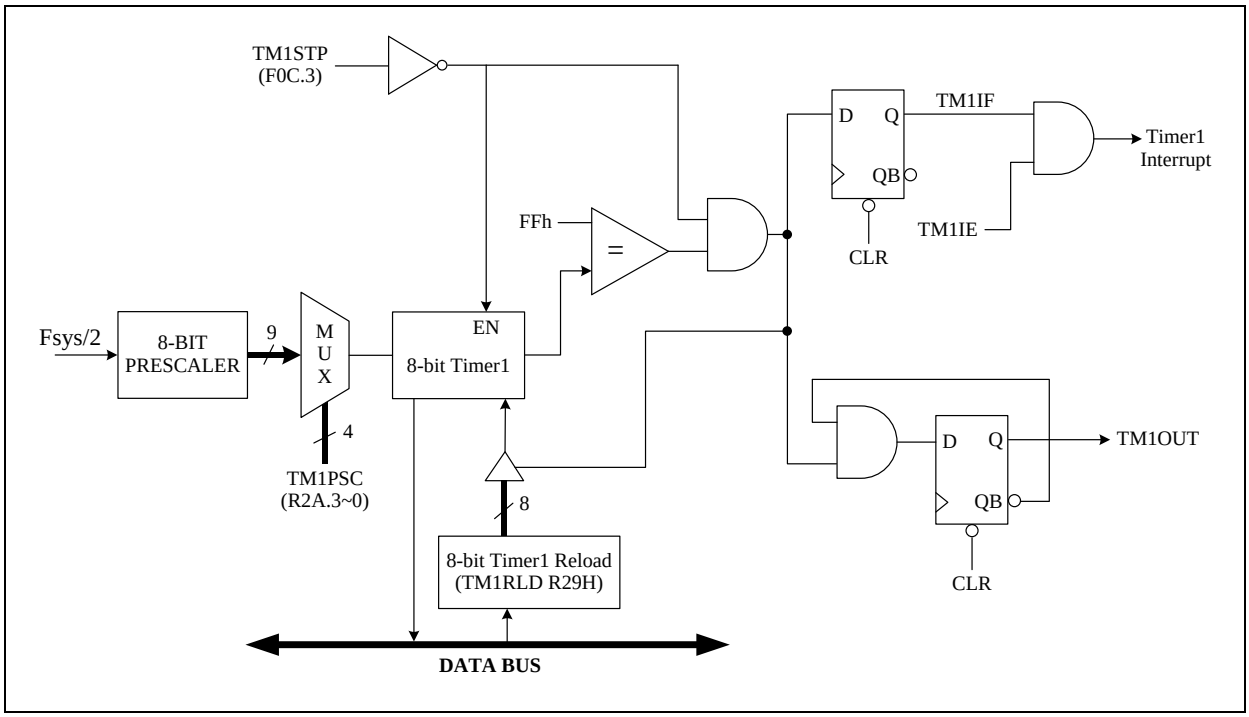
R02.5 **TM0EDG:** Timer0 prescaler counting edge for TM0CKI pin
0: rising edge 1: falling edge

R02.4 **TM0CKS:** Timer0 prescaler clock source
0: Instruction cycle 1: TM0CKI pin (PA2 pin)

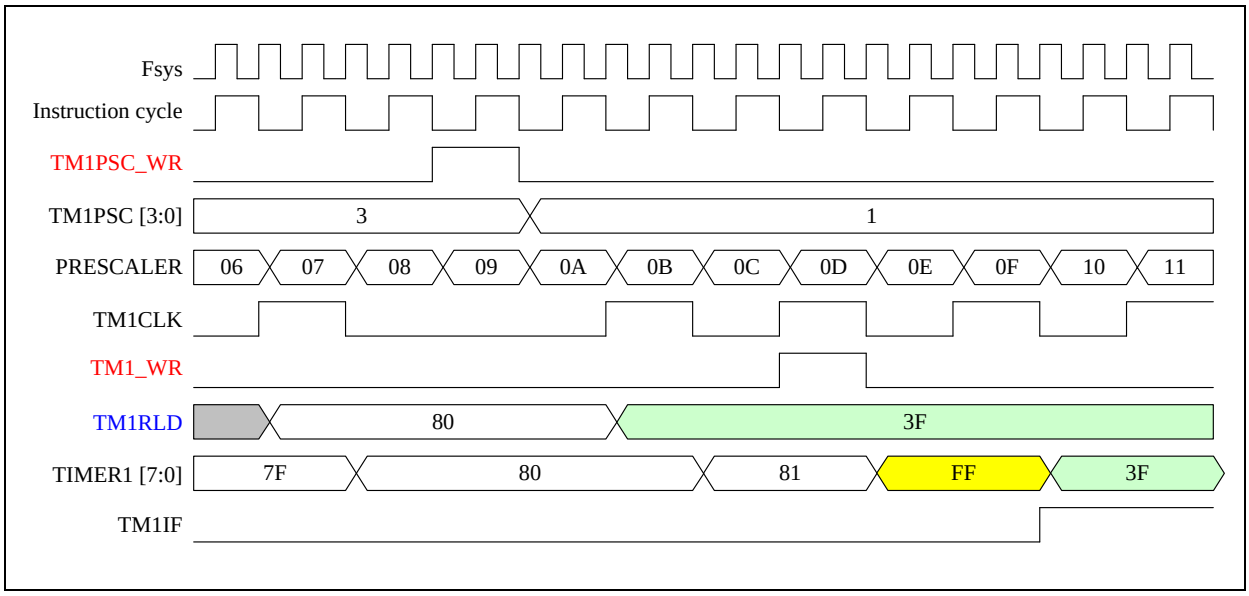
R02.3~0 **TM0PSC:** Timer0 prescaler. Timer0 prescaler clock source divided by
0000: /1 0001: /2 0010: /4 0011: /8 0100: /16
0101: /32 0110: /64 0111: /128 1xxx: /256

7.3 Timer1

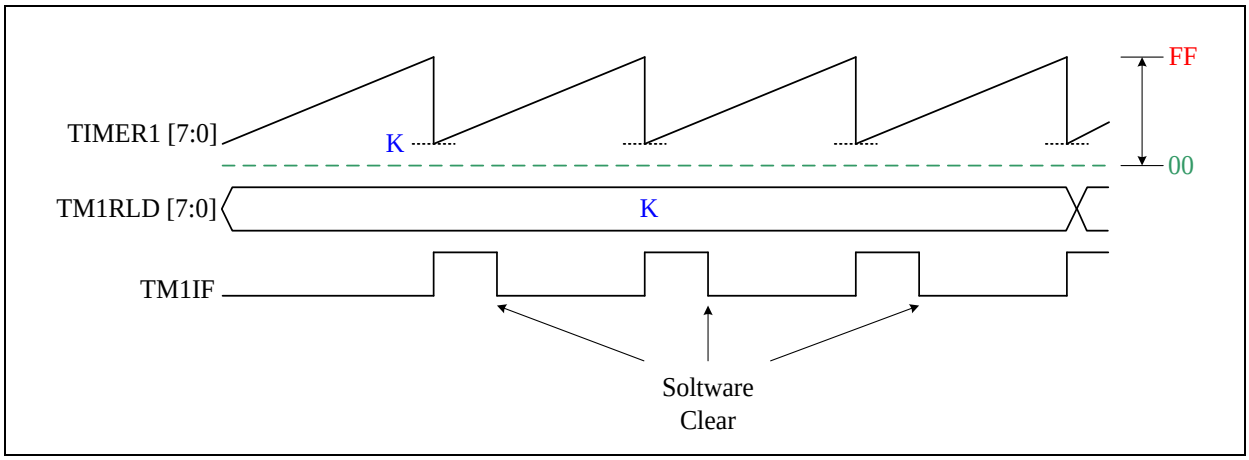
The Timer1 is an 8-bit wide register of F-Plane. It can be read or written as any other register of F-Plane. Besides, Timer1 increases itself periodically and automatically reloads a new "offset value" (TM1RLD) while it rolls over based on the pre-scaled instruction clock. The Timer1 increase rate is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



Timer1 Block Diagram



Timer1 Timing Diagram



Timer1 Reload Diagram

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

```
; Setup TM1 clock source, divider and enable TM1OUT
        MOVLW   x000 0101 B
        MOVWR   R2A             ; R2A.3~0=5 (TM1PSC) , Select TM1 clock=Fsys/64.
        BSF     TM1OE           ; F1B.5=1, Enable TM1OUT function pin (PD0).

; Set TM1 timer offset and stops TM1 counting
        BSF     TM1STP          ; Stop TM1 counting (Default "0").
        MOVLW   F0H
        MOVWF   TM1             ; Write F0H into TM1 counter (F0A, F-Plane)

; Enable TM0 timer and interrupt function.
        MOVLW   11 0 11111B     ; Clear TM1 request interrupt flag by byte operation
        MOVWF   INTIF           ; F-Plane 09H

        MOVLW   00 1 00000B     ; Enable TM1 interrupt function.
        MOVWF   INTIE           ;

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

Example:

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

TM1RLD=0xF0,

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

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

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

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	TM3IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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	TM3IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

R28	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1	TM1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

R28 **TM1:** Timer1 content

F0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0C	CLKFLT	VCCFLT	TM3SET	TM3STP	TM1STP	TM0STP	PED	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	1	1

F0C.3 **TM1STP:** Timer1 counter stop
0: Release
1: Stop counting

F1B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF1B	TCOE	TRCOE	TM1OE	PWM1COE	PWM1BOE	PWM1AOE	PWM0O2OE	PWM0O1OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

R2A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR2A	–	–	–	–	TM1PSC			
R/W	–	–	–	–	W	W	W	W
Reset	–	–	–	–	0	0	0	0

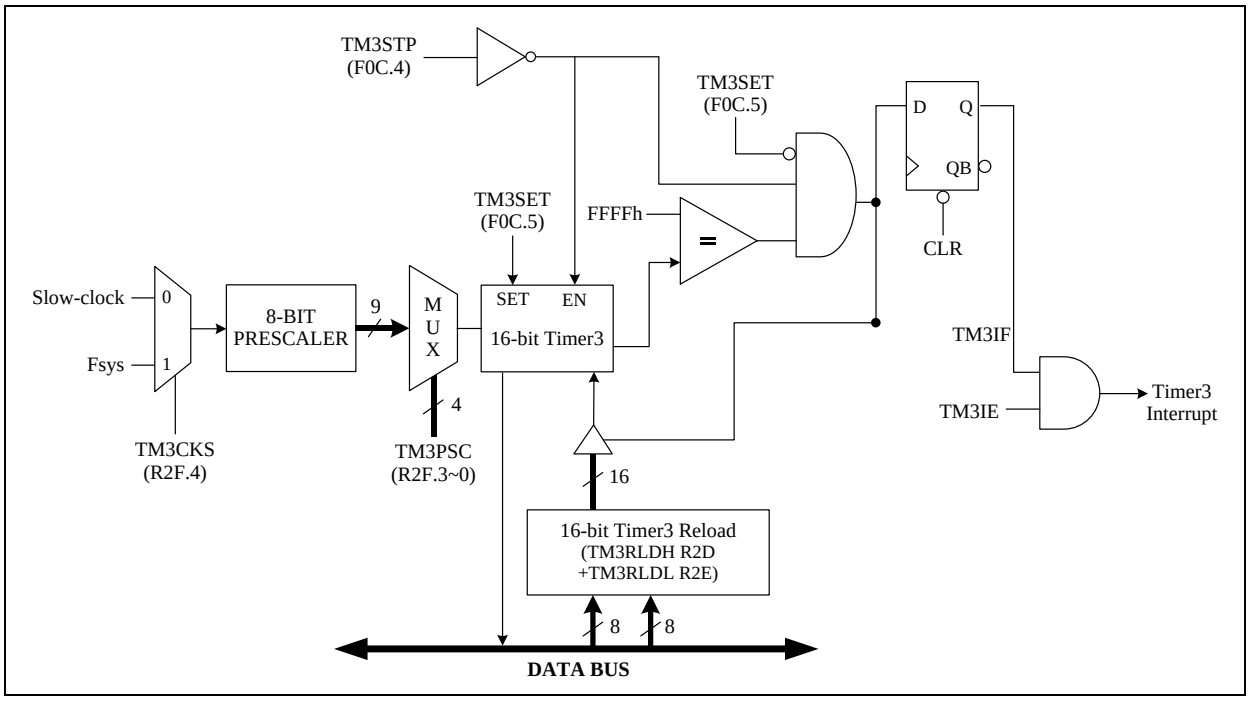
R2A.3~0 **TM1PSC:** Timer1 prescaler. Timer1 clock source divided by
0000: Fsys/2 0101: Fsys/64
0001: Fsys/4 0110: Fsys/128
0010: Fsys/8 0111: Fsys/256
0011: Fsys/16 1xxx: Fsys/512
0100: Fsys/32

R29	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1RLD	TM1RLD							
R/W	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

R29.7~0 **TM1RLD:** Timer1 reload offset value while it rolls over

7.4 Timer3:16-bit Timer

The Timer3 is an 16-bit wide register of R-Plane. It can be read as any other register of R-Plane. Besides, Timer3 increases itself periodically and automatically reloads a new "offset value" (TM3RLD) while it rolls over based on the pre-scaled Fsys or Slow-clock. The Timer3 increasing rate is determined by TM3PSC register in R-Plane. Setting the TM3STP bit will stop Timer3 counting. Setting the TM3SET=1 will hold Timer3 on the FFFFh, but it will not make TM3IF=1. After resetting TM3SET to 0, Timer3 will no longer be held on the FFFFh, and reload a new value (TM3RLD) in the next step.



Timer3 Block Diagram

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	TM3IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F08.6 **TM3IE:** Timer3 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	TM3IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F09.6 **TM3IF:** Timer3 interrupt event pending flag
This bit is set by H/W while Timer3 overflows, write 0 to this bit will clear this flag

R2B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3H	TM3H							
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

R2B **TM3H:** Timer3 content high byte [15:8]

R2C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3L	TM3L							
R/W	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

R2C **TM3L:** Timer3 content low byte [7:0]

R2D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3RLDH	TM3RLDH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

R2D **TM3RLDH:** Timer3 reload offset value high byte [15:8] while it rolls over

R2E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3RLDL	TM3RLDL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

R2E **TM3RLDL:** Timer3 reload offset value low byte [7:0] while it rolls over

R2F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3CTL	–	–	–	TM3CKS	TM3PSC			
R/W	–	–	–	R/W	R/W	R/W	R/W	R/W
Reset	–	–	–	0	0	0	0	0

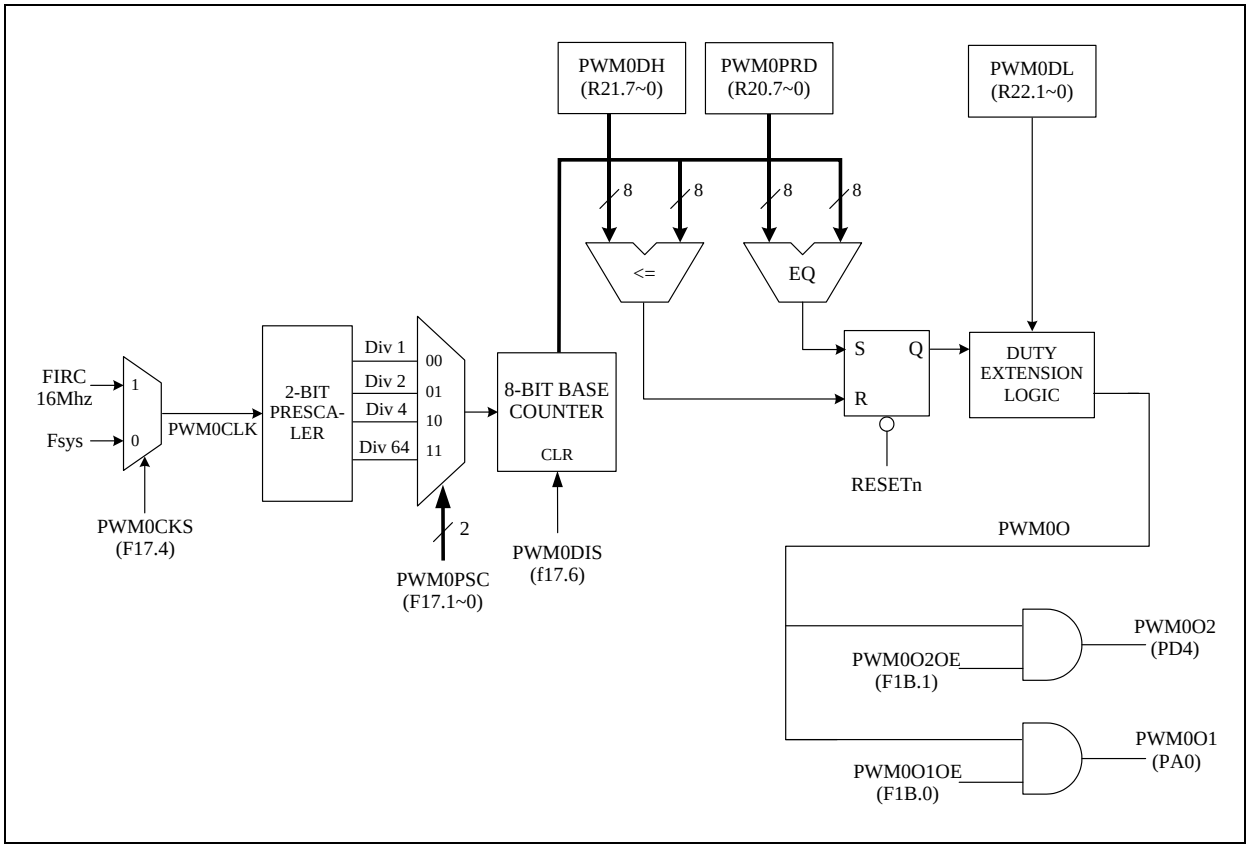
R2F.4 **TM3CKS:** Timer3 prescaler clock source
0: Instruction cycle 1: Slow-clock

R2F.3~0 **TM3PSC:** Timer3 prescaler. Timer3 prescaler clock source divided by
0000: /1 0001: /2 0010: /4 0011: /8 0100: /16
0101: /32 0110: /64 0111: /128 1xxx: /256

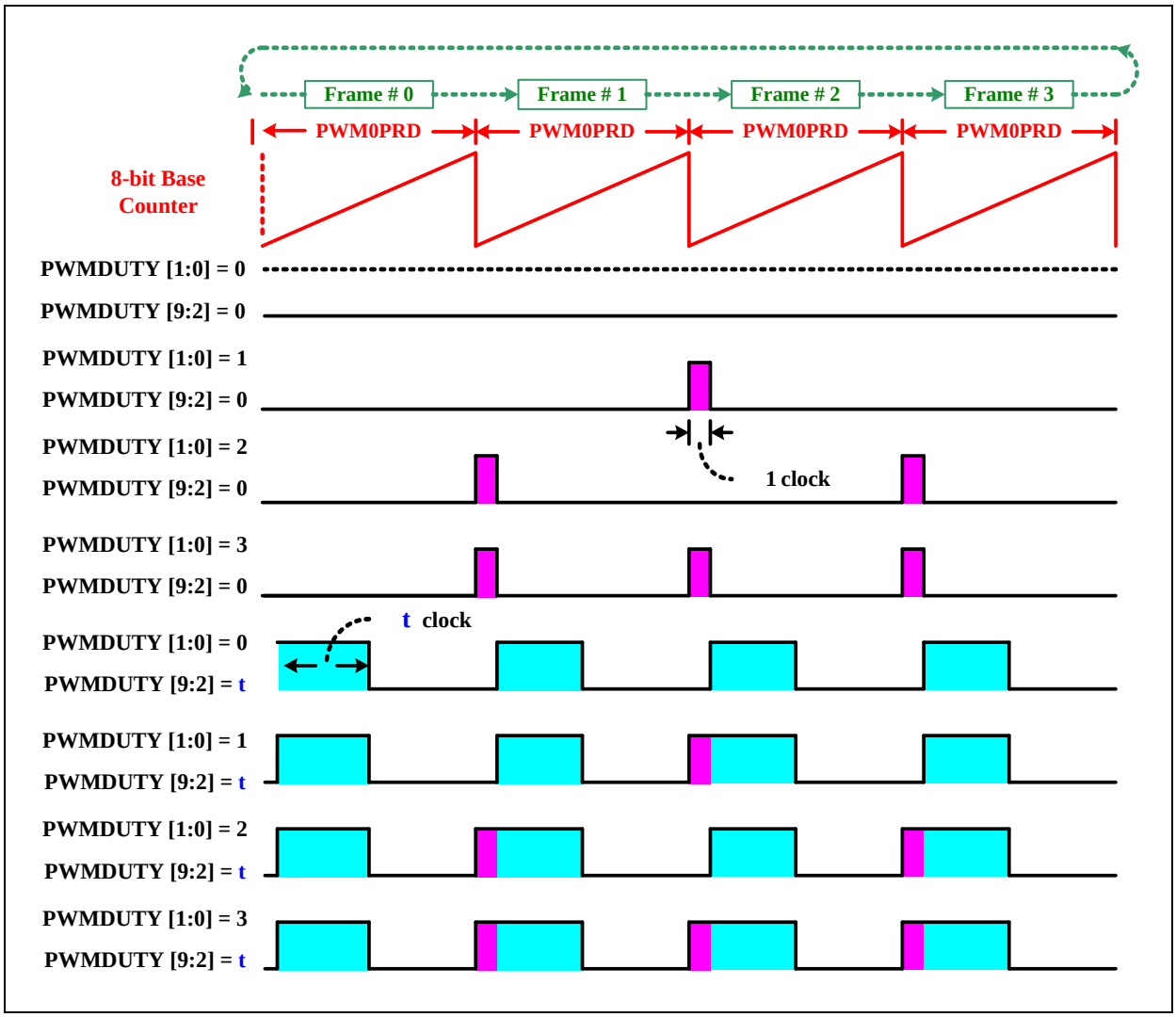
7.5 PWM0: (8+2) bits PWM

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

The PWM0 period can be set by writing period value to PWM0PRD register (R20) . Note that changing the PWM0PRD will immediately change the PWM0PRD values, which are different from PWM0DH/PWM0DL which has buffer to update the duty at the end of current period. The Programmer must pay attention to the current time to change PWMAPRD by observing the following figure. There is a digital comparator that compares the PWM0 counter and PWM0RD, if PWM0 counter is larger than PWM0PRD after setting the PWM0PRD, a fault long PWM cycle will be generated because PWM0 counter must count to overflow then keep counting to PWM0PRD to finish the cycle.



PWM0 Block Diagram



PWM0 8+2 Timing Diagram

Example:

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

◇ Example:

; Setup PWM0 clock prescaler

```
BSF     PWM0CKS     ; PWM0 clock source=FIRC 16 MHz
MOVLW   00010011B
MOVWF   F17         ; PWM0 prescaler/64

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

MOVLW   00000000B
MOVWR   PWM0DL      ; Set PWM0DL duty=00H

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

BSF     PWM0O1OE    ; Enable PWM0O1 output
```

Example:

PWM0 clock source=FIRC16 MHz, PWM0PSC=/64, PWMAPRD=80H,

PWM0DL=00H, PWM0DH=20H

PWM0 output frequency=16 MHz/64/ (PWMPROD+1) =16 MHz/64/129=1938 Hz.

PWM0P output duty=32:129=24.8%.

R21	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DH	PWM0DH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

R21.7~0 **PWM0DH:** PWM0 duty 8-bit MSB

R22	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DL	–	–	–	–	–	–	PWM0DL	
R/W	–	–	–	–	–	–	R/W	R/W
Reset	–	–	–	–	–	–	0	0

R22.1~0 **PWM0DL:** PWM0 duty 2-bit LSB

F17	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF17	PWM1DIS	PWM0DIS	PWM1CKS	PWM0CKS	PWM1PSC		PWM0PSC	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F17.6 **PWM0DIS:** PWM0 Clock Disable
0:Clock Enable 1:Clock Disable

F17.4 **PWM0CKS:** PWM0 Clock Source
0: Fsys 1:FIRC16M

F17.1~0 **PWM0PSC:** PWM1A/1B/1C Clock Source Prescaler
00: DIV1 01: DIV2 10:DIV4 11:DIV64

F1B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF1B	TCOE	TRCOE	TM1OE	PWM1COE	PWM1BOE	PWM1AOE	PWM0O2OE	PWM0O1OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F1B.1 **PWM0O2OE:** PWM0O2 Output Enable
0: Disable 1:Enable

F1B.0 **PWM0O1OE:** PWM0O1 Output Enable
0: Disable 1:Enable

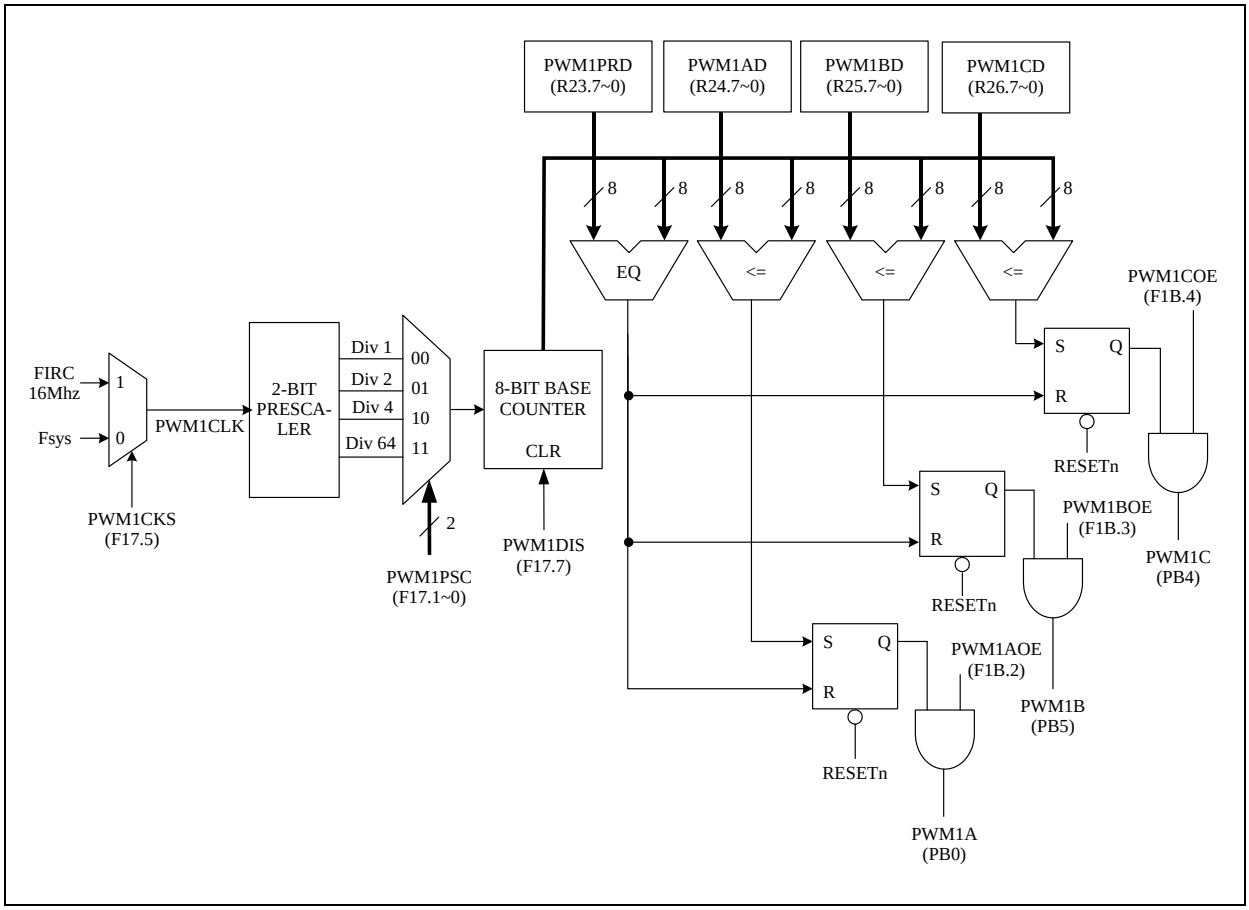
R20	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0PRD	PWM0PRD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

R20.7~0 **PWM0PRD:** PWM0 period data

7.6 PWM1A/PWM1B/PWM1C: 8 bits PWMs

PWM1A/PWM1B/PWM1C are 3 PWMs which have independent duty and common period. The PWMs can generate various frequency waveform with 256 duty resolution based on PWM1CLK, which can select Fsys or FIRC 16 MHz, decided by PWM1CKS (F17.5) . The advantage of higher PWM frequency is that the post RC filter can transform the PWM signal to more stable DC voltage level. The PWM output signal reset to low level whenever the 8-bit base counter matches the 8-bit of PWM duty register PWM1AD/PWM1BD/PWM1CD.

The PWM1A/1B/1C common period can be set by writing period value to PWM1PRD register (R23). Note that changing the PWM0PRD will immediately change the PWM1PRD values. The Programmer must pay attention to the current time to change PWM1PRD by observing the following figure. There is a digital comparator that compares the PWM1A/1B/1C counter and PWM1RD, if PWM1A/1B/C counter is larger than PWM1PRD after setting the PWM1PRD, a fault long PWM cycle will be generated because PWM1A/1B/1C counter must count to overflow then keep counting to PWM1PRD to finish the cycle.



PWM1 Block Diagram

R23	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1PRD	PWM1PRD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

R23.7~0 **PWM1PRD:** PWM1 period data

R24	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1AD	PWM1AD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

R24.7~0 **PWM1A:** PWM1A duty 8-bit

R25	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1BD	PWM1BD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

R25.7~0 **PWM1B:** PWM1B duty 8-bit

R26	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CD	PWM1CD							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

R26.7~0 **PWM1C:** PWM1C duty 8-bit

F17	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF17	PWM1DIS	PWM0DIS	PWM1CKS	PWM0CKS	PWM1PSC		PWM0PSC	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F17.7 **PWM1DIS:** PWM1A/1B/1C Clock Disable
0:Clock Enable 1:Clock Disable

F17.5 **PWM1CKS:** PWM1A/1B/1C Clock Source
0: Fsys 1:FIRC16M

F17.3~2 **PWM1PSC:** PWM1A/1B/1C Clock Source Prescaler
00: DIV1 01: DIV2 10:DIV4 11:DIV64

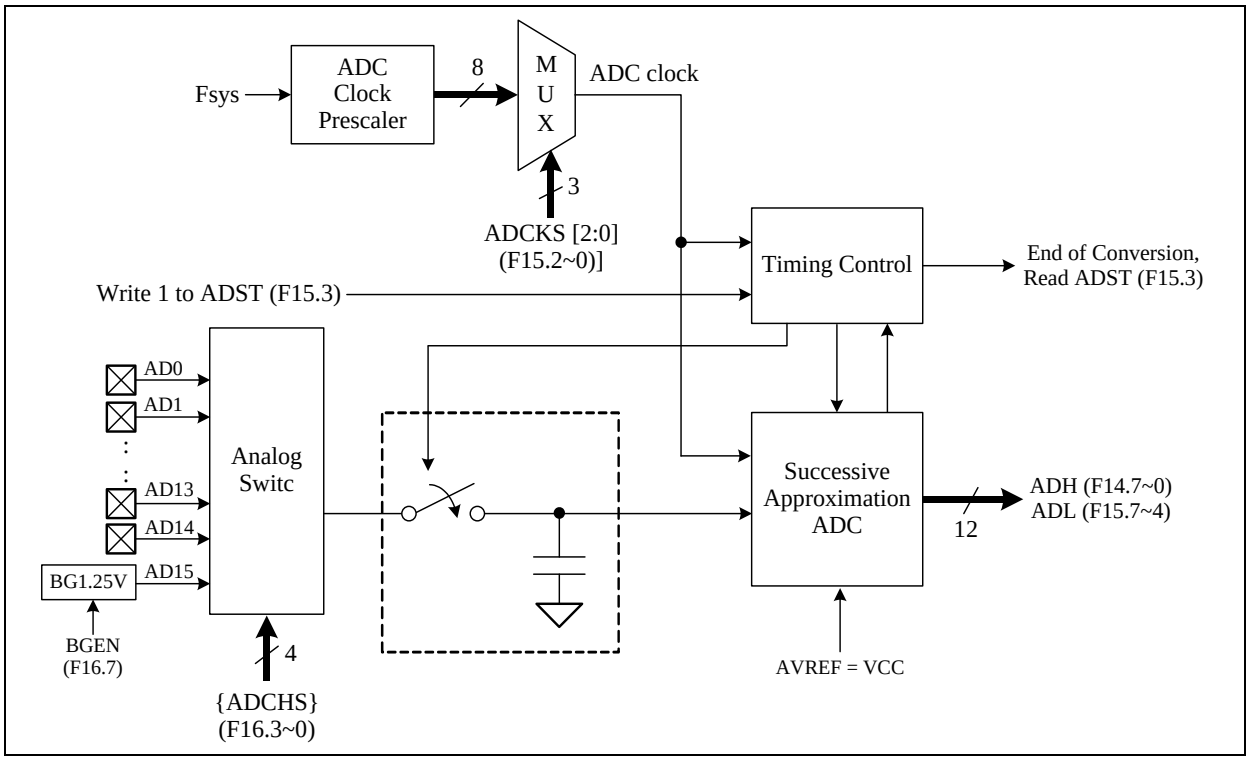
F1B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF1B	TCOE	TRCOE	TM1OE	PWM1COE	PWM1BOE	PWM1AOE	PWM0O2OE	PWM0O1OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F1B.4 **PWM1COE:** PWM1C Output Enable
0: Disable 1:Enable

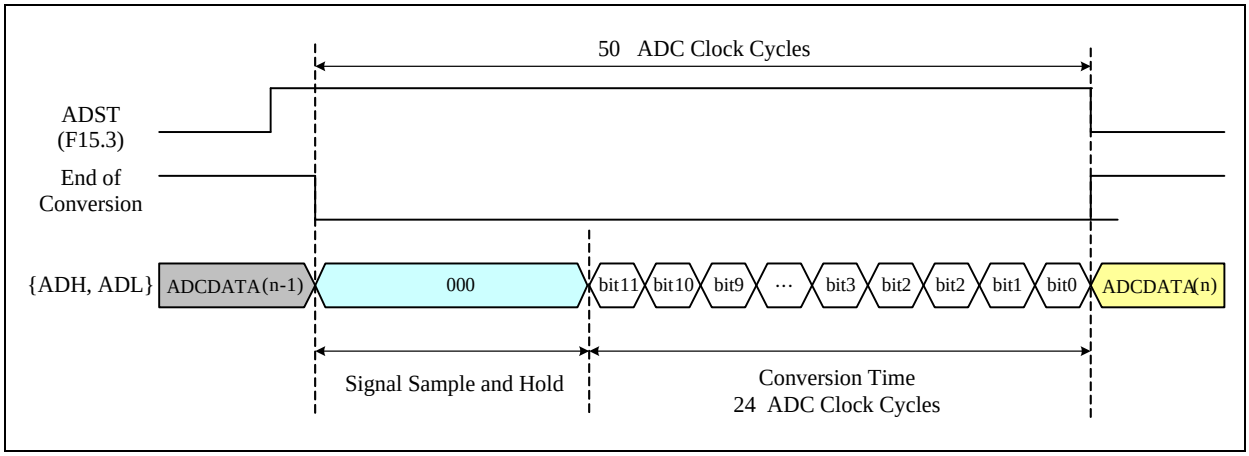
F1B.3 **PWMIBOE:** PWM1B Output Enable
0: Disable 1:Enable

F1B.2 **PWM1AOE:** PWM1A Output Enable
0: Disable 1:Enable

7.7 Analog-to-Digital Converter



The 12-bit ADC (Analog to Digital Converter) consists of a 16-channel analog input multiplexer, control register, clock generator, 12-bit successive approximation register, and output data register. To use the ADC, user needs to set ADCKS (F15.2~0) to choose a proper ADC clock frequency, which must be less than 1 MHz. User then launches the ADC conversion by setting the ADST (F15.3) control bit. After end of conversion, H/W automatic clears the ADST (F15.3) bit. User can poll this bit to know the conversion status. The PxMODE control registers are used for ADC pin configuration, user must set the Pin Mode=3 when the pin is used as an ADC input. The setting can disable the pin logical input path to save power consumption. User needs to set {ADCHS} (F16.3~0) to choose the input channel of ADC. One of them, AD15 is VBG1.25V input for ADC. ADC reference voltage is VCC.



Example:

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

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

◇ Example:

```
        MOVLW   00000101B       ; Fsys=4 MHz
        MOVWF   ADCTL           ; F15.2~0 (ADCKS) =ADC clock=Fsys/4=1 MHz

        MOVLW   01110101B       ; ADC2 (PA2) Pin Mode=3=ADC input
        MOVWR   PAMODL;

        MOVLW   01100010B
        MOVWF   MF16            ; F16.3~0 (ADCHS [3:0]) =2, ADC select ADC2 (PA2 pin).

        BSF     ADST            ; F15.3 (ADST) , ADC start conversion.

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

        MOVFW   ADH             ; F14.7~0, Read ADC result [11:4] into W
        MOVFW   ADCTL           ; F15.7~4, Read ADC result [3:0] into W

        :
        :
```

F14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADH	ADH							
R/W	R	R	R	R	R	R	R	R
Reset	–	–	–	–	–	–	–	–

F14.7~0 **ADH:** ADC output data MSB, ADQ [11:4]

F15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCTL	ADL				ADST	ADCKS		
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset	–	–	–	–	0	0	0	0

F15.7~4 **ADL:** ADC output data LSB, ADQ [3:0]

F15.3 **ADST:** ADC start bit.
0: H/W clear after end of conversion
1: ADC start conversion

F15.2~0 **ADCKS:** ADC clock frequency selection:
000: Fsys/256 100: Fsys/16
001: Fsys/128 101: Fsys/8
010: Fsys/64 110: Fsys/4
011: Fsys/32 111: Fsys/2

F16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF16	BGEN	LVR2SAV	LDOSAV	MODE3V	ADCHS			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	0	0	0	0	0

F16.4 **BGEN:** Band Gap BG1.25V enable
0: Disable
1: Enable and Auto disable in STOP/IDLE mode

F16.3~0 **ADCHS:** ADC channel select

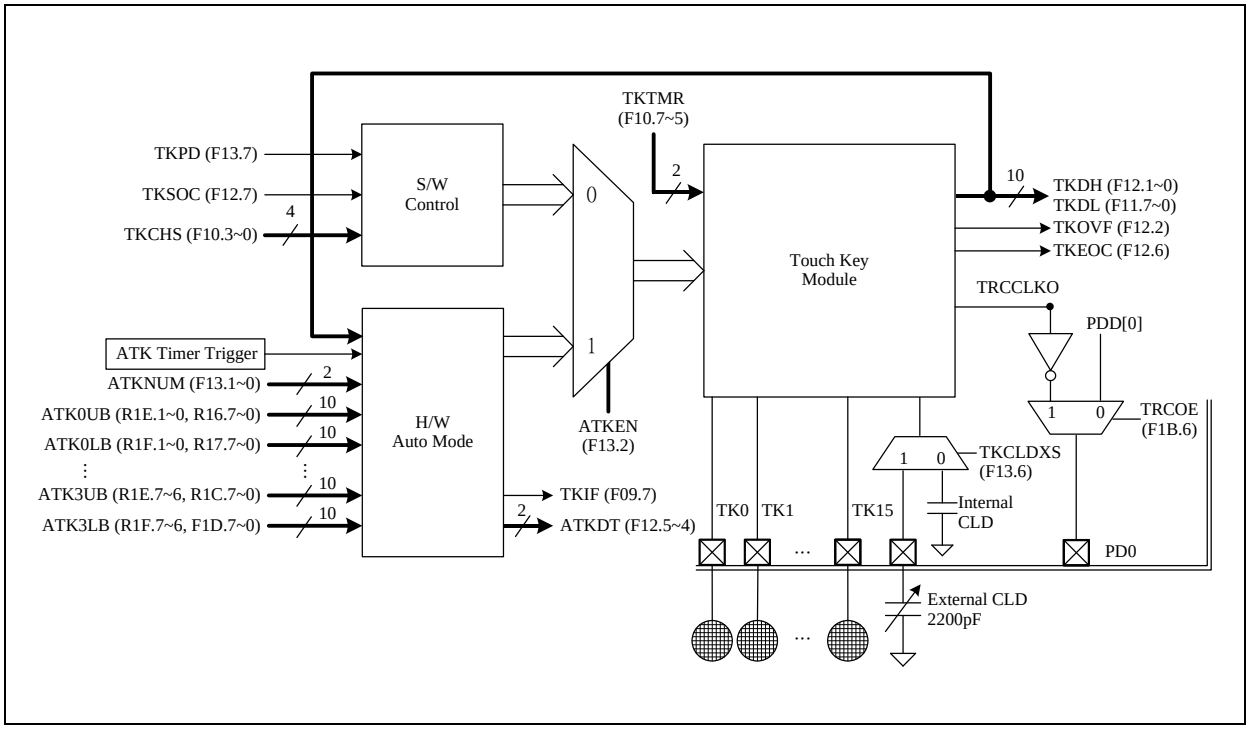
0000: ADC0 (PA6)	0110: ADC6 (PD6)	1100: ADC12 (PB3)
0001: ADC1 (PA1)	0111: ADC7 (PB0)	1101: ADC13 (PB6)
0010: ADC2 (PA2)	1000: ADC8 (PA0)	1110: ADC14 (PB7)
0011: ADC3 (PB1)	1001: ADC9 (PD5)	1111: VBG 1.25V
0100: ADC4 (PD7)	1010: ADC10 (PD4)	
0101: ADC5 (PA5)	1011: ADC11 (PB2)	

7.8 Touch Key

The Touch Key offers an easy, simple and reliable method to implement finger touch applications. In most applications, it only requires an capacitor component (CLD) , and the user can freely choose to use the built-in capacitor (Internal CLD) or an external capacitor component on CLD pin (External CLD) by TKCLDXS (F13.6) . The device support 16 channels touch key detection with S/W manual mode and 4 channels with H/W auto mode (ATK) . Only one mode can be active at a time.

To use the Touch Key, user must setup the Pin Mode 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, **TM57M5541** will disable the pull up resistor automatically. **TM57M5541** can also output the Touch Key clock to PD0 when TRCOE (F1B.6) is set, the negative Touch Key clock will output to PD0.

Pin Mode Setting for Touch Key	TK0~15
Pin is not Touch Key	Mode0, 1, 2, 3
Pin is Touch Key, Idling	Mode0
Pin is Touch Key, S/W Scanning	Mode0
Pin is Touch Key, H/W Auto Scan	Mode0



Touch Key Block Diagram

7.8.1 S/W Manual Mode Touch Key Detection

All Touch Key (TK0~TK15) can be used for S/W mode, it can be select by TKCHS (F10.3~0) bits. To start the S/W mode, user assigns ATKEN=0 (F13.2) and TKPD=0 (F13.7) , then set the TKSOC (F12.7) bit to start touch key conversion, the TKSOC bit will 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. After TKEOC's (F11.3) edge rising, user must wait at least 10 μs for next conversion. The touch key counting values is stored into the 10 bits touch key data count " TKDH (F12.1~0) , TKDL (F11.7~0) " . If TKOVF (F12.2) is set, it means the conversion transaction exceeds period time. Reduce/Increase TKTMR (F10.7~5) 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 TKREF (F10.4) =1 and start the S/W scan mode can get the TK Data Count 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=TK5 (PA5).

```
        MOVLW   xxxx00xxB       ; PAMODH [3:2] =00b
        MOVWR   PAMODH          ; Set PA5MOD as Mode0 for touch key input
        BSF     PAD, 5          ; Set PA5 is input with pull-up

        MOVLW   0 10 0 0101B    ; TKSOC=0
        MOVWF   TKCTL           ; TKTMR=2, TKCHS=5 (TK5)

        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 [1:0]
        MOVFW   TKDL            ; Read TKDL [7:0]
```

7.8.2 H/W Auto Mode Touch Key Detection

Only continuous 1 to 4 Touch Key are eligible for H/W auto mode. Setting ATKNUM (F13.1~0) can select the number from 1 to 4. Setting TKCHS (F10.3~0) can choose ATK0. For example, ATKNUM= 10B (ATK0~ATK2) , TKCHS=1001B (TK9) , then ATK0=TK9, ATK1=TK10, ATK2=TK11. This function can work in IDLE mode and save the S/W effort as well as minimize the chip current consumption. If ATKEN is set to "1", the internal ATK timer will generate an overflow flag after ATK timer 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 ATK0~ATK3's TK Data Count at every 30/60/120/240 ms rate by ATKSIT (F13.5~4) . If a keys' TK Data Count is less than the pre-set compare lower boundary (ATKnLB) or more than the pre-set compare upper boundary (ATKnUB) , H/W will record the compare result in the ATKDT (F12.5~4) , and H/W will also generate interrupt flag TKIF (F9.7) after touch key module ends conversion. It generates auto touch key interrupt and wake up CPU if the TKIE (F8.7) bit is set. User can switch the TK module to S/W Manual Mode after the TK interrupt and identify/confirm the Key touch event.

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	TKIE	TM3IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F08.7 **TKIE**: 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	TM3IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F09.7 **TKIF**: Touch Key interrupt event pending flag, set by H/W
ATKEN=1: Key's TK Data Count is over the pre-set compare threshold range
ATKEN=0: End of TK conversion

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

F10.7~5 **TKTMR**: Touch key conversion time
000: shortest
…
111: longest

F10.4 **TKREF**: Touch Key Channel Select REFC
1: Touch key channel select REFC (Internal reference capacitor)
0: Touch key channel select=TKCHS (Touch key channel depends on TKCHS)

F10.3~0 **TKCHS**: Touch key channel select / ATK0 select

0000: TK0 (PA6)	1000: TK8 (PD6)
0001: TK1 (PA1)	1001: TK9 (PB5)
0010: TK2 (PA2)	1010: TK10 (PB4)
0011: TK3 (PB1)	1011: TK11 (PE1)
0100: TK4 (PD7)	1100: TK12 (PE0)
0101: TK5 (PA5)	1101: TK13 (PB7)
0110: TK6 (PA0)	1110: TK14 (PB6)
0111: TK7 (PB0)	1111: TK15 (PB3)

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	TKOVF	TKDH	
R/W	R/W	R	R	R	R	R	R	
Reset	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

F12.5~4 **ATKDT**:Touch key auto scan result
00: ATK0 has a touch event
01: ATK1 has a touch event
10: ATK2 has a touch event
11: ATK3 has a touch event

F12.3 **ATKPOL**: Auto Touch Key trigger Polarity
0: Low Boundary
1: Up Boundary

F12.2 **TKOVF**: Touch key counter overflow flag
0: not overflow
1: overflow

F12.1~0 **TKDH**: Touch key data MSB [9:8]

F13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCTL2	TKPD	TKCLDXS	ATKSIT		–	ATKEN	ATKNUM	
R/W	R/W	R/W	R/W	R/W	–	R/W	R/W	R/W
Reset	1	0	0	0	–	0	1	1

F13.7 **TKPD**: Touch key power down
0: Touch key running
1: Touch key power down

F13.6 **TKCLDXS**: Touch Key CLD External Selection
0: Internal CLD
1: External CLD

F13.5~4 **ATKSIT:** Auto Touch Key Scan Interval Time, @3V
00: 30ms
01: 60ms
10 :120ms
11: 240ms

F13.2 **ATKEN**: Auto Touch Key Enable
0: Disable
1: Enable

F13.1~0 **ATKNUM**: Touch key auto scan channel number
00: ATK0
01: ATK0~ATK1
10: ATK0~ATK2
11: ATK0~ATK3

F1B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF1B	TCOE	TRCOE	TM1OE	PWM1COE	PWM1BOE	PWM1AOE	PWM0O2OE	PWM0O1OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F1B.6 **TRCOE:** TRCOUT Output Enable
0: Disable
1:Enable

R16	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

R16.7~0 **ATK0UBL:** Auto Touch Key ATK0 upper boundary LSB [7:0]

R17	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

R17.7~0 **ATK0LBL:** Auto Touch Key ATK0 lower boundary LSB [7:0]

R18	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

R18.7~0 **ATK1UBL:** Auto Touch Key ATK1 upper boundary LSB [7:0]

R19	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

R19.7~0 **ATK1LBL:** Auto Touch Key ATK1 lower boundary LSB [7:0]

R1A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATK2UBL	ATK2UBL							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

R1a.7~0 **ATK2UBL:** Auto Touch Key ATK2 upper boundary LSB [7:0]

R1B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATK2LBL	ATK2LBL							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

R1b.7~0 **ATK2LBL:** Auto Touch Key ATK2 lower boundary LSB [7:0]

R1C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATK3UBL	ATK3UBL							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

R1c.7~0 **ATK3UBL:** Auto Touch Key ATK3 upper boundary LSB [7:0]

R1D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATK3LBL	ATK3LBL							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

R1D.7~0 **ATK3LBL:** Auto Touch Key ATK3 lower boundary LSB [7:0]

R1E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATKUBH	ATK3UBH		ATK2UBH		ATK1UBH		ATK0UBH	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

R1E.7~6 **ATK3UBH:** Auto Touch Key ATK3 upper boundary MSB [9:8]
R1E.5~4 **ATK2UBH:** Auto Touch Key ATK2 upper boundary MSB [9:8]
R1E.3~2 **ATK1UBH:** Auto Touch Key ATK1 upper boundary MSB [9:8]
R1E.1~0 **ATK0UBH:** Auto Touch Key ATK0 upper boundary MSB [9:8]

R1F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATKLBH	ATK3LBH		ATK2LBH		ATK1LBH		ATK0LBH	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

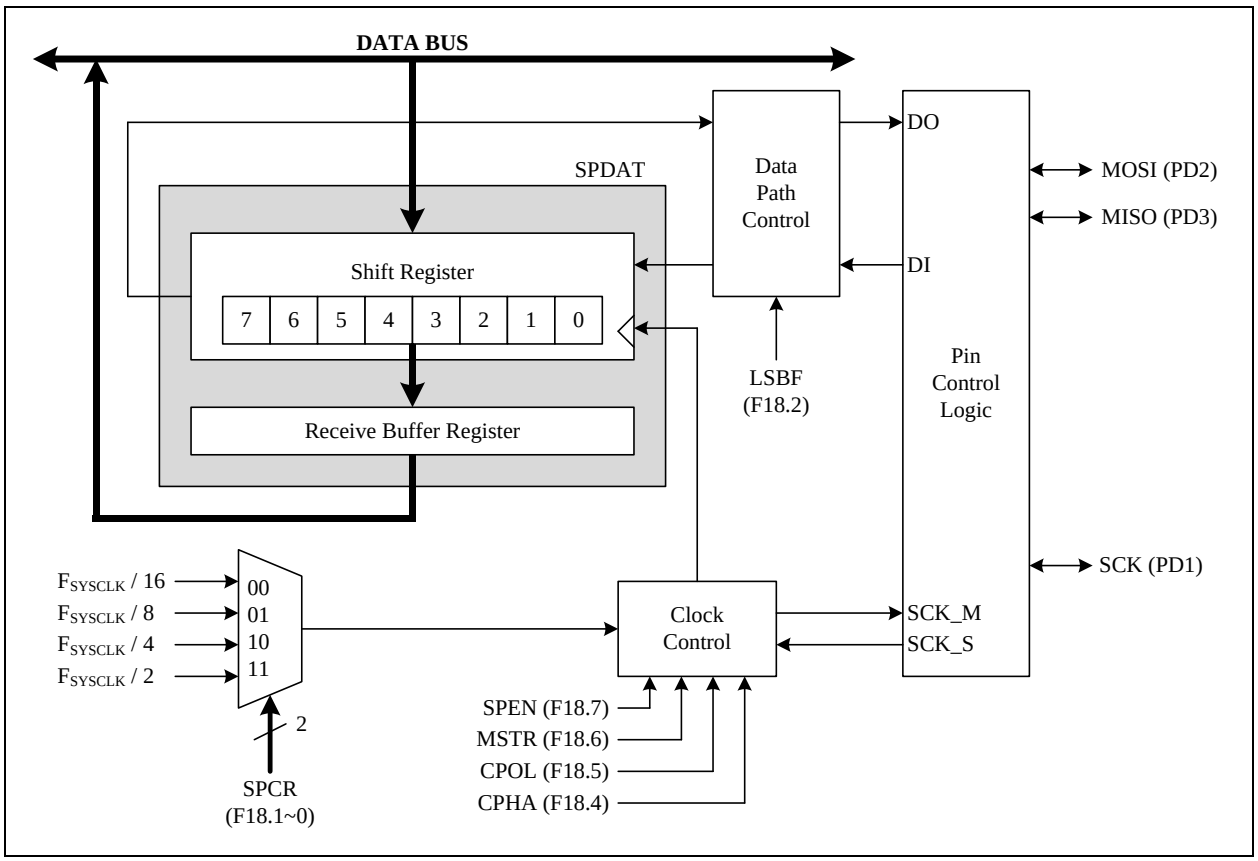
R1F.7~6 **ATK3LBH:** Auto Touch Key ATK3 lower boundary MSB [9:8]
R1F.5~4 **ATK2LBH:** Auto Touch Key ATK2 lower boundary MSB [9:8]
R1F.3~2 **ATK1LBH:** Auto Touch Key ATK1 lower boundary MSB [9:8]
R1F.1~0 **ATK0LBH:** Auto Touch Key ATK0 lower boundary MSB [9:8]

7.9 Serial Peripheral Interface (SPI)

The SPI module is capable of full-duplex, synchronous, serial communication between the MCU and peripheral devices. The peripheral devices can be other MCUs, A/D converter, sensors, or Flash memory, etc. The SPI runs at a baud rate up to the system clock divided by two. Firmware can read the status flags, or the operation can be interrupt driven.

The features of the SPI module include:

- Master or Slave mode operation
- 3-wire mode operation
- Full-duplex operation
- Programmable transmit bit rate
- Single Buffer receive
- Serial clock phase and polarity options
- MSB-first or LSB-first shifting selectable



SPI System Block Diagram

The MOSI (PD2) signal is an output when SPI is operating in Master mode and an input when SPI is operating in Slave mode. The MISO (PD3) signal is an input when SPI is operating in Master mode and an output when SPI is operating in Slave mode. Data is transferred MSB or LSB first by setting the LSBF bit. The SCK (PD1) signal is an output from a Master device and an input to Slave devices. It is used to synchronize the data on the MOSI and MISO lines of Master and Slave. SPI generates the signal with eight programmable clock rates in Master mode.

Master Mode

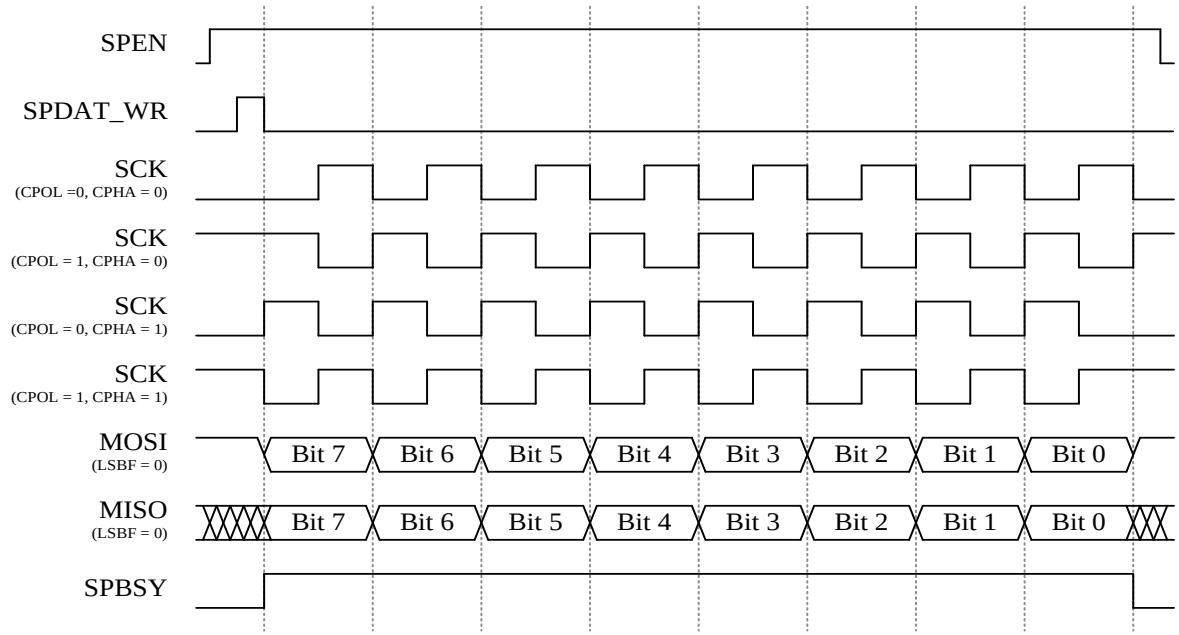
The SPI operates in Master mode by setting the MSTR bit in the SPCON. To start transmit, writing a data to the SPDAT. If SPBSY=0, the data will be transferred to the shift register and starts shift out on the MOSI line. The data of the Slave shift in from the MISO line at the same time. When the SPIF bit becomes set at the end of transfer, the receive data is written to receiver buffer and the RCVBF bit in the SPSTA is set. To prevent an overrun condition, software must read the SPDAT before next byte enters the shift register. The SPBSY bit will be set when writing a data to SPDAT to start transmit, and be cleared at the end of the eighth SCK period in Master mode.

Slave Mode

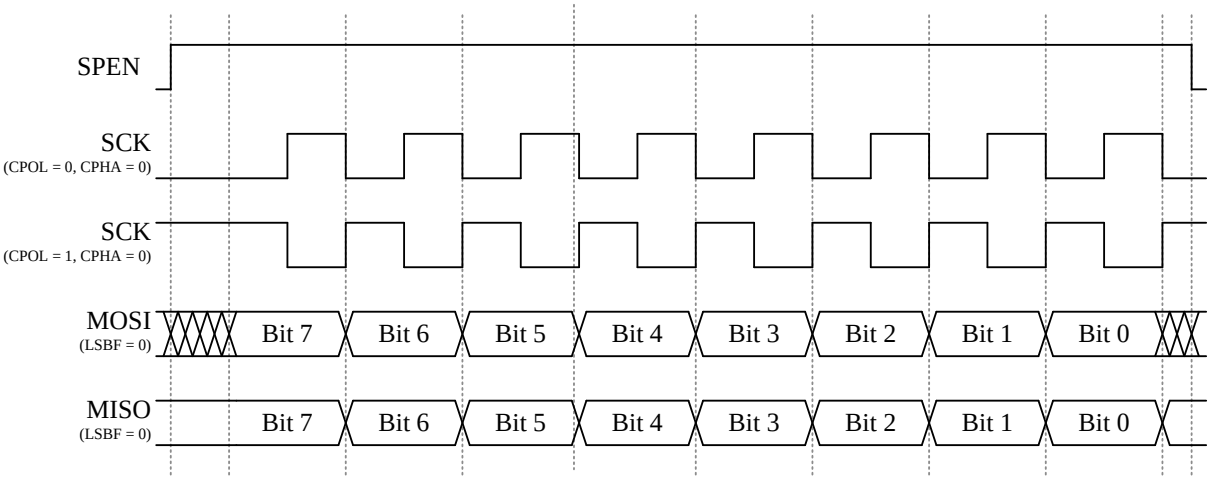
The SPI operates in Slave mode by clearing the MSTR bit in the SPCON. The transmission will start when the SPEN bit in the SPCON is set. The data from a Master will shift into the shift register through the MOSI line, and shift out from the shift register on the MISO line. When a byte enters the shift register, the data will be transferred to receiver buffer if RCVBF=0. If RCVBF=1, the newer received data will not be transferred to receiver buffer and the RCVOVF bit is set. After a byte enters the shift register, the SPIF and RCVBF bits are set. To prevent an overrun condition, software must read the SPDAT and write 0 to RCVBF before next byte enters the shift register. The maximum SCK frequency allowed in Slave mode is $F_{SYS}/4$.

Serial Clock

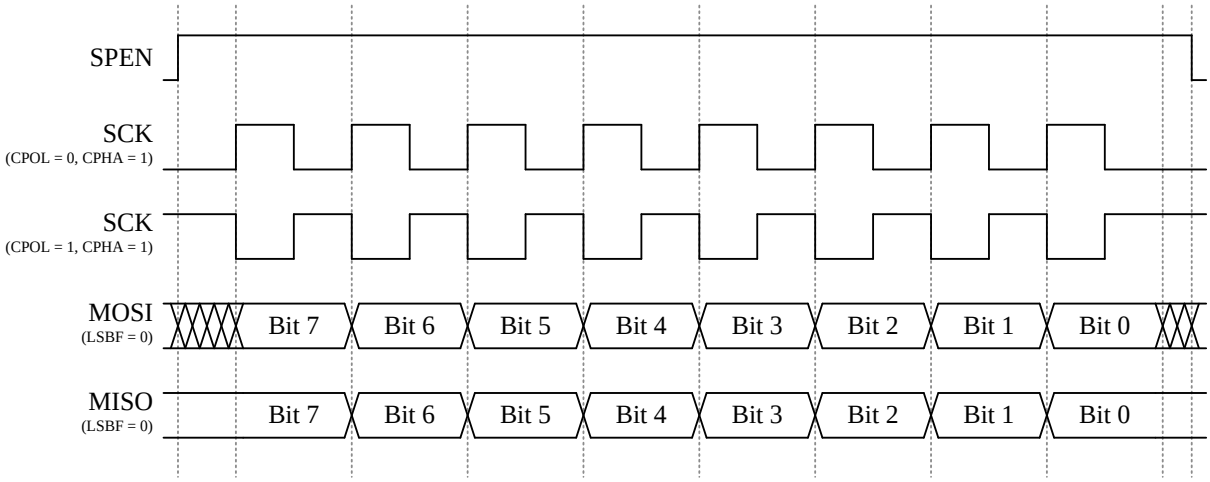
The SPI has four clock types by setting the CPOL and CPHA bits in the SPCON register. The CPOL bit defines the level of the SCK in SPI idle state. The level of the SCK in idle state is low when CPOL=0, and is high when CPOL=1. The CPHA bit defines the edges used to sample and shift data. The SPI sample data on the first edge of SCK period and shift data on the second edge of SCK period when CPHA=0. The SPI sample data on the second edge of SCK period and shift data on first edge of SCK period when CPHA=1. Figures below show the detail timing in Master and Slave modes. Both Master and Slave devices must be configured to use the same clock type before the SPEN bit is set. The SPCR controls the Master mode serial clock frequency. This register is ignored when operating in Slave mode. The SPI clock can select System clock divided by 2, 4, 8, or 16 in Master mode.



Master Mode Timing



Slave Mode Timing (CPHA=0)



Slave Mode Timing (CPHA=1)

In both Master and Slave modes, the SPIF interrupt flag is set by H/W at the end of a data transfer. If write data to SPDAT when SPBSY=1, the WCOL interrupt flag will be set by H/W. When this occurs, the data write to SPDAT will be ignored, and shift register will not be written.

F18	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPCON	SPEN	MSTR	CPOL	CPHA	–	LSBF	SPCR	
R/W	R/W	R/W	R/W	R/W	–	R/W	R/W	
Reset	0	0	0	0	–	0	0	0

F18.7 **SPEN:** SPI Enable.
0: SPI Disable
1: SPI Enable, PD1~3 are SPI functional pins.

F18.6 **MSTR:** Master Mode Enable.
0: Slave Mode
1: Master Mode

F18.5 **CPOL:** SPI Clock Polarity
0: SCK is low in idle state
1: SCK is high in idle state

F18.4 **CPHA:** SPI Clock Phase
0: Data sampled on first edge of SCK period
1: Data sampled on second edge of SCK period

F18.2 **LSBF:** LSB First.
0: MSB first
1: LSB first

F18.1~0 **SPCR:** SPI Clock Rate.
00: Fsys/2
01: Fsys/4
10: Fsys/8
11: Fsys/16

F19	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPSTA	SPIF	WCOL	–	RCVOVF	RCVBF	SPBSY	–	–
R/W	R/W	R/W	–	R/W	R/W	R	–	–
Reset	0	0	–	0	0	–	–	–

F19.7 **SPIF:** SPI Interrupt Flag
Set by H/W at the end of a data transfer. Cleared by H/W when interrupt is vectored into. Write 0 to this bit will clear this flag.

F19.6 **WCOL:** Write Collision Interrupt Flag
Set by H/W if write data to SPDAT when SPBSY=1. Write 0 to this bit or rewrite data to SPDAT when SPBSY=0 will clear this flag.

F19.4 **RCVOVF:** Receive Buffer Overrun Flag
Set by H/W at the end of a data transfer and RCVBF=1. Write 0 to this bit will clear this flag.

F19.3 **RCVBF:** Receive Buffer Full Flag
Set by H/W at the end of a data transfer. Write 0 to this bit will clear this flag.

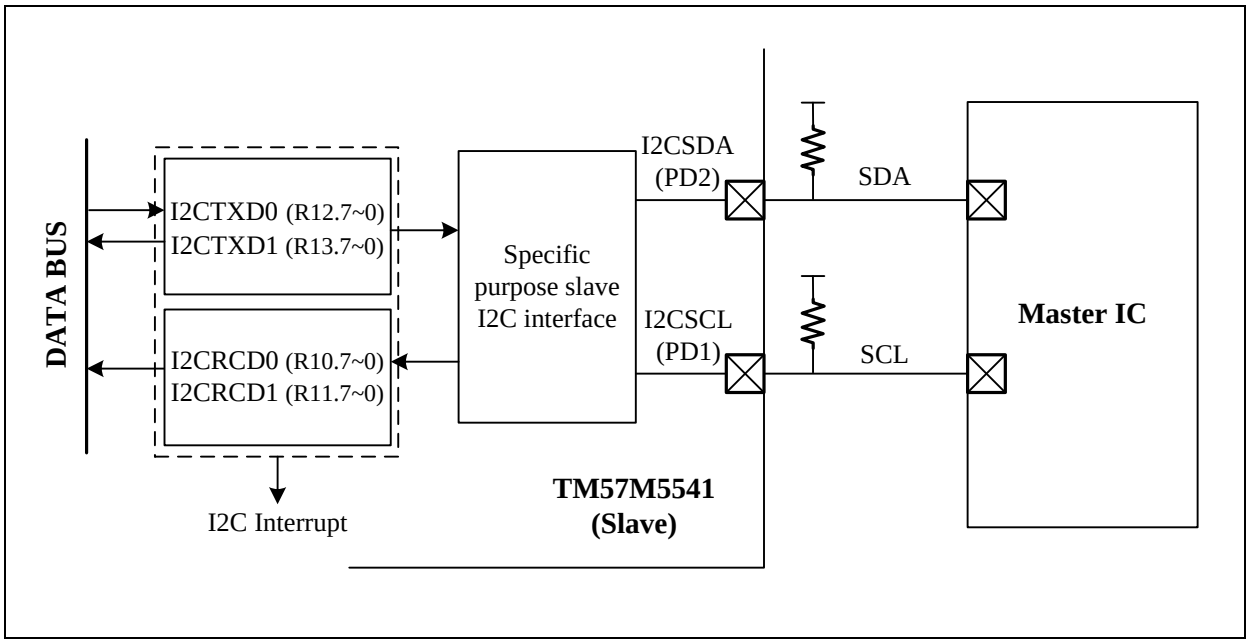
F19.2 **SPBSY:** SPI Busy Flag (Read Only)
Set by H/W when a SPI transfer is in progress.

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

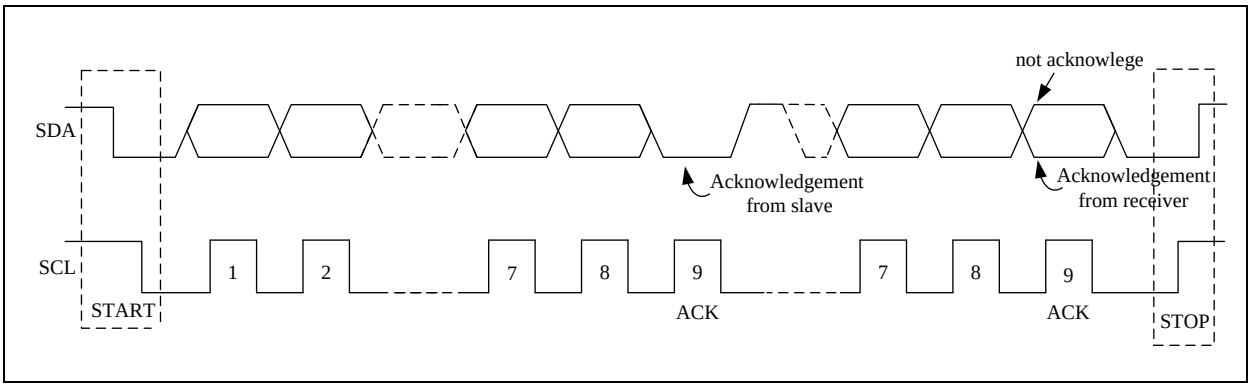
F1A.7~0 **SPDAT:** SPI Transmit and Receive Data
The SPDAT register is used to transmit and receive data. Writing data to SPDAT place the data into shift register and start a transfer when in Master mode. Reading SPDAT returns the contents of the receive buffer.

7.10 Specific Purpose Slave I2C Interface

Specific purpose slave I2C interface in TM57M5541 could be used for data transmission. This interface is based on a standard I2C (Inter-Integrated Circuit) , and This chip is always as a slave mode. When the master node (another IC or device) sends the correct ID through I2C, it can read data from the register I2CTXD0 (R12.7~0) and I2CTXD1 (R13.7~0) or write data to the register I2CRCD0 (R10.7~0) and I2CRCD1 (R11.7~0) .

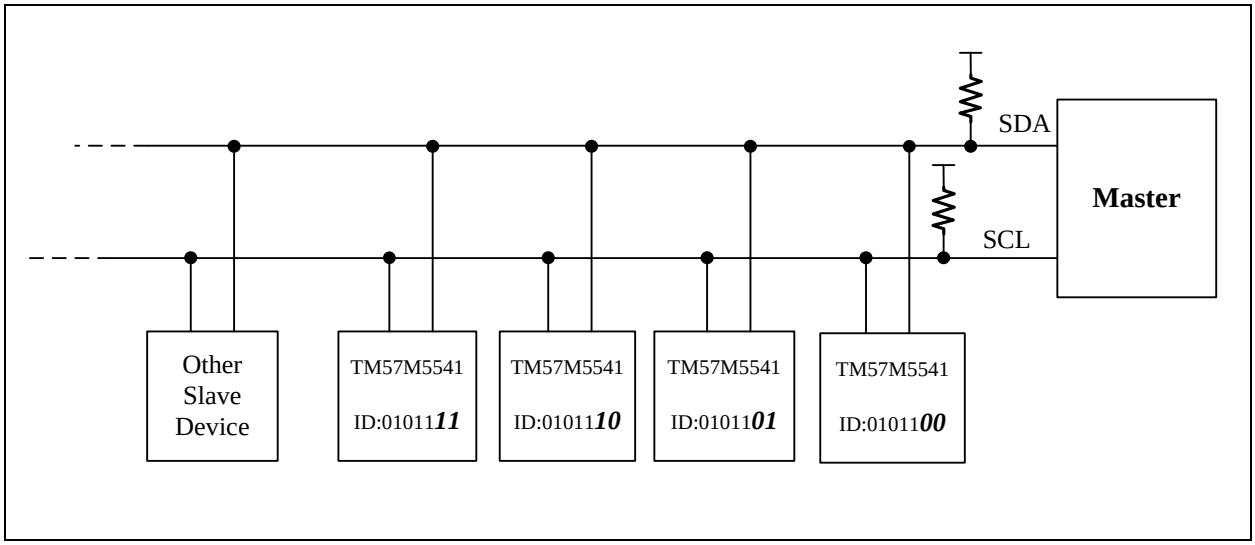


Slave I2C Interface Block Diagram

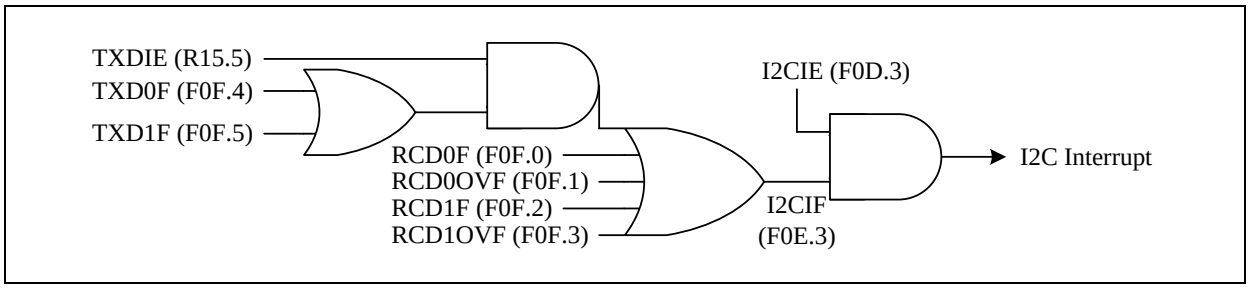


I2C Protocol

To use the slave I2C interface, the I2CEN (R14.6) bit has to be set. TM57M5541 supports 4 slave device IDs by setting I2CID (R14.4~3) . TM57M5541 can generate the transmitting flag TXD0F (F0F.4) and TXD1F (F0F.5) when data transmitting finished. It generates the receiving flag RCD0F (F0F.0) and RCD1F (F0F.2) when data receiving finished. It can also generate the receiving overflow flag RCD0OVF (F0F.1) and RCD1OVF (F0F.3) when data receiving finished but the receiving flag is not cleared. If one of those I2C flags is set, the I2C interrupt flag I2CIF (F0E.3) will be generated. It generates I2C interrupt if the I2CIE (F0D.3) bit is set. The transmitting interrupt can be disabled by setting TXDIE (R15.5) . Refer to the following table and figure.



I2C Parallel Connection Application Circuit



Slave I2C Interrupt Block Diagram

RCDxOVF	RCDxF	I2CIF	STATE
0	0	0	IDLE
0	1	1	Data received to I2CRCDx register
1	1	1	Data overflow occurred at I2CRCDx register

TXDIE	TXDxF	I2CIF	STATE
0	x	x	Disable the transmitting interrupt
1	0	0	IDLE
1	1	1	Data in I2CTXDx is transmitting finish

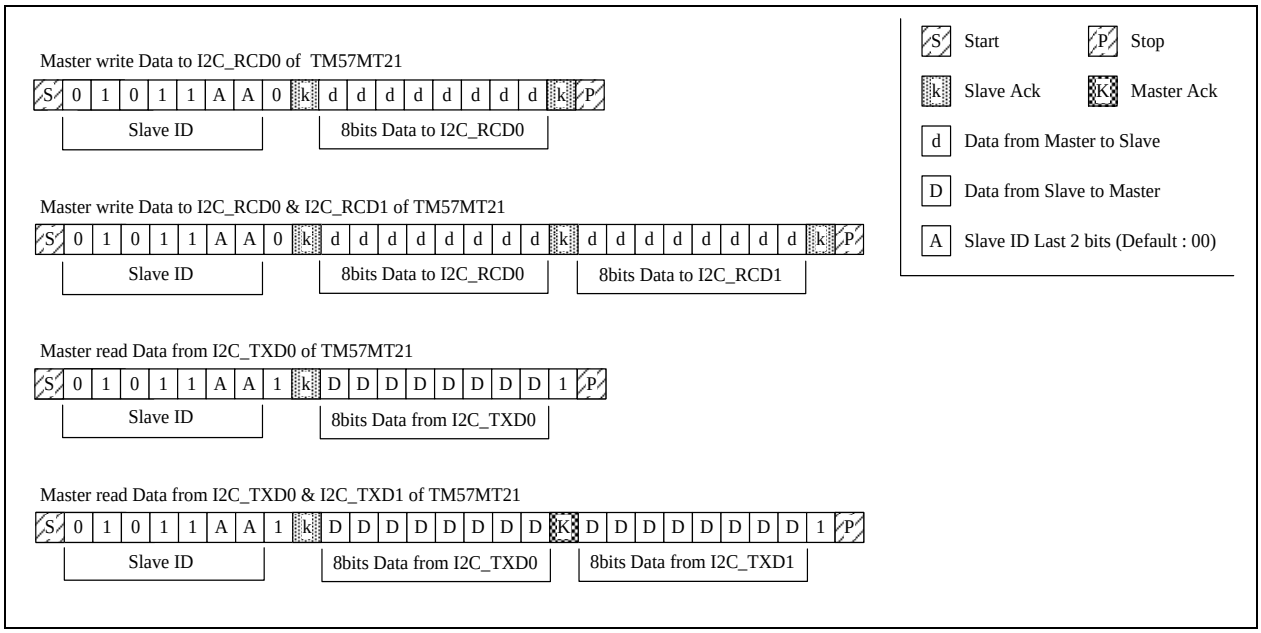


Table of TM57M5541 I2C Commands

F0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE1	–	–	LVDIE	ADCIE	I2CIE	SPIE	INT4IE	INT3IE
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	0	0	0	0

F08.3 **I2CIE**: I2C Receive/Transmit Data finished Interrupt Enable
0: disable
1: enable

F0E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF1	–	–	LVDIF	ADCIF	I2CIF	SPIF	INT4IF	INT3IF
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	0	0	0	0

F09.3 **I2CIF**: I2C interrupt event pending flag
This bit is set by H/W while
a. I2CRCD0 or I2CRCD1 receive data finished
b. I2CRCD0 or I2CRCD1 data overflow occurred
c. I2CTXD0 or I2CTXD1 data transmit finished
write 0 to this bit will clear this flag and slave I2C related flags

R10	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CRCD0	I2CRCD0							
R/W	R							
Reset	0	0	0	0	0	0	0	0

R10.7~0 **I2CRCD0:** The receiving register 0 of slave I2C

R11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CRCD1	I2CRCD1							
R/W	R							
Reset	0	0	0	0	0	0	0	0

R11.7~0 **I2CRCD1:** The receiving register 1 of slave I2C

R12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CTXD0	I2CTXD0							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

R12.7~0 **I2CTXD0:** The transmitting register 0 of slave I2C

R13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CTXD1	I2CTXD1							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

R13.7~0 **I2CTXD1:** The transmitting register 1 of slave I2C

R14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CCTL	–	I2CEN	TXDIE	I2CID		LVDEN	LVDSEL	
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	0	0	0	0	0	0	0

R14.6 **I2CEN:** Slave I2C interface enable
0: disable
1: enable

R14.5 **TXDIE**: Slave I2C transmitting interrupt enable
0: disable
1: enable

R14.4~3 **I2CID**: Slave I2C ID last 2 bits

F0F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CFLAG	–	–	TXD1F	TXD0F	RCD1OVF	RCD1F	RCD0OVF	RCD0F
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	0	0	0	0

F0F.5 **TXD1F**: Slave I2C transmitting data register 1 flag
This bit is set by H/W while I2CTXD1 data transmitting finished, write 0 to this bit will clear this flag

F0F.4 **TXD0F**: Slave I2C transmitting data register 0 flag
This bit is set by H/W while I2CTXD0 data transmitting finished, write 0 to this bit will clear this flag

F0F.3 **RCD1OVF**: Slave I2C receiving data register 1 overflow
This bit is set by H/W while receiving data to I2CRCD1 overflow, write 0 to this bit will clear this flag

F0F.2 **RCD1F**: Slave I2C receiving data register 1 flag
This bit is set by H/W while data receiving to I2CRCD1 finished, write 0 to this bit will clear this flag

F0F.1 **RCD0OVF**: Slave I2C receiving data register 0 overflow
This bit is set by H/W while receiving data to I2CRCD0 overflow, write 0 to this bit will clear this flag

F0F.0 **RCD0F**: Slave I2C receiving data register 0 flag
This bit is set by H/W while data receiving to I2CRCD0 finished, write 0 to this bit will clear this flag

MEMORY MAP

F-Plane

Name	Address	R/W	Rst	Description
(F00) INDF				**Function related to: RAM W/R**
INDF	00.7~0	R/W	-	Not a physical register, addressing INDF actually point to the register whose address is contained in the FSR register
(F01) TM0				**Function related to: Timer0**
TM0	01.7~0	R/W	0	Timer0 content
(F02) PCL				**Function related to: PROGRAM COUNT**
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
GB0	03.6	R/W	0	General purpose bit
RAMBK	03.5	R/W	0	FRAM Bank
TO	03.4	R	0	WDT timeout flag, cleared by PWRST, 'SLEEP' or 'CLRWDT' instruction
PD	03.3	R	0	Power down flag, set by 'SLEEP', cleared by 'CLRWDT' instruction
Z	03.2	R/W	0	Zero flag
DC	03.1	R/W	0	Decimal Carry flag
C	03.0	R/W	0	Carry flag
(F04) FSR				**Function related to: RAM W/R**
GB2	04.7	R/W	0	General purpose bit
FSR	04.6~0	R/W	-	File Select Register, indirect address mode pointer
(F05) PAD				**Function related to: Port A**
PAD	05.7~0	R	-	Port A pin or "data register" state
		W	FF	Port A output data register
(F06) PBD				**Function related to: Port B**
PBD	06.7~0	R	-	Port B pin or "data register" state
		W	FF	Port B output data register
(F07) PDD				**Function related to: Port D**
PDD	07.2~0	R	-	Port D pin or "data register" state
		W	FF	Port D output data register
(F08) INTIE				**Function related to: Interrupt Enable**
TKIE	08.7	R/W	0	Touch key interrupt enable, 1=enable, 0=disable
TM3IE	08.6	R/W	0	TM3 interrupt enable, 1=enable, 0=disable
TM1IE	08.5	R/W	0	Timer1 interrupt enable, 1=enable, 0=disable
TM0IE	08.4	R/W	0	Timer0 interrupt enable, 1=enable, 0=disable
WKTIE	08.3	R/W	0	Wakeup Timer interrupt enable, 1=enable, 0=disable Set 0 to clear & disable WKT timer
INT2IE	08.2	R/W	0	INT2 pin (PA7) interrupt enable, 1=enable, 0=disable
INT1IE	08.1	R/W	0	INT1 pin (PA1) interrupt enable, 1=enable, 0=disable
INT0IE	08.0	R/W	0	INT0 pin (PA6) interrupt enable, 1=enable, 0=disable

Name	Address	R/W	Rst	Description
(F09) INTIF				**Function related to: Interrupt Flag**
TKIF	09.7	R	-	Touch Key interrupt event pending flag, set by H/W ATKEN=1: Key's TK Data Count is over the pre-set compare threshold range ATKEN=0: end of TK conversion
		W	0	write 0: clear this flag; write 1: no action
TM3IF	09.6	R	-	TM3 interrupt event pending flag, set by H/W while Timer3 overflows
		W	0	write 0: clear this flag; write 1: no action
TM1IF	09.5	R	-	Timer1 interrupt event pending flag, set by H/W while Timer1 overflows
		W	0	write 0: clear this flag; write 1: no action
TM0IF	09.4	R	-	Timer0 interrupt event pending flag, set by H/W while Timer0 overflows
		W	0	write 0: clear this flag; write 1: no action
WKTIF	09.3	R	-	WKT interrupt event pending flag, set by H/W while WKT time out
		W	0	write 0: clear this flag; write 1: no action
INT2IF	09.2	R	-	INT2 (PA7) interrupt event pending flag, set by H/W at INT2 pin's falling edge
		W	0	write 0: clear this flag; write 1: no action
INT1IF	09.1	R	-	INT1 (PA1) interrupt event pending flag, set by H/W at INT1 pin's falling/rising edge
		W	0	write 0: clear this flag; write 1: no action
INT0IF	09.0	R	-	INT0 (PA6) interrupt event pending flag, set by H/W at INT0 pin's falling/rising edge
		W	0	write 0: clear this flag; write 1: no action
(F0A) PCH				**Function related to: PROGRAM COUNT**
PCH	0A.2~0	R/W	0	Programming Counter MSB [10~8]
(F0B) CLKCTL				**Function related to: Fsys**
SCKTYPE	0b.7	R/W	0	Slow-clock Type 0: SIRC, 1: SXT
FCKTYPE	0b.6	R/W	0	Fast-clock Type 0: FIRC, 1: FXT
-	0b.5	-	-	Reserved
SLOWSTP	0b.4	R/W	0	Stop Slow-clock in Stop Mode 0: no Stop 1: Stop
FASTSTP	0b.3	R/W	0	Stop Fast-clock 0:no Stop 1:Stop
CPUCKS	0b.2	R/W	0	Select Fast-clock 0: Fsys=Slow-clock 1: Fsys=Fast-clock
CPUPSC	0b.1~0	R/W	11	Fsys Prescaler, 0: div 16, 1: div 4, 2: div 2, 3: div 1

Name	Address	R/W	Rst	Description
(F0C) MF0C				**Function related to: TM0/TM1/TM3/Port E/Filter**
CLKFLT	0c.7	R/W	0	CLK Filte : Please keep the value of this register = 0
VCCFLT	0c.6	R/W	0	Power noise Filter 0: disable 1: enable
TM3SET	0c.5	R/W	0	After TM3SET=1, TM3 NEXT STEP=FFFFh
TM3STP	0c.4	R/W	0	Stop Timer3
TM1STP	0c.3	R/W	0	Stop Timer1
TM0STP	0c.2	R/W	0	Stop Timer0
PED	0c.1~0	R	-	Port E pin or “data register” state
		W	FF	Port E output data register
(F0D) INTIE1				**Function related to: Interrupt Enable**
LVDIE	0d.5	R/W	0	LVD Interrupt Enable 0: Disable 1: Enable
ADCIE	0d.4	R/W	0	ADC Interrupt Enable 0: Disable 1: Enable
I2CIE	0d.3	R/W	0	I2C Interrupt Enable 0: Disable 1: Enable
SPIE	0d.2	R/W	0	SPI Interrupt Enable 0: Disable 1: Enable
INT4IE	0d.1	R/W	0	XINT4 (PD2) Interrupt Enable 0: Disable 1: Enable
INT3IE	0d.0	R/W	0	XINT3 (PD1) Interrupt Enable 0: Disable 1: Enable
(F0E) INTIF1				**Function related to: Interrupt Flag**
LVDIF	0e.5	R	-	Low voltage detection interrupt flag, set by H/W while Vcc ≦ LVD
		W	0	write 0: clear this flag; write 1: no action
ADCIF	0e.4	R	-	ADC interrupt flag, set by H/W after end of ADC conversion
		W	0	write 0: clear this flag; write 1: no action
I2CIF	0e.3	R	-	I2C interrupt event pending flag, set by H/W while a. I2CRCD0 or I2CRCD1 receive new data finished b. I2CRDC0 or I2CRCD1 data overflow occurred c. I2CTXD0 or I2CTXD1 data transmit finished
		W	0	write 0: clear this flag; write 1: no action
-	0e.2	-	-	Reserved
INT4IF	0e.1	R	-	INT4 (PD2) interrupt event pending flag, set by H/W at INT4 pin’s falling/rising edge
		W	0	write 0: clear this flag; write 1: no action
INT3IF	0e.0	R	-	INT3 (PD1) interrupt event pending flag, set by H/W at INT3 pin’s falling/rising edge
		W	0	write 0: clear this flag; write 1: no action

Name	Address	R/W	Rst	Description
(F0F) I2CFLAG				**Function related to: Slave I2C**
TXD1F	0f.5	R	-	Slave I2C transmitting data register 1 flag, set by H/W while I2CTXD1 data transmitting finished
		W	0	0: clear this flag 1: no action
TXD0F	0f.4	R	-	Slave I2C transmitting data register 0 flag, set by H/W while I2CTXD0 data transmitting finished
		W	0	0: clear this flag 1: no action
RCD1OVF	0f.3	R	-	Slave I2C receiving data register 1 overflow, set by H/W while receiving data to I2CRCD1 overflow
		W	0	0: clear this flag 1: no action
RCD1F	0f.2	R	-	Slave I2C receiving data register 1 flag, set by H/W while data receiving to I2CRCD1 finished
		W	0	0: clear this flag 1: no action
RCD0OVF	0f.1	R	-	Slave I2C receiving data register 0 overflow, set by H/W while receiving data to I2CRCD0 overflow
		W	0	0: clear this flag 1: no action
RCD0F	0f.0	R	-	Slave I2C receiving data register 0 flag, set by H/W while data receiving to I2CRCD0 finished
		W	0	0: clear this flag 1: no action
(F10) TKCTL				**Function related to: Touch Key**
TKTMR	10.7~5	R/W	011	Touch key conversion time 000: shortest … 111: longest
TKREF	10.4	R/W	1	Touch Key Channel Select REFC
TKCHS	10.3~0	R/W	0	Touch key channel select (ATKEN=0) or ATK0 select (ATKEN=1) 0000: TK0 (PA6) 1000: TK8 (PD6) 0001: TK1 (PA1) 1001: TK9 (PB5) 0010: TK2 (PA2) 1010: TK10 (PB4) 0011: TK3 (PB1) 1011: TK11 (PE1) 0100: TK4 (PD7) 1100: TK12 (PE0) 0101: TK5 (PA5) 1101: TK13 (PB7) 0110: TK6 (PA0) 1110: TK14 (PB6) 0111: TK7 (PB0) 1111: TK15 (PB3)
(F11) TKDL				**Function related to: Touch Key**
TKDL	11.7~0	R	-	Touch key data LSB [7~0]

Name	Address	R/W	Rst	Description
(F12) TKDH				**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	-	Touch key end of conversion 0: conversion is in process 1: end of conversion
ATKDT	12.5~4	R	-	Touch key auto scan result 00: ATK0 has a touch event 01: ATK1 has a touch event 10: ATK2 has a touch event 11: ATK3 has a touch event
ATKPOL	12.3	R	-	Auto Touch Key trigger Polarity 0: Low Boundary, 1: Up Boundary
TKOVF	12.2	R	-	Touch key counter overflow flag 0: not overflow 1: overflow
TKDH	12.1~0	R	-	Touch key data MSB [9~8]
(F13)TKCTL2				**Function related to: Touch Key**
TKPD	13.7	R/W	1	Touch Key Power Down 0: TK Enable 1: TK Power Down
TKCLDXS	13.6	R/W	0	Touch Key CLD External Selection 0: Internal CLD 1: External CLD
ATKSIT	13.5~4	R/W	0	Auto Touch Key Scan Interval Time, @3V 00: 30ms 01: 60ms 10: 120ms 11: 240ms
-	13.3	-	-	Reserved
ATKEN	13.2	R/W	0	Auto Touch Key Enable 0: Disable 1: Enable
ATKNUM	13.1~0	R/W	11	Auto Touch Key Scan Channel Number 00: ATK0 01: ATK0~ATK1 10: ATK0~ATK2 11: ATK0~ATK3
(F14)ADH				**Function related to: ADC**
ADH	14.7~0	R	-	ADC output data MSB, ADQ [11:4]
(F15) ADCTL				**Function related to: ADC**
ADL	15.7~4	R	-	ADC output data LSB, ADQ [3:0]
ADST	15.3	R/W	0	ADC start bit. 0: H/W clear after end of conversion 1: ADC start conversion
ADCKS	15.2~0	R/W	0	ADC clock frequency selection: 000: Fsys/256 100: Fsys/16 001: Fsys/128 101: Fsys/8 010: Fsys/64 110: Fsys/4 011: Fsys/32 111: Fsys/2

Name	Address	R/W	Rst	Description
(F16) MF16				**Function related to: ADC/LDO/MODE3V/LVR/VBG**
BGEN	16.7	R/W	1	VBG1.25V 0: disable 1: Enable and auto disable in STOP/IDLE mode
LVR2SAV	16.6	R/W	1	1: LVR2 (2.3V/2.9V) auto power off in STOP/IDLE mode
LDOSAV	16.5	R/W	1	1: LDO auto power off in STOP/IDLE mode
MODE3V	16.4	R/W	0	LDO power down 1: 3V mode (Vcc < 3.6V) 0: 5Vmode (Vcc > 3.6V)
ADCHS	16.3~0	R/W	0	**ADCHS:** ADC channel select 0000: ADC0 (PA6) 0110: ADC6 (PD6) 1100: ADC12(PB3) 0001: ADC1 (PA1) 0111: ADC7 (PB0) 1101: ADC13(PB6) 0010: ADC2 (PA2) 1000: ADC8(PA0) 1110: ADC14(PB7) 0011: ADC3 (PB1) 1001: ADC9(PD5) 1111: VBG 1.25V 0100: ADC4 (PD7) 1010: ADC10(PD4) 0101: ADC5 (PA5) 1011: ADC11(PB2)
(F17) PWMCTL				**Function related to: PWM**
PWM1DIS	17.7	R/W	0	PWM1A/1B/1C Clock Disable 0: Clock Enable 1: Clock Disable
PWM0DIS	17.6	R/W	0	PWM0 Clock Disable 0: Clock Enable 1: Clock Disable
PWM1CKS	17.5	R/W	0	PWM1A/1B/1C Clock Source 0: Fsys 1: FIRC16M
PWM0CKS	17.4	R/W	0	PWM0 Clock Source 0: Fsys 1: FIRC16M
PWM1PSC	17.3~2	R/W	0	PWM1A/1B/1C Clock Source Prescaler 00: DIV1 01: DIV2 10: DIV4 11: DIV64
PWM0PSC	17.1~0	R/W	0	PWM0 Clock Source Prescaler 00: DIV1 01: DIV2 10: DIV4 11: DIV64
(F18) SPCON				**Function related to: SPI**
SPEN	18.7	R/W	0	SPI enable 0: SPI disable 1: SPI enable
MSTR	18.6	R/W	0	Master mode enable 0: Slave mode 1: Master mode
CPOL	18.5	R/W	0	SPI clock polarity 0: SCK is low in idle state 1: SCK is high in idle state
CPHA	18.4	R/W	0	SPI clock phase 0: Data sample on first edge of SCK period 1: Data sample on second edge of SCK period
-	18.3	-	-	Reserved
LSBF	18.2	R/W	0	LSB first 0: MSB first 1: LSB first
SPCR	18.1~0	R/W	0	SPI clock rate 00: Fsys/2 01: Fsys/4 10: Fsys/8 11: Fsys/16

Name	Address	R/W	Rst	Description
(F19) SPSTA				**Function related to: SPI**
SPIF	19.7	R/W	0	SPI interrupt flag This is set by H/W at the end of a data transfer. Cleared by H/W when an interrupt is vectored into. Writing 0 to this bit will clear this flag.
WCOL	19.6	R/W	0	Write collision interrupt flag Set by H/W if write data to SPDAT when SPBSY is set. Write 0 to this bit or rewrite data to SPDAT when SPBSY is cleared will clear this flag.
-	19.5	-	-	Reserved
RCVOVF	19.4	R/W	0	Received buffer overrun flag Set by H/W at the end of a data transfer and RCVBF is set. Write 0 to this bit will clear this flag.
RCVBF	19.3	R/W	0	Receive buffer full flag Set by H/W at the end of a data transfer. Write 0 to this bit will clear this flag.
SPIBSY	19.2	R	0	SPI busy flag Set by H/W when a SPI transfer is in progress.
-	19.1~0			Reserved
(F1A) SPDAT				**Function related to: SPI**
SPDAT	1a.7~0	R/W	0	SPI transmit and receive data The SPDAT register is used to transmit and receive data. Writing data to SPDAT place the data into shift register and start a transfer when in master mode. Reading SPDAT returns the contents of the receive buffer.
(F1B) MF1B				**Function related to: TCOUT/Timer1/Touch Key/PWM**
TCOE	1b.7	R/W	0	TCOUT (PD6) Output Enable 0: Disable 1: Enable
TRCOE	1b.6	R/W	0	TRCOUT(PD0) Output Enable 0: Disable 1: Enable
TM1OE	1b.5	R/W	0	TM1OUT (PD0) Output Enable 0: Disable 1: Enable
PWM1COE	1b.4	R/W	0	PWM1COE (PB4) Output Enable 0: Disable 1: Enable
PWM1BOE	1b.3	R/W	0	PWM1BOE (PB5) Output Enable 0: Disable 1: Enable
PWM1AOE	1b.2	R/W	0	PWM1AOE (PB0) Output Enable 0: Disable 1: Enable
PWM0O2OE	1b.1	R/W	0	PWM0O2 (PD4) Output Enable 0: Disable 1: Enable
PWM0O1OE	1b.0	R/W	0	PWM0O1 (PA0) Output Enable 0: Disable 1: Enable
(F1C) RSR				**Function related to: RRAM W/R**
RSR	1c.7~0	R/W	0	R-Plane File Select Register
(F1D) DPL				**Function related to: Table Read**
DPL	1d.7~0	R/W	0	Table read low address, data ROM pointer (DPTR) low byte

Name	Address	R/W	Rst	Description
(F1E) DPH				**Function related to: Table Read**
DPH	1e.3~0	R/W	0	Table read high address, data ROM pointer (DPTR) high byte
(F1F) IRCF				**Function related to: Internal RC**
IRCF	1f.4~0	R/W		FIRC frequency adjustment: 00H: Lowest frequency 1FH: Highest frequency
User Data Memory				
FRAM	20~27	R/W	-	RAM common area (8 Bytes)
	28~7F	R/W	-	RAM Bank0 area (RAMBK=0, 88 bytes)
	28~7F	R/W	-	RAM Bank1 area (RAMBK=1, 88 bytes)

R-Plane

Name	Address	R/W	Rst	Description
(R00) INDR				**Function related to: RRAM R/W**
INDR	00.7~0	R/W	-	Not a physical register, addressing INDR actually point to the register whose address is contained in the RSR register
(R01) TM0RLD				**Function related to: TM0**
TM0RLD	01.7~0	R/W	0	Timer0 reload Data
(R02) TM0CTL				**Function related to: TM0**
	02.7~6			
TM0EDG	02.5	R/W	0	Timer0 prescaler counting edge for TM0CKI pin 0: rising edge 1: falling edge
TM0CKS	02.4	R/W	0	Timer0 prescaler clock source 0: Instruction cycle 1: TM0CKI pin (PA2 pin)
TM0PSC	02.3~0	R/W	0	Timer0 prescaler. Timer0 prescaler clock source divided by 0000: /1 0101: /32 0001: /2 0110: /64 0010: /4 0111: /128 0011: /8 1xxx: /256 0100: /16
(R03) PWRDN				**Function related to: Power Down**
PWRDN	03	W	-	write this register to enter STOP/IDLE Mode (i.e. 'SLEEP' instruction)
(R04) WDTCLR				**Function related to: WDT**
WDTCLR	04	W	-	write this register to clear WDT timer (i.e. 'CLRWDT' instruction)
(R05) PAMODH				**Function related to: Port A**
PA7MOD	05.6	R/W	0	PA7 I/O mode control 0: with pull-up (PAD [7] =1) 1: without pull-up
PA6MOD	05.5~4	R/W	01	PA6~PA4 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PA5MOD	05.3~2	R/W	01	
PA4MOD	05.1~0	R/W	01	
(R06) PAMODL				**Function related to: Port A**
PA3MOD	06.7~6	R/W	01	PA3~PA0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PA2MOD	06.5~4	R/W	01	
PA1MOD	06.3~2	R/W	01	
PA0MOD	06.1~0	R/W	01	
(R07) PBMODH				**Function related to: Port B**
PB7MOD	07.7~6	R/W	01	PB7~PB4 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PB6MOD	07.5~4	R/W	01	
PB5MOD	07.3~2	R/W	01	
PB4MOD	07.1~0	R/W	01	
(R08) PBMODL				**Function related to: Port B**
PB3MOD	08.7~6	R/W	01	PB3~PB0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PB2MOD	08.5~4	R/W	01	
PB1MOD	08.3~2	R/W	01	
PB0MOD	08.1~0	R/W	01	

Name	Address	R/W	Rst	Description
(R09) PDMODH				**Function related to: Port D**
PD7MOD	09.7~6	R/W	01	PD7~PD4 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PD6MOD	09.5~4	R/W	01	
PD5MOD	09.3~2	R/W	01	
PD4MOD	09.1~0	R/W	01	
(R0A) PDMODL				**Function related to: Port D**
PD3MOD	0a.7~6	R/W	01	PD3~PD0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PD2MOD	0a.5~4	R/W	01	
PD1MOD	0a.3~2	R/W	01	
PD0MOD	0a.1~0	R/W	01	
(R0B) MR0B				**Function related to: STATUS/INT0/WDT/WKT**
HWAUTO	0b.7	R/W	0	Save/Restore STATUS w/o TO, PD 0:disable 1: Enable
INT0EDG	0b.6	R/W	0	INT0 pin (PA6) edge interrupt event 0: falling edge to trigger 1: rising edge to trigger
	0b.5~4		-	Reserved
WDTPSC	0b.3~2	R/W	11	WDT pre-scale selections: 00: 128mS 01: 256mS 10: 1024mS 11: 2048mS
WKTPSC	0b.1~0	R/W	11	WKT pre-scale selections: 00: 16mS 01: 32mS 10: 64mS 11: 128mS
(R0C) PEMODL				**Function related to: Port E**
	0c.7~4	R/W	-	
PE1MOD	0c.3~2	R/W	01	PE1~PE0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PE0MOD	0c.1~0	R/W	01	
(R10) I2CRCD0				**Function related to: Slave I2C**
I2CRCD0	10.7~0	R	0	The receiving register 0 of slave I2C
(R11) I2CRCD1				**Function related to: Slave I2C**
I2CRCD1	11.7~0	R	0	The receiving register 1 of slave I2C
(R12) I2CTXD0				**Function related to: Slave I2C**
I2CTXD0	12.7~0	R/W	0	The transmitting register 0 of slave I2C
(R13) I2CTXD1				**Function related to: Slave I2C**
I2CTXD1	13.7~0	R/W	0	The transmitting register 1 of slave I2C

Name	Address	R/W	Rst	Description
(R14) MR14				**Function related to: I2C/LVD**
	14.7			
I2CEN	14.6	R/W	0	Slave I2C interface enable 0: disable 1: enable
TXDIE	14.5	R/W	0	Slave I2C transmitting interrupt enable 0: disable 1: enable
I2CID	14.4~3	R/W	0	Slave I2C ID last 2 bits
LVDEN	14.2	R/W	0	LVD Enable 0: disable 1: enable
LVDSEL	14.1~0	R/W	0	LVD Voltage Select 00: 2.2V 01: 2.5V 10: 3.1V 11: 4.5V
(R15) MR15				**Function related to: INT1/INT3/INT4**
INT4EDG	15.2	R/W	0	INT4 pin (PD2) edge interrupt event 0: falling edge to trigger 1: rising edge to trigger
INT3EDG	15.1	R/W	0	INT3 pin (PD1) edge interrupt event 0: falling edge to trigger 1: rising edge to trigger
INT1EDG	15.0	R/W	0	INT1 pin (PA1) edge interrupt event 0: falling edge to trigger 1: rising edge to trigger
(R16) ATK0UBL				**Function related to: Touch Key**
ATK0UBL	16.7~0	R/W	FF	Auto Touch Key ATK0 upper boundary LSB [7:0]
(R17) ATK0LBL				**Function related to: Touch Key**
ATK0LBL	17.7~0	R/W	00	Auto Touch Key ATK0 lower boundary LSB [7:0]
(R18) ATK1UBL				**Function related to: Touch Key**
ATK1UBL	18.7~0	R/W	FF	Auto Touch Key ATK1 upper boundary LSB [7:0]
(R19) ATK1LBL				**Function related to: Touch Key**
ATK1LBL	19.7~0	R/W	00	Auto Touch Key ATK1 lower boundary LSB [7:0]
(R1A) ATK2UBL				**Function related to: Touch Key**
ATK2UBL	1a.7~0	R/W	FF	Auto Touch Key ATK2 upper boundary LSB [7:0]
(R1B) ATK2LBL				**Function related to: Touch Key**
ATK2LBL	1b.7~0	R/W	00	Auto Touch Key ATK2 lower boundary LSB [7:0]
(R1C) ATK3UBL				**Function related to: Touch Key**
ATK3UBL	1c.7~0	R/W	FF	Auto Touch Key ATK3 upper boundary LSB [7:0]
(R1D) ATK3LBL				**Function related to: Touch Key**
ATK3LBL	1d.7~0	R/W	00	Auto Touch Key ATK3 lower boundary LSB [7:0]
(R1E) ATKUBH				**Function related to: Touch Key**
ATK3UBH	1e.7~6	R/W	11	Auto Touch Key ATK3 upper boundary MSB [9:8]
ATK2UBH	1e.5~4	R/W	11	Auto Touch Key ATK2 upper boundary MSB [9:8]
ATK1UBH	1e.3~2	R/W	11	Auto Touch Key ATK1 upper boundary MSB [9:8]
ATK0UBH	1e.1~0	R/W	11	Auto Touch Key ATK0 upper boundary MSB [9:8]

Name	Address	R/W	Rst	Description
(R1F) ATKLBH				**Function related to: Touch Key**
ATK3LBH	1f.7~6	R/W	0	Auto Touch Key ATK3 lower boundary MSB [9:8]
ATK2LBH	1f.5~4	R/W	0	Auto Touch Key ATK2 lower boundary MSB [9:8]
ATK1LBH	1f.3~2	R/W	0	Auto Touch Key ATK1 lower boundary MSB [9:8]
ATK0LBH	1f.1~0	R/W	0	Auto Touch Key ATK0 lower boundary MSB [9:8]
(R20) PWM0PRD				**Function related to: PWM0**
PWM0PRD	20.7~0	R/W	FF	PWM0 period data
(R21) PWM0DH				**Function related to: PWM0**
PWM0DH	21.7~0	R/W	0	PWM0 Duty MSB 8bit
(R22) PWM0DL				**Function related to: PWM0**
PWM0DL	22.1~0	R/W	0	PWM0 Duty LSB 2bit
(R23) PWM1PRD				**Function related to: PWM1**
PWM1PRD	23.7~0	R/W	FF	PWM1 (PWM1A/PWM1B/PWM1C) period data
(R24) PWM1AD				**Function related to: PWM1A**
PWM1AD	24.7~0	R/W	0	PWM1A Duty
(R25) PWM1BD				**Function related to: PWM1B**
PWM1BD	25.7~0	R/W	0	PWM1B Duty
(R26) PWM1CD				**Function related to: PWM1C**
PWM1CD	26.7~0	R/W	0	PWM1C Duty
(R28) TM1				**Function related to: Timer1**
TM1	28.7~0	R/W	0	Timer1 content
(R29) TM1RLD				**Function related to: Timer1**
TM1RLD	29.7~0	R/W	0	Timer1 reload Data
(R2A) TM1PSC				**Function related to: Timer1**
TM1PSC	2a.3~0	R/W	0	Timer1 prescaler. Timer1 clock source 0000: Fsys/2 0001: Fsys/4 0010: Fsys/8 0011: Fsys/16 0100: Fsys/32 0101: Fsys/64 0110: Fsys/128 0111: Fsys/256 1xxx: Fsys/512
(R2B) TM3H				**Function related to: Timer3**
TM3H	2b.7~0	R	0	Timer3 content MSB [15:8]
(R2C) TM3L0				**Function related to: Timer3**
TM3L	2c.7~0	R	0	Timer3 content LSB [7:0]
(R2D) TM3RLDH				**Function related to: Timer3**
TM3RLDH	2d.7~0	R/W	0	Timer3 reload Data MSB [15:8]
(R2E) TM3RLDL				**Function related to: Timer3**
TM3RLDL	2e.7~0	R/W	0	Timer3 reload Data LSB [7:0]

Name	Address	R/W	Rst	Description
(R2F) TM3CTL				**Function related to: Timer3**
TM3CKS	2f.4	R/W	0	Timer3 Clock select 0: Fsys/2 1: Slow-clock
TM3PSC	2f.3~0	R/W	0	Timer3 prescaler. Timer3 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
User Data Memory				
RRAM	40~ff	R/W	-	RAM common area (192 bytes)

INSTRUCTION SET

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

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

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

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

Mnemonic		Op Code	Cycle	Flag Affect	Description
Byte-Oriented File Register Instruction					
ADDWF	f, d	00 0111 dfff ffff	1	C, DC, Z	Add W and "f"
ANDWF	f, d	00 0101 dfff ffff	1	Z	AND W with "f"
CLRF	F	00 0001 1fff ffff	1	Z	Clear "f"
CLRW		00 0001 0100 0000	1	Z	Clear W
COMF	f, d	00 1001 dfff ffff	1	Z	Complement "f"
DECF	f, d	00 0011 dfff ffff	1	Z	Decrement "f"
DECFSZ	f, d	00 1011 dfff ffff	1 or 2	-	Decrement "f", skip if zero
INCF	f, d	00 1010 dfff ffff	1	Z	Increment "f"
INCFSZ	f, d	00 1111 dfff ffff	1 or 2	-	Increment "f", skip if zero
IORWF	f, d	00 0100 dfff ffff	1	Z	OR W with "f"
MOVFW	f	00 1000 0fff ffff	1	-	Move "f" to W
MOVWF	f	00 0000 1fff ffff	1	-	Move W to "f"
MOVWR	r	01 1110 00rr rrrr	1	-	Move W to "r"
MOVRW	r	01 1111 00rr rrrr	1	-	Move "r" to W
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
SLEEP		01 1110 0000 0011	1	TO, PD	Go into Power-down mode, Clock oscillation stops
TABRH		00 0000 0101 1000	2	-	Lookup ROM high data to W
TABRL		00 0000 0101 0000	2	-	Lookup ROM low data to W
XORLW	k	01 1101 kkkk kkkk	1	Z	XOR Literal "k" with W

ADDLW Add Literal "k" and W

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

ADDWF Add W and "f"

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

ANDLW Logical AND Literal "k" with W

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

ANDWF AND W with "f"

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

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

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

BTFSC	Test "b" bit of "f", skip if clear(0)	
Syntax	BTFSC f [,b]	
Operands	f : 00h ~ 3Fh, b : 0 ~ 7	
Operation	Skip next instruction if (f.b) =0	
Status Affected	-	
OP-Code	01 010b bbff ffff	
Description	If bit 'b' in register 'f' is 1, then the next instruction is executed. If bit 'b' in register 'f' is 0, then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.	
Cycle	1 or 2	
Example	LABEL1 BTFSC FLAG, 1 TRUE GOTO SUB1 FALSE ...	B : PC =LABEL1 A : if FLAG.1 =0, PC =FALSE if FLAG.1 =1, PC =TRUE

BTFSS	Test "b" bit of "f", skip if set(1)	
Syntax	BTFSS f [,b]	
Operands	f : 00h ~ 3Fh, b : 0 ~ 7	
Operation	Skip next instruction if (f.b) =1	
Status Affected	-	
OP-Code	01 011b bbff ffff	
Description	If bit 'b' in register 'f' is 0, then the next instruction is executed. If bit 'b' in register 'f' is 1, then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.	
Cycle	1 or 2	
Example	LABEL1 BTFSS FLAG, 1 TRUE GOTO SUB1 FALSE ...	B : PC =LABEL1 A : if FLAG.1 =0, PC =TRUE if FLAG.1 =1, PC =FALSE

CALL	Call subroutine "k"	
Syntax	CALL k	
Operands	k : 000h ~ FFFh	
Operation	Operation: TOS ← (PC) + 1, PC.11~0 ← k	
Status Affected	-	
OP-Code	10 kkkk kkkk kkkk	
Description	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The 12-bit immediate address is loaded into PC bits <11:0>. CALL is a two-cycle instruction.	
Cycle	2	
Example	LABEL1 CALL SUB1	B : PC =LABEL1 A : PC =SUB1, TOS =LABEL1 + 1

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

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

CLRWDT	Clear Watchdog Timer	
Syntax	CLRWDT	
Operands	-	
Operation	WDT/WKT Timer ← 00h	
Status Affected	TO, PD	
OP-Code	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

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

MOVRW	Move "r" to W	
Syntax	MOVWR r	
Operands	r : 20h	
Operation	(W) ← (r)	
Status Affected	-	
OP-Code	01 1111 00rr rrrr	
Description	Move data from register 'r' to register W.	
Cycle	1	
Example	MOVRW EEPDT	B : EEPDT =0xFE, W =0x4F A : EEPDT =0xFE, W =0xFE

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

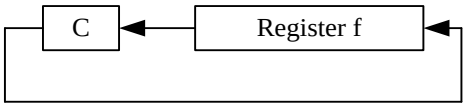
RET	Return from Subroutine	
Syntax	RET	
Operands	-	
Operation	PC ← TOS	
Status Affected	-	
OP-Code	00 0000 0100 0000	
Description	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.	
Cycle	2	
Example	RET	A : PC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

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

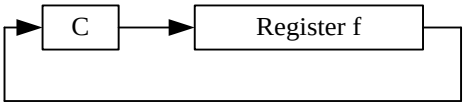
RETLW Return with Literal in W

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

RLF Rotate Left "f" through Carry

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

RRF Rotate Right "f" through Carry

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

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

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

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

TABRH Return DPTR high byte to W

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

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

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

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

TABRL Return DPTR low byte to W

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

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

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

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

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_{CC} +0.3	
Output voltage	V_{SS} -0.3 to V_{CC} +0.3	
Output current high per 1 PIN	-25	mA
Output current high per all PIN	-80	
Output current low per 1 PIN	+30	
Output current low per all PIN	+150	
Maximum operating voltage	5.5	V
Operating temperature	-40 to +85	℃
Storage temperature	-65 to +150	

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

Parameter	Sym	Conditions		Min	Typ	Max	Unit
Input High Voltage	V_{IH}	All Input, except PA7	V_{CC} =3~5V	0.6Vcc	–	Vcc	V
		PA7	V_{CC} =3~5V	0.7Vcc	–	Vcc	V
Input Low Voltage	V_{IL}	All Input, except PA7	V_{CC} =3~5V	Vss	–	0.2Vcc	V
		PA7	V_{CC} =3~5V	Vss	–	0.2Vcc	V
Output High Current	I_{OH}	All Output, except HS Pin	V_{CC} =5V,V_{OH} =4.5V	6	11	–	mA
			V_{CC} =3V,V_{OH} =2.7V	3	5	–	
		All HS Pin	V_{CC} =5V,V_{OH} =4.5V	6	11	–	
			V_{CC} =3V,V_{OH} =2.7V	3	5	–	
Output Low Current	I_{OL}	All Output, except HS Pin	V_{CC} =5V,V_{OL} =0.5V	12	22	–	mA
			V_{CC} =3V,V_{OL} =0.3V	7	12	–	
		All HS Pin	V_{CC} =5V,V_{OL} =0.5V	35	58	–	
			V_{CC} =3V,V_{OL} =0.3V	18	30	–	
Input Leakage Current (pin high)	I_{ILH}	All Input	V_{IN} =V_{CC}	–	–	1	uA
Input Leakage Current (pin low)	I_{ILL}	All Input	V_{IN} =0V	–	–	–1	uA
Power Supply Current (No Load)	I_{CC}	FAST mode FIRC 16 MHz	V_{CC} =5V	–	3.9	–	mA
		FAST mode FIRC 8 MHz	V_{CC} =5V		2.4		
		FAST mode FIRC 4 MHz	V_{CC} =5V	–	1.4	–	
		FAST mode FIRC 1 MHz	V_{CC} =5V		0.85		
		FAST mode FIRC 1 MHz	V_{CC} =3V		0.6		
		SLOW mode SIRC 8 KHz BGEN =0 MODE3V =0	V_{CC} =5.0V	–	190	–	uA

Parameter	Sym	Conditions		Min	Typ	Max	Unit
Power Supply Current (No Load)	I_{CC}	SLOW mode SIRC 8 KHz BGEN =0 MODE3V =1 LVR disable	V_{CC} =3.0V	–	7	–	uA
		IDLE mode SIRC 8 KHz BGEN =0 LDOSAV =1 LVR2SAV=1	V_{CC} =3.0V LVR enable	–	5	–	uA
			V_{CC} =3.0V LVR disable	–	3.5	–	
		STOP mode MODE3V =0 LDOSAV =1 LVR2SAV =1	V_{CC} =5.0V LVR disable	–	0.1	1	uA
			V_{CC} =5.0V, LVR enable	–	6	–	
		STOP mode MODE3V =1 LDOSAV =1 LVR2SAV =1	V_{CC} =3.0V, LVR disable	–	0.1	1	uA
			V_{CC} =3.0V, LVR enable	–	1.5	–	
System Operating Voltage	V_{SYS}	MODE3V =0	Fsys =4 MHz	LVR_{th}	–	5.5	V
			Fsys =8 MHz	LVR_{th}	–	5.5	
			Fsys =16 MHz	2.6	–	5.5	
		MODE3V =1	Fsys =4 MHz	LVR_{th}	–	3.6	
			Fsys =8 MHz	LVR_{th}	–	3.6	
			Fsys =16 MHz	2.6	–	3.6	
Pull-up Resistor	R_{UP}	V_{IN} =0 V Ports A/B/D/E	V_{CC} =5.0V	–	26	–	KΩ
			V_{CC} =3.0V	–	46	–	
		V_{IN} =0 V PA7	V_{CC} =5.0V	–	30	–	KΩ
			V_{CC} =3.0V	–	56	–	

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

Parameter	Condition	Min	Typ	Max	Unit
FIRC Frequency (*)	0℃ ~ 85℃, V_{CC} =4.0 V	-2.5%	16	+2.5%	MHz
	25℃, V_{CC} =3.0 ~ 5.0 V	-3%	16	+3%	
	25℃, V_{CC} =2.5 ~ 5.0 V	-5%	16	+5%	

(*) FIRC frequency can be divided by 1/2/4/16.

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

Parameter	Conditions	Min	Typ	Max	Unit
RESET Input Low width	Input V_{CC} =5 V ±10 %	3	–	–	μs
WDT time	V_{CC} =3 V, WDTPSC =11	-20%	2240	+20%	ms
	V_{CC} =5 V, WDTPSC =11		2048		
WKT time	V_{CC} =3 V, WKTPSC =11	-20%	136	+20%	ms
	V_{CC} =5 V, WKTPSC =11		128		
CPU start up time	V_{CC} = 5 V	–	15	–	ms

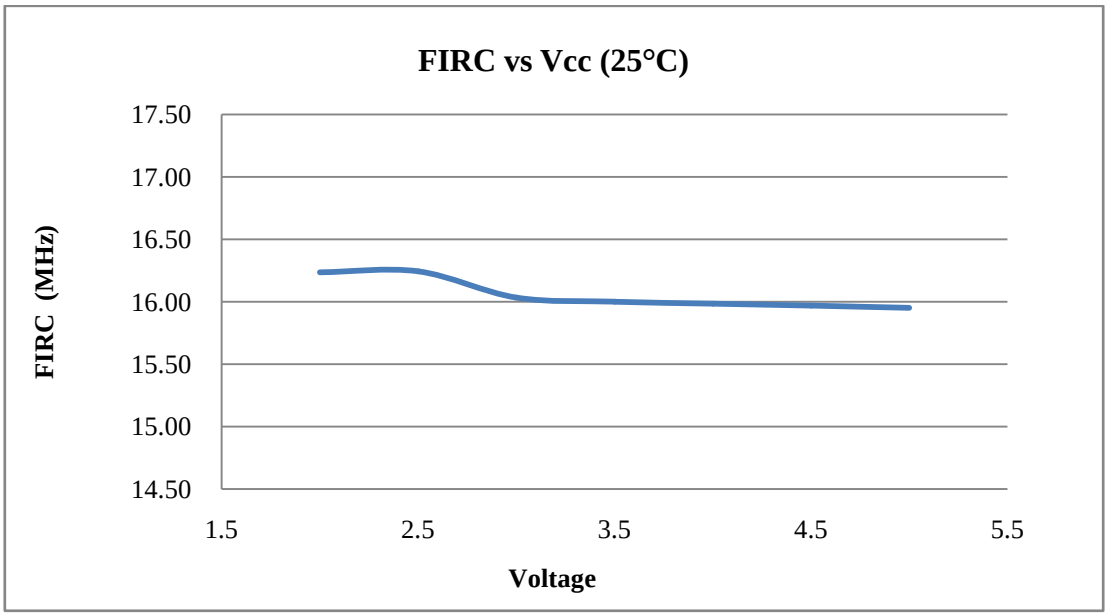
5. LVR Circuit Characteristics (T_A =25℃)

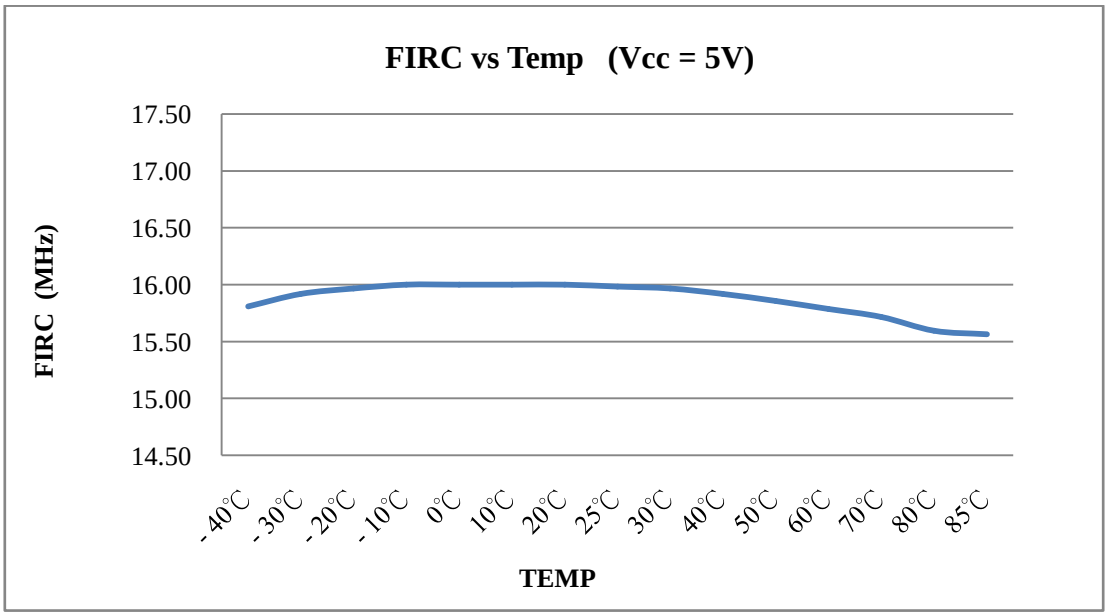
Parameter	Symbol	Min	Typ	Max	Unit
LVR Reference Voltage	LVR_{th}	–	2.0	–	V
		–	2.3	–	
		–	2.9	–	
LVR Hysteresis Voltage	V_{HYST}	–	±0.1	–	V
Low Voltage Detection time	t_{LVR}	100	–	–	μs

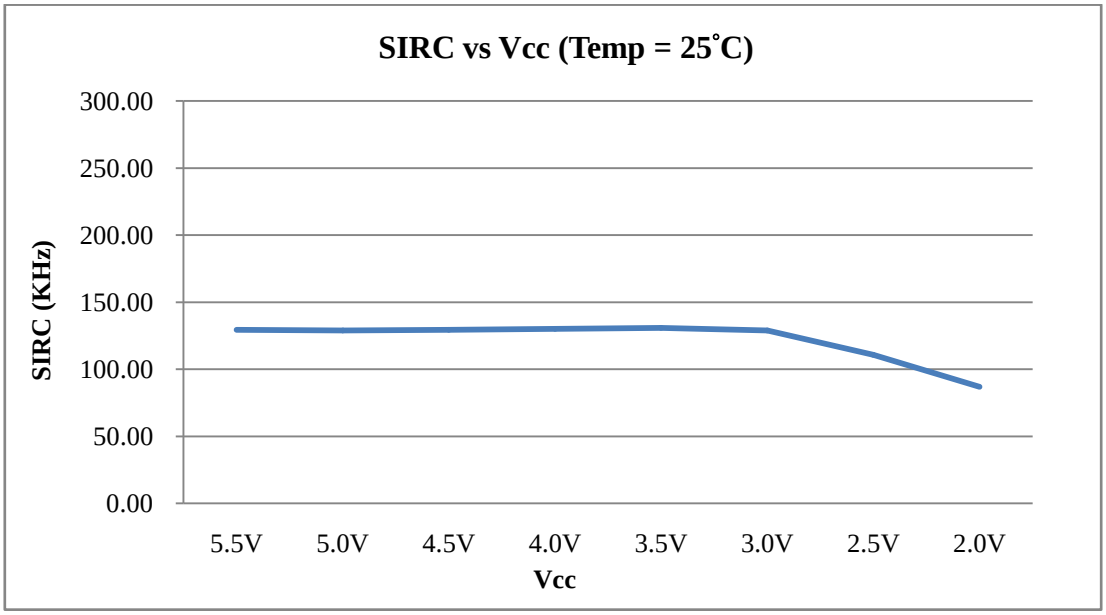
6. ADC Electrical Characteristics (T_A =25℃, VCC =2.2V to 5.5V, VSS =0V)

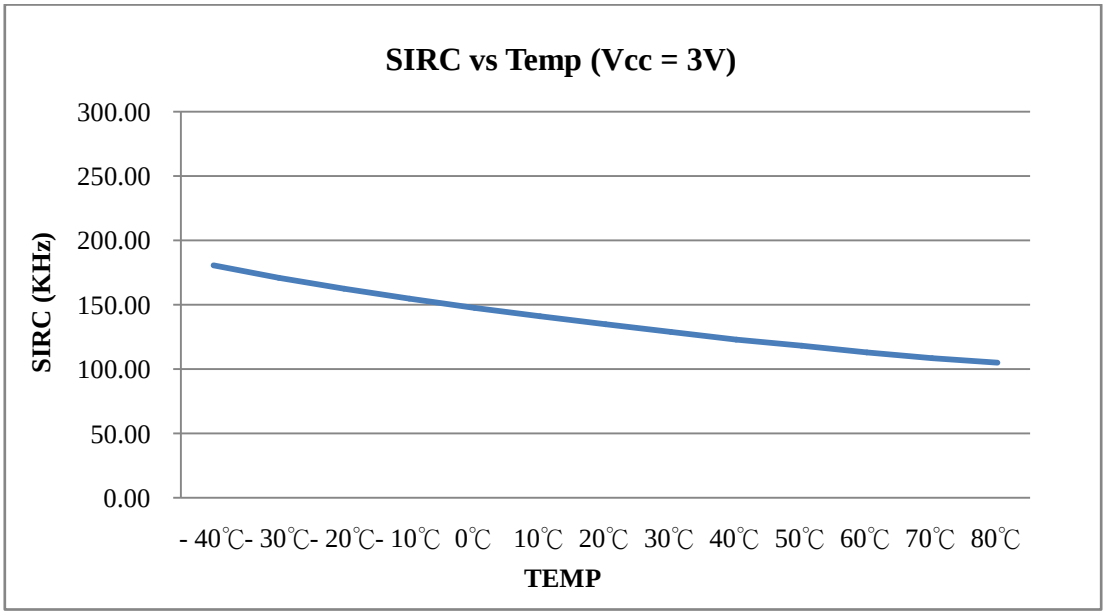
Parameter	Conditions	Min	Typ	Max	Units
Total Accuracy	V_{CC} =5V, V_{SS} =0V, f_{ADC} =1 MHz	–	±2.5	±12	LSB
Integral Non-Linearity		–	±3.2	±15	
Differential Non-linearity		–	±1	±4	
Max Input Clock (f_{ADC})	–	–	–	1	MHz
Conversion Time	f_{ADC} =1 MHz	–	50	–	μs
Input Voltage	–	V_{SS}	–	**Vcc**	V

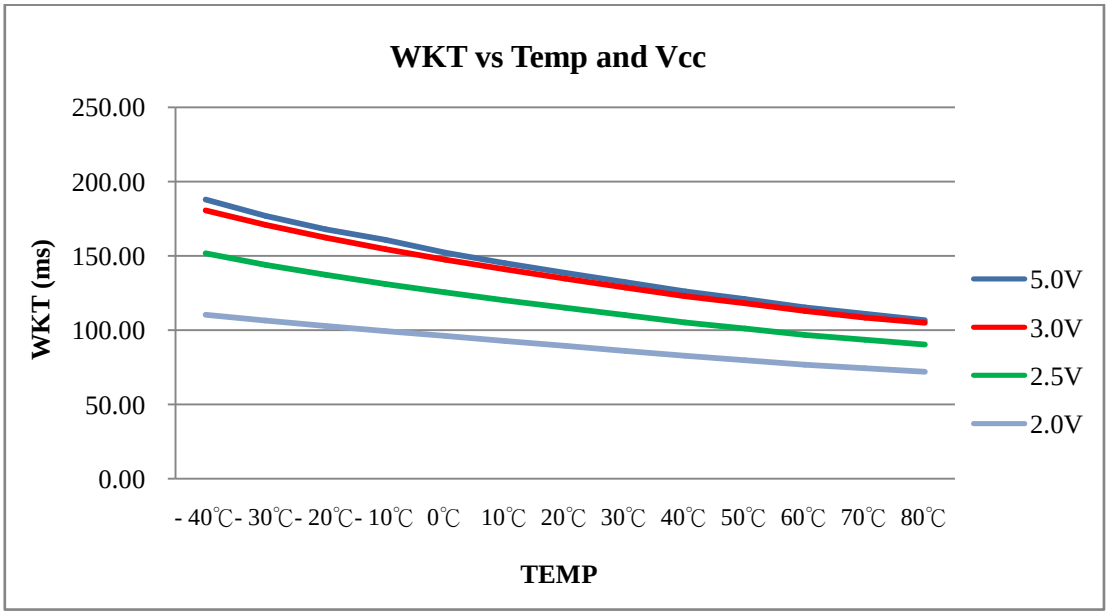
7. Characteristic Graphs

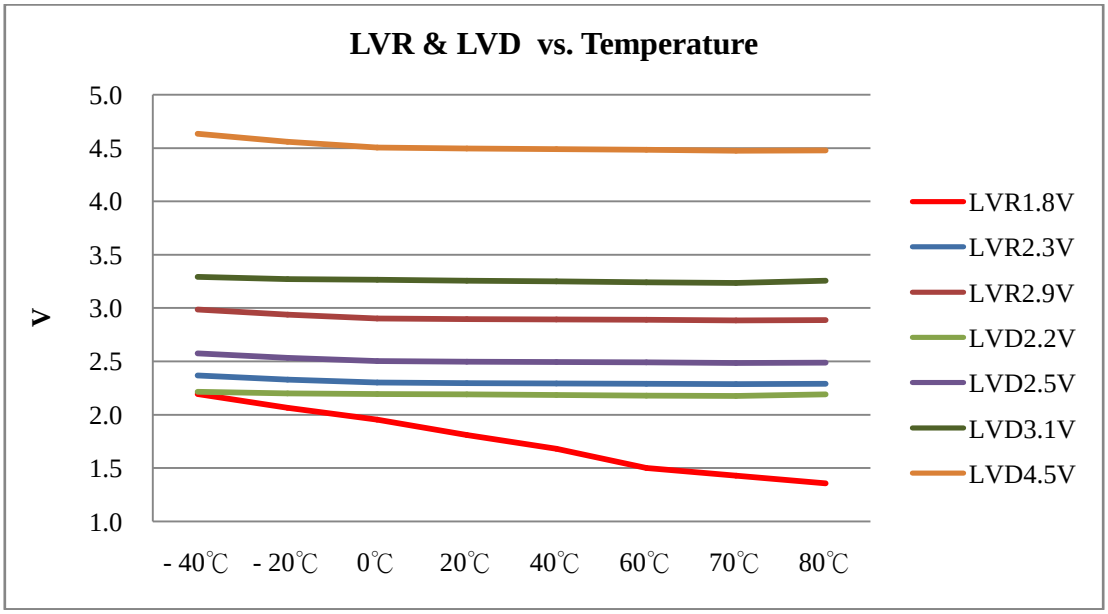


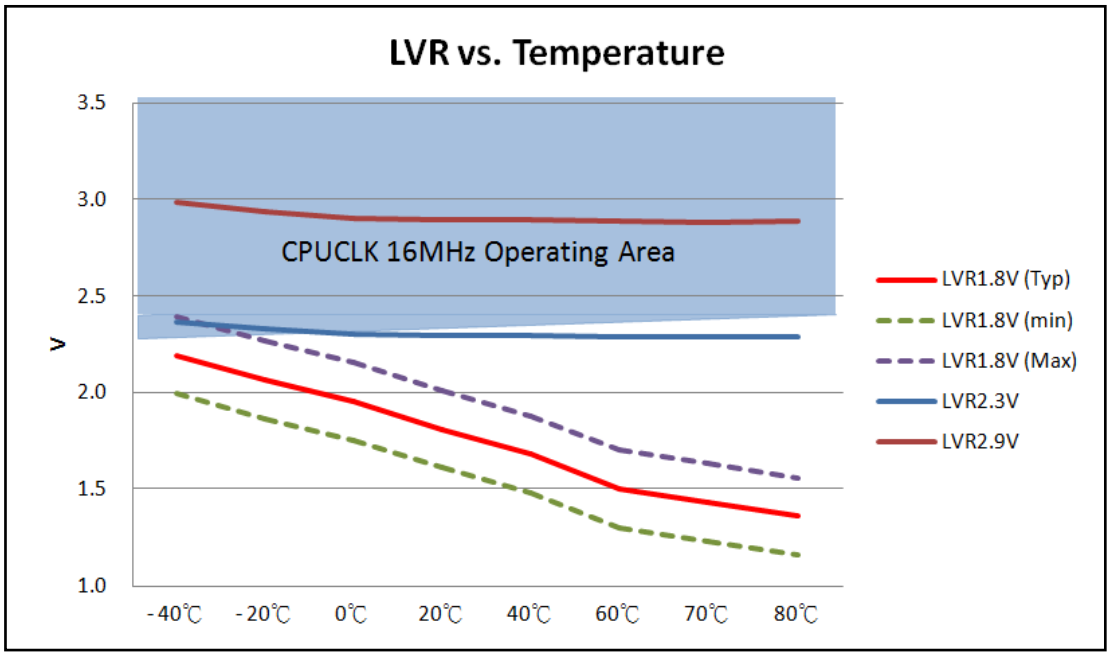


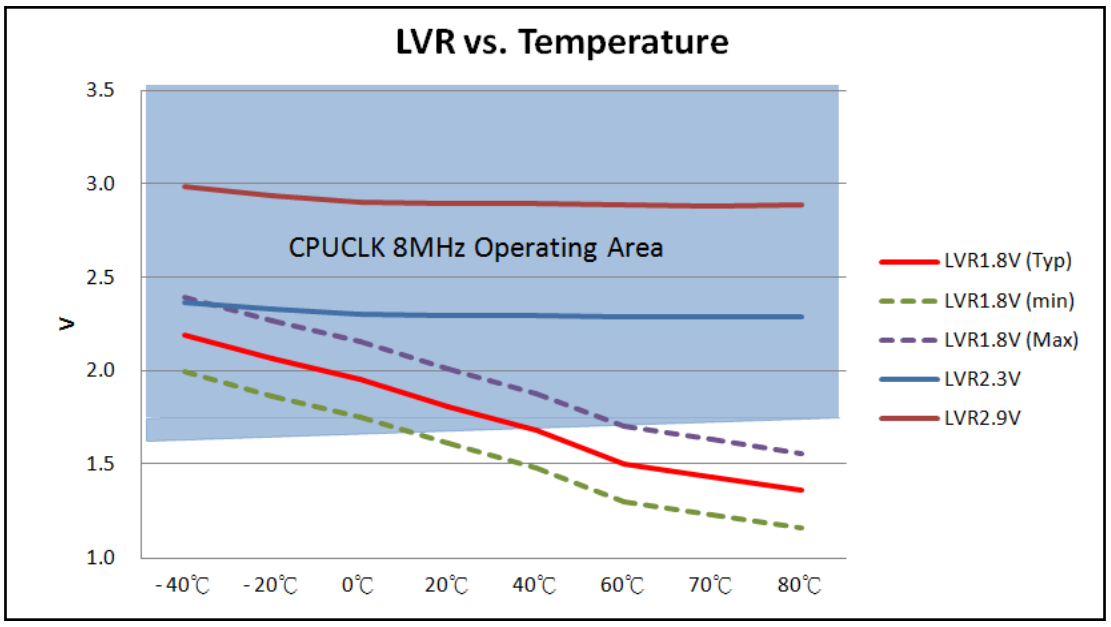

SIRC vs Vcc (Temp = 25˚C)
300.00
250.00
200.00
150.00
100.00
50.00
0.00
SIRC (KHz)
5.5V
5.0V
4.5V
4.0V
3.5V
3.0V
2.5V
2.0V
Vcc

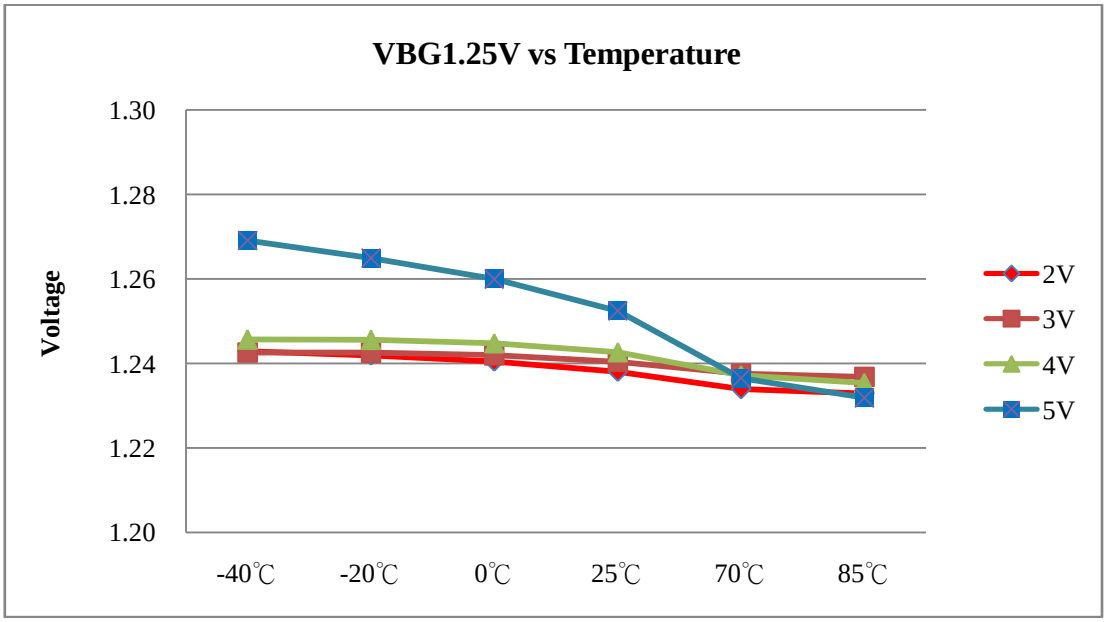

SIRC vs Temp (Vcc = 3V)
300.00
250.00
200.00
150.00
100.00
50.00
0.00
SIRC (KHz)
- 40℃
- 30℃
- 20℃
- 10℃
0℃
10℃
20℃
30℃
40℃
50℃
60℃
70℃
80℃
TEMP


WKT vs Temp and Vcc
250.00
200.00
150.00
100.00
50.00
0.00
WKT (ms)
- 40℃ - 30℃ - 20℃ - 10℃ 0℃ 10℃ 20℃ 30℃ 40℃ 50℃ 60℃ 70℃ 80℃
TEMP
5.0V
3.0V
2.5V
2.0V


LVR & LVD vs. Temperature
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
V
- 40℃ - 20℃ 0℃ 20℃ 40℃ 60℃ 70℃ 80℃
LVR1.8V
LVR2.3V
LVR2.9V
LVD2.2V
LVD2.5V
LVD3.1V
LVD4.5V


LVR vs. Temperature
3.5
3.0
2.5
2.0
1.5
1.0
V
CPUCLK 16MHz Operating Area
-40℃
-20℃
0℃
20℃
40℃
60℃
70℃
80℃
LVR1.8V (Typ)
LVR1.8V (min)
LVR1.8V (Max)
LVR2.3V
LVR2.9V


LVR vs. Temperature
3.5
3.0
2.5
2.0
1.5
1.0
V
CPUCLK 8MHz Operating Area
-40℃
-20℃
0℃
20℃
40℃
60℃
70℃
80℃
LVR1.8V (Typ)
LVR1.8V (min)
LVR1.8V (Max)
LVR2.3V
LVR2.9V

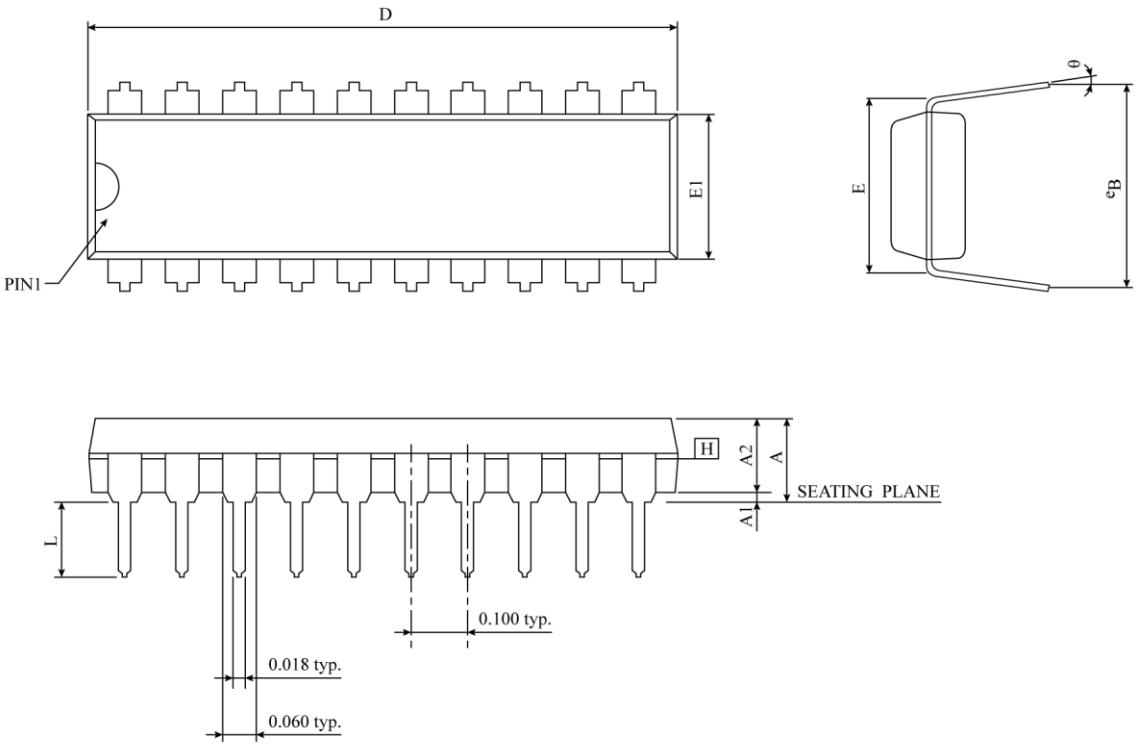

VBG1.25V vs Temperature
Voltage
1.30
1.28
1.26
1.24
1.22
1.20
-40℃
-20℃
0℃
25℃
70℃
85℃
2V
3V
4V
5V

PACKAGING INFORMATION

The ordering information:

Ordering number	Package
TM57M5541-MTP	Wafer/Dice blank chip
TM57M5541-COD	Wafer/Dice with code
TM57M5541-MTP-21	SOP 20-pin (300 mil)
TM57M5541-MTP-05	DIP 20-pin (300 mil)
TM57M5541-MTP-22	SOP 24-pin (300 mil)
TM57M5541-MTP-08	DIP 28-pin (300 mil)
TM57M5541-MTP-23	SOP 28-pin (300 mil)

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

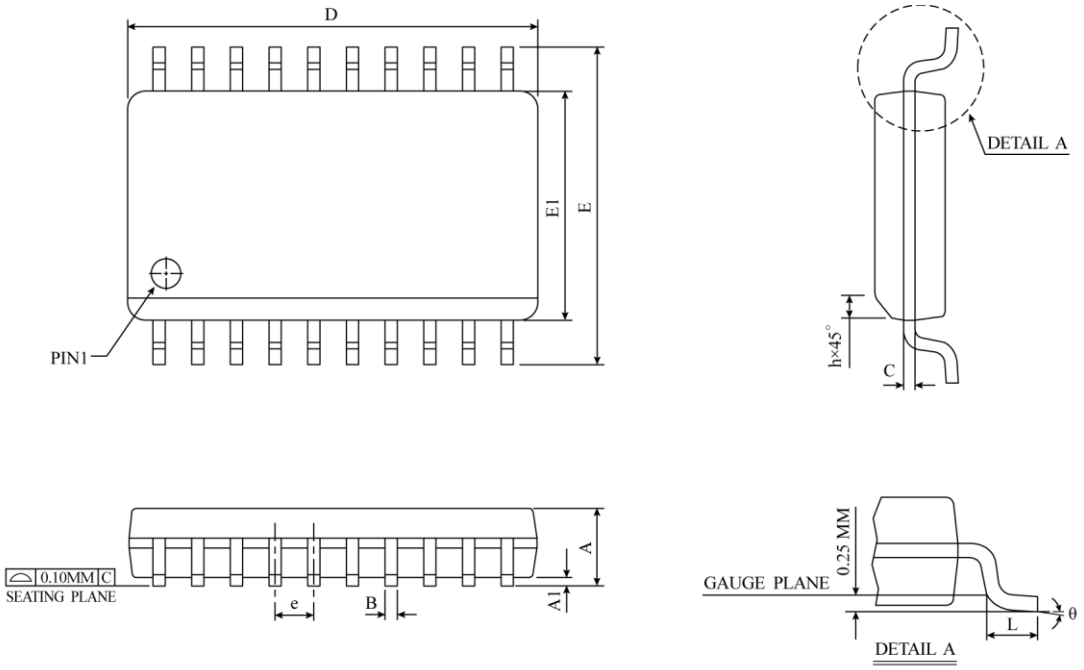


SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	4.445	-	-	0.175
A1	0.381	-	-	0.015	-	-
A2	3.175	3.302	3.429	0.125	0.130	0.135
D	25.705	26.061	26.416	1.012	1.026	1.040
E	7.620	7.747	7.874	0.300	0.305	0.310
E1	6.223	6.350	6.477	0.245	0.250	0.255
L	3.048	3.302	3.556	0.120	0.130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

NOTES：

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

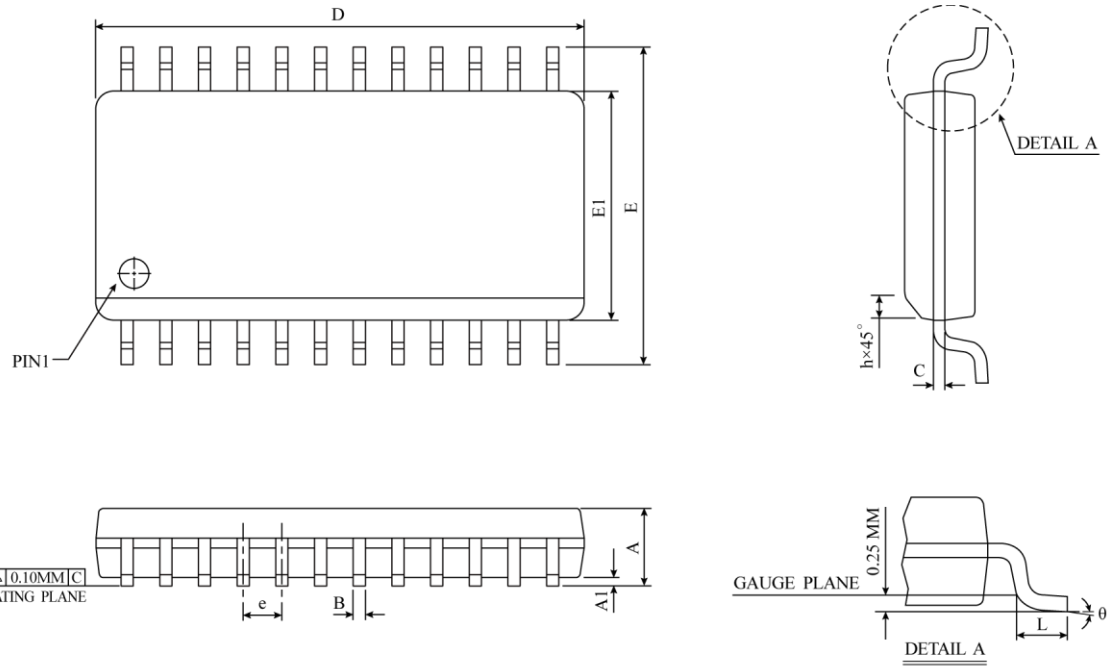
- **SOP-20 (300mil) Package Dimension**



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

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

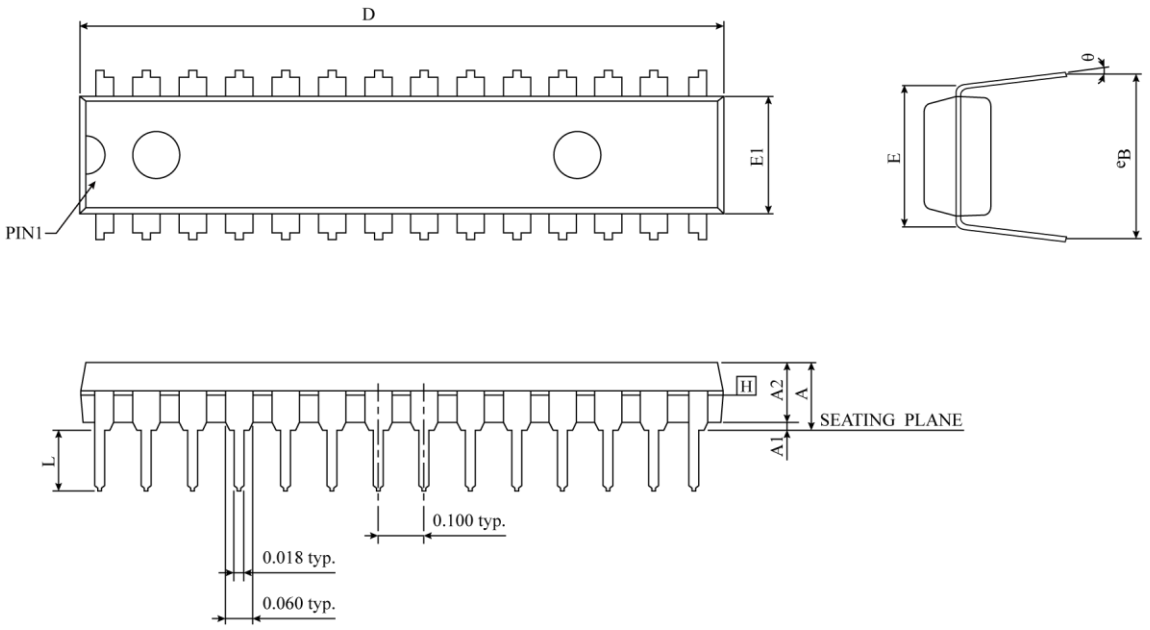
- **SOP-24 (300mil) Package Dimension**



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

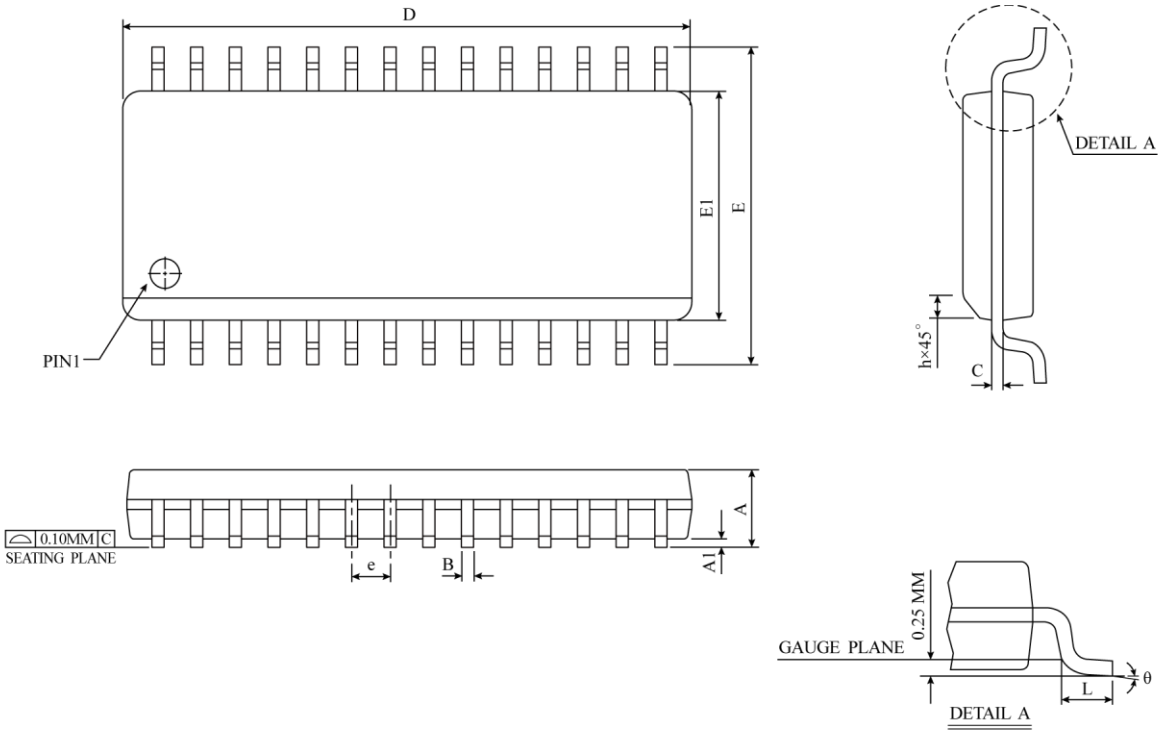
△ * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.
MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL
NOT EXCEED 0.15 MM (0.006 INCH) PER SIDE.

- **Skinny DIP-28 (300mil) Package Dimension**



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	4.445	-	-	0.175
A1	0.381	-	-	0.015	-	-
A2	3.175	3.302	3.429	0.125	0.130	0.135
D	35.179	35.370	35.56	1.385	1.393	1.400
E	7.874 BSC			0.310 BSC		
E1	7.188	7.315	7.442	0.283	0.288	0.293
L	3.048	3.302	3.556	0.120	0.130	0.140
^{e}B	8.382	8.954	9.525	0.330	0.353	0.375
θ	0°	7.5°	15°	0°	7.5°	15°
JEDEC	MS-015 (AH)					

- **SOP-28 (300mil) Package Dimension**



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	2.35	2.50	2.65	0.0926	0.0985	0.1043
A1	0.10	0.20	0.30	0.0040	0.0079	0.0118
B	0.33	0.42	0.51	0.0130	0.0165	0.0200
C	0.23	0.28	0.32	0.0091	0.0108	0.0125
D	17.70	17.90	18.10	0.6969	0.7047	0.7125
E	10.00	10.33	10.65	0.3940	0.4425	0.4910
E1	7.40	7.50	7.60	0.2914	0.2953	0.2992
e	1.27 BSC			0.050 BSC		
h	0.25	0.50	0.75	0.0100	0.0195	0.0290
L	0.40	0.84	1.27	0.0160	0.0330	0.0500
θ	0°	4°	8°	0°	4°	8°
JEDEC	MS-013 (AE)					

△ * 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.