十速

TM57M5406/08

DATA SHEET

Rev 0.90

AMENDMENT HISTORY

Version	Date	Description
V0.90	May, 2018	New release.

CONTENTS

FEATURES

1. ROM: 2K x 14 bits MTP (Multi Time Programmable ROM)
2. RAM: 368 x 8 bits
3. STACK: 6 Levels
4. I/O Ports: Three bit-programmable I/O ports (Max. 18 pins)
5. Three Independent Timers
   - Timer0
     - 8-bit Timer0 with divided by 1~256 pre-scale option / auto-reload / counter / interrupt / stop function
   - Timer1
     - 8-bit Timer1 with divided by 1~256 pre-scale option / auto-reload / interrupt / stop function / stop function
   - T2
     - 15-bit T2 with 4 interrupt interval time options
     - IDLE mode wake-up timer or used as one simple 15-bit time base
     - Clock source: Slow-clock (SIRC) or Fsys/128
6. PWMx5
   - PWM0A/0B
     - 8+2 bits PWM0 with one group shared duty-adjustable and period-adjustable function
     - PWM0 clock source: System clock (Fsys) or FIRC (12 MHz), with 1~128 pre-scalers
   - PWM1A/PWM1B/PWM1C
     - 8 bits PWM1 with three groups independent duty-adjustable function and shared period-adjustable controlled
     - PWM1 clock source: System clock (Fsys) or FIRC (12 MHz), with 1~128 pre-scalers
7. 12-bit ADC Converter with 4 input channels and 1 internal reference voltage
   - ADC reference voltage = Internal reference voltage LDO ±2% @25°C, VCC=3V~5V
8. 8-channel Touch Key with 2 TK-modules
   - each module included:
     - 4-channel Touch Key
     - 4-bit TK reference clock capacitor adjustment
     - 12-bit TK scan length adjustment
     - independent control and TK data

9. I2C Interface

- Specific purpose slave I2C interface with interrupt function

10. PB2~PB7 individual pin change wake up

11. System Oscillation Sources

- Fast-clock
  - FIRC (Fast Internal RC) : 12 MHz
- Slow-clock
  - SIRC (Slow Internal RC) : 128 KHz @VCC=3V

12. System Clock Prescaler

- System Oscillation Sources can be divided by 16 / 4 / 2 / 1 as System Clock (Fsys)

13. Power Saving Operation Modes

- FAST Mode: Fast-clock keeps CPU running
- SLOW Mode: Fast-clock stops, Slow-clock keeps CPU running
- IDLE Mode: Fast-clock and CPU stop, Wake-up Timer keep running
- STOP Mode: All Clocks Stop

14. Dual System Clock

- FIRC + SIRC

15. Reset Sources

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

16. 3-Level Low Voltage Reset: 2.9V / 2.3V / 2.0V

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

- Fsys = 3 MHz, 2.0V ~ 5.5V
- Fsys = 6 MHz, 2.3V ~ 5.5V
- Fsys = 12 MHz, 2.9V ~ 5.5V

18. Interrupts

- Three External Interrupt pins
  - Two pin is falling edge triggered
  - One pin is rising or falling edge triggered
- Timer0 / Timer1 / T2 / Wake-up Timer Interrupt
- Touch Key / ADC Interrupt
- I2C Interrupt

19. Wake-up Timer (WKT)

- Clocked by built-in RC oscillator with 4 adjustable interrupt times
  16 ms / 32 ms / 64 ms / 128 ms @VCC=5V

20. Watchdog Timer (WDT)

- Clocked by built-in RC oscillator with 4 adjustable reset times
  128 ms / 256 ms / 1024 ms / 2048 ms @VCC=5V
- Watchdog timer can be disabled / enabled in Power-down mode

21. I/O Port Modes

- Open-Drain Output
- CMOS Push-Pull Output
- Schmitt Trigger Input with pull-up resistor option

22. High-Sink and High-Drive I/O: PA0 ~ PA4 (PWM0A, PWM1A, PWM1B, PWM1C, PWM0B)

23. Programming connectivity support 5-wire (ICP) or 8-wire program

24. Operating Temperature Range : -40℃ to + 85℃

25. Table Read Instruction: 16-bit ROM data lookup table

26. Instruction set: 39 Instructions

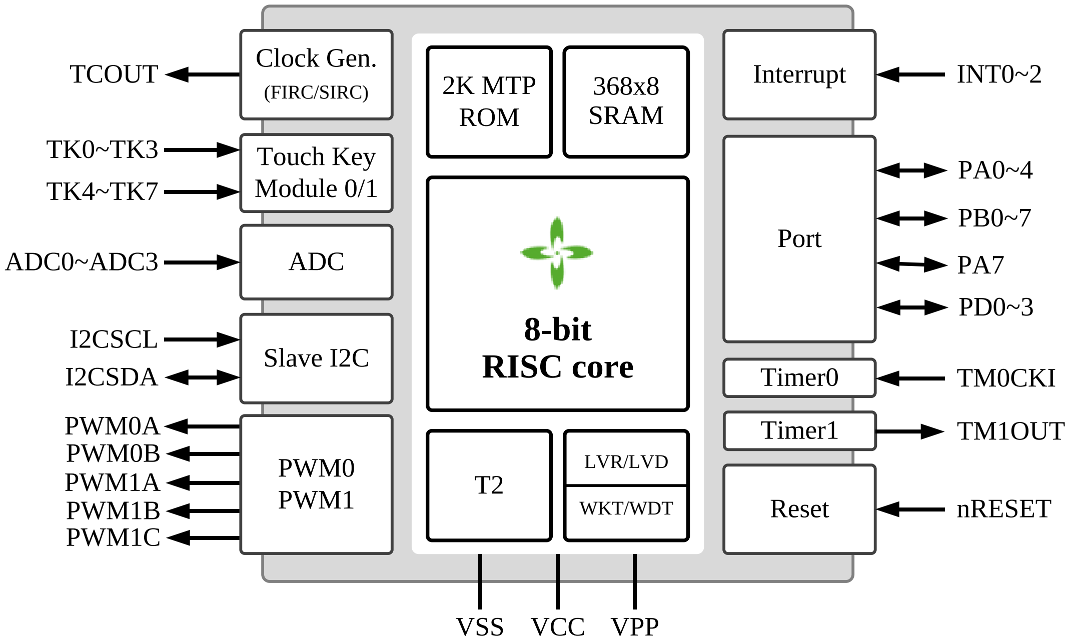
27. Package Types:

- TM57M5408: 20-pin SOP (300 mil) / 20-pin DIP (300 mil)
- TM57M5406: 16-pin SOP (150 mil) / 16-pin DIP (300 mil)

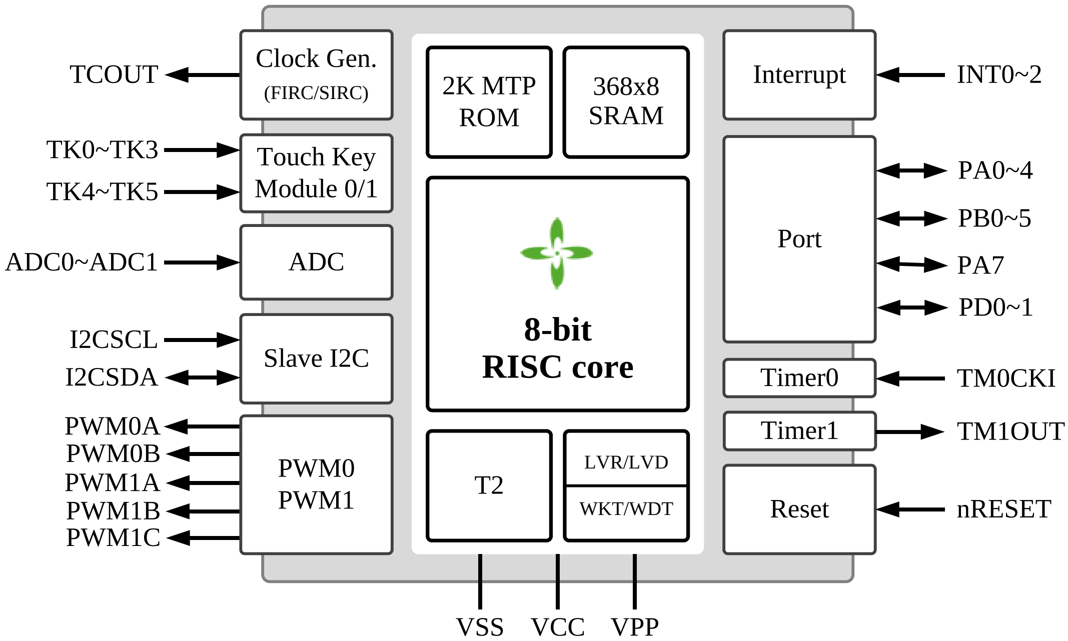
28. Supported EV board on ICE

- EV board: EV8225

BLOCK DIAGRAM

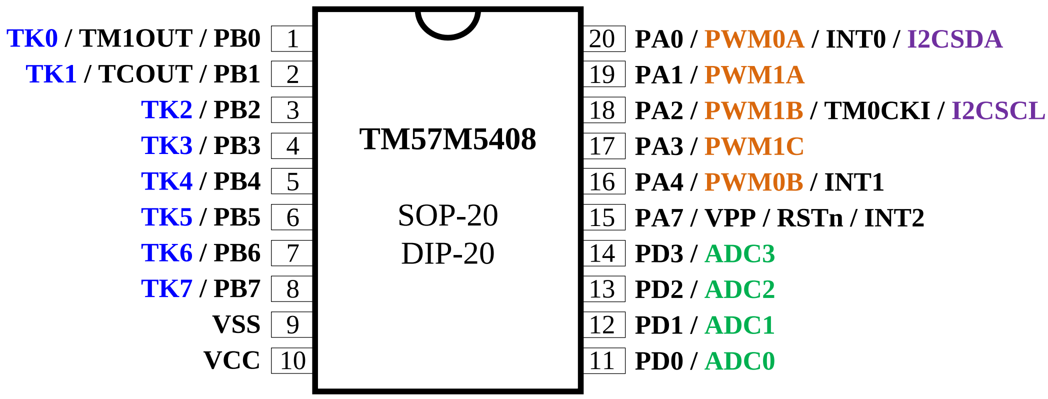


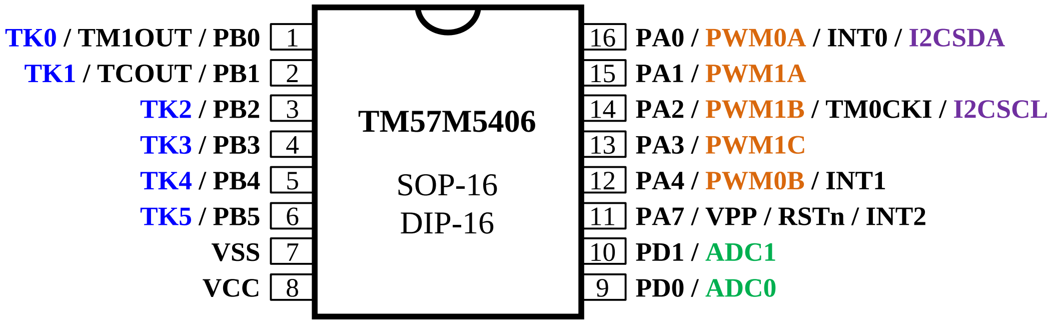
TM57F5408 Block Diagram



TM57F5406 Block Diagram

PIN ASSIGNMENT





PIN DESCRIPTION

Name	In/Out	Pin Description
PA0–PA3	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 resistor is always assignable by software.
PB0–PB7	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.
PD0–PD3	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.
nRESET	I	External active low reset
TM1OUT	O	Toggle output as timer1 overflow
TCOUT	O	Fsys/2 clock output
VCC, VSS	P	Power Voltage input pin and ground
VPP	I	PROM programming high voltage input
INT0–INT2	I	External interrupt input
PWM0A PWM0B PWM1A PWM1B PWM1C	O	PWM0 / PWM1 output
TM0CKI	I	Timer0's input in counter mode
TK0–TK7	I	Touch Key input
ADC0–ADC3	I	ADC channels input
I2CSCL	I	I2C serial clock input
I2CSDA	I/O	I2C serial data pin

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

Pin Number		Pin Name	Type	GPIO				Function AfterReset	Alternate Function				
				Input		Output							
20-SOP/DIP (M5408)	16-SOP/DIP (M5406)			Weak Pull-up	Ext. Interrupt	O.D	P.P		PWM	ADC	Touch Key	I2C	MISC
1	1	PB0/TM1OUT/TK0	I/O			○	○	PB0			○		TM1OUT
2	2	PB1/TCOUT/TK1	I/O			○	○	PB1			○		TCOUT
3	3	PB2/TK2	I/O	○		○	○	PB2			○		
4	4	PB3/TK3	I/O	○		○	○	PB3			○		
5	5	PB4/TK4	I/O	○		○	○	PB4			○		
6	6	PB5/TK5	I/O	○		○	○	PB5			○		
7	-	PB6/TK6	I/O	○		○	○	PB6			○		
8	-	PB7/TK7	I/O	○		○	○	PB7			○		
9	7	VSS	P										
10	8	VCC	P										
11	9	PD0/ADC0	I/O			○	○	PD0			○		
12	10	PD1/ADC1	I/O			○	○	PD1			○		
13	-	PD2/ADC2	I/O			○	○	PD2			○		
14	-	PD3/ADC3	I/O			○	○	PD3			○		
15	11	PA7/INT2/nRESET/VPP	I/O	○	○	○		PA7					nRESET
16	12	PA4/INT1/PWM0B	I/O	○	○	○	○	PA4	○				
17	13	PA3/PWM1C	I/O			○	○	PA3	○				
18	14	PA2/TM0CKI/I2CSCL/PWM1B	I/O			○	○	PA2	○			○	TM0CKI
19	15	PA1/PWM1A	I/O			○	○	PA1	○				
20	16	PA0/INT0/I2CSDA/PWM0A	I/O	○	○	○	○	PA0	○			○	

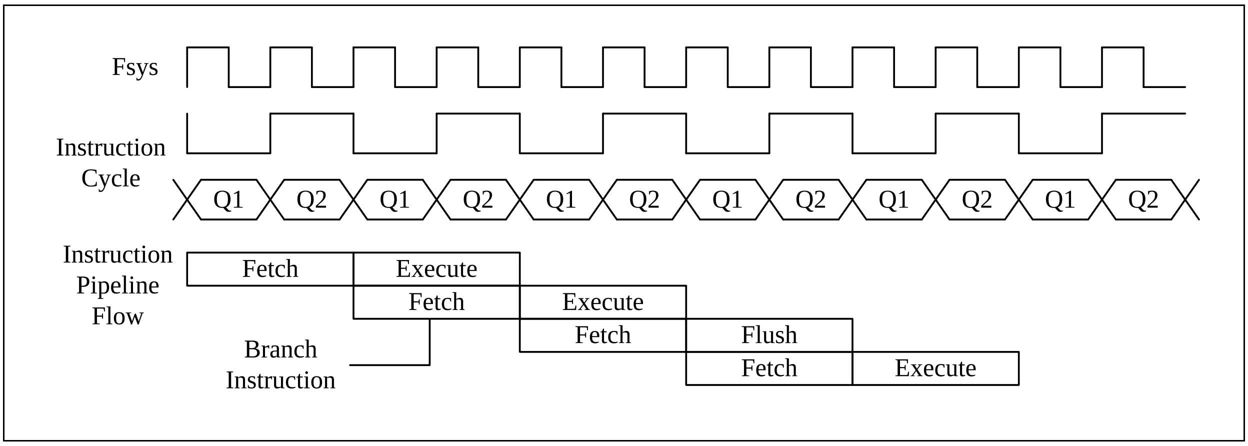
Symbol：P.P. = COM Push-Pull Output
O.D. = Open Drain Output

FUNCTION DESCRIPTION

1. CPU Core

1.1. Clock Scheme and Instruction Cycle

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



Terminology definitions:

(1) **Fsys**: System clock. The main clock that drive the core logic and most peripherals. The clock source can be either Fast-clock or Slow-clock which can be set by registers.

(2) **Instruction Cycle** = Fsys/2

FIRC: Fast Internal RC oscillator

SIRC: Slow Internal RC oscillator

1.2. 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.

1.3. Programming Counter (PC) and Stack

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

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

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

```
            ORG       000H            ; Reset Vector
            GOTO      START           ; Goto user program address
START:
            MOVLW     00H
            MOVWF     INDEX           ; Set lookup table's address (INDEX)
LOOP:
            MOVFW     INDEX           ; Move INDEX value to W register
            CALL      TABLE           ; To Lookup data (W = 55H when INDEX = 00H)
            …
            INCF      INDEX, 1        ; Increment the INDEX for next address
            …
            GOTO      LOOP            ; Goto LOOP label

            ORG       X00H            ; X = 1, 2, 3, …, 6, 7
TABLE:
            ADDWF     PCL, 1          ; (Addr = X00H) Add the W with PCL, the result
                                      ; back in PCL
            RETLW     55H             ; W = 55H when return
            RETLW     56H             ; W = 56H when return
            RETLW     58H             ; W = 58H when return
```

Note: TM57M5406/08 defines 256 ROM addresses as one page, so that TM57M5406/08 has 8 pages, 000H~0FFH, 100H~1FFH, 200H~2FFH, …, and 700H~7FFH. On the other words, PC [10:8] can be defined as page. A lookup table must be located at the same page to avoid getting wrong data. Thus, the lookup table has maximum 255 data for above example with starting a lookup table at X00H (X = 1, 2, 3, …, 6, 7) . If a lookup table has fewer data, it needs not set the starting address at X00H, just only confirm all lookup table data are located at the same page.

◇ Example2: To look up the PROM data located "TABLE2"

```
            ORG         000H                ; Reset Vector
            GOTO        START               ; Goto user program address
START:
            MOVLW       (TABLE2>>8) & 0xff
            MOVWF       DPH                 DPH register (F1E.2~0)
            MOVLW       (TABLE2) & 0xff
            MOVWF       DPL                 DPL register (F1D.7~0)
            TABRL                           W=86H
            TABRH                           W=19H

            ORG         368H
TABLE2:     .DT 0x1986
            .DT 0x3719
            .DT 0x2983
```

F02	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCL	PCL							
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

F02.7~0 **PCL:** Low-byte of Program Counter (PC[7:0])

F0A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCH	–	–	–	–	–	PCH		
R/W	–	–	–	–	–	R/W	R/W	R/W
Reset	–	–	–	–	–	0	0	0

F0A.2~0 **PCH:** 3 MSBs of Program Counter (PC[10:8])

F1D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPL	DPL							
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

F1D.7~0 **DPL:** Table read low address, data ROM pointer (DPTR[7:0])

F1E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPH	–	–	–	–	–	–	DPH	
R/W	–	–	–	–	–	–	R/W	R/W
Reset	–	–	–	–	–	–	0	0

F1E.1~0 **DPH:** 2 MSBs of Table read high address, data ROM pointer (DPTR[9:8])

1.4. STATUS Register (F-Plane 03H)

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

<table>
<tr><th>STATUS</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td>Reset Value</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R</td><td>R</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td>Bit</td><td colspan="8">Description</td></tr>
<tr><td>7</td><td colspan="8">GB1: General Purpose Bit 1</td></tr>
<tr><td>6</td><td colspan="8">GB0: General Purpose Bit 0</td></tr>
<tr><td>5</td><td colspan="8">RAMBK: FRAM Bank Selection
0: FRAM Bank0
1: FRAM Bank1</td></tr>
<tr><td>4</td><td colspan="8">TO: Time Out Flag
0: after Power On Reset, LVR Reset, or CLRWDT / SLEEP instruction
1: WDT time out occurs</td></tr>
<tr><td>3</td><td colspan="8">PD: Power Down Flag
0: after Power On Reset, LVR Reset, or CLRWDT instruction
1: after SLEEP instruction</td></tr>
<tr><td>2</td><td colspan="8">Z: Zero Flag
0: the result of a logic operation is not zero
1: the result of a logic operation is zero</td></tr>
<tr><td rowspan="3">1</td><td colspan="8">DC: Decimal Carry Flag or Decimal/Borrow Flag</td></tr>
<tr><td colspan="4">ADD instruction</td><td colspan="4">SUB instruction</td></tr>
<tr><td colspan="4">0: no carry
1: a carry from the low nibble bits of the result occurs</td><td colspan="4">0: a borrow from the low nibble bits of the result occurs
1: no borrow</td></tr>
<tr><td rowspan="3">0</td><td colspan="8">C: Carry Flag or Borrow Flag</td></tr>
<tr><td colspan="4">ADD instruction</td><td colspan="4">SUB instruction</td></tr>
<tr><td colspan="4">0: no carry
1: a carry occurs from the MSB</td><td colspan="4">0: a borrow occurs from the MSB
1: no borrow</td></tr>
</table>

◇ Example: Write immediate data into STATUS register

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

◇ Example: Bit addressing set and clear STATUS register

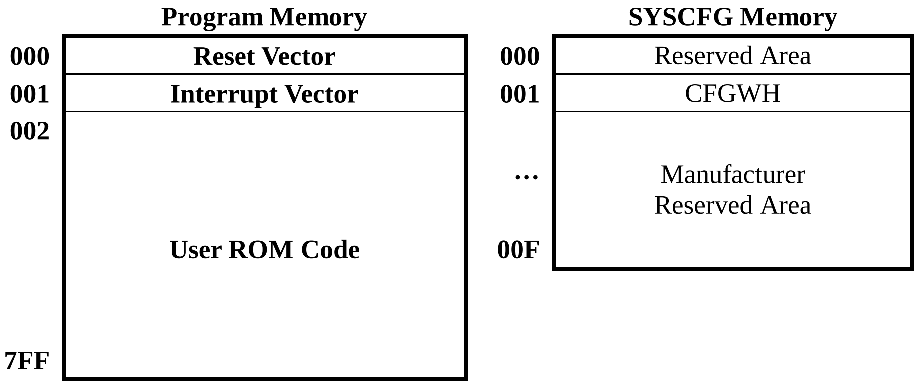
```
BSF       STATUS, 0       ; Set C = 1
BCF       STATUS, 0       ; Clear C = 0
```

◇ Example: Determine the C flag by BTFSS instruction

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

2. Program ROM (MTP)

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



The System Configuration Register (SYSCFG) is located at MTP INFO area. The SYSCFG determines the option for initial condition of MCU. It is written by MTP Writer only. User can select chip operation mode by SYSCFG register.

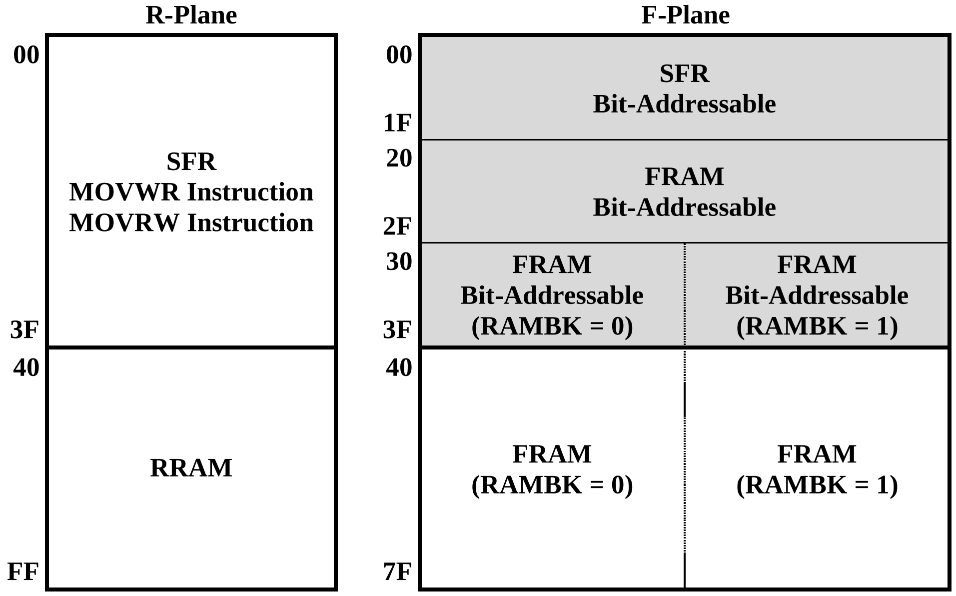
Bit		**13~0**	
Default Value		**00_0000_0000_0000**	
Bit		Description	
CFGWH	13	**PROTECT**: Code protection selection	
		1	Enable
		0	Disable
	12	**XRSTE**: External Pin (PA7) Reset Enable	
		1	Enable
		0	Disable (PA7 as input I/O pin)
	11-10	**LVR**: Low Voltage Reset Mode	
		11	2.0V
		10	2.0V
		01	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	

3. Data Memory (RAM and SFR)

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

The lower locations of F-Plane are reserved for Special-Function-Register (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 supports the MOVWR, MOVRW byte operating instructions.



F-Plane	8/0	9/1	A/2	B/3	C/4	D/5	E/6	F/7
00h	INDF	TM0	PCL	STATUS	FSR	PAD	PBD	PDD
08h	INTIE	INTIF	PCH	CLKCTL	MF0C	PWM0DH	PWM0DL	MF0F
10h		ADCH	ADTKD	ADCTL	TM1	TKFLG	TKCTL	TKCHS
18h	I2CTXD0	I2CTXD1	I2CCTL	I2CFLG	RSR	DPL	DPH	IRCF

R-Plane	8/0	9/1	A/2	B/3	C/4	D/5	E/6	F/7
00h	INDR	TM0RLD	TM0CTL	PWRDN	WDTCLR	PAMODH	PAMODL	PBMODH
08h	PBMODL		PDMODL	MR0B		PWM0PRD	PWM1PRD	
18h	PWMCTL	PWM1AD	PWM1BD	PWM1CD	TM1CTL	TM1RLD	MR16	
20h	I2CRCD0	I2CRCD1		TKM10DH	TKM0DL	TKM1DL		
28h	TKM0CTL	TKM0TMR	TKM1CTL	TKM1TMR				

◇ Example: Write immediate data into R-Plane register

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

◇ Example: Write immediate data into F-Plane register

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

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

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

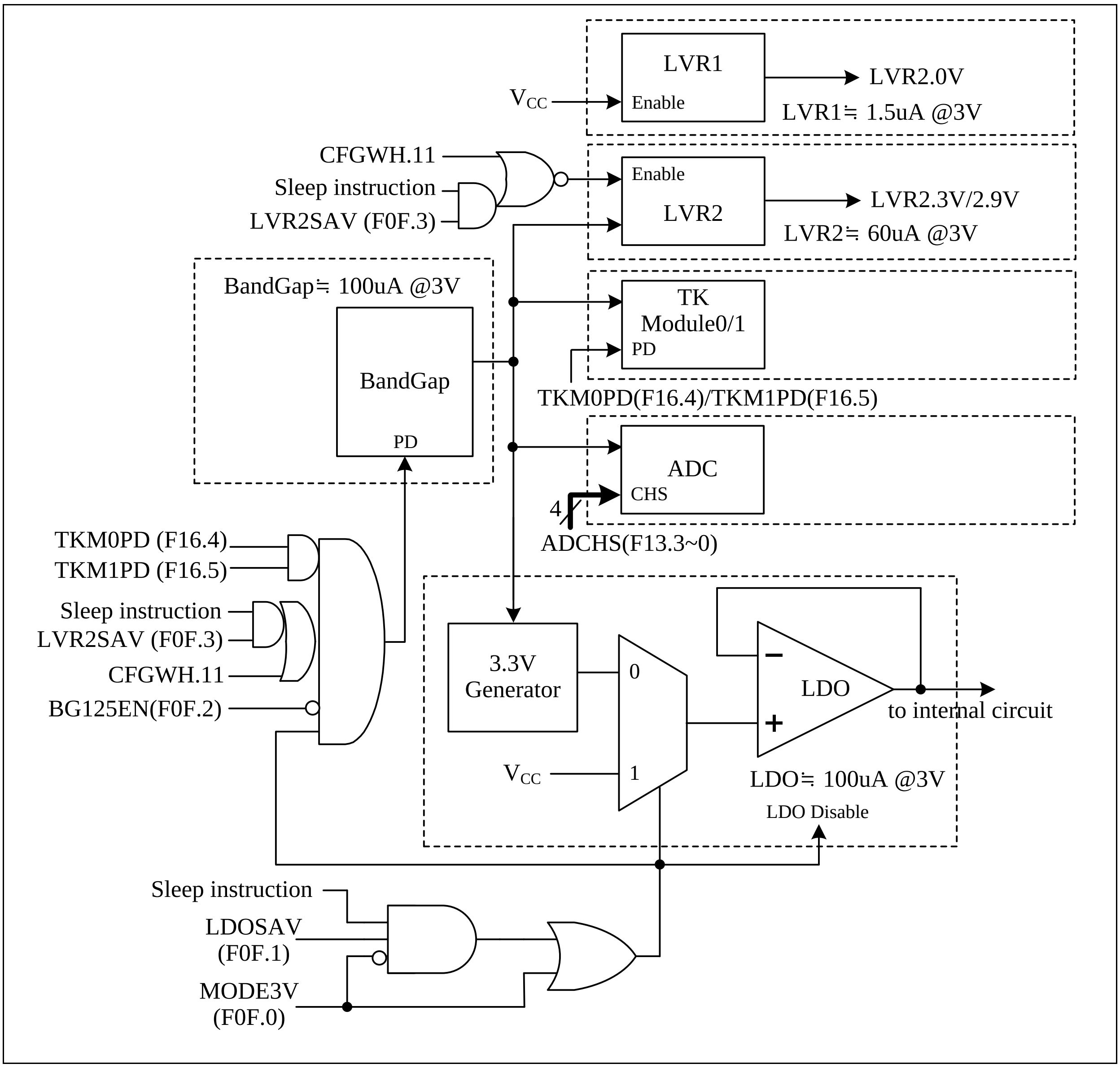
◇ Example: Clear FRAM Bank0 data by indirect addressing mode

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

4. Power Management

The Chip 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 user enable one of LDO, LVR2, BG125EN, TK Module0 or TK Module1, BandGap (≒100µA) will be also enabled. To further save power consumption in STOP/IDLE mode, user must disable LDO/LVR2/BG125EN/TK at the same time.



MODE3V=0

Operation Mode	LDOSAV (F0F.1)	LVR CFGWH.11~10	LVR2SAV (F0F.3)	BG125EN (F0F.2)	TKM0PD (F16.4)	TKM1PD (F16.5)	BandGap (100uA)	LDO (100uA)	LVR1 (2uA)	LVR2 (60uA)	Function
Fast / Slow	X	00	X	X	X	X	ON	ON	ON	ON	LVR2.9V
	X	01	X	X	X	X	ON	ON	ON	ON	LVR2.3V
	X	1X	X	X	X	X	ON	ON	ON	OFF	LVR2.0V
Idle / Stop	0	00	0	X	X	X	ON	ON	ON	ON	LVR2.9V
	0	01	0	X	X	X	ON	ON	ON	ON	LVR2.3V
	0	0X	1	X	X	X	ON	ON	ON	OFF	LVR2.0V
	0	1X	X	X	X	X	ON	ON	ON	OFF	LVR2.0V
	1	00	0	X	X	X	ON	OFF	ON	ON	LVR2.9V
	1	01	0	X	X	X	ON	OFF	ON	ON	LVR2.3V
	1	0X	1	0	0	0	OFF	OFF	ON	OFF	LVR2.0V
	1	1X	X	0	0	0	OFF	OFF	ON	OFF	LVR2.0V
	1	XX	1	X	X	X	ON	OFF	ON	OFF	LVR2.0V
	X	XX	X	1	X	X	ON	X	ON	X	LVR2.0V
	X	XX	X	X	1	X	ON	X	ON	X	LVR2.0V
	X	XX	X	X	X	1	ON	X	ON	X	LVR2.0V

MODE3V=1

Operation Mode	LDOSAV (F0F.1)	LVR CFGWH.11~10	LVR2SAV (F0F.3)	BG125EN (F0F.2)	TKM0PD (F16.4)	TKM1PD (F16.5)	BandGap (100uA)	LDO (100uA)	LVR1 (2uA)	LVR2 (60uA)	Function
Fast / Slow	X	00	X	X	X	X	ON	OFF	ON	ON	LVR2.9V
	X	01	X	X	X	X	ON	OFF	ON	ON	LVR2.3V
	X	1X	X	X	X	X	ON	OFF	ON	OFF	LVR2.0V
	X	1X	X	0	0	0	OFF	OFF	ON	OFF	LVR2.0V
Idle / Stop	X	00	0	X	X	X	ON	OFF	ON	ON	LVR2.9V
	X	01	0	X	X	X	ON	OFF	ON	ON	LVR2.3V
	X	0X	1	0	0	0	OFF	OFF	ON	OFF	LVR2.0V
	X	1X	X	0	0	0	OFF	OFF	ON	OFF	LVR2.0V
	X	XX	X	1	X	X	ON	OFF	ON	X	LVR2.0V
	X	XX	X	X	1	X	ON	OFF	ON	X	LVR2.0V
	X	XX	X	X	X	1	ON	OFF	ON	X	LVR2.0V

F0F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0F	–	–	TKM1SOC	TKM1SOC	LVR2SAV	BG125EN	LDOSAV	MODE3V
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	1	1	1	0

F0F.3 **LVR2SAV:** LVR2 power save mode enable
0: disable LVR2 power save mode
1: LVR2 (2.3V/2.9V) auto power off in STOP/IDLE mode

F0F.2 **BG125EN:** Internal Bandgap voltage 1.25V enable
0: Internal Bandgap voltage disable
1: Internal Bandgap voltage enable

F0F.1 **LDOSAV:** LDO power save mode enable
0: disable LVR2 power save mode
1: LDO auto power off in STOP/IDLE mode

F0F.0 **MODE3V:** 3V mode selection control bit
If this bit is set, the chip can be only operated in the condition of VCC<3.6V, and LDO is turned off to save current

F16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCTL	–	–	TKM1PD	TKM0PD	TKIE	TK3V	TKFDB	TKFSL
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	1	1	0	0	0	0

F16.5 **TKM1PD**: Touch Key Module1 power down
0: Touch Key Module1 running
1: Touch Key Module1 power down

F16.4 **TKM0PD**: Touch Key Module0 power down
0: Touch Key Module0 running
1: Touch Key Module0 power down

5. Reset

The TM57M5406/08 can be RESET in four ways.

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

After Power-On-Reset, all system and peripheral control registers are then set to their default hardware Reset values. The LVR level is selected by the CFGWH register. The Low Voltage Reset features static reset when supply voltage is below a threshold level. There are three threshold levels 2.0V, 2.3V and 2.9V can be selected. The External Pin Reset and Watchdog Reset can be disabled or enabled by the CFGWH register. These two resets also set all the control registers to their default reset value. The TO/PD flag is not affected by these resets.

5.1. 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 CFGWH register.

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

LVR Selection Table:

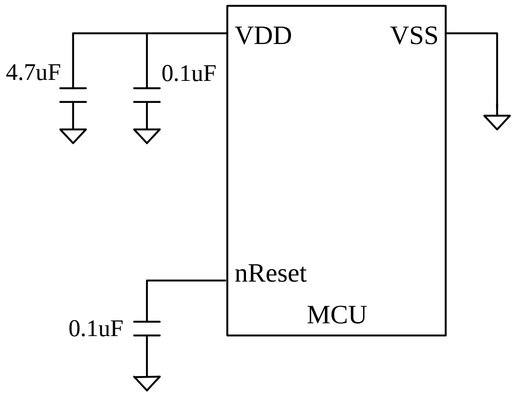
LVR level	Operating voltage
LVR2.0	5.5V > VCC > 2.2V
LVR2.3	5.5V > VCC > 2.4V
LVR2.9	5.5V > VCC > 3.1V or V_{CC} =5.0V

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

5.3. External Pin Reset

The External Pin Reset can be disabled or enabled by the CFGWH. 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.

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



5.4. Watchdog Timer Reset

WDT overflow Reset can be disabled or enabled by the CFGWH 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 Timer is cleared by device Reset or instruction CLRWDT. WDT Timer Reset also set all the control registers to their default reset value. The TO/PD flags are not affected by these resets.

◇ Example: Defining Reset Vector

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

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

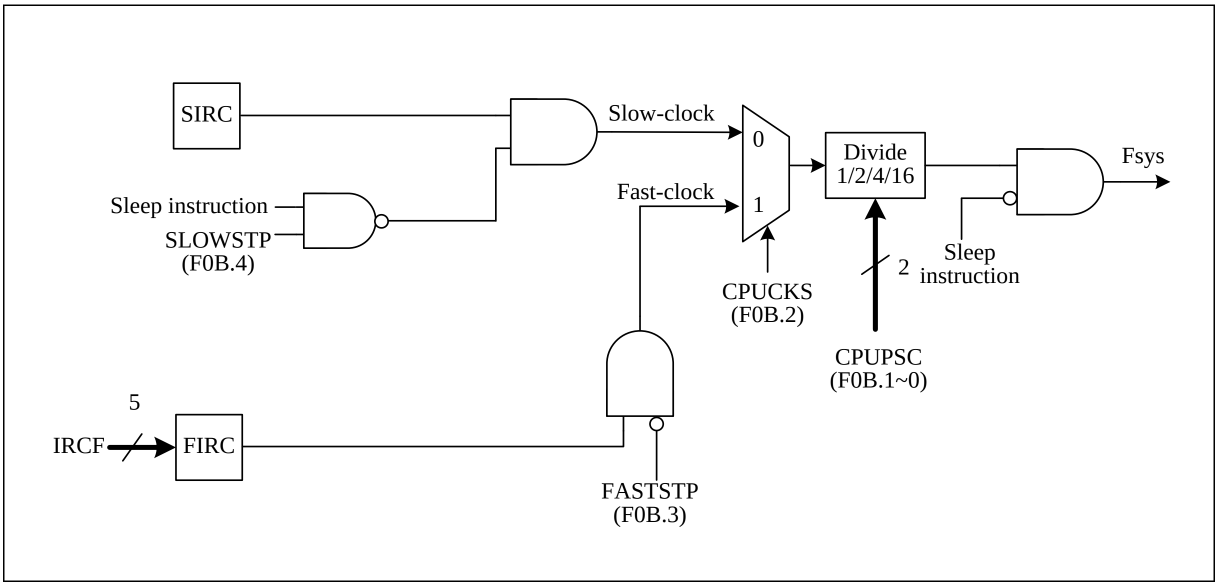
6. Clock Circuitry and Operation Mode

6.1. System Clock

The device is designed with dual-clock system. There are two kinds of clock source SIRC (Slow Internal RC) and FIRC (Fast Internal RC). Each clock source can be applied to CPU kernel as system clock. When in IDLE mode, only Slow-clock can be configured to keep oscillating to provide clock source to T2. Refer to the figure below.

After Reset, the device is running at 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, a 12 MHz System clock rate requires V_{CC} > 3.0V.

The CLKCTL (F0B) SFR controls the System clock operating. H/W automatically blocks the S/W abnormally setting for this register. Never write both FASTSTP=1 & CPUCKS=1. It is recommended to write CLKCTL bit by bit.



Clock Scheme Block Diagram

The frequency of FIRC (Fast Internal RC) can be adjusted by IRCF (F1F) . When IRCF=10h, frequency is the lowest. When IRCF=0Fh, 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=12 MHz after Power on Reset.

FAST Mode:

In this mode, the program is executed using Fast-clock as CPU clock (Fsys). The Timer0/1 blocks are driven by Fast-clock. T2 can be driven by Slow-clock or Fsys/128 by setting T2CKS (F0C.6).

SLOW Mode:

In this mode, Fast-clock is stopped and Slow-clock is enabled for power saving. All peripheral blocks (Timer0/1, PWM, T2, etc...) clock sources are Slow-clock in the SLOW mode.

IDLE Mode:

When SLOWSTP (F0B.4) is cleared, the Chip will enter the "IDLE Mode" after executing the SLEEP instruction. In this mode, the Slow-clock will continue running to provide clock to T2 block (if T2CKS = 0).

Another way to keep Slow-clock oscillation in IDLE mode is setting WKTIE=1 (F08.3) to keeping WKT running before executing the SLEEP instruction or WDTE=11 (CFGWH.9~8) to keeping WDT running. In such condition, the Slow-clock keeps working and wakes up CPU periodically no matter SLOWSTP is set or cleared.

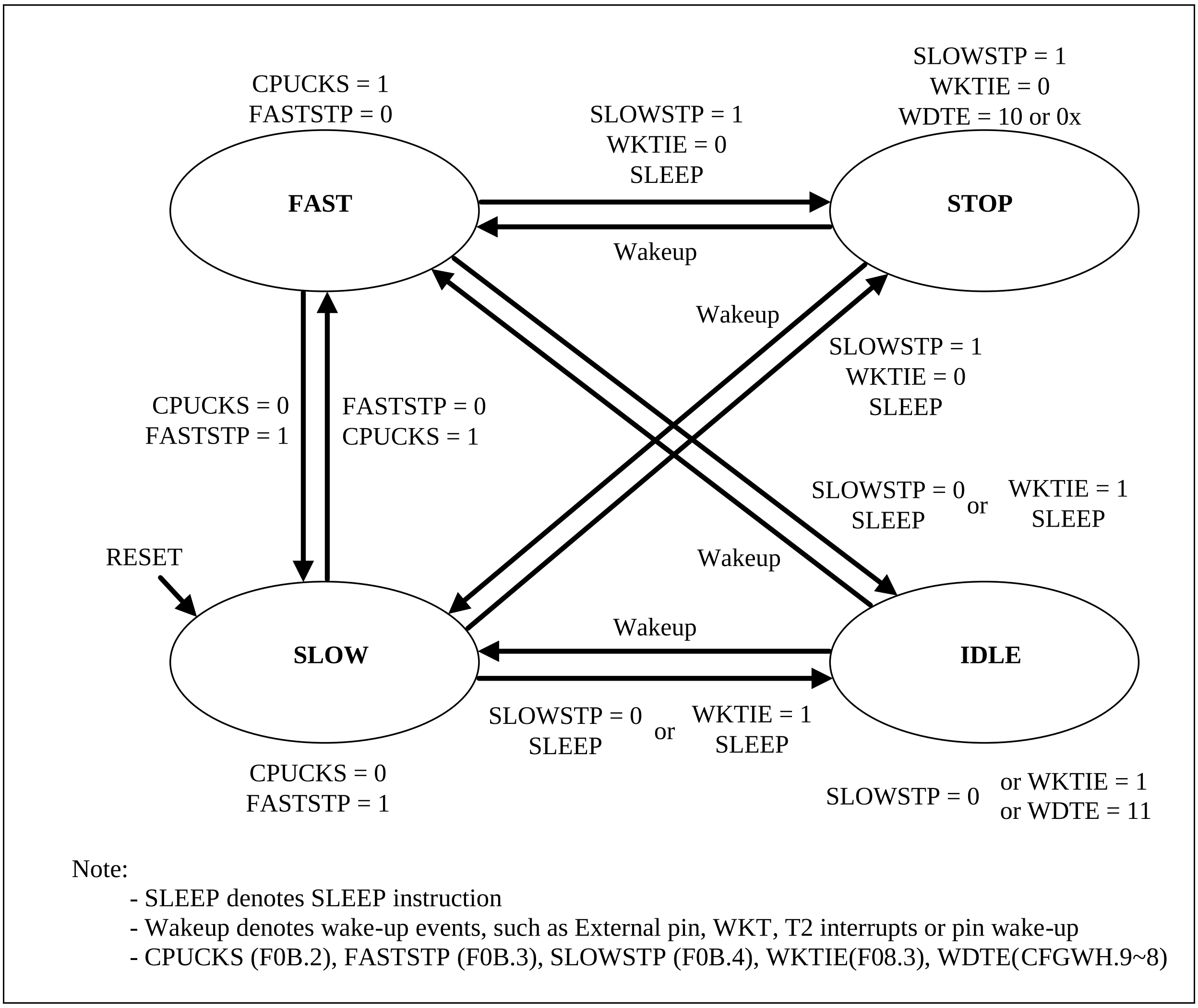
T2 and WKT/WDT are independent and have their own control registers. It is possible to keep both T2 and WKT working and wake-up in the IDLE mode, which is useful for low power mode Touch Key detection.

STOP Mode:

When SLOWSTP (F0B.4) is set, WKTIE (F08.3) is cleared and WDTE=10 or 0X, all blocks will be turned off and the Chip will enter the "STOP Mode" after executing the SLEEP instruction. STOP mode is similar to IDLE mode. The difference is all clock oscillators either Fast-clock or Slow-clock are stopped and no clocks are generated.

6.2. Dual System Clock Modes Transition

The device is designed with



CPU Mode & Clock Functions Table:

Mode	Oscillator	Fsys	Fast-clock	Slow-clock	TM0/1	WKT WDT	T2	ADC/TK	Wakeup event
FAST	FIRC	Fast-clock	Run	Run	Run	Run/Stop	Run	Run	X
SLOW	SIRC	Slow-clock	Stop	Run	Run	Run/Stop	Run	Run	X
IDLE	SIRC	Stop	Stop	Run	Stop	Run/Stop	Run/Stop	Stop	WKT/T2/IO
STOP	Stop	Stop	Stop	Stop	Stop	Stop	Stop	Stop	IO

FAST Mode transits to SLOW Mode:

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

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

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

Note: Stop Fast-clock (FASTSTP = 1) is optional. If PWM clock source is set to FIRC, FASTSTP must be cleared. Otherwise FIRC will not oscillate.

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

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

SLOW Mode transits to FAST Mode:

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

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

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

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

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

IDLE Mode Setting:

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

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

(2) Execute SLEEP instruction

IDLE mode can be woken up by interrupts XINT, WKT, T2, I2C or PB2-PB7 low level wake up.

◇ Example: Switch operating mode to IDLE mode

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

STOP Mode Setting:

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

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

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

(3) Execute SLEEP instruction

STOP mode can be woken up by interrupt XINT or PB2-PB7 pins low level wake up.

◇ Example: Switch operating mode to STOP mode

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

F0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCTL	–	–	–	SLOWSTP	FASTSTP	CPUCKS	CPUPSC	
R/W	–	–	–	R/W	R/W	R/W	R/W	
Reset	–	–	–	0	0	0	1	1

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

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

F0B.2 **CPUCKS**: System clock source select
0: Slow-clock
1: Fast-clock

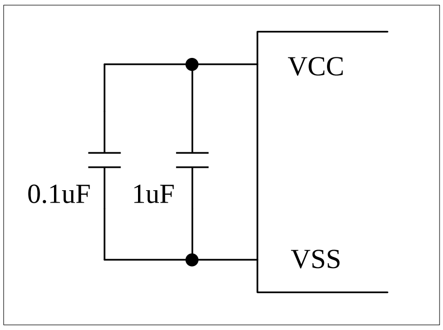
F0B.1~0 **CPUPSC**: System clock source prescaler. System clock source
00: divided by 16
01: divided by 4
10: divided by 2
11: divided by 1

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

6.3. System Clock Oscillator

In the Fast Internal RC (FIRC) mode, the on-chip oscillator generates 12 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.



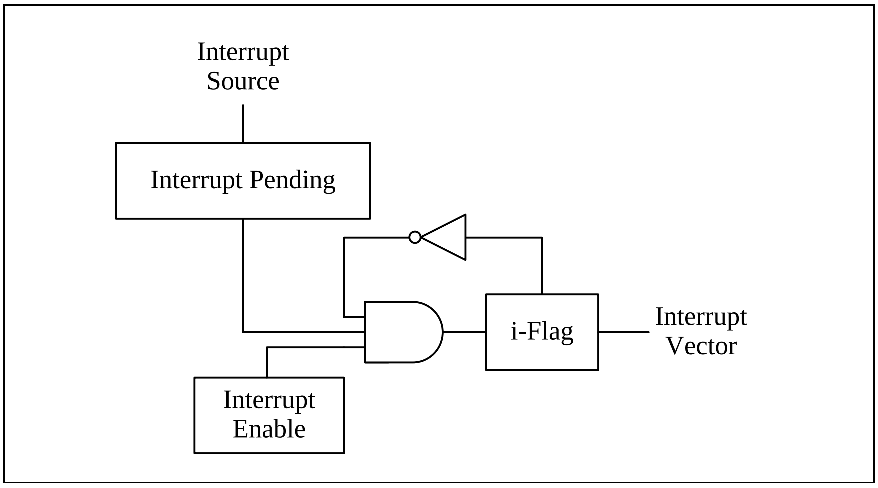
Internal RC Mode

7. Interrupt

The TM57M5406/08 has 1 level, 1 vector and 10 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 TM57M5406/08 has only 1 vector, there is not an interrupt priority register. The interrupt priority is determined by F/W.

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

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



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

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

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

          ORG       002H
START:
          MOVLW     xxxxxx00B
          MOVWR     PAMODL            ; Select INT0 Pin Mode as Mode0
                                      ; Open drain output low or input with Pull-up
          MOVLW     xxxxxxx1B
          MOVWF     PAD               ; Release INT0, it becomes Schmitt-trigger
                                      ; input with input pull-up resistor
          MOVLW     010xxxxxB
          MOVWR     MR0B              ; Set INT0 interrupt trigger as rising edge
          MOVLW     11111110B
          MOVWF     INTIF             ; Clear INT0 interrupt request flag
          MOVLW     00000001B
          MOVWF     INTIE             ; Enable INT0 interrupt
MAIN:
          …
          GOTO      MAIN

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

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

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADIE	T2IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F08.7 **ADIE**: ADC interrupt enable
0: disable
1: enable

F08.6 **T2IE**: T2 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) pin interrupt enable
0: disable
1: enable

F08.1 **INT1IE**: INT1 (PA4) pin interrupt enable
0: disable
1: enable

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

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADIF	T2IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F09.7 **ADIF**: ADC interrupt event pending flag
This bit is set while ADC is end of conversion, write 0 to this bit will clear this flag or sets the ADST bit to clear this flag.

F09.6 **T2IF**: T2 interrupt event pending flag
This bit is set by H/W while T2 overflows, write 0 to this bit will clear this flag

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

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

F09.3 **WKTIF**: WKT interrupt event pending flag
This bit is set by H/W while WKT time out, write 0 to this bit will clear this flag

F09.2 **INT2IF**: INT2 (PA7) interrupt event pending flag
This bit is set by H/W at INT2 pin's falling edge, write 0 to this bit will clear this flag

F09.1 **INT1IF**: INT1 (PA4) interrupt event pending flag
This bit is set by H/W at INT1 pin's falling edge, write 0 to this bit will clear this flag

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

F15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKFLG	–	–	TKM1IF	TKM0IF	TKIF	–	–	–
R/W	–	–	R/W	R/W	R/W	–	–	–
Reset	–	–	0	0	0	–	–	–

F15.5 **TKM1IF**: Touch Key Module1 interrupt event pending flag
This bit is set while Touch Key Module1 is end of conversion, write 0 to this bit will clear this flag or sets the TKM1SOC bit to clear this flag.

F15.4 **TKM0IF**: Touch Key Module0 interrupt event pending flag
This bit is set while Touch Key Module0 is end of conversion, write 0 to this bit will clear this flag or sets the TKM0SOC bit to clear this flag.

F15.3 **TKIF**: Touch Key interrupt event pending flag
This bit is set by H/W while Touch Key Module0 or Module1 are end of conversion, write 0 to this bit will clear all of Touch Key flag

F16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCTL	–	–	TKM1PD	TKM0PD	TKIE	TK3V	TKFDB	TKFSL
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	1	1	0	0	0	0

F16.3 **TKIE**: Touch Key interrupt enable
0: disable
1: enable

F1A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CCTL	–	–	–	I2CIE	I2CEN	–	I2CID	
R/W	–	–	–	R/W	R/W	–	R/W	
Reset	–	–	–	0	0	–	0	0

F1A.4 **I2CIE:** Slave I2C interrupt enable
0: disable
1: enable

F1B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CFLAG	–	I2CIF	TXD1F	TXD0F	RCD1OVF	RCD1F	RCD0OVF	RCD0F
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	0	0	0	0	0	0	0

F1B.6 **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

F1B.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

F1B.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

F1B.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

F1B.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

F1B.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

F1B.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

R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	HWAUTO	INT0EDG	T2PSC		WDTPSC		WKTPSC	
R/W	R/W	R/W	R/W		R/W		R/W	
Reset	0	0	0	0	1	1	1	1

R0B.4 **INT0EDG:** INT0 pin (PA0) edge interrupt event
0: falling edge to trigger
1: rising edge to trigger

8. I/O Port

8.1. PA0-4, PB0-7, PD0-3

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~PA4, PB0~PB7 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	PWM/TCOUT/TM1OUT/ADC	X	–	N	N
	Wakeup	0	–	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, IIC interface and Touch Key.

Pin Name	Wake-up	CKO	TK	ADC	others	Mode3
PA0	INT0				I2CSDA	PWM0A
PA1						PWM1A
PA2					I2CSCL/TM0CKI	PWM1B
PA3						PWM1C
PA4	INT1					PWM0B
PA7	INT2					–
PB0		TM1OUT	TK0			TM1OUT
PB1		TCOUT	TK1			TCOUT
PB2	Wakeup		TK2			Wakeup
PB3	Wakeup		TK3			Wakeup
PB4	Wakeup		TK4			Wakeup
PB5	Wakeup		TK5			Wakeup
PB6	Wakeup		TK6			Wakeup
PB7	Wakeup		TK7			Wakeup
PD0				ADC0		ADC0
PD1				ADC1		ADC1
PD2				ADC2		ADC2
PD3				ADC3		ADC3

PortA/B/D multi-function Table

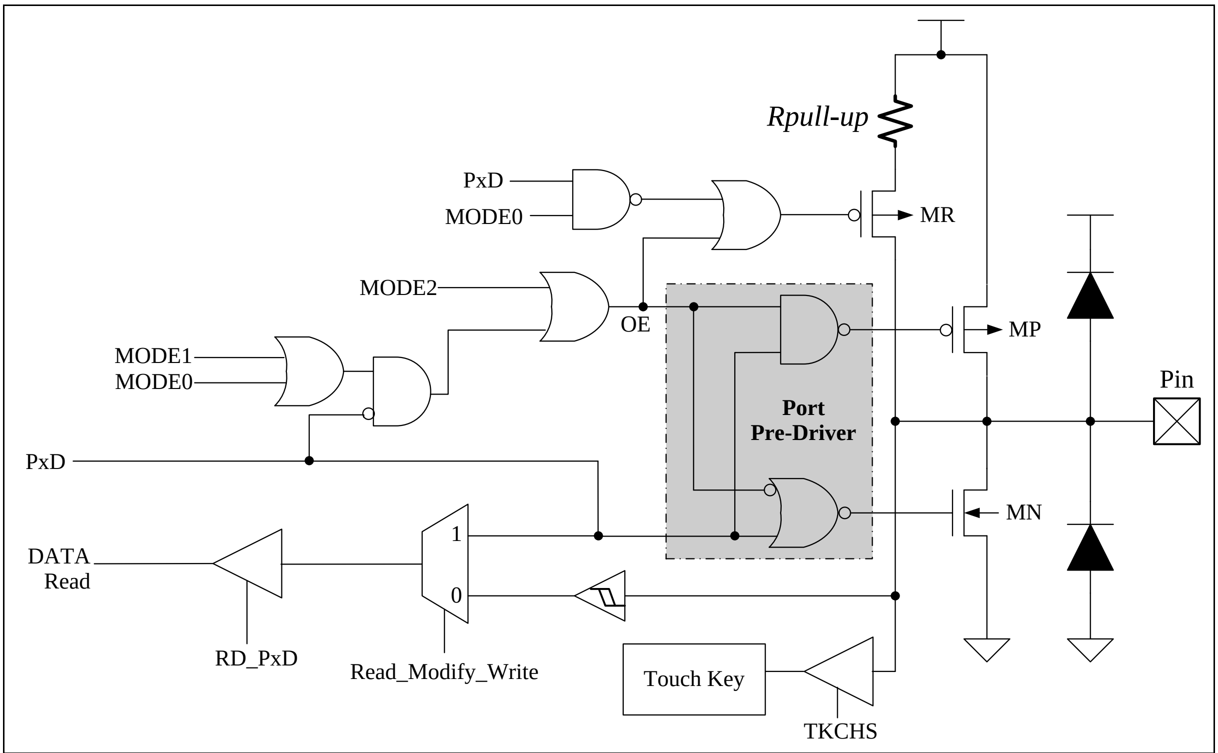
The necessary SFR setting for PA0-4, PB0-7 pin's alternative function is list below.

Alternative Function	Mode	PxD SFR data	Pin State	Other necessary SFR setting
INT0, INT1, INT2 TM0CKI	**0**	1	Input with Pull-up	INTxIE TM0CTL
	1	1	Input	
TK0~TK7	**0**	1	Touch Key Idling, Pull-up	TKCHS
			Touch Key Scanning	
ADC0~ADC3	**3**	X	ADC Channel input	ADCHS
I2CSCL	**0**	1	Input with Pull-up	I2CCTL
	1	1	Input	
I2CSDA	**0**	X	Input with Pull-up / Open Drain Output	
	1	X	Input / Open Drain Output	
PWM0A, PWM0B PWM1A, PWM1B, PWM1C	**3**	X	PWM Output (CMOS Push-Pull)	
TM1OUT, TCOUT	**3**	X	CMOS Push-Pull	
Wake-up	**3**	0	Input	
		1	Input with Pull-up	

Mode Setting for PA0-4, PB0-7, PD0-3 Alternative Function

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

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



General Pin Structure

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

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

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

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

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

```
MOVLW    xxxxxx10B
MOVWR    PAMODL
```

◇Example: Set PA0 as PWM0A push-pull output mode (Mode3)

```
MOVLW    xxxxxx11B
MOVWR    PAMODL          ; Set PA0 as mode3
```

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

F05.7~0 **PAD**: PA7~PA0 data

F06	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBD	PBD							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

F06.7~0 **PBD**: PB7~PB0 data

F07	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDD	–	–	–	–	PDD			
R/W	–	–	–	–	R/W			
Reset	–	–	–	–	1	1	1	1

F07.3~0 **PDD**: PD3~PD0 data

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

R05.6 **PA7MOD**: PA7 pull-up resistor enable
0: the pin pull-up resistor is enabled
1: the pin pull-up resistor is disabled

R05.1~0 **PA4MOD**: PA4 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PA4 as PWM0B push-pull output

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

R06.7~6 **PA3MOD**: PA3 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PA3 as PWM1C pin

R06.5~4 **PA2MOD**: PA2 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PA2 as PWM1B push-pull output

R06.3~2 **PA1MOD**: PA1 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2,
11: Mode3, PA1 as PWM1A push-pull output

R06.1~0 **PA0MOD**: PA0 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PA0 as PWM0A push-pull output

R07	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBMODH	PB7MOD		PB6MOD		PB5MOD		PB4MOD	
R/W	R/W		R/W		R/W		R/W	
Reset	0	1	0	1	0	1	0	1

R07.7~6 **PB7MOD**: PB7 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PB7 with wakeup function

R07.5~4 **PB6MOD**: PB6 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PB6 with wakeup function

R07.3~2 **PB5MOD**: PB5 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PB5 with wakeup function

R07.1~0 **PB4MOD**: PB4 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PB4 with wakeup function

R08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBMODL	PB3MOD		PB2MOD		PB1MOD		PB0MOD	
R/W	R/W		R/W		R/W		R/W	
Reset	0	1	0	1	0	1	0	1

R08.7~6 **PB3MOD**: PB3 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PB3 with wakeup function

R08.5~4 **PB2MOD**: PB2 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PB2 with wakeup function

R08.3~2 **PB1MOD**: PB1 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PB1 as TCOUT push-pull output

R08.1~0 **PB0MOD**: PB0 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PB0 as TM1OUT push-pull output

R0A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDMODL	PD3MOD		PD2MOD		PD1MOD		PD0MOD	
R/W	R/W		R/W		R/W		R/W	
Reset	0	1	0	1	0	1	0	1

R08.7~6 **PD3MOD**: PD3 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PD3 as ADC3 channel input

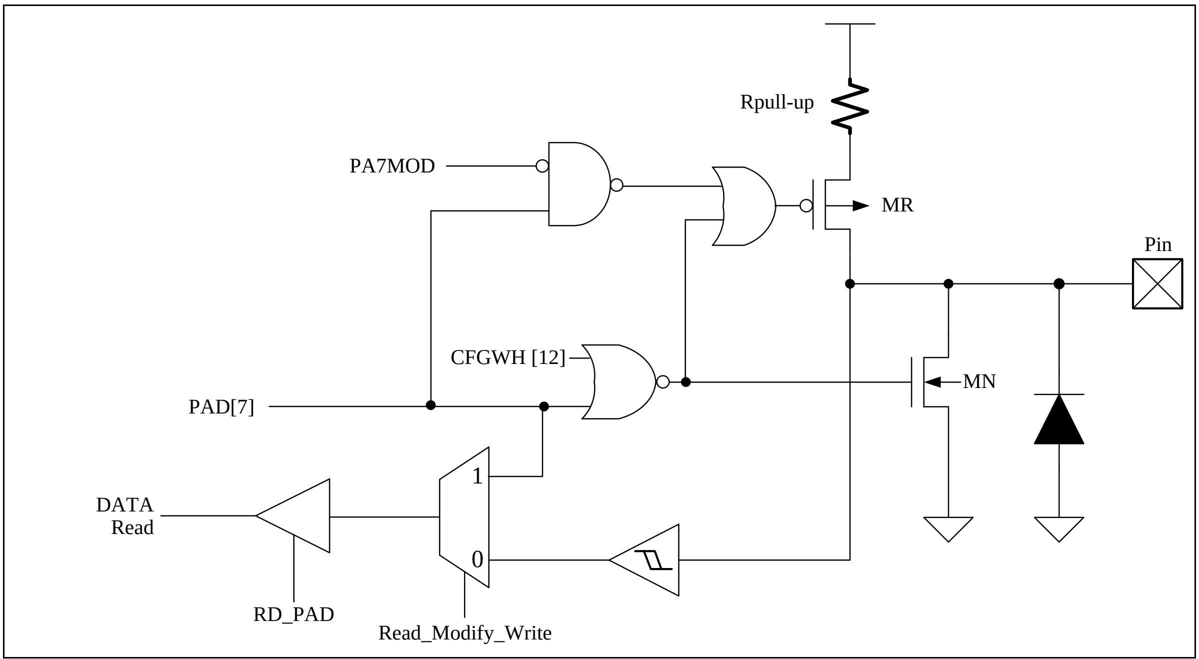
R08.5~4 **PD2MOD**: PD2 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PD2 as ADC2 channel input

R08.3~2 **PD1MOD**: PD1 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PD1 as ADC1 channel input

R08.1~0 **PD0MOD**: PD0 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PD0 as ADC0 channel input

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

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

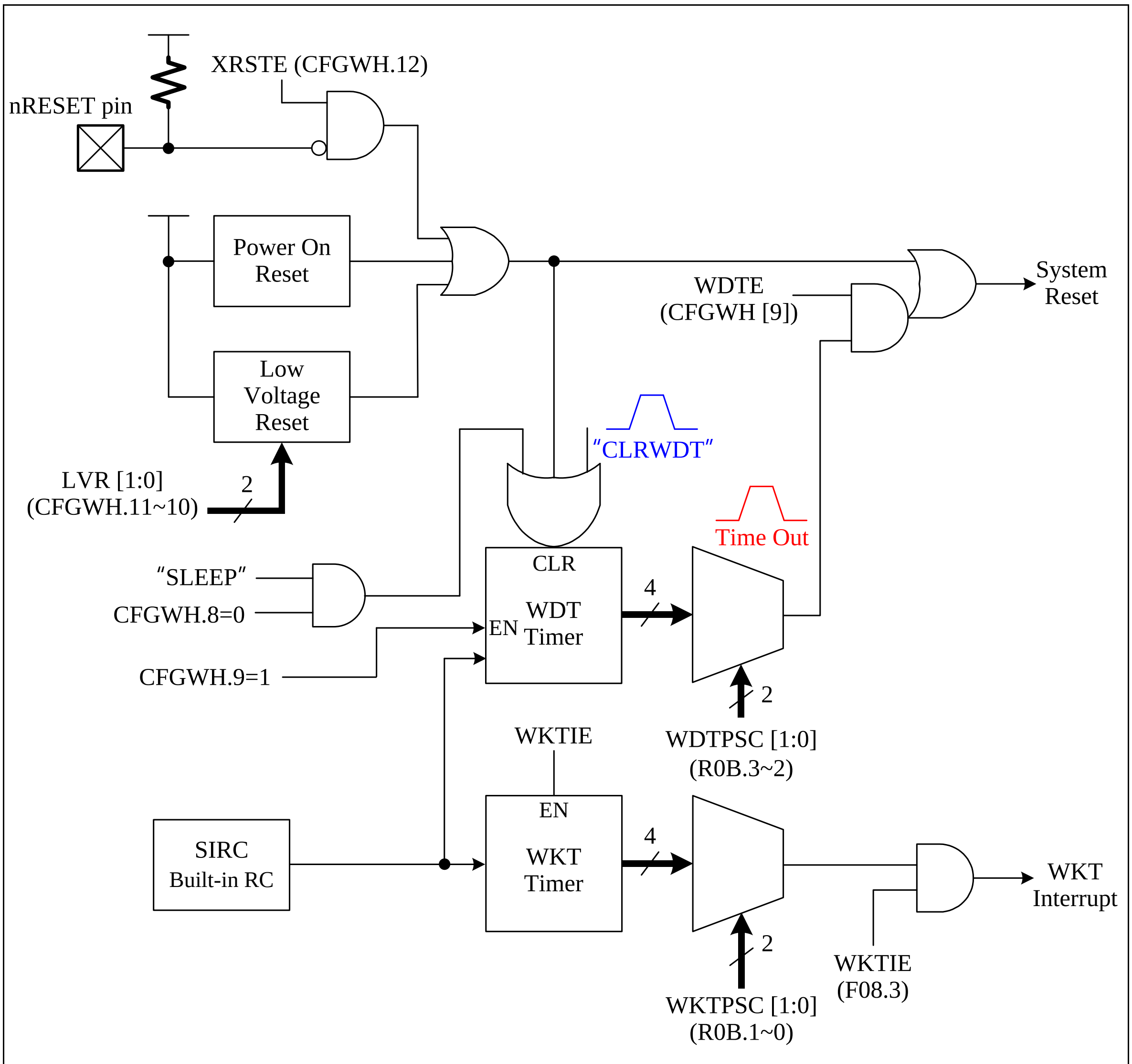
R05.6 **PA7MOD**: PA7 pull-up resistor enable
0: the pin pull-up resistor is enabled
1: the pin pull-up resistor is disabled

9. Peripheral Functional Block

9.1. Watchdog Timer (WDT) / Wakeup Timer (WKT)

The WDT has an independent internal RC Timer. The overflow period of WDT can be selected by WDTPSC. The WDT timer can be cleared by the CLRWDT instruction. If the WDT is enabled (CFGWH[9:8], WDTE=1X), the WDT generates the chip reset signal. The WDT works in both normal mode and power-down mode. Users can keep the WDT alive in power-down mode by setting WDTE=11.

The WKT has an independent internal RC Timer. The overflow period of WKT can be selected by WKTPSC. The WKT only generates overflow time out interrupt. The WKT works in both normal mode and IDLE mode. During IDLE mode, user can further choose to enable or disable the WKT by "WKTIE". If WKTIE is cleared in IDLE mode, the WKT internal RC Timer stops for power saving. In other words, user keeps the WKT alive in IDLE mode by setting WKTIE=1. Refer to the following table and figure



WDT/WKT Block Diagram

The WDT's behavior in different Mode is shown as below table.

Mode	WDTE[1]	WDTE[0]	WDT
Normal Mode	0	0	Stop
	0	1	Stop
	1	0	Run
	1	1	Run
Power-down Mode (SLEEP)	0	0	Stop
	0	1	Stop
	1	0	Stop
	1	1	Run

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

◇ Example: Clear watchdog timer by executing CLRWDT instruction.

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

◇ Example: Clear watchdog timer by writing WDTCLR register.

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

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

```
        MOVLW     00000111B             ; Write any value into WDTCLR register.
        MOVWR     R0B                   Select WDT Time out=256 ms @5V
        …
        SLEEP
```

◇ Example: Set WKT period and interrupt function.

```
        MOVLW     00000110B
        MOVWR     R0B                   ; Select WKT period=64 ms @5V.
        MOVLW     11110111B
        MOVWF     INTIF                 ; Don't use bit operation "BCF WKTIF"
                                        ; to clear interrupt flag
        MOVLW     00001000B
        MOVWF     F08                   ; Enable WKT interrupt function
```

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

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

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADIE	T2IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADIF	T2IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F09.3 **WKTIF**: WKT interrupt event pending flag
This bit is set by H/W while WKT time out, write 0 to this bit will clear this flag

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

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

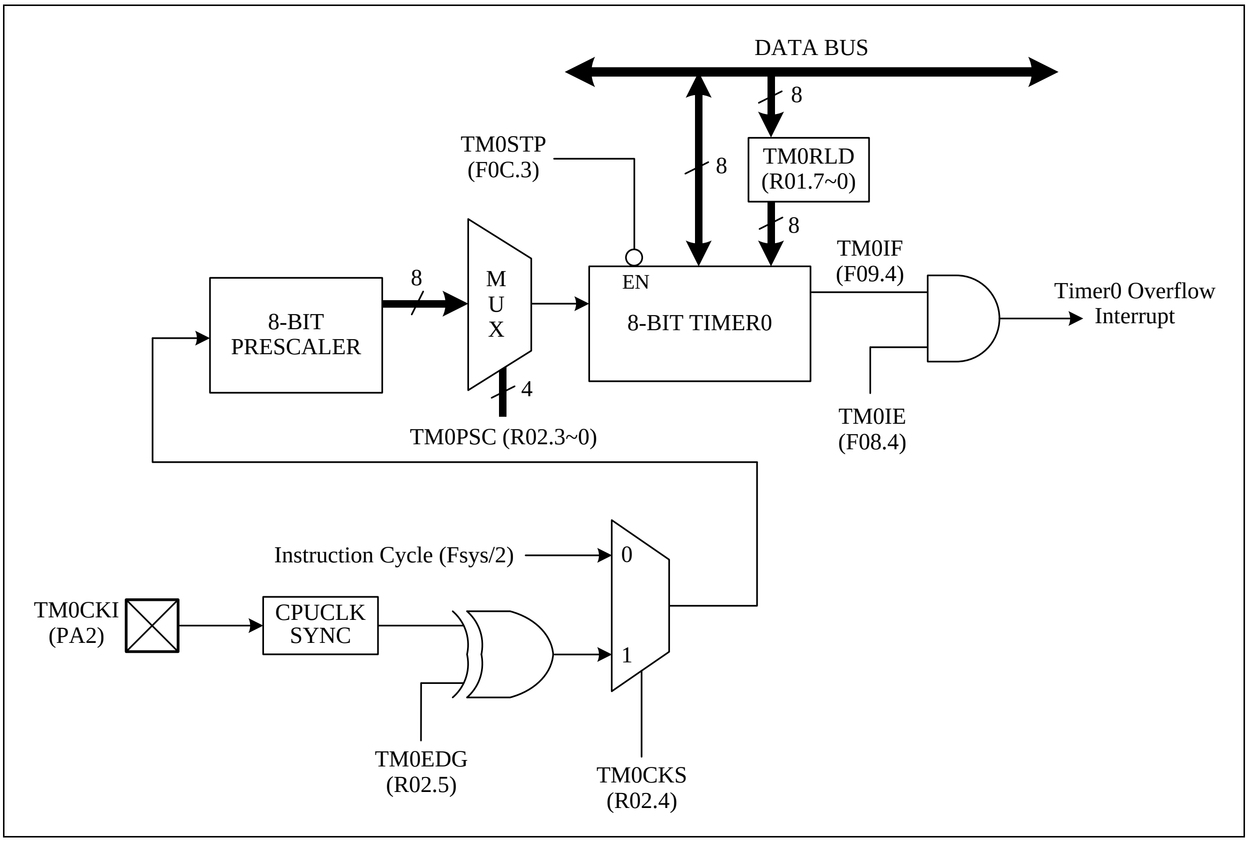
R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	HWAUTO	INT0EDG	T2PSC		WDTPSC		WKTPSC	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	1	1

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

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

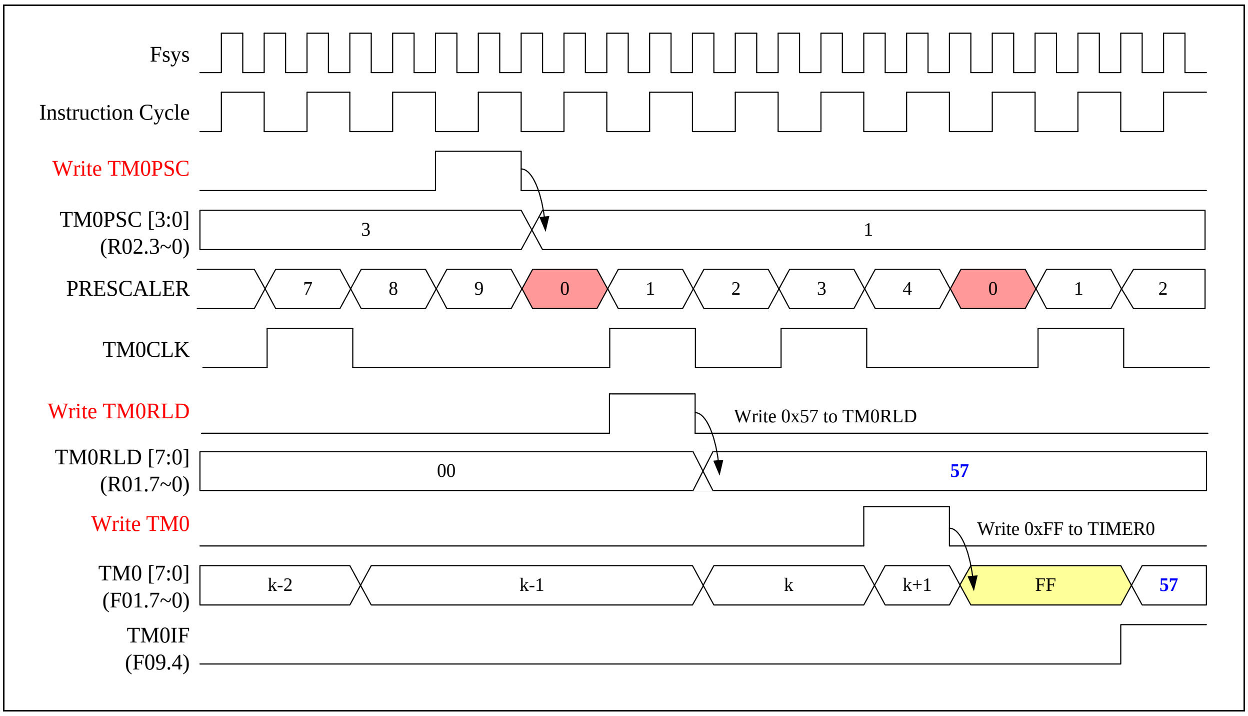
The Timer0 is an 8-bit wide register of F-Plane 01h (TM0). It can be read or written as any other register of F-Plane. Besides, Timer0 increases itself periodically and automatically rolls over based on the pre-scaled clock source. When Timer0 rolls over, TM0 will load offset value from TM0RLD. The clock source of Timer0 can be selected to 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

Timer Mode:

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



Timer0 works in Timer mode (TM0CKS = 0)

The equation of Timer0 interrupt frequency is as following:

Timer0 interrupt frequency = Fsys / 2 / TM0PSC / (256-TM0RLD)

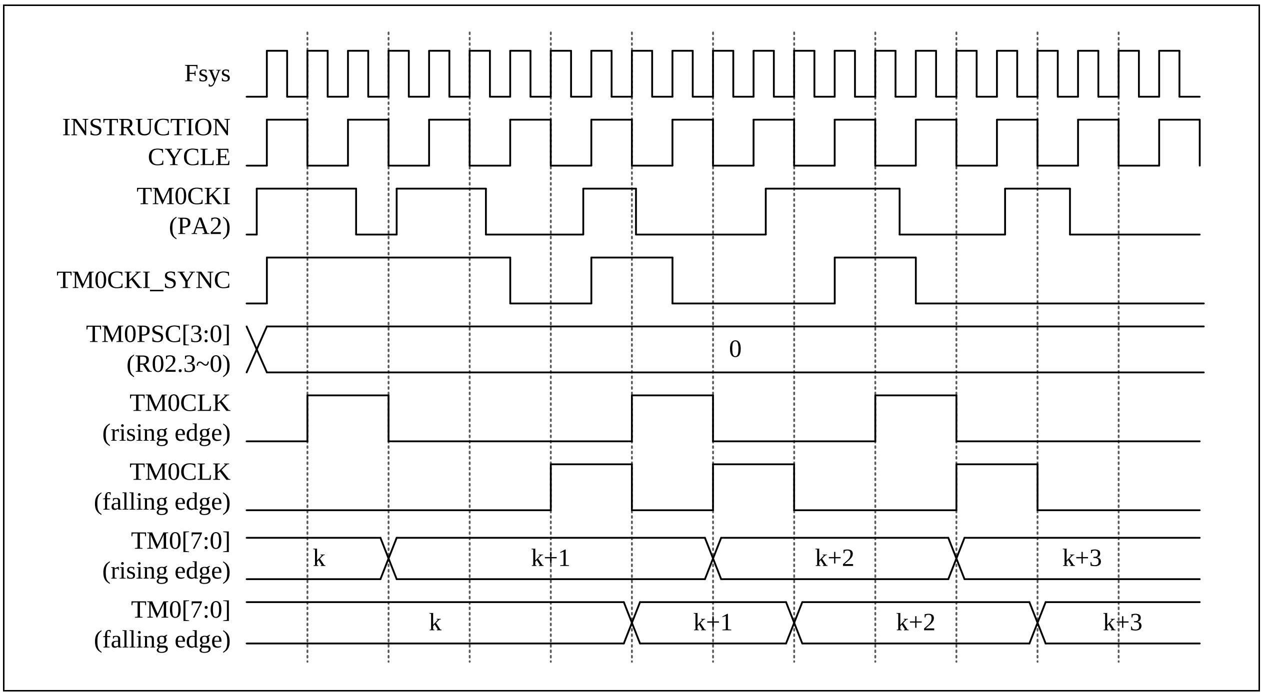
◇ Example: Setup Timer0 work in Timer mode, if Fsys = 12 MHz

```
; Setup Timer0 clock source and divider
        BSF       CPUCKS        ; Set Fast-clock as system clock
        MOVLW     00x00101B     ; TM0CKS = 0, Timer0 clock is instruction cycle
        MOVWR     TM0CTL        ; TM0PSC = 0101b, divided by 32

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

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

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

Timer0 interrupt frequency = Fsys / 2 / TM0PSC / (256-TM0RLD),
Fsys = 12MHz, TM0PSC = div 32
Timer0 interrupt frequency = 12 MHz /2 / 32 / (256-128) = 1.46 KHz

Counter Mode:

Counter Mode, the clock source is TM0CKI (PA2). 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. The following timing diagram describes the Timer0 works in Counter mode.



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

◇ Example: Setup Timer0 works in Counter mode

```
; Setup Timer0 clock source and divider
        MOVLW     00110000B     ; TM0EDG = 1, counting edge is falling edge
        MOVWR     TM0CTL        ; TM0CKS = 1, Timer0 clock is TM0CKI
                                ; TM0PSC = 0000b, divided by 1
; Setup Timer0
        BSF       TM0STP        ; Timer0 stops counting
        CLRF      TM0           ; Clear Timer0 content

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

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

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

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADIE	T2IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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	ADIF	T2IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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	T2CKS	T2CLR	–	TM0STP	TM1STP	PWM1CLR	PWM0CLR
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W	R/W
Reset	0	0	0	–	0	0	0	0

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

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

R01.7~0 **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	–	–	W	W	W			
Reset	–	–	0	0	0	0	0	0

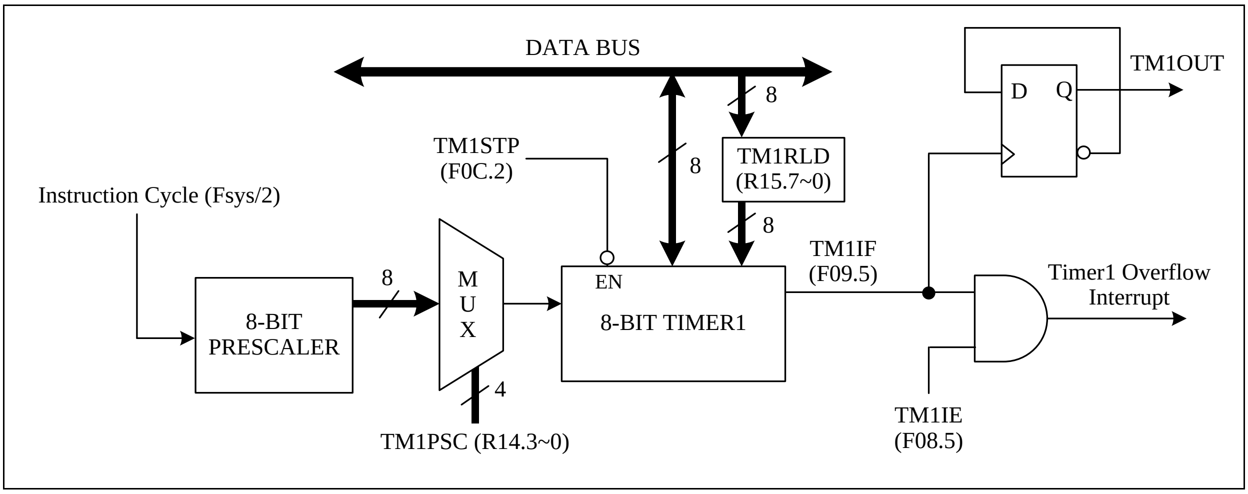
R02.5 **TM0EDG:** TM0CKI (PA2) edge selection for Timer0 prescaler count
0: TM0CKI rising edge for Timer0 prescaler count
1: TM0CKI falling edge for Timer0 prescaler count

R02.4 **TM0CKS:** Timer0 clock source select
0: Instruction Cycle (Fsys/2) as Timer0 prescaler clock
1: TM0CKI (PA2) as Timer0 prescaler clock

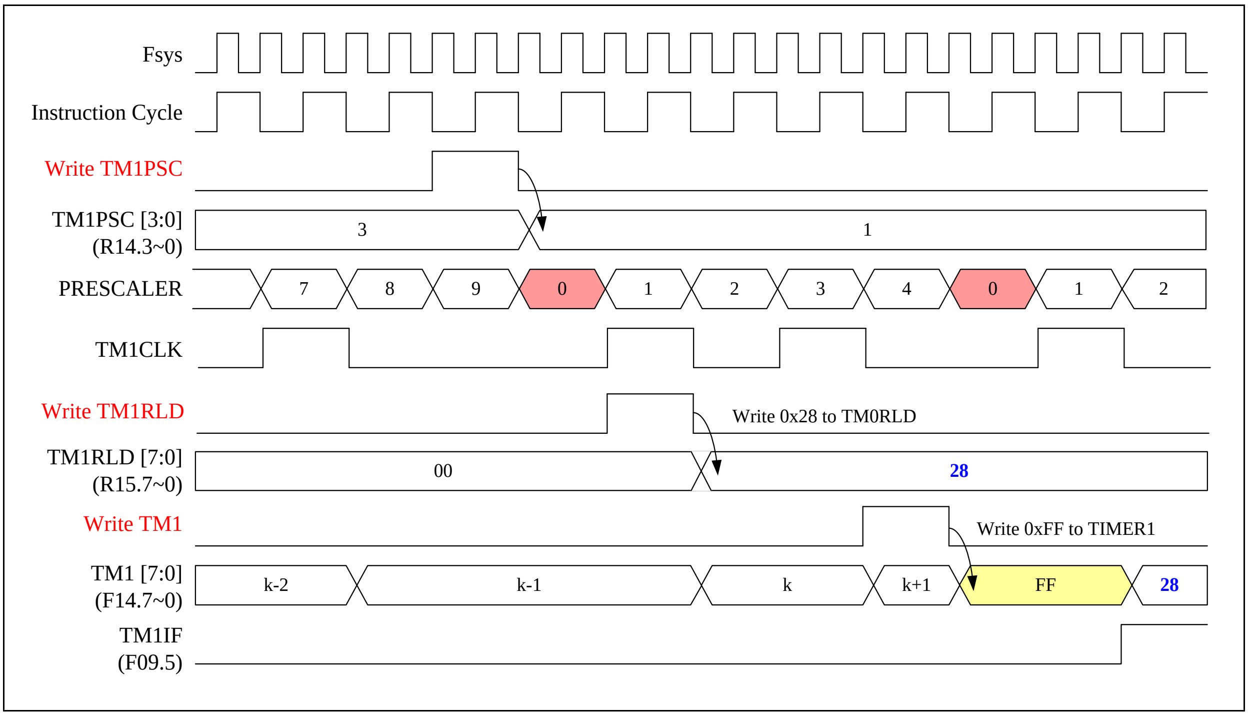
R02.3~0 **TM0PSC:** Timer0 prescaler. Timer0 clock source
0000: divided by 1 0001: divided by 2
0010: divided by 4 0011: divided by 8
0100: divided by 16 0101: divided by 32
0110: divided by 64 0111: divided by 128
1xxx: divided by 256

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

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



Timer1 Block Diagram



Timer1 works in Timer mode

The equation of Timer1 interrupt frequency is as following:

Timer1 interrupt frequency = Fsys / 2 / TM1PSC / (256-TM1RLD)

◇ Example: CPU is running in SLOW mode, Fsys = Slow-clock / CPUPSC= 128KHz / 2 = 64KHz

```
; Setup Timer1 clock source and divider
        MOVLW    00000010B      ; Set Slow-clock as system clock
        MOVWF    CLKCTL         ; CPUPSC = 10b, divided by 2
        MOVLW    00000101B
        MOVWR    TM1CTL         ; TM1PSC = 0101b, divided by 32

; Setup Timer1 reload data
        MOVLW    FFH
        MOVWR    TM1RLD         ; Set Timer1 reload data = 255

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

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

Timer1 clock source is Fsys/2 = 64 KHz / 2 = 32 KHz, Timer1 divided by 32
Timer1 interrupt frequency = 64KHz / 2 / 32 / (256-255) = 1 Hz

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADIE	T2IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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	ADIF	T2IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

F0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0C	CLKFLT	T2CKS	T2CLR	–	TM0STP	TM1STP	PWM1CLR	PWM0CLR
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W	R/W
Reset	0	0	0	–	0	0	0	0

F0C.2 **TM1STP:** Timer1 counter stop
0: Timer1 is counting
1: Timer1 stop counting

F14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1	TM1							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F14.7~0 **TM1:** Timer1 content

R14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1CTL	–	–	–	–	TM1PSC			
R/W	–	–	–	–	W			
Reset	–	–	–	–	0	0	0	0

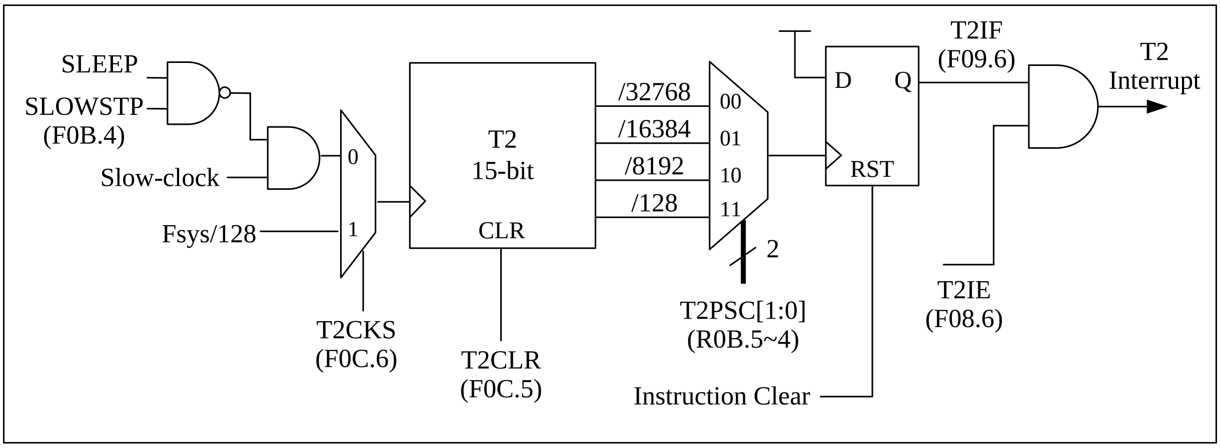
R14.3~0 **TM1PSC:** Timer1 prescaler. Timer1 clock source (Fsys/2)
0000: divided by 1
0001: divided by 2
0010: divided by 4
0011: divided by 8
0100: divided by 16
0101: divided by 32
0110: divided by 64
0111: divided by 128
1xxx: divided by 256

R15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1RLD	TM1RLD							
R/W	W							
Reset	0	0	0	0	0	0	0	0

R15.7~0 **TM1RLD:** Timer1 reload data

9.4. T2: 15-bit Timer

The T2 is a 15-bit counter and the clock sources are from either Fsys/128 or Slow-clock. The clock source is used to generate time base interrupt and T2 counter block clock. It is selected by T2CKS. The T2's 15-bit content cannot be read by instructions. It generates interrupt flag T2IF with the clock divided by 32768, 16384, 8192, or 128 depends on the T2PSC[1:0] bits. The following figure shows the block diagram of T2.



T2 Block Diagram

◇Example: CPU is running at FAST mode, Fsys = Fast-clock / CPUPSC = FIRC 12 MHz / 4,

T2 clock source is Fsys/128

```
; Setup FIRC frequency
        MOVLW     00000001B
        MOVWF     CLKCTL              ; Fsys is 3 MHz

; Setup T2 clock source and divider
        BSF       T2CKS               ; T2CKS = 1, T2 clock source is Fsys/128
        MOVLW     00011111
        MOVWR     R0B                 ; T2PSC = 01b, divided by 16384
        BSF       T2CLR               ; T2CLR = 1, clear T2 counter

; Enable T2 interrupt function
        MOVLW     10111111B
        MOVWF     INTIF               ; Clear T2 request interrupt flag
        BSF       T2IE                ; Enable T2 interrupt function
```

F0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0C	CLKFLT	T2CKS	T2CLR	–	TM0STP	TM1STP	PWM1CLR	PWM0CLR
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W	R/W
Reset	0	0	0	–	0	0	0	0

F0C.0 **T2CLR:** T2 counter clear
0: T2 is counting
1: T2 is cleared immediately, this bit is auto cleared by H/W

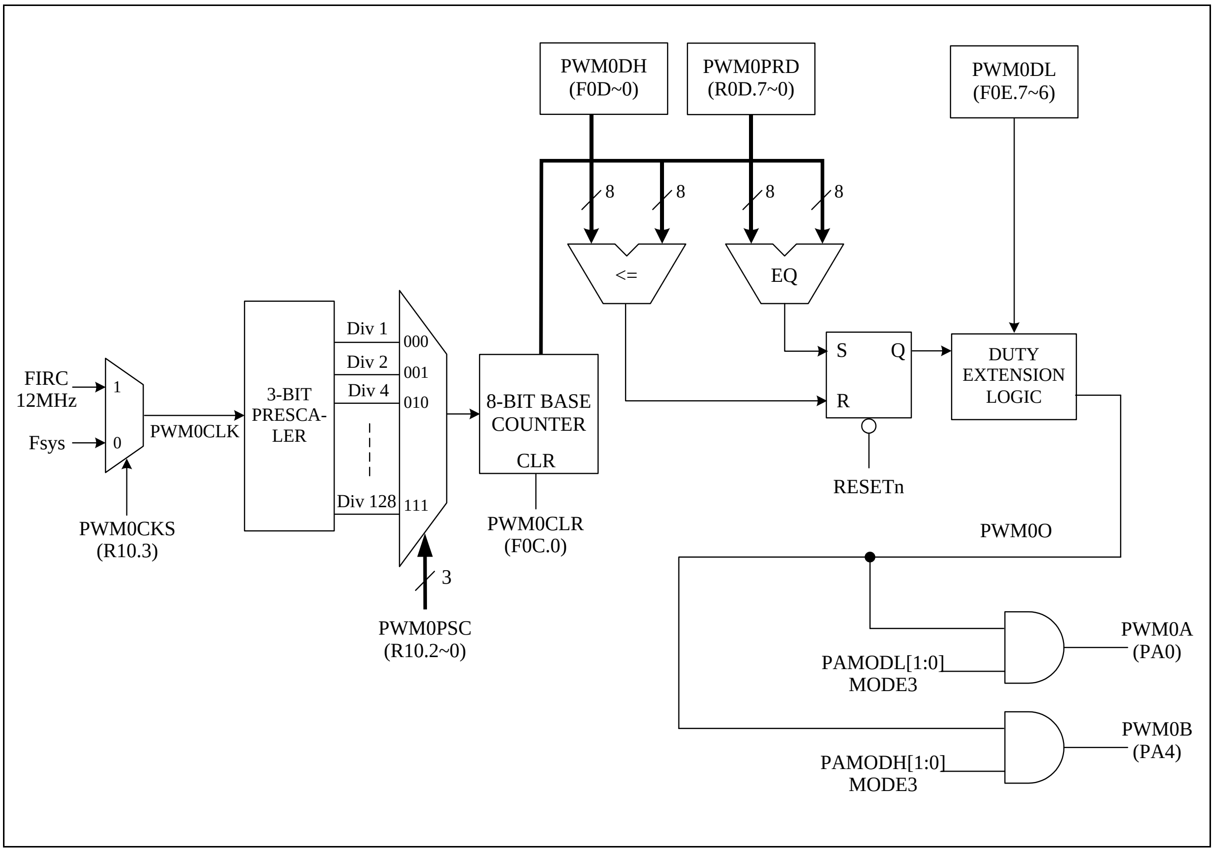
R0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MR0B	HWAUTO	INT0EDG	T2PSC		WDTPSC		WKTPSC	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	1	1

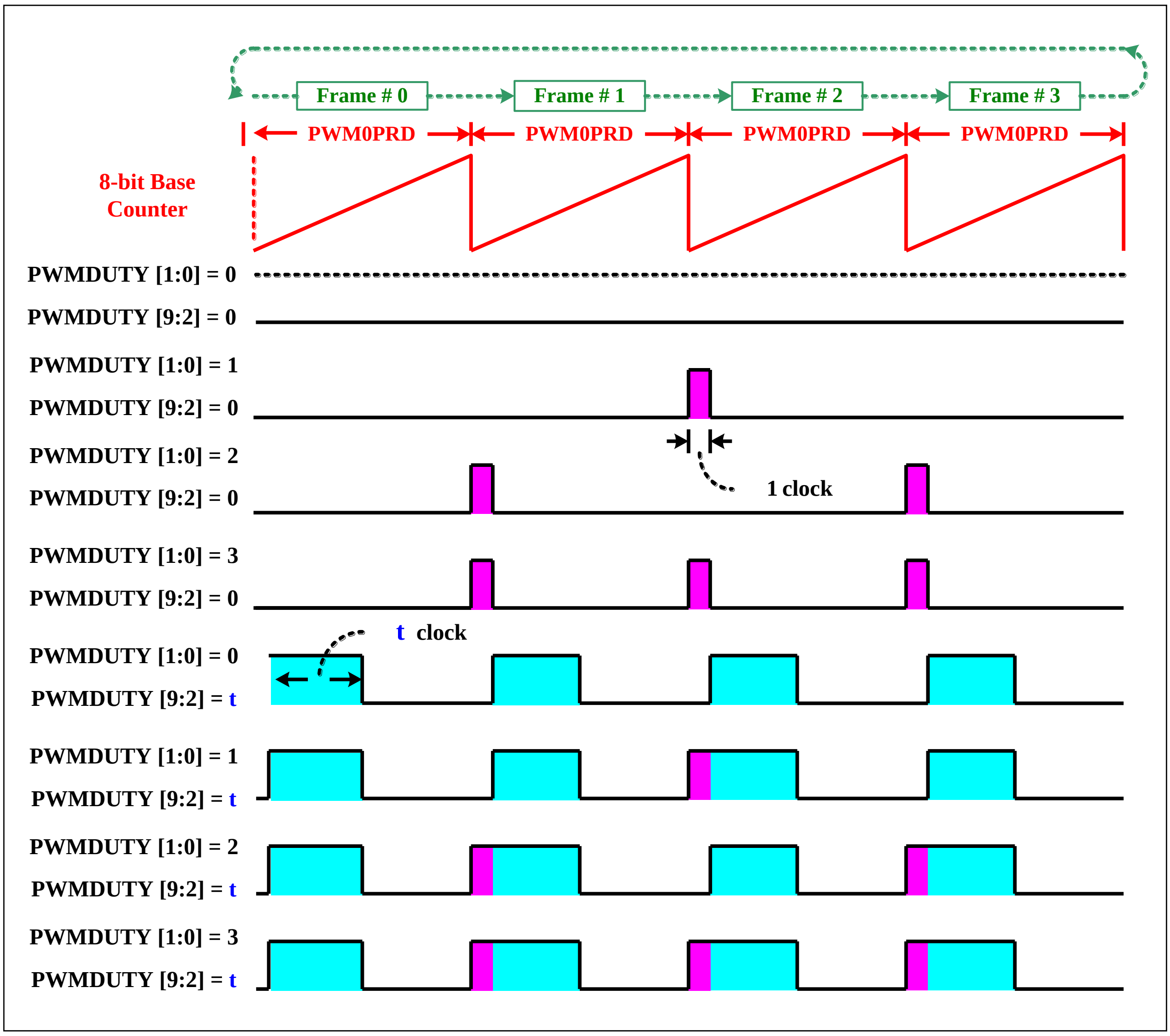
R0B.5~4 **T2PSC:** T2 pre-scale
00: div 32768
01: div 16384
10: div 8192
11: div 128

9.5. PWM0: (8+2) bits PWM

The PWM can generate various frequencies with 1024 duty resolution based on PWM0CLK, which can select Fsys or FIRC 12 MHz, decided by PWM0CKS. 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. When the base counter rolls over, the 2-bit LSB of PWM duty register PWM0DL 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. 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 PWM0PRD 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 8+2 Timing Diagram

◇ Example: [CPU running at Fast mode, Fsys=FIRC 12 MHz]

```
; Setup PWM0 clock prescaler
        MOLVW     00001110       ; PWM0 clock source=FIRC 12 MHz
        MOVWR     R10            ; PWM0 prescaler/64
        MOVLW     80H
        MOVWR     PWM0PRD        ; Set PWM0 period=80H
        MOVLW     00000000B
        MOVWF     PWM0DL         ; Set PWM0DL duty=00H
        MOVLW     20H
        MOVWF     PWM0DH         ; Set PWM0DH duty=20H
        BCF       PWM0CLR        ; Release PWM0 Clear flag
        MOVLW     00000011
        MOVWR     PAMODL         ; Enable PWM0A (PA0) output
        MOVLW     00000011
        MOVWR     PAMODH         ; Enable PWM0B (PA4) output
```

Example:

PWM0 clock source=FIRC 12 MHz, PWM0PSC=/64, PWM0PRD=80H,

PWM0DL=00H, PWM0DH=20H

PWM0 output frequency=12 MHz/64/ (PWMPROD+1) =12 MHz/64/129=1453.5 Hz.

PWM0 output duty=32:129=24.8%

F0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0C	CLKFLT	T2CKS	T2CLR	–	TM0STP	TM1STP	PWM1CLR	PWM0CLR
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W	R/W
Reset	0	0	0	–	0	0	0	0

F0C.0 **PWM0CLR:** PWM0 Clear and Hold
- 0: PWM0 Running
- 1: PWM0 Clear and Hold

F0D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DH	PWM0DH							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

F0D.7~0 **PWM0DH**: PWM0 duty 8-bit MSB

F0E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DL	PWM0DL		–	–	–	–	–	–
R/W	R/W		–	–	–	–	–	–
Reset	0	0	–	–	–	–	–	–

F0E.7~0 **PWM0DL**: PWM0 duty 2-bit LSB

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

R05.1~0 **PA4MOD**: PA4 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PA4 as PWM0B push-pull output

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

R06.1~0 **PA0MOD**: PA0 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PA0 as PWM0A push-pull output

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

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

R10	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMCTL	PWM1CKS	PWM1PSC			PWM0CKS	PWM0PSC		
R/W	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	0

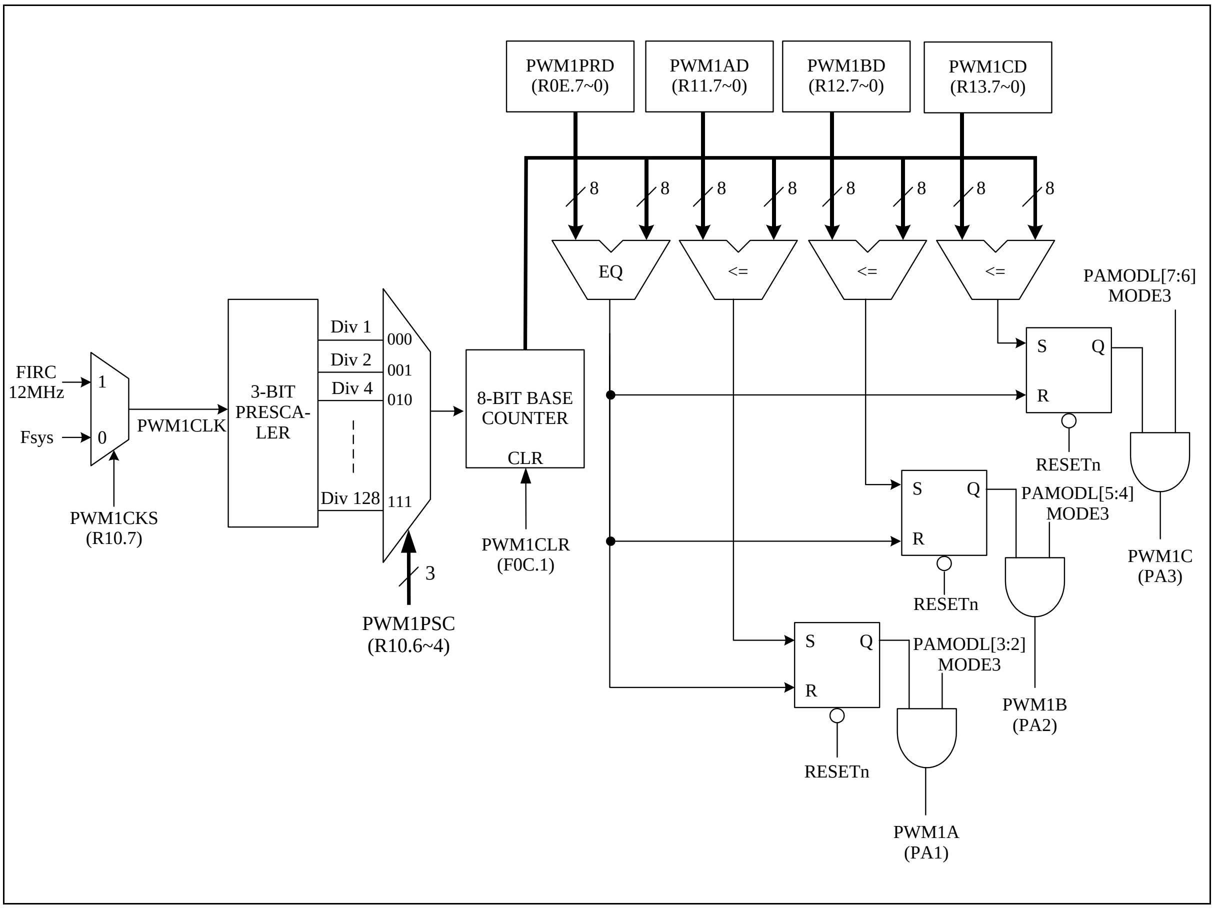
R10.3 **PWM0CKS**: PWM0 clock source
0: Fsys
1: FIRC (16MHz)

R10.2~0 **PWM0PSC**: PWM0 prescaler
000: divided by 1
001: divided by 2
010: divided by 4
…
111: divided by 128

9.6. PWM1A/PWM1B/PWM1C: 8 bits PWMs

PWM1A/PWM1B/PWM1C has independent duty and common period. The PWM1 can generate various frequencies with 256 duty resolution based on PWM1CLK, which can select Fsys or FIRC 12 MHz, decided by PWM1PSC. 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 PWM1PRD register. 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 PWM1PRD. 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.



F0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0C	CLKFLT	T2CKS	T2CLR	–	TM0STP	TM1STP	PWM1CLR	PWM0CLR
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W	R/W
Reset	0	0	0	–	0	0	0	0

F0C.1 **PWM1CLR:** PWM1 Clear and Hold
0: PWM1 Running
1: PWM1 Clear and Hold

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

R06.7~6 **PA3MOD**: PA3 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PA3 as PWM1C push-pull output

R06.5~4 **PA2MOD**: PA2 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PA2 as PWM1B push-pull output

R06.3~2 **PA1MOD**: PA1 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, PA1 as PWM1A push-pull output

R0E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1PRD	PWM1PRD							
R/W	W							
Reset	1	1	1	1	1	1	1	1

R0E.7~0 **PWM1PRD**: PWM1 period data

R10	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMCTL	PWM1CKS	PWM1PSC			PWM0CKS	PWM0PSC		
R/W	R/W	R/W			R/W	R/W		
Reset	0	0	0	0	0	0	0	0

R10.7 **PWM1CKS**: PWM1 clock source
0: Fsys
1: FIRC (12MHz)

R10.6~4 **PWM1PSC**: PWM1 prescaler
000: divided by 1
001: divided by 2
010: divided by 4
…
111: divided by 128

R11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1AD	PWM1AD							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

R11.7~0 **PWM1AD**: PWM1A duty

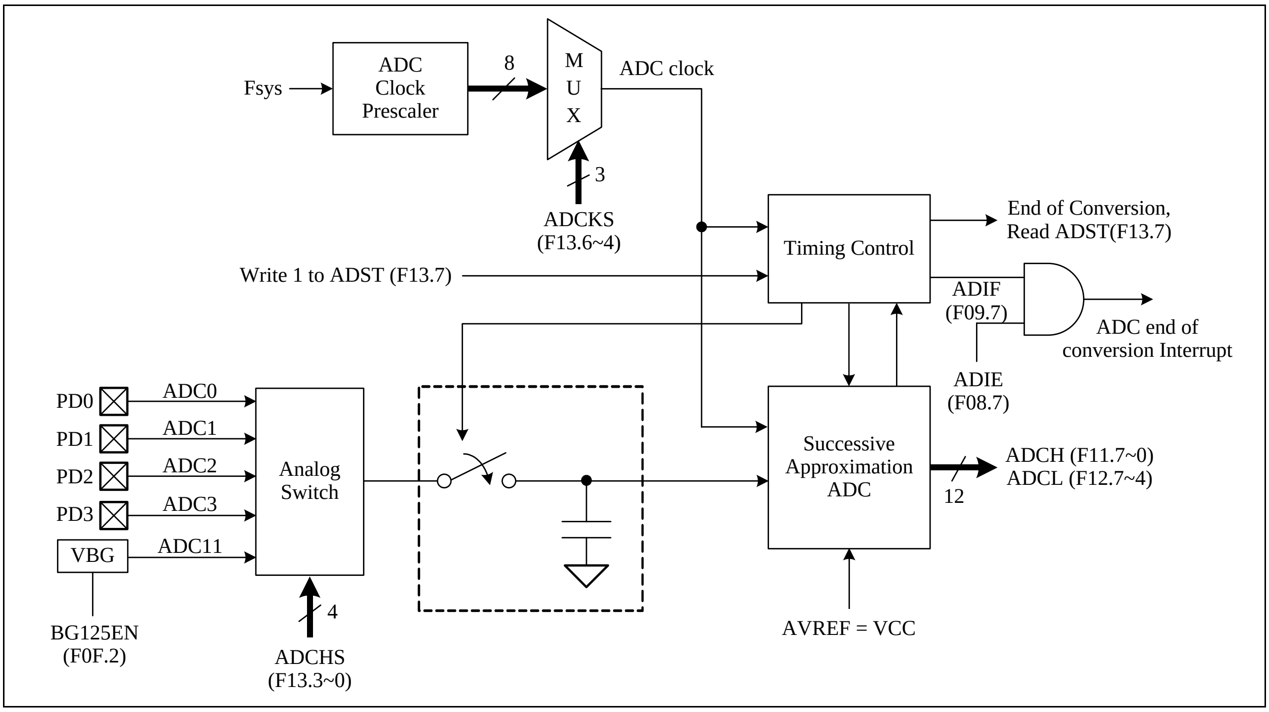
R12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1BD	PWM1BD							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

R12.7~0 **PWM1BD**: PWM1B duty

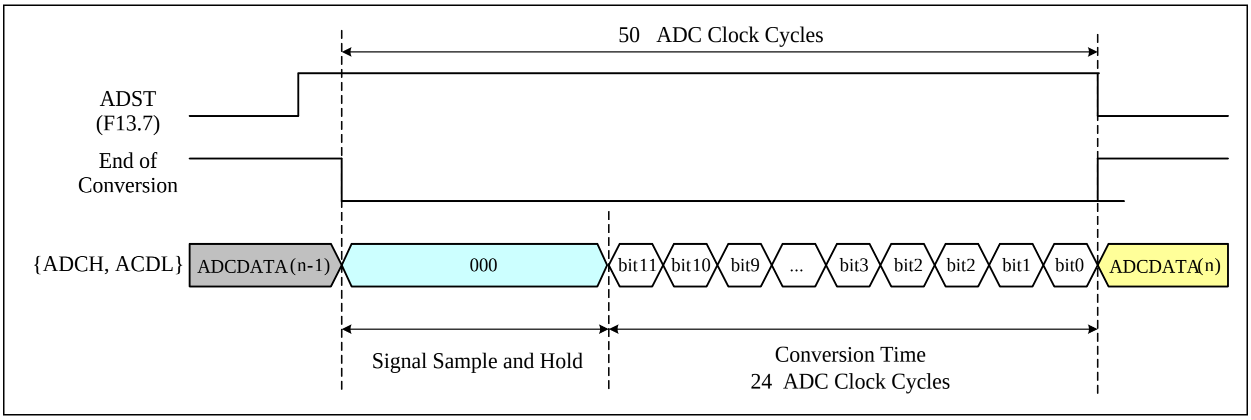
R13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CD	PWM1CD							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

R13.7~0 **PWM1CD**: PWM1C duty

9.7. Analog-to-Digital Converter



The 12-bit ADC (Analog to Digital Converter) consists of a 5-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 (F13.6~4) to choose a proper ADC clock frequency, which must be less than 1 MHz. User then launches the ADC conversion by setting the ADST (F13.7) control bit. After end of conversion, H/W automatic clears the ADST (F13.7) bit and set the ADC end of conversion flag ADIF (F08.7) bit. User can poll the ADST bit to know the conversion status or set the ADIE (F08.7) to generate ADC interrupt after conversion. About pin mode configuration, user needs to set the I/O mode as Mode3 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 (F13.3~0) to choose the input channel of ADC. One of them, ADC11 is Bandgap voltage 1.25V input for ADC (make sure BG125EN=1 before used). But, for better results, user needs delay 30 uS after setting the ADC input channel = ADC11, then begin to use ADC again. In TM57M5406/08, ADC reference voltage is VCC. It should be noted that the voltage of ADC input channel can’t exceed 0.95*VCC.



Example:

[CPU running at FAST mode , Fsys = FIRC 12 MHz / 2]

ADC clock frequency = 750 KHz, ADC channel = ADC2 (PD2).

◇ Example:

```
; Setup ADC clock
            MOVLW  xxx001 10B      ; F0B.2 = 1 (CPUCKS), Fsys = Fast-clock
            MOVWF  CLKCTL          ; F0B.1~0 = 2 (CPUPSC), divided by 2
                                   ; Fsys = 12 MHz/2 = 6 MHz
            MOVLW  x101xxxxB
            MOVWF  ADCTL                 ; F13.6~4 (ADCKS), ADC clock = Fsys/8 = 750 KHz

; Setup Pin mode
            MOVLW  xx11xxxxB       ; R0A.5~4 = 3 (PD2MOD), PD2 Pin mode = Mode3
            MOVWR  PDMODL

            MOVLW  01010010B       ; F13.3~0 = 2 (ADCHS), ADC select ADC2 (PD2 pin)
            MOVWF  ADCTL
            CALL   DELAY30uS


            BSF    ADST            ; F13.7 (ADST), ADC start conversion
            CALL   DELAY2uS

WAIT_ADC:
            BTFSC  ADST            ; Wait ADC conversion finish
            GOTO   WAIT_ADC

            MOVFW  ADCH            ; F11.7~0, Read ADC result [11:4] into W
            MOVFW  ADTKD           ; F12.7~4, Read ADC result [3:0] into W

            :
            :
```

F08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE	ADIE	T2IE	TM1IE	TM0IE	WKTIE	INT2IE	INT1IE	INT0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F08.7 **ADIE**: ADC interrupt enable
0: disable
1: enable

F09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF	ADIF	T2IF	TM1IF	TM0IF	WKTIF	INT2IF	INT1IF	INT0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F09.7 **ADIF**: ADC interrupt event pending flag
This bit is set while ADC is end of conversion, write 0 to this bit will clear this flag or sets the ADST bit to clear this flag.

F0F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0F	–	–	TKM1SOC	TKM1SOC	LVR2SAV	BG125EN	LDOSAV	MODE3V
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	1	1	1	0

F0F.2 **BG125EN:** Internal Bandgap voltage 1.25V enable
0: Internal Bandgap voltage disable
1: Internal Bandgap voltage enable

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

F11.7~0 **ADCH:** ADC output data MSB[11~4]

F12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADTKD	ADCL				–	–	TKM1EOC	TKM0EOC
R/W	R				–	–	R	R
Reset	–	–	–	–	–	–	–	–

F12.7~4 **ADCL:** ADC output data LSB[3~0]

F13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCTL	ADST	ADCKS			ADCHS			
R/W	R/W	R/W			R/W			
Reset	0	0	0	0	0	0	0	0

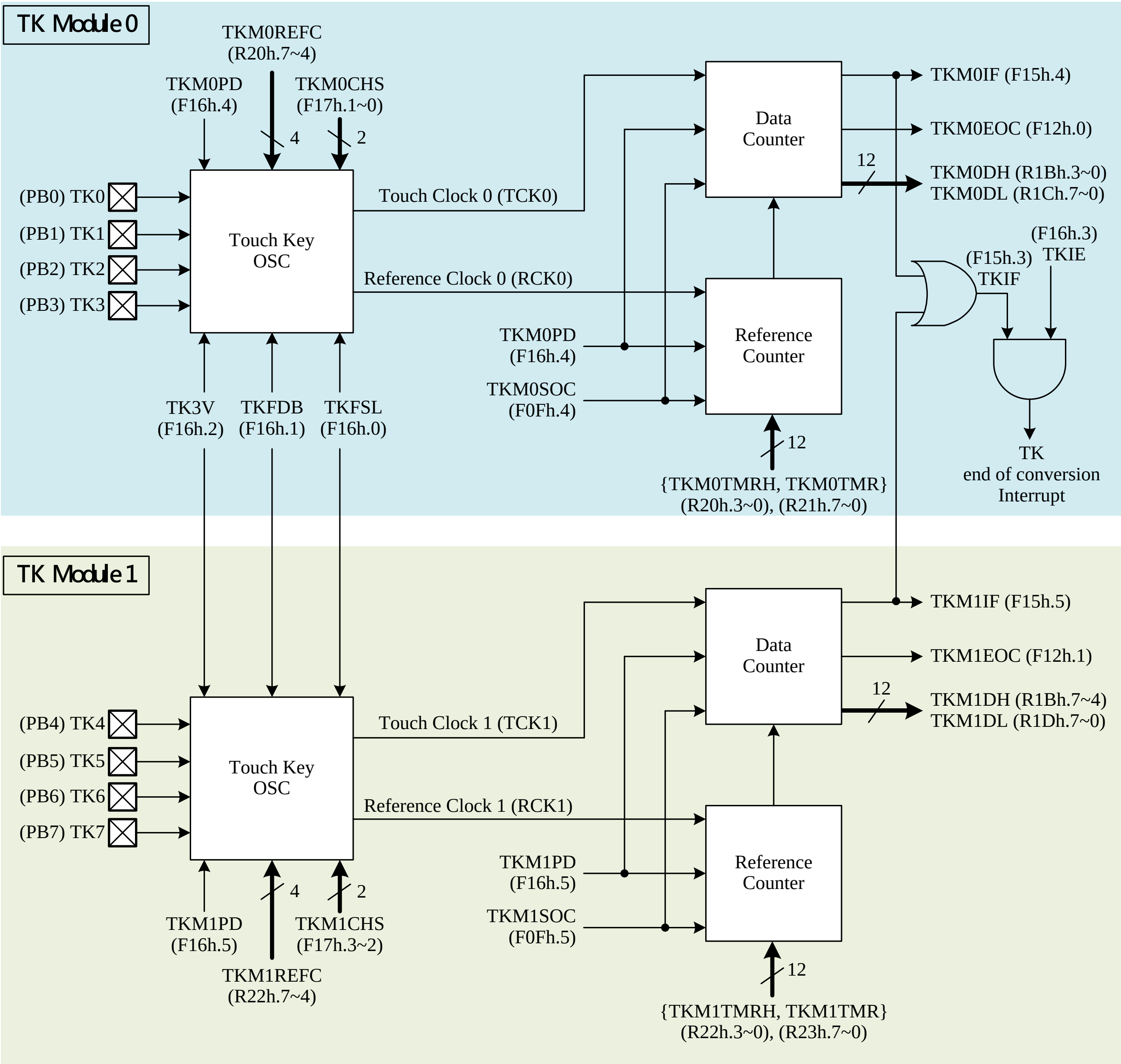
F13.7 **ADST:** ADC start bit.
0: H/W clear after end of conversion
1: ADC start conversion

F13.6~4 **ADCKS:** ADC clock frequency (Fadc) select
000: Fsys/256
001: Fsys/128
010: Fsys/64
011: Fsys/32
100: Fsys/16
101: Fsys/8
110: Fsys/4
111: Fsys/2

F13.3~0 **ADCHS:** ADC channel select
0000: ADC0 (PD0)
0001: ADC1 (PD1)
0010: ADC2 (PD2)
0011: ADC3 (PD3)
1011: VBG (1.25V)
others: Reserved

9.8. Touch Key

The Touch Key offers an easy, simple and reliable method to implement finger touch detection. In most applications, it doesn't require any external component. The device support 2 modules, 8 channels touch key detection.



Touch Key Block Diagram

To use the Touch Key, user must setup the pin mode correctly as below table. Setting Mode0 for an Idling Touch Key pin can pull up the pin and reduce the Key's mutual interference.

Pin Mode setting for Touch Key	TK0~TK7
Pin is Touch Key, Idling	PB.0~7 =Mode0
Pin is Touch Key, Scanning	

There are two Touch Key Modules (Module0 and Module1) in the TM57M5406/08. Each Module can work independently. In the Touch Key Module, there are two oscillators: Reference Clock (RCK) and Touch Clock (TCK). They are connected to the Reference Counter and Data Counter respectively. The frequency of RCK can be adjusted by setting TKREFC, TKFDB (frequency double) and TKFSL (frequency slow). Reference Counter is used to control conversion time. From starting touch key conversion to end, it will take 0 to 4096 RCK oscillation cycles by setting TKTMR. After end of conversion, user can get TKDATA (TKDH, TKDL) from Data counter. TKDATA is affected by finger touching. As finger touching TCK is getting slower, the value of TKDATA is smaller than the no finger touching. According to the difference of TKDATA, user can check if it is touched of not. A suitable TKTMR and TKREFC setting can adjust TKDATA to adapt the system board circumstances. To get the best TKREFC setting, user can try different TKREFC value (with TKFDB=0), then find the one which makes the TKDATA and TKTMR as close as possible.

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

TKM0IF	TKM1IF	TKIF	STATE
0	0	0	IDLE
1	0	1	TK Module0 is end of conversion
0	1	1	TK Module1 is end of conversion
1	1	1	TK Modue0 and Module1 are both end of conversion

Touch Key Interrupt Flag Description

◇ Example: Use TK Module0 & Module1, Touch Key channel = TK2 (PB2) & TK4 (PB4).

```
.ORG        001H
INT:
            BTFSC       TKIF
            CALL        INT_TK              ; check TKIF
            RETI
INT_TK:
            BTFSC       TKM0IF              ; check TKM0IF
            CALL        INT_TKM0
            BTFSC       TKM1IF              ; check TKM1IF
            CALL        INT_TKM1
            RET
INT_TKM0:
            MOVLW       11101111B           ; clear TKM0IF
            MOVWF       TKFLG
            MOVRW       TKM10DH             ;move TKDATA into W register
            MOVRW       TKM0DL
            RET
INT_TKM1:
            MOVLW       11011111B           ; clear TKM1IF
            MOVWF       TKFLG
            MOVRW       TKM10DH             ;move TKDATA into W register
            MOVRW       TKM1DL
```

```
            RET

SET_MODE:
            MOVLW       xx00xxxxB           ; PBMODL[5:4] = 00b
            MOVWR       PBMODL              ; Set PB2MOD as Mode0 for touch key input
            BSF         PBD, 2              ; Set PB2 is input with pull-up

            MOVLW       xxxxxx00B           ; PBMODH[1:0] = 00b
            MOVWR       PBMODH              ; Set PB4MOD as Mode0 for touch key input
            BSF         PBD, 4              ; Set PB4 is input with pull-up
TK_INIT:
            MOVLW       10H
            MOVWR       TKM0CTL             ;TKM0REFC=1
            MOVLW       100
            MOVWR       TKM0TMR             ;TKM0TMR=100

            MOVLW       20H
            MOVWR       TKM1CTL             ;TKM1REFC=2
            MOVLW       200
            MOVWR       TKM1TMR             ;TKM1TMR=200

            MOVLW       xxxx00 10B
            MOVWF       TKCHS               ;TKM0CHS=2, TKM1CHS=0

            BSF         TKM0PD              ; enable TK Module0
            BSF         TKM1PD              ; enable TK Module1
            BCF         TK3V
            BCF         TKFDB
            BCF         TKFSL

            MOVLW       11110111B           ; clear TKIF
            MOVWF       TKFLG
            BSF         TKIE                ; enable TK interrupt
START:
            BSF         TKM0SOC             ; start TKM0 conversion
            BSF         TKM1SOC             ; start TKM1 conversion
                        .
                        .
                        .
                        .
```

F0F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MF0F	–	–	TKM1SOC	TKM1SOC	LVR2SAV	BG125EN	LDOSAV	MODE3V
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	1	1	1	0

F0F.5 **TKM1SOC:** Start Touch Key Module1 conversion
Set the TKM1SOC bit to start Touch Key Module1 conversion, and the TKM1SOC bit will be cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag.

F0F.4 **TKM0SOC:** Start Touch Key Module0 conversion
Set the TKM0SOC bit to start Touch Key Module0 conversion, and the TKM0SOC bit will be cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag.

F12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADTKD	ADCL				–	–	TKM1EOC	TKM0EOC
R/W	R				–	–	R	R
Reset	–	–	–	–	–	–	–	–

F12.1 **TKM1EOC:** Touch Key Module1 end of conversion flag, TKM1EOC may have 3uS delay after TKSOC=1, so F/W must wait enough time before polling this Flag.
0: Indicates conversion is in progress
1: Indicates conversion is finished

F12.0 **TKM0EOC:** Touch Key Module0 end of conversion flag, TKM0EOC may have 3uS delay after TKSOC=1, so F/W must wait enough time before polling this Flag.
0: Indicates conversion is in progress
1: Indicates conversion is finished

F15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKFLG	–	–	TKM1IF	TKM0IF	TKIF	–	–	–
R/W	–	–	R/W	R/W	R/W	–	–	–
Reset	–	–	0	0	0	–	–	–

F15.5 **TKM1IF**: Touch Key Module1 interrupt event pending flag
This bit is set while Touch Key Module1 is end of conversion, write 0 to this bit will clear this flag or sets the TKM1SOC bit to clear this flag.

F15.4 **TKM0IF**: Touch Key Module0 interrupt event pending flag
This bit is set while Touch Key Module0 is end of conversion, write 0 to this bit will clear this flag or sets the TKM0SOC bit to clear this flag.

F15.3 **TKIF**: Touch Key interrupt event pending flag
This bit is set by H/W while Touch Key Module0 or Module1 are end of conversion, write 0 to this bit will clear **all** of Touch Key flag

F16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCTL	–	–	TKM1PD	TKM0PD	TKIE	TK3V	TKFDB	TKFSL
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	1	1	0	0	0	0

F16.5 **TKM1PD**: Touch Key Module1 power down
0: Touch Key Module1 running
1: Touch Key Module1 power down

F16.4 **TKM0PD**: Touch Key Module0 power down
0: Touch Key Module0 running
1: Touch Key Module0 power down

F16.3 **TKIE**: Touch Key interrupt enable
0: disable
1: enable

F16.2 **TK3V**: Touch Key operation voltage select
0: for VCC>3.6V operation
1: for VCC<3.6V operation
F16.1 **TKFDB**: Touch Key reference clock (RCK) double frequency enable
0: select normal RCK
1: select double RCK (RCK0 & RCK1 shared the same control bit)
F16.0 **TKFSL**: Touch Key reference clock (RCK) slow frequency enable
0: select normal RCK
1: select slower RCK (RCK0 & RCK1 shared the same control bit)

F17	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKCHS	–	–	–	–	TKM1CHS		TKM0CHS	
R/W	–	–	–	–	R/W		R/W	
Reset	–	–	–	–	0	0	0	0

F17.3~2 **TKM1CHS**: Touch Key Module1 channel select
00: TK4 (PB4)
01: TK5 (PB5)
10: TK6 (PB6)
11: TK7 (PB7)
F17.1~0 **TKM0CHS**: Touch Key Module0 channel select
00: TK0 (PB0)
01: TK1 (PB1)
10: TK2 (PB2)
11: TK3 (PB3)

R1B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKM10DH	TKM1DH				TKM0DH			
R/W	R				R			
Reset	–	–	–	–	–	–	–	–

R1B.7~4 **TKM1DH**: Touch Key Module1 data MSB[11~8]
R1B.3~0 **TKM0DH**: Touch Key Module0 data MSB[11~8]

R1C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKM0DL	TKM0DL							
R/W	R							
Reset	–	–	–	–	–	–	–	–

R1C.7~0 **TKM0DL**: Touch Key Module0 data LSB[7~0]

R1D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKM1DL	TKM1DL							
R/W	R							
Reset	–	–	–	–	–	–	–	–

R1D.7~0 **TKM1DL**: Touch Key Module1 data LSB[7~0]

R20	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKM0CTL	TKM0REFC				TKM0TMRH			
R/W	R/W				R/W			
Reset	1	0	0	0	0	0	0	0

R20.7~4 **TKM0REFC**: Touch Key Module0 reference clock (RCK0) capacitor select
000: smallest (RCK0 frequency fastest, conversion time shortest)
.
.
.
111: biggest (RCK0 frequency slowest, conversion time longest)

R20.3~0 **TKM0TMRH**: Touch Key Module0 reference counter MSB[11~8]

R21	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKM0TMR	TKM0TMR							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

R21.7~0 **TKM0TMR**: Touch Key Module0 reference counter LSB[7~0]

R22	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKM1CTL	TKM1REFC				TKM1TMRH			
R/W	R/W				R/W			
Reset	1	0	0	0	0	0	0	0

R22.7~4 **TKM1REFC**: Touch Key Module1 reference clock (RCK1) capacitor select
000: smallest (RCK1 frequency fastest, conversion time shortest)
.
.
.
111: biggest (RCK1 frequency slowest, conversion time longest)

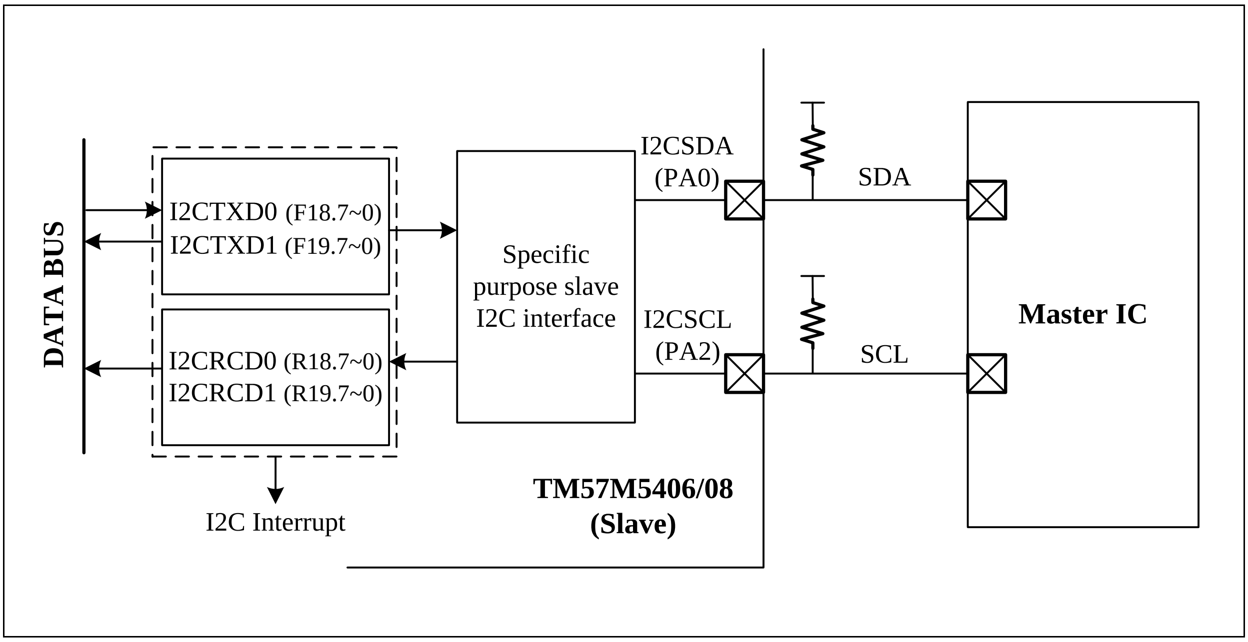
R22.3~0 **TKM1TMRH**: Touch Key Module1 reference counter MSB[11~8]

R23	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKM1TMR	TKM1TMR							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

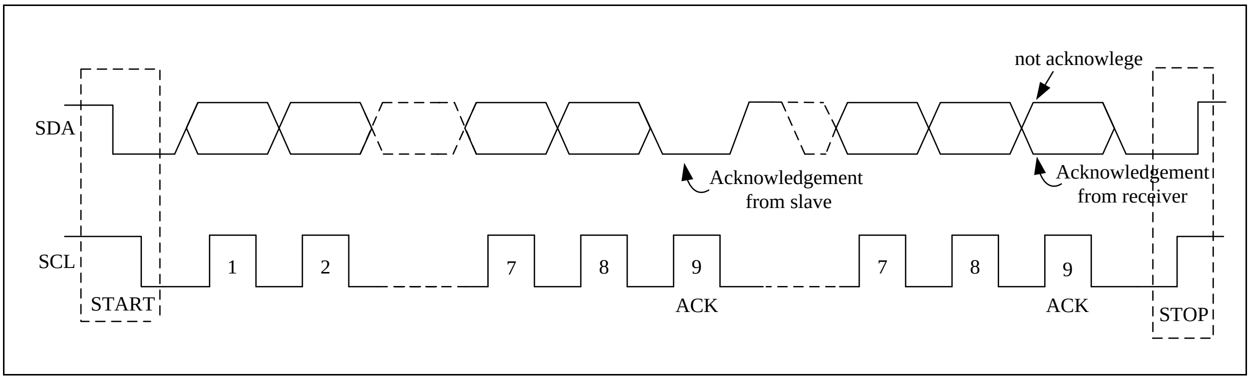
R23.7~0 **TKM1TMR**: Touch Key Module1 reference counter LSB[7~0]

9.9. Specific Purpose Slave I2C Interface

Specific purpose slave I2C interface in TM57M5406/08 could be used for data transmission. This interface is based on a standard I2C (Inter-Integrated Circuit), and TM57M5406/08 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 (F18.7~0) and I2CTXD1 (F19.7~0) of TM57M5406/08 or write data to the register I2CRCD0 (R18.7~0) and I2CRCD1 (R19.7~0) of TM57M5406/08.

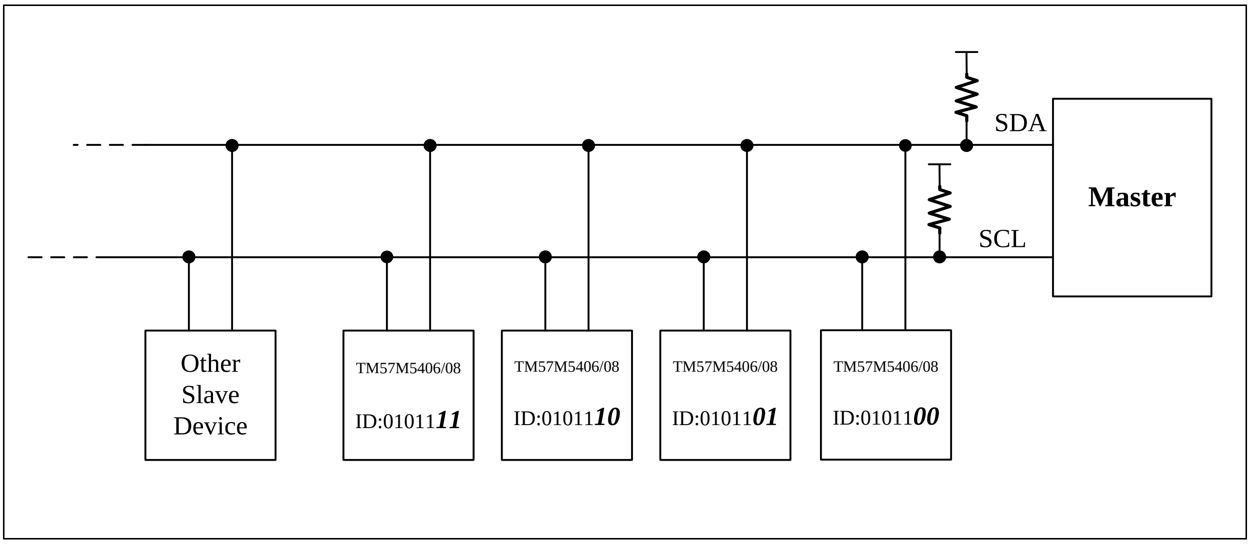


Slave I2C Interface Block Diagram

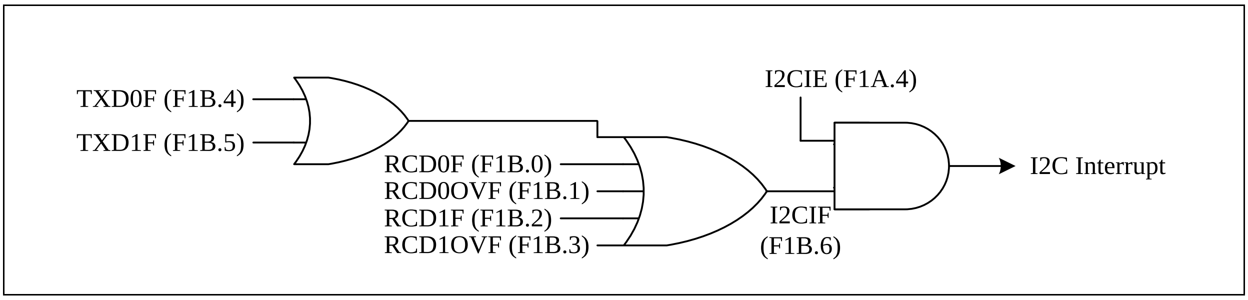


I2C Protocol

To use the slave I2C interface, the I2CEN (F1A.3) bit has to be set. TM57M5406/08 supports 4 slave device IDs by setting I2CID (F1A.1~0). TM57M5406/08 can generate the transmitting flag TXD0F (F1B.4) and TXD1F (F1B.5) when data transmitting finished. It generates the receiving flag RCD0F (F1A.0) and RCD1F (F1A.2) when data receiving finished. It can also generate the receiving overflow flag RCD0OVF (F1A.1) and RCD1OVF (F1A.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 (F1B.6) will be generated. It generates I2C interrupt if the I2CIE (F1A.4) bit is set. 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

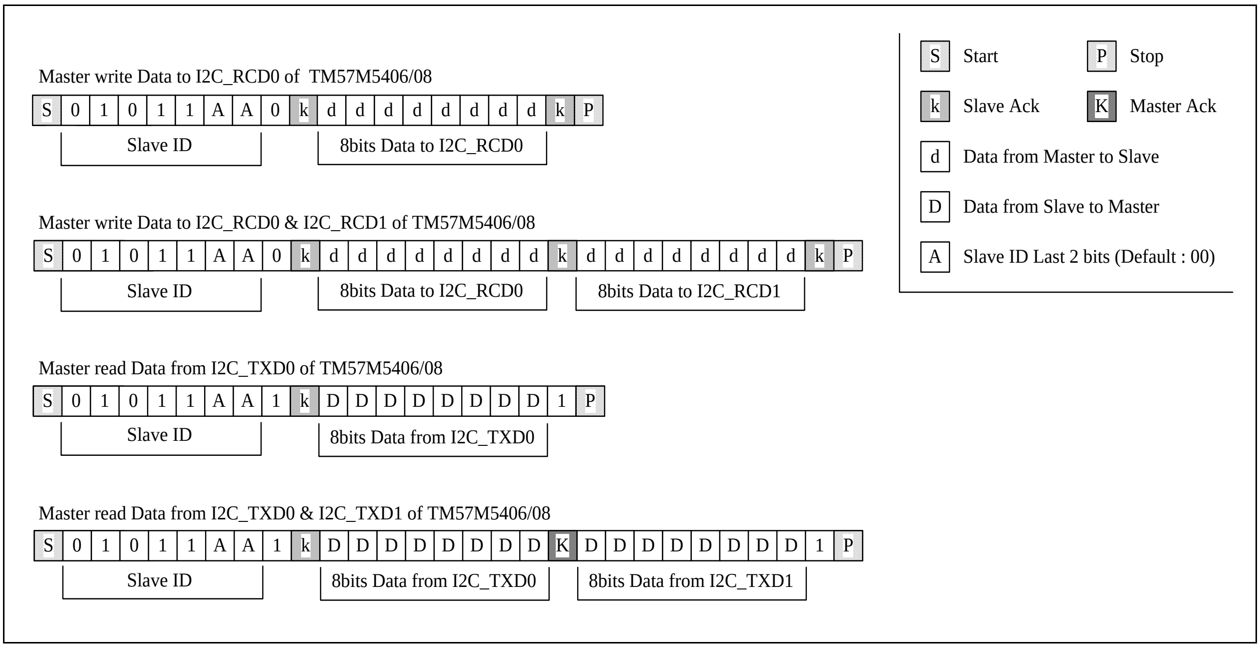


Table of TM57M5406/08 I2C Commands

F18	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

F18.7~0 **I2CTXD0:** The transmitting register 0 of slave I2C

F19	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

F19.7~0 **I2CTXD1:** The transmitting register 1 of slave I2C

F1A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CCTL	–	–	–	I2CIE	I2CEN	–	I2CID	
R/W	–	–	–	R/W	R/W	–	R/W	
Reset	–	–	–	0	0	–	0	–

F1A.4 **I2CIE:** Slave I2C interrupt enable
0: disable
1: enable

F1A.3 **I2CEN:** Slave I2C interface enable
0: disable
1: enable

F1A.1~0 **I2CID**: Slave I2C ID last 2 bits

F1B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CFLAG	–	I2CIF	TXD1F	TXD0F	RCD1OVF	RCD1F	RCD0OVF	RCD0F
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	0	0	0	0	0	0	0

F1B.6 **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

F1B.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

F1B.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

F1B.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

F1B.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

F1B.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

F1B.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

R18	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

R18.7~0 **I2CRCD0:** The receiving register 0 of slave I2C

R19	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

R19.7~0 **I2CRCD1:** The receiving register 1 of slave I2C

MEMORY MAP

F-Plane

Name	Address	R/W	Rst	Description
(F00) INDF				**Function related to: RAM W/R**
INDF	00.7~0	R/W	-	Not a physical register, addressing INDF actually point to the register whose address is contained in the FSR register
(F01) TM0				**Function related to: Timer0**
TM0	01.7~0	R/W	0	Timer0 content
(F02) PCL				**Function related to: Program Counter**
PCL	02.7~0	R/W	0	Programming Counter LSB [7~0]
(F03) STATUS				**Function related to: STATUS**
GB1	03.7	R/W	0	General purpose bit 1
GB0	03.6	R/W	0	General purpose bit 0
RAMBK	03.5	R/W	0	FRAM Bank selection, 0: Bank0, 1: Bank1
TO	03.4	R	0	WDT timeout flag
PD	03.3	R	0	Power-down mode flag
Z	03.2	R/W	0	Zero flag
DC	03.1	R/W	0	Decimal Carry flag or Decimal / Borrow flag
C	03.0	R/W	0	Carry flag or / Borrow flag
(F04) FSR				**Function related to: RAM W/R / Table Read**
GB2	04.7	R/W	0	General purpose bit 2
FSR	04.6~0	R/W	-	File Select Register, indirect address mode pointer
(F05) PAD				**Function related to: Port A**
PAD	05.7	R	-	PA7 pin or "data register" state
		W	1	0: PA7 is open-drain output mode 1: PA7 is Schmitt-trigger input mode
	05.6~0	R	-	Port A pin or "data register" state
		W	7F	Port A output data register
(F06) PBD				**Function related to: Port B**
PBD	06.7~0	R	-	Port B pin or "data register" state
		W	FF	Port B output data register
(F07) PDD				**Function related to: Port D**
PDD	07.3~0	R	-	Port D pin or "data register" state
		W	F	Port D output data register

Name	Address	R/W	Rst	Description
(F08) INTIE				**Function related to: Interrupt Enable**
ADIE	08.7	R/W	0	ADC interrupt enable 0: disable 1: enable
T2IE	08.6	R/W	0	T2 interrupt enable 0: disable 1: enable
TM1IE	08.5	R/W	0	Timer1 interrupt enable 0: disable 1: enable
TM0IE	08.4	R/W	0	Timer0 interrupt enable 0: disable 1: enable
WKTIE	08.3	R/W	0	Wakeup Timer interrupt enable 0: disable 1: enable
INT2IE	08.2	R/W	0	INT2 (PA7) pin interrupt enable 0: disable 1: enable
INT1IE	08.1	R/W	0	INT1 (PA3) pin interrupt enable 0: disable 1: enable
INT0IE	08.0	R/W	0	INT0 (PA0) pin interrupt enable 0: disable 1: enable
(F09) INTIF				**Function related to: Interrupt Flag**
ADIF	09.7	R	-	This bit is set while ADC is end of conversion, write 0 to this bit will clear this flag or sets the ADST bit to clear this flag.
		W	0	0: clear this flag 1: no action
T2IF	09.6	R	-	T2 interrupt event pending flag, set by H/W while T2 overflows
		W	0	0: clear this flag 1: no action
TM1IF	09.5	R	-	Timer1 interrupt event pending flag, set by H/W while Timer1 overflows
		W	0	0: clear this flag 1: no action
TM0IF	09.4	R	-	Timer0 interrupt event pending flag, set by H/W while Timer0 overflows
		W	0	0: clear this flag 1: no action
WKTIF	09.3	R	-	WKT interrupt event pending flag, set by H/W while WKT time out
		W	0	0: clear this flag 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	0: clear this flag 1: no action
INT1IF	09.1	R	-	INT1 (PA4) interrupt event pending flag, set by H/W at INT1 pin's falling edge
		W	0	0: clear this flag 1: no action
INT0IF	09.0	R	-	INT0 (PA0) interrupt event pending flag, set by H/W at INT0 pin's falling / rising edge
		W	0	0: clear this flag 1: no action

Name	Address	R/W	Rst	Description
(F0A) PCH				**Function related to: PROGRAM COUNT**
PCH	0a.2~0	R/W	0	Programming Counter MSB [10~8]
(F0B) CLKCTL				**Function related to: Fsys**
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
(F0C) MF0C				**Function related to: TM0/TM1/T2/PWM0/PWM1/Filter**
CLKFLT	0c.7	R/W	0	Clock noise Filter 0: disable 1: enable
T2CKS	0c.6	R/W	0	T2 clock source selection 0: Slow-clock 1: Fsys/128
T2CLR	0c.5	R/W	0	T2 counter clear 0: T2 is counting 1: T2 is cleared immediately, this bit is auto cleared by H/W
-	0c.4	-	-	-
TM0STP	0c.3	R/W	0	Timer0 counter stop 0: Timer0 is counting 1: Timer0 stops counting
TM1STP	0c.2	R/W	0	Timer1 counter stop 0: Timer1 is counting 1: Timer1 stops counting
PWM1CLR	0c.1	R/1	0	PWM1 clear and hold 0: PWM1 is running 1: PWM1 is cleared and hold
PWM0CLR	0c.0	R/W	0	PWM0 clear and hold 0: PWM0 is running 1: PWM0 is cleared and hold
(F0D) PWM0DH				**Function related to: PWM0**
PWM0DH	0d.7~0	R/W	0	PWM0 Duty MSB 8bit
(F0E) PWM0DL				**Function related to: PWM0**
PWM0DL	0e.7~6	R/W	0	PWM0 Duty LSB 2bit
(F0F) MF0F				**Function related to: LDO/MODE3V/LVR**
TKM1SOC	0f.5	R/W	0	Start Touch Key Module1 conversion Set the TKM1SOC bit to start Touch Key Module1 conversion, and the TKM1SOC bit will be cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag.
TKM0SOC	0f.4	R/W	0	Start Touch Key Module0 conversion Set the TKM0SOC bit to start Touch Key Module0 conversion, and the TKM0SOC bit will be cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag.

Name	Address	R/W	Rst	Description
LVR2SAV	0f.3	R/W	1	LVR2 power save mode enable 0: disable LVR2 power save mode 1: LVR2 (2.3V/2.9V) auto power off in STOP/IDLE mode
BG125EN	0f.2	R/W	1	Internal Bandgap voltage 1.25V enable 0: Internal Bandgap voltage disable 1: Internal Bandgap voltage enable
LDOSAV	0f.1	R/W	1	LDO power save mode enable 0: disable LVR2 power save mode 1: LDO auto power off in STOP/IDLE mode
MODE3V	0f.0	R/W	0	3V mode selection control bit If this bit is set, the chip can be only operated in the condition of VCC<3.6V, and LDO is turned off to save current
(F11) ADCH				**Function related to: ADC**
ADCH	11.7~0	R	-	ADC output data MSB[11~4]
(F12) ADTKD				**Function related to: ADC/TK**
ADCL	12.7~4	R	-	ADC output data LSB[3~0]
TKM1EOC	12.1	R	-	Touch Key Module1 end of conversion flag, TKM1EOC may have 3uS delay after TKSOC=1, so F/W must wait enough time before polling this Flag. 0: Indicates conversion is in progress 1: Indicates conversion is finished
TKM0EOC	12.0	R	-	Touch Key Module0 end of conversion flag, TKM0EOC may have 3uS delay after TKSOC=1, so F/W must wait enough time before polling this Flag. 0: Indicates conversion is in progress 1: Indicates conversion is finished
(F13) ADCTL				**Function related to: ADC**
ADST	13.7	R/W	0	ADC start bit. 0: H/W clear after end of conversion 1: ADC start conversion
ADCKS	13.6~4	R/W	0	ADC clock frequency (Fadc) select 000: Fsys/256 001: Fsys/128 010: Fsys/64 011: Fsys/32 100: Fsys/16 101: Fsys/8 110: Fsys/4 111: Fsys/2
ADCHS	13.3~0	R/W	0	ADC channel select 0000: ADC0 (PD0) 0001: ADC1 (PD1) 0010: ADC2 (PD2) 0011: ADC3 (PD3) 1011: VBG (1.25V) others: Reserved
(F14) TM1				**Function related to: Timer1**
TM1	14.7~0	R/W	0	Timer1 content

Name	Address	R/W	Rst	Description
(F15) TKFLG				**Function related to: TK**
TKM1IF	15.2	R	-	Touch Key Module1 interrupt event pending flag, set by H/W while Touch Key Module1 is end of conversion
		W	0	write 0 to this bit will clear this flag or sets the TKM1SOC bit to clear this flag; write 1: no action
TKM0IF	15.1	R	-	Touch Key Module0 interrupt event pending flag, set by H/W while Touch Key Module0 is end of conversion
		W	0	write 0 to this bit will clear this flag or sets the TKM0SOC bit to clear this flag; write 1: no action
TKIF	15.0	R	-	Touch Key interrupt event pending flag, set by H/W while Touch Key Module0 or Module1 are end of conversion
		W	0	write 0 to this bit will clear all of Touch Key flag; write 1: no action
(F16) TKCTL				**Function related to: TK**
TKM1PD	16.5	R/W	1	Touch Key Module1 power down 0: Touch Key Module1 running 1: Touch Key Module1 power down
TKM0PD	16.4	R/W	1	Touch Key Module0 power down 0: Touch Key Module0 running 1: Touch Key Module0 power down
TKIE	16.3	R/W	0	Touch Key interrupt enable 0: disable 1: enable
TK3V	16.2	R/W	0	Touch Key operation voltage select 0: for VCC>3.6V operation 1: for VCC<3.6V operation
TKFDB	16.1	R/W	0	Touch Key reference clock (RCK) double frequency enable 0: select normal RCK 1: select double RCK (RCK0 & RCK1 shared the same control bit)
TKFSL	16. 0	R/W	0	Touch Key reference clock (RCK) slow frequency enable 0: select normal RCK 1: select slower RCK (RCK0 & RCK1 shared the same control bit)
(F17) TKCHS				**Function related to: TK**
TKM1CHS	17.3~2	R/W	0	Touch Key Module1 channel select 00: TK4 (PB4) 01: TK5 (PB5) 10: TK6 (PB6) 11: TK7 (PB7)
TKM0CHS	17.1~0	R/W	0	Touch Key Module0 channel select 00: TK0 (PB0) 01: TK1 (PB1) 10: TK2 (PB2) 11: TK3 (PB3)
(F18) I2CTXD0				**Function related to: Slave I2C**
I2CTXD0	18.7~0	R/W	0	The transmitting register 0 of slave I2C
(F19) I2CTXD1				**Function related to: Slave I2C**
I2CTXD1	19.7~0	R/W	0	The transmitting register 1 of slave I2C
(F1A) I2CCTL				**Function related to: Slave I2C**
I2CIE	1a.4	R/W	0	I2C Interrupt Enable 0: Disable 1: Enable
I2CEN	1a.3	R/W	0	Slave I2C interface enable 0: disable 1: enable
-	1a.2	-	-	-
I2CID	1a.1~0	R/W	0	Slave I2C ID last 2 bits

Name	Address	R/W	Rst	Description
(F1B) INTIF				**Function related to: Interrupt Flag**
I2CIF	1b.6	R	-	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
		W	0	0: clear this flag 1: no action
TXD1F	1b.5	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
TXD0F	1b.4	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
RCD1OVF	1b.3	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
RCD1F	1b.2	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
RCD0OVF	1b.1	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
RCD0F	1b.0	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
(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
(F1E) DPH				**Function related to: Table Read**
DPL	1e.7~0	R/W	0	Table read low address, data ROM pointer (DPTR) high byte
(F1F) IRCF				**Function relate to: Trim FIRC**
IRCF	1f.4~0	R/W	-	FIRC frequency adjustment:
User Data Memory				
FRAM	20~30	R/W	-	FRAM common area (8 bytes)
	30~7f	R/W	-	FRAM Bank0 area
	30~7f	R/W	-	FRAM Bank1 area

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: Timer0**
TM0RLD	01.7~0	R/W	0	Timer0 reload Data
(R02) TM0CTL				**Function related to: Timer0**
TM0EDG	02.5	R/W	0	TM0CKI (PA2) edge selection for Timer0 prescaler count 0: TM0CKI rising edge for Timer0 prescaler count 1: TM0CKI falling edge for Timer0 prescaler count
TM0CKS	02.4	R/W	0	Timer0 clock source select 0: Instruction Cycle (Fsys/2) as Timer0 prescaler clock 1: TM0CKI (PA2) as Timer0 prescaler clock
TM0PSC	02.3~0	R/W	0	Timer0 prescaler. Timer0 clock source 0000: divided by 1 0001: divided by 2 0010: divided by 4 0011: divided by 8 0100: divided by 16 0101: divided by 32 0110: divided by 64 0111: divided by 128 1xxx: divided by 256
(R03) PWRDN				**Function related to: POWER DOWN**
PWRDN	03	W	-	Write this register to enter Power-down (STOP / IDLE) Mode
(R04) WDTCLR				**Function related to: WDT**
WDTCLR	04	W	-	Write this register to clear WDT timer
(R05) PAMODH				**Function related to: Port A**
PA7MOD	05.6	R/W	0	PA7 pull-up resistor enable 0: the pin pull-up resistor is enabled 1: the pin pull-up resistor is disabled
PA4MOD	05.1~0	R/W	01	PA4 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, PA4 as PWM0B push-pull output

Name	Address	R/W	Rst	Description
(R06) PAMODL				**Function related to: Port A**
PA3MOD	06.7~6	R/W	01	PA3 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, PA3 as PWM1C push-pull output
PA2MOD	06.5~4	R/W	01	PA2 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, PA2 as PWM1B push-pull output
PA1MOD	06.3~2	R/W	01	PA1 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, PA1 as PWM1A push-pull output
PA0MOD	06.1~0	R/W	01	PA0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, PA0 as PWM0A push-pull output
(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, input with wake-up function
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~PB2 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, input with wake-up function
PB2MOD	08.5~4	R/W	01	
PB1MOD	08.3~2	R/W	01	PB1 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, PB1 as TM1OUT push-pull output
PB0MOD	08.1~0	R/W	01	PB0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, PB0 as TCOUT push-pull output

Name	Address	R/W	Rst	Description
(R0A) PDMODL				**Function related to: Port D**
PD3MOD	0a.7~6	R/W	01	PD3 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, PD3 as ADC3 channel input
PD2MOD	0a.5~4	R/W	01	PD2 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, PD2 as ADC2 channel input
PD1MOD	0a.3~2	R/W	01	PD1 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, PD1 as ADC1 channel input
PD0MOD	0a.1~0	R/W	01	PD0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3, PD0 as ADC0 channel input
(R0B) MR0B				**Function related to: INT0/TCOUT/WDT**
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 (PA0) edge interrupt event 0: falling edge to trigger 1: rising edge to trigger
T2PSC	0b.5~4	R/W	00	T2 prescaler. T2 clock source 00: divided by 32768 01: divided by 16384 10: divided by 8192 11: divided by 128
WDTPSC	0b.3~2	R/W	11	WDT pre-scale option 00: 128ms 01: 256ms 10: 1024ms 11: 2048ms
WKTPSC	0b.1~0	R/W	11	WKT pre-scale option 00: 16ms 01: 32ms 10: 64ms 11: 128ms
(R0D) PWM0PRD				**Function related to: PWM0**
PWM0PRD	0d.7~0	R/W	FF	PWM0 period data
(R0E) PWM1PRD				**Function related to: PWM1**
PWM1PRD	0e.7~0	R/W	FF	PWM1 period data
(R10) PWMCTL				**Function related to: PWM0/1**
PWM1CKS	10.7	R/W	0	PWM0 clock source 0: Fsys 1: FIRC (12MHz)
PWM1PSC	10.6~4	R/W	00	PWM0 prescaler 000: divided by 1 001: divided by 2 010: divided by 4 … 111: divided by 128

Name	Address	R/W	Rst	Description
PWM0CKS	10.3	R/W	0	PWM1 clock source 0: Fsys 1: FIRC (12MHz)
PWM0PSC	10.2~0	R/W	00	PWM1 prescaler 000: divided by 1 001: divided by 2 010: divided by 4 … 111: divided by 128
(R11) PWM1A				**Function related to: PWM1**
PWM1AD	11.7~0	R/W	00	PWM1A Duty
(R12) PWM1B				**Function related to: PWM1**
PWM1BD	12.7~0	R/W	00	PWM1B Duty
(R13) PWM1D				**Function related to: PWM1**
PWM1CD	13.7~0	R/W	00	PWM1C Duty
(R14) TM1CTL				**Function related to: Timer1**
TM1PSC	14.3~0	R/W	0	Timer1 prescaler. Timer1 clock source (Fsys/2) 0000: divided by 1 0100: divided by 16 1xxx: divided by 256 0001: divided by 2 0101: divided by 32 0010: divided by 4 0110: divided by 64 0011: divided by 8 0111: divided by 128
(R15) TM1RLD				**Function related to: Timer1**
TM1RLD	15.7~0	R/W	0	Timer1 reload Data
(R18) I2CRCD0				**Function related to: Slave I2C**
I2CRCD0	18.7~0	R	0	The receiving register 0 of slave I2C
(R19) I2CRCD1				**Function related to: Slave I2C**
I2CRCD1	19.7~0	R	0	The receiving register 1 of slave I2C
(R1B) TKM10DH				**Function related to: TK**
TKM1DH	1b.7~4	R	-	Touch Key Module1 data MSB[11~8]
TKM0DH	1b.3~0	R	-	Touch Key Module0 data MSB[11~8]
(R1C) TKM0DL				**Function related to: TK**
TKM0DL	1c.7~0	R	-	Touch Key Module0 data LSB[7~0]
(R1D) TKM1DL				**Function related to: TK**
TKM1DL	1d.7~0	R	-	Touch Key Module1 data LSB[7~0]
(R20) TKM0CTL				**Function related to: TK**
TKM0REFC	20.7~4	R/W	8	Touch Key Module0 reference clock (RCK0) capacitor select 0000: smallest (RCK0 frequency fastest, conversion time shortest) … 1111: biggest (RCK0 frequency slowest, conversion time longest)
TKM0TMRH	20.3~0	R/W	0	Touch Key Module0 reference counter MSB[11~8]
(R21) TKM0TMR				**Function related to: TK**
TKM0TMR	21.7~0	R/W	FF	Touch Key Module0 reference counter LSB[7~0]
(R22) TKM1CTL				**Function related to: TK**
TKM1REFC	22.7~4	R/W	8	Touch Key Module1 reference clock (RCK1) capacitor select 0000: smallest (RCK1 frequency fastest, conversion time shortest) … 1111: biggest (RCK1 frequency slowest, conversion time longest)
TKM1TMRH	22.3~0	R/W	0	Touch Key Module1 reference counter MSB[11~8]
(R21) TKM0TMR				**Function related to: TK**
TKM0TMR	21.7~0	R/W	FF	Touch Key Module1 reference counter LSB[7~0]
User Data Memory				
RRAM	40~ff	R/W	-	RRAM common area (192 bytes)

Instruction Set

Each instruction is a 14-bit word divided into an OPCODE, 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 listed in the following table.

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

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

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

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

MOVRW — Move "r" to W

Syntax	MOVRW r	
Operands	f : 00h ~ FFh	
Operation	(W) ← (r)	
Status Affected	-	
OP-Code	01 1111 rrrr rrrr	
Description	The contents of register 'r' are moved to W register.	
Cycle	1	
Example	MOVRW RSR	B : RSR = 0xC2, W = ? A : RSR = 0xC2, W = 0xC2

MOVWR — Move W to "r"

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

NOP — No Operation

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

RET — Return from Subroutine

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

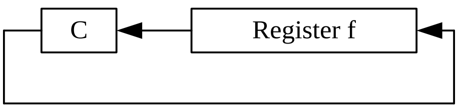
RETI — Return from Interrupt

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

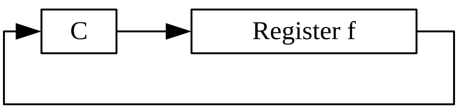
RETLW — Return with Literal in W

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

RLF — Rotate Left "f" through Carry

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

RRF — Rotate Right "f" through Carry

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

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

Syntax	SLEEP	
Operands	-	
Operation	-	
Status Affected	TO, PD	
OP-Code	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   (TAB1>>8)&0xFF
MOVWF   DPH              ;Where DPH is F-Plane register

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

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

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

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

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

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	3.6	V
Operating temperature	-40 to +85	°C
Storage temperature	-65 to +150	

2. DC Characteristics (T_A = 25℃, V_{CC} = 2.0V to 5.5V)

Parameter	Symbol	Conditions		Min	Typ	Max	Unit
Operating Voltage	V_{CC}	FAST mode, 25°C, Fsys = 12 MHz		2.9	–	5.5	V
		FAST mode, 25°C, Fsys = 6 MHz		2.3	–	5.5	
		FAST mode, 25°C, Fsys = 3 MHz		2.0	–	5.5	
		SLOW mode, 25°C, Fsys = 128 KHz		2.0	–	5.5	
Input High Voltage	V_{IH}	All Input, except PA7	V_{CC} = 5V	0.6V_{CC}	–	–	V
			V_{CC} = 3V	0.6V_{CC}	–	–	
		PA7	V_{CC} = 5V	0.8V_{CC}	–	–	
			V_{CC} = 3V	0.8V_{CC}	–	–	
Input Low Voltage	V_{IL}	All Input	V_{CC} = 5V	–	–	0.2V_{CC}	V
			V_{CC} = 3V	–	–	0.2V_{CC}	
I/O Port Source Current	I_{OH}	All Output, except PA7	V_{CC} = 5V, V_{OH} = 0.9V_{CC}	5	10	–	mA
			V_{CC} = 3V, V_{OH} = 0.9V_{CC}	2	4	–	
I/O Port Sink Current	I_{OL}	All Output, except PortA	V_{CC} = 5V, V_{OL} = 0.1V_{CC}	9	18	–	mA
			V_{CC} = 3V, V_{OL} = 0.1V_{CC}	5	10	–	
		PA4~PA0	V_{CC} = 5V, V_{OL} = 0.1V_{CC}	15	30	–	mA
			V_{CC} = 3V, V_{OL} = 0.1V_{CC}	9	18	–	
		PA7	V_{CC} = 5V, V_{OL} = 0.1V_{CC}	8	16	–	mA
			V_{CC} = 3V, V_{OL} = 0.1V_{CC}	5	10	–	
Input Leakage Current (pin high)	I_{ILH}	All Input	V_{IN} = V_{CC}	–	–	1	μA
Input Leakage Current (pin low)	I_{ILL}	All Input	V_{IN} = 0V	–	–	-1	

Supply Current	I_{DD}	Fast, V_{CC} = 5V MODE3V = 0	FRC = 12 MHz	–	2.3	–	mA
			FRC = 6 MHz	–	1.4	–	
			FRC = 3 MHz	–	1.0	–	
		Fast, V_{CC} = 3V MODE3V = 0	FRC = 12 MHz	–	2.0	–	
			FRC = 6 MHz	–	1.3	–	
			FRC = 3 MHz	–	0.9	–	
		Fast, V_{CC} = 3V MODE3V = 1	FRC = 12 MHz	–	1.9	–	
			FRC = 6 MHz	–	1.2	–	
			FRC = 3 MHz	–	0.8	–	
		Slow, V_{CC} = 5V MODE3V = 0	SRC = 128 KHz	–	320	–	μA
		Slow, V_{CC} = 3V MODE3V = 0	SRC = 128 KHz	–	290	–	
		Slow, V_{CC} = 3V MODE3V = 1 BG125EN = 1	SRC = 128 KHz	–	180	–	
		Slow, V_{CC} = 3V MODE3V = 1 BG125EN = 0	SRC = 128 KHz	–	24	–	
		Idle, V_{CC} = 5V MODE3V = 0	SRC = 128 KHz	–	275	–	
		Idle, V_{CC} = 3V MODE3V = 0	SRC = 128 KHz	–	245	–	
		Idle, V_{CC} = 3V MODE3V = 1	SRC = 128 KHz	–	6	–	
		Stop, V_{CC} = 5V MODE3V = 0	LDOSAV=0 LVR2SAV=0	–	270	–	
		Stop, V_{CC} = 3V MODE3V = 0	LDOSAV=0 LVR2SAV=0	–	240	–	
		Stop, V_{CC} = 3V MODE3V = 1	LVR2SAV=0	–	1.8	–	
			LVR2SAV=1	–	1.5	–	
System Clock Frequency	Fsys	V_{CC} = 2.9V		–	–	12	MHz
		V_{CC} = 2.3V		–	–	6	
		V_{CC} = 2.0V		–	–	3	
LVR Reference Voltage	V_{LVR}	T_A = 25°C		-3%	2.9	+3%	V
				-3%	2.3	+3%	
				-8%	2.0	+8%	
LVR Hysteresis Voltage	V_{HYST}	T_A = 25°C		–	±0.1	–	V
Low Voltage Detection time	t_{LVR}	T_A = 25°C		100	–	–	μs
Pull-Up Resistor	R_P	V_{IN} = 0 V, All Pins except PA7	V_{CC} = 5V	–	33	–	KΩ
			V_{CC} = 3V		62		
		V_{IN} = 0 V, PA7	V_{CC} = 5V	–	28	–	
			V_{CC} = 3V		52		

3. Clock Timing (T_A = -40°C to +85°C, V_{CC}=2.0V~5.5V)

Parameter	Condition	Min	Typ	Max	Unit
Fast Internal RC Frequency	25℃, V_{CC} = 3.0 ~ 5.5V	11.64	12	12.36	MHz
	25°C,V_{CC} = 2.0 ~ 3.0V	11.40	12	12.48	
	-40℃ ~ 85℃, V_{CC} = 2.0 ~ 5.5V	11.16	12	12.6	

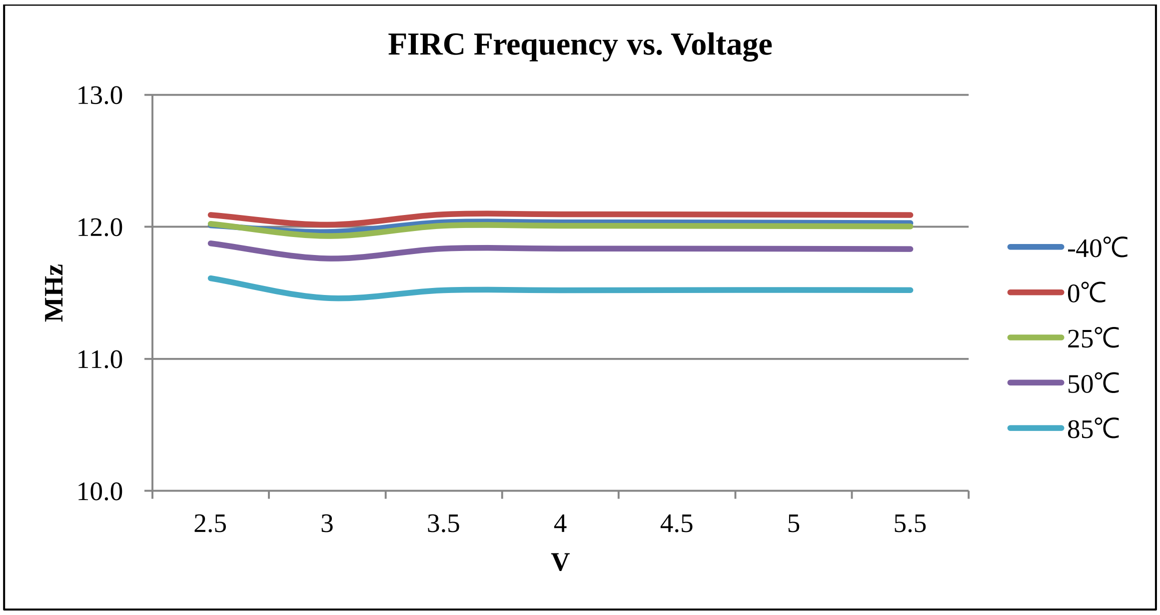
4. Reset Timing Characteristics (T_A = -40°C to +85°C, V_{CC} = 3.0V~5.0V)

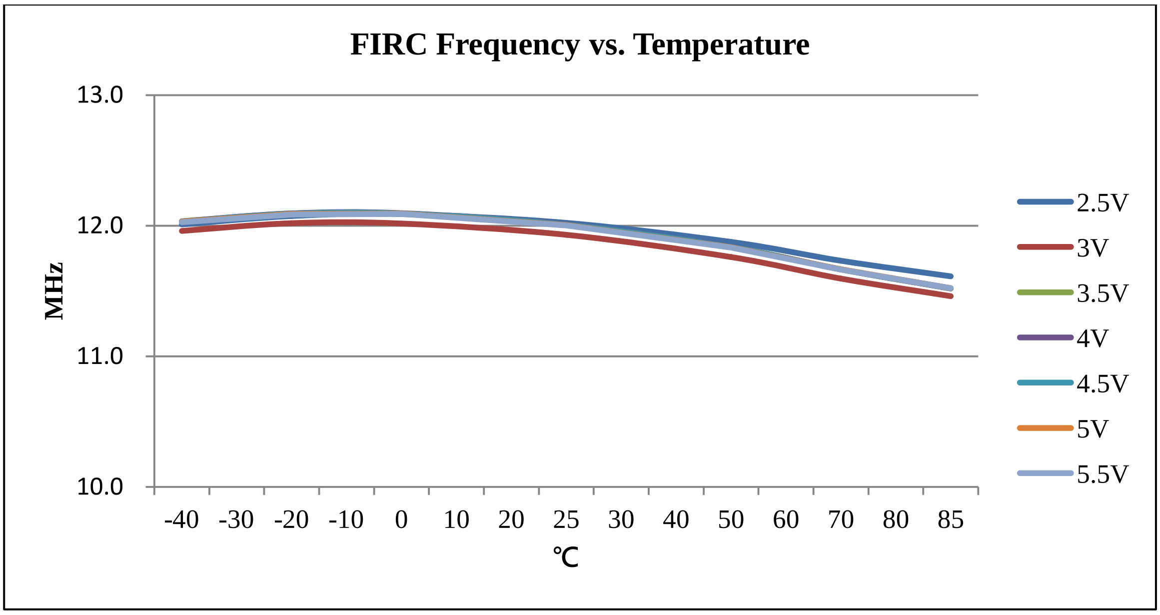
Parameter	Conditions	Min	Typ	Max	Unit
RESET Input Low width	V_{CC} = 5.0V ±10 %	3	–	–	μs
WKT time	V_{CC} = 5V, WKTPSC = 3	–	16	–	ms
	V_{CC} = 3V, WKTPSC = 3	–		–	
WDT time	V_{CC} = 5V, WDTPSC = 3	–	128	–	ms
	V_{CC} = 3V, WDTPSC = 3	–		–	
CPU start up time	V_{CC} = 5V	–	15	–	ms
	V_{CC} = 3V	–		–	

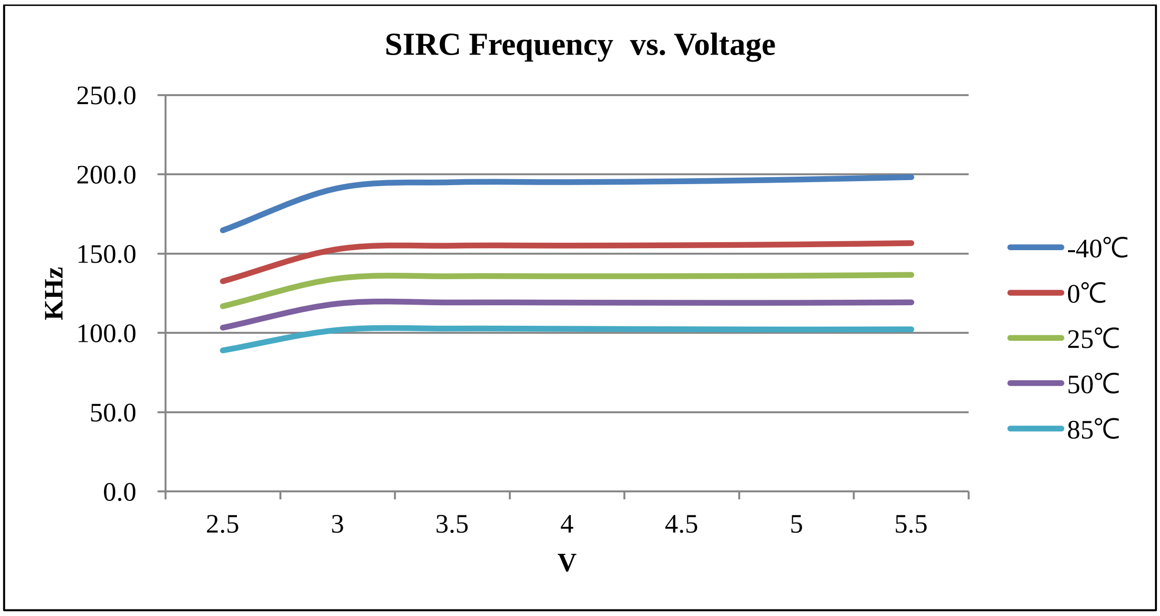
5. ADC Electrical Characteristics (T_A=25°C, V_{CC}=3.0V~5.5V, V_{SS}=0V)

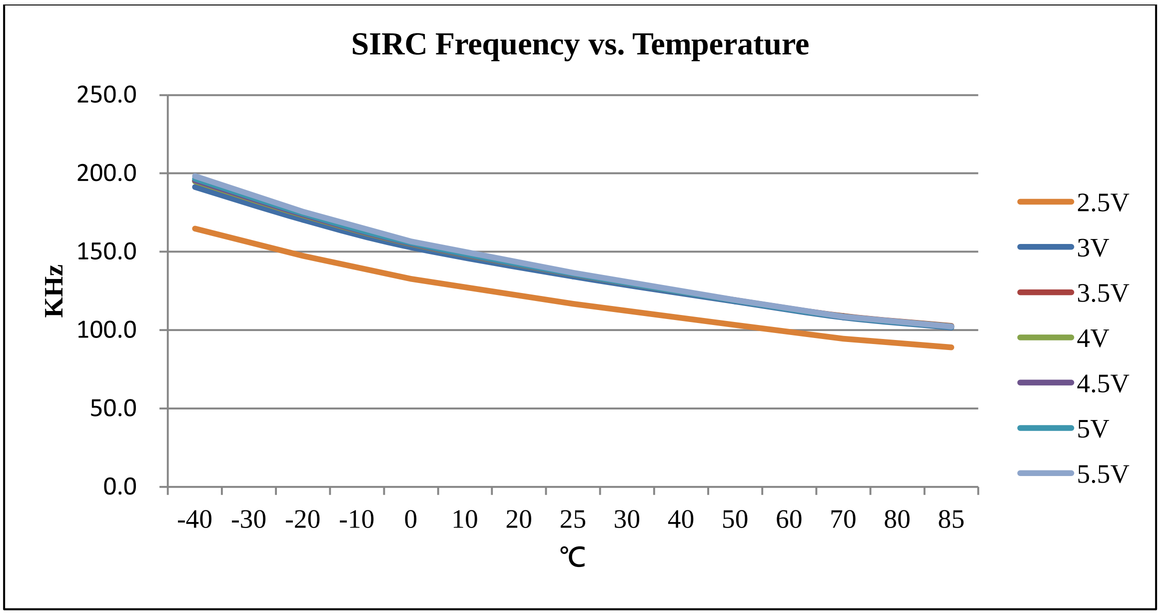
Parameter	Conditions	Min.	Typ.	Max.	Unit
Total Accuracy	V_{CC} = 5.12V,V_{SS} = 0V	–	±2.5	±4	LSB
Integral Non-Linearity		–	±3.2	±5	
Max Input Clock (f_{ADC})	–	–	–	1	MHz
Conversion Time	f_{ADC} = 1 MHz	–	50	–	μs
BandGap Voltage Reference	V_{CC} = 3V	1.14	1.22	1.30	V
	V_{CC} = 5V	1.15	1.25	1.35	
Input Voltage	–	V_{SS}	–	V_{CC}	V

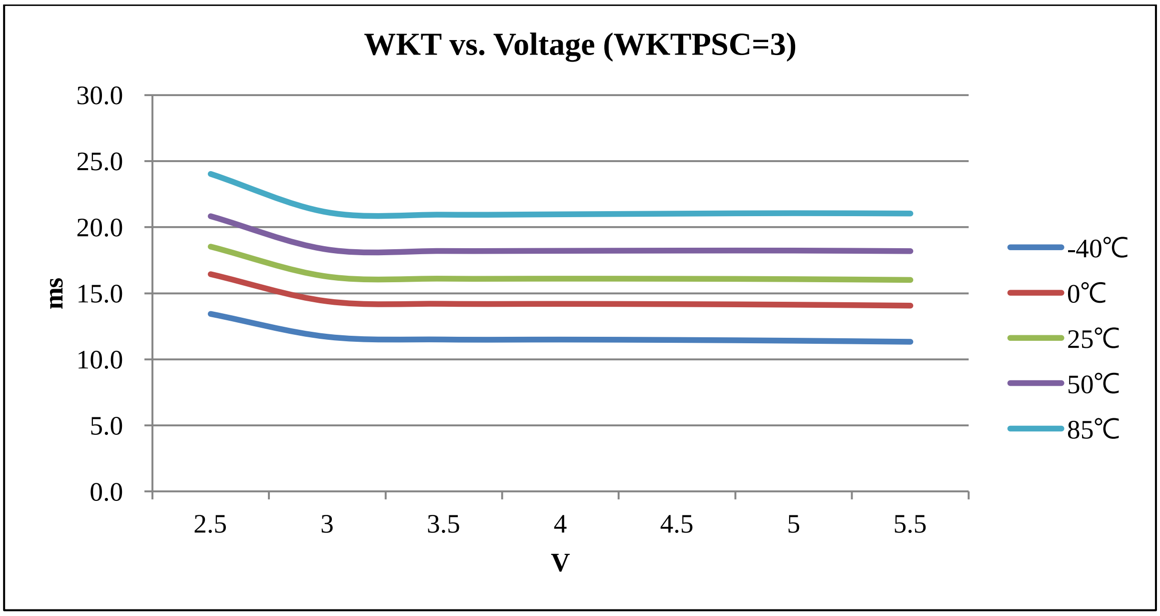
6. Characteristic Graphs

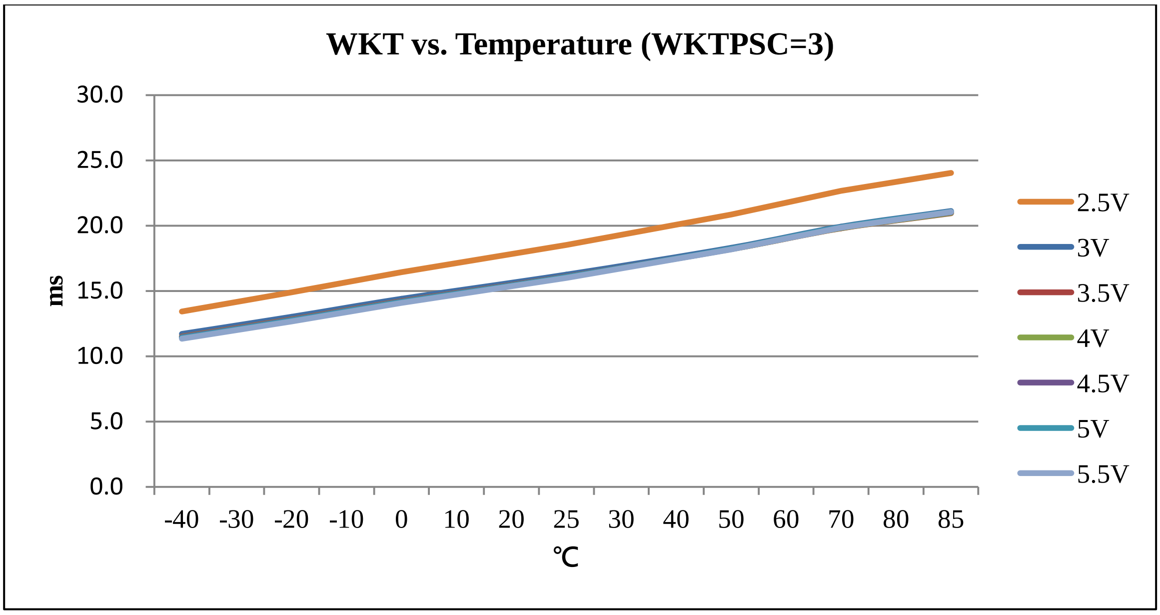


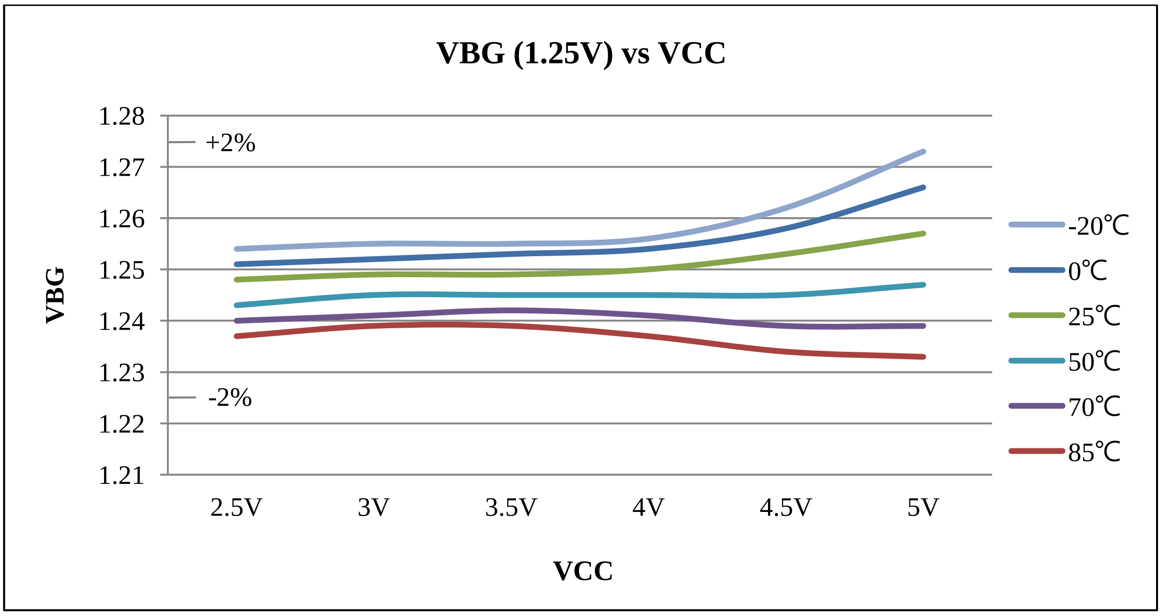


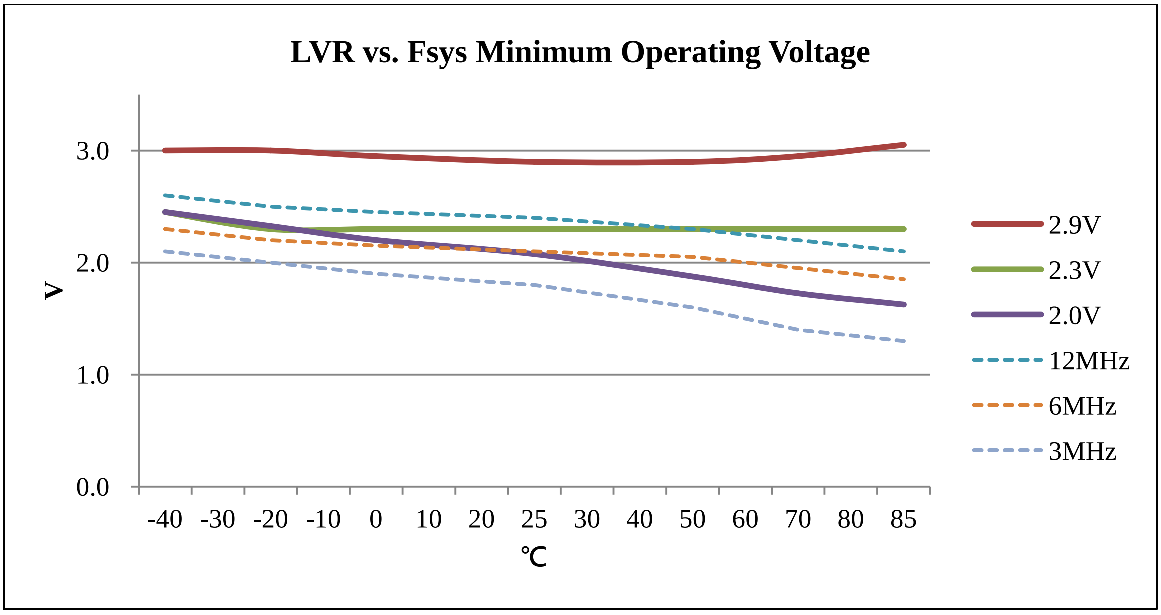

SIRC Frequency vs. Voltage
250.0
200.0
150.0
100.0
50.0
0.0
KHz
2.5
3
3.5
4
4.5
5
5.5
V
-40℃
0℃
25℃
50℃
85℃


SIRC Frequency vs. Temperature
250.0
200.0
150.0
100.0
50.0
0.0
KHz
-40 -30 -20 -10 0 10 20 25 30 40 50 60 70 80 85
℃
2.5V
3V
3.5V
4V
4.5V
5V
5.5V


WKT vs. Voltage (WKTPSC=3)
30.0
25.0
20.0
15.0
10.0
5.0
0.0
ms
2.5
3
3.5
4
4.5
5
5.5
V
-40℃
0℃
25℃
50℃
85℃


WKT vs. Temperature (WKTPSC=3)
30.0
25.0
20.0
15.0
10.0
5.0
0.0
ms
-40
-30
-20
-10
0
10
20
25
30
40
50
60
70
80
85
℃
2.5V
3V
3.5V
4V
4.5V
5V
5.5V





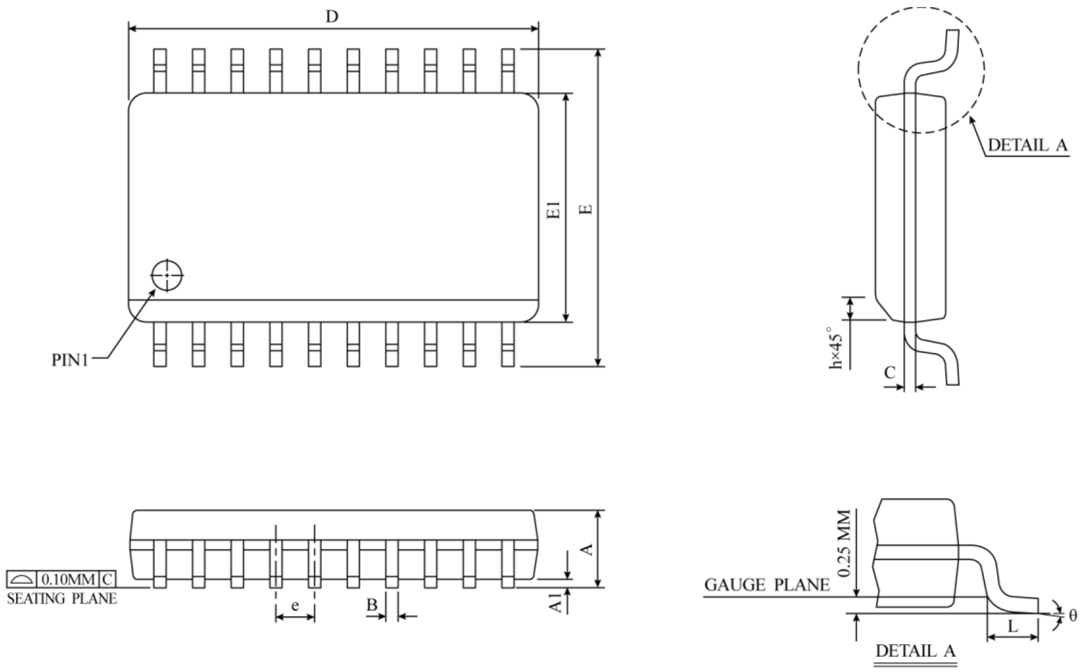
Note: Due to the variation of manufacturing process, this LVR2.0 will slightly vary between different chips.

Packaging information

The ordering information:

Ordering number	Package
TM57M5406/08-MTP	Wafer / Dice blank chip
TM57M5406/08-COD	Wafer / Dice with code
TM57M5408-MTP-21	SOP 20-pin (300 mil)
TM57M5408-MTP-05	DIP 20-pin (300 mil)
TM57M5406-MTP-16	SOP 16-pin (150 mil)
TM57M5406-MTP-03	DIP 16-pin (300 mil)

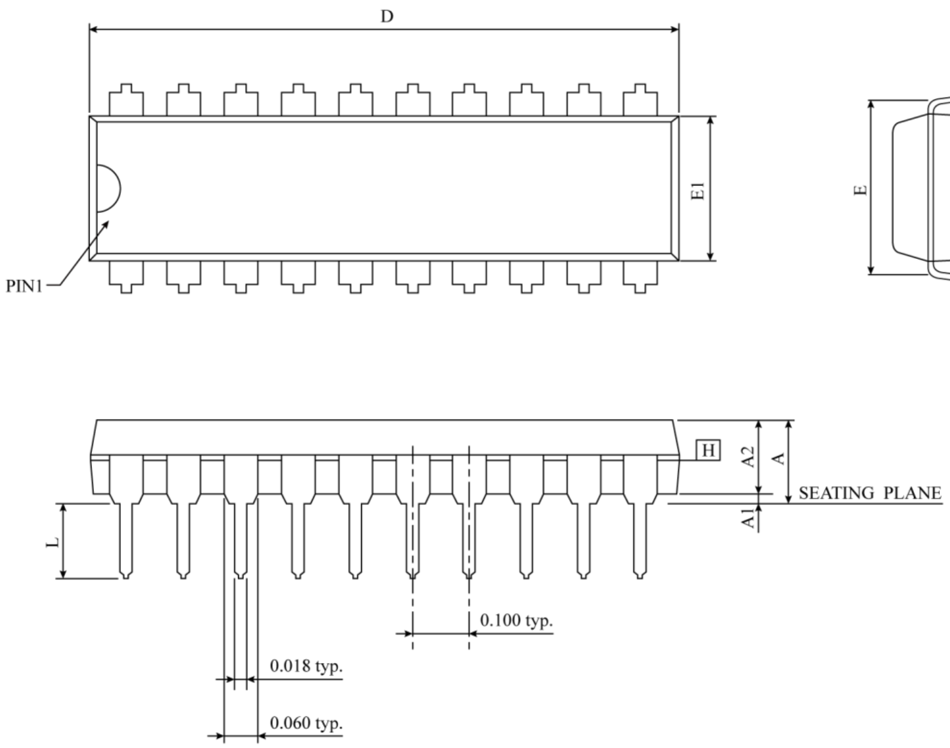
20-SOP Package Dimension



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

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

20-DIP Package Dimension

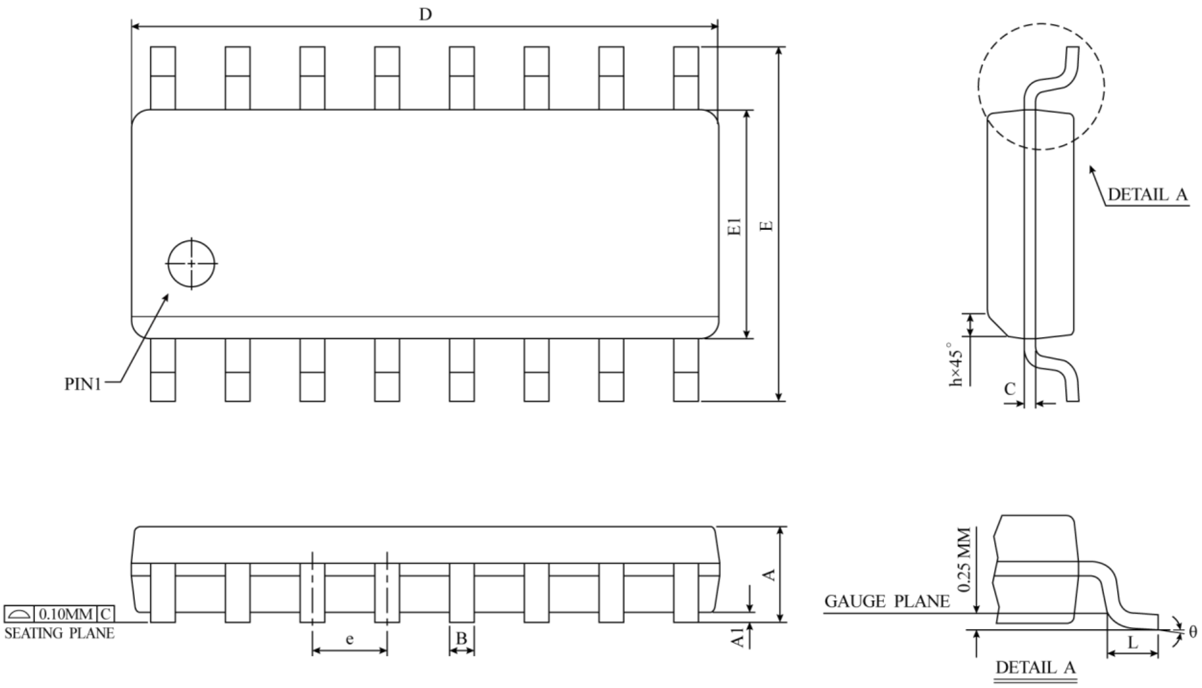


SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	4.445	-	-	0.175
A1	0.381	-	-	0.015	-	-
A2	3.175	3.302	3.429	0.125	0.130	0.135
D	25.705	26.061	26.416	1.012	1.026	1.040
E	7.620	7.747	7.874	0.300	0.305	0.310
E1	6.223	6.350	6.477	0.245	0.250	0.255
L	3.048	3.302	3.556	0.120	0.130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

NOTES：

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

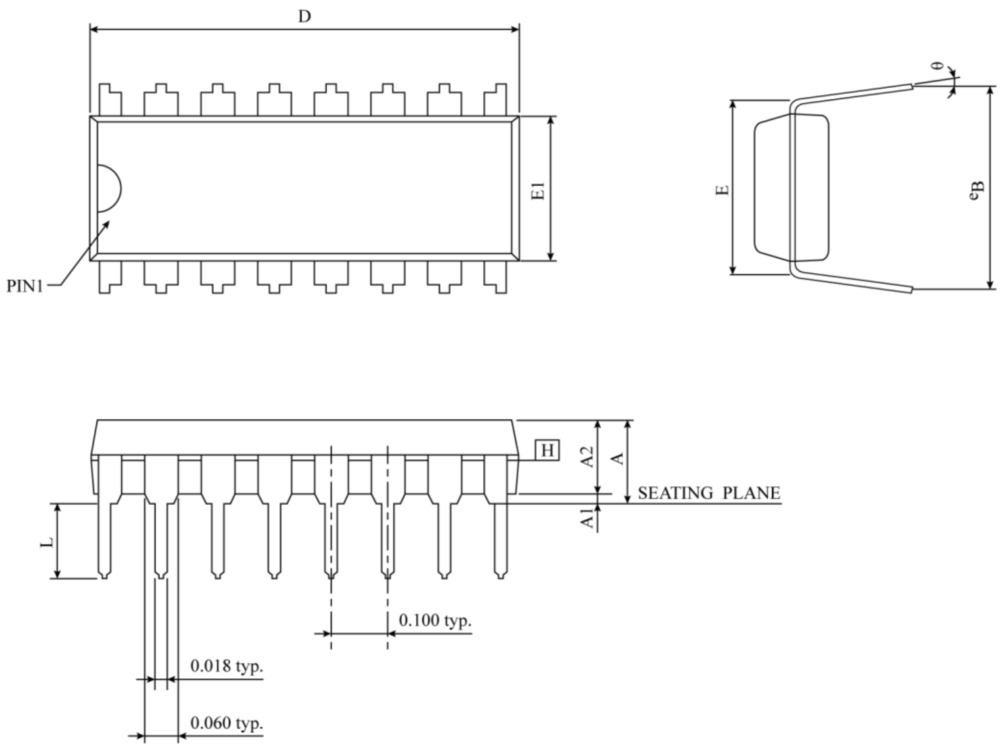
16-SOP Package Dimension



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

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

16-DIP Package Dimension



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	4.369	-	-	0.172
A1	0.381	0.673	0.965	0.015	0.027	0.038
A2	3.175	3.302	3.429	0.125	0.130	0.135
D	18.669	19.177	19.685	0.735	0.755	0.775
E	7.620 BSC			0.300 BSC		
E1	6.223	6.350	6.477	0.245	0.250	0.255
L	2.921	3.366	3.810	0.115	0.133	0.150
e_B	8.509	9.017	9.525	0.335	0.355	0.375
θ	0°	7.5°	15°	0°	7.5°	15°
JEDEC	MS-001 (BB)					

NOTES：

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