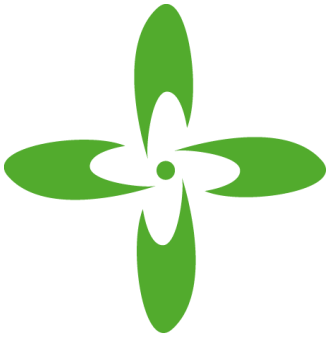


十速

TM56M7842
TM56E7842

DATA SHEET

Rev 0.92

PRECAUTIONS

1. The four instructions ADDWXC/SUBXW/SUBXWB/SUBWXB do not support indirect addressing; they only support direct addressing.
2. RAM Bank4 only supports indirect addressing operations via FSR1 and INDF1.
3. During simulation, operations on FSR1 and INDF1 must be performed in full-speed execution (Free Run or GO) mode.

CONTENTS

FEATURES

1. **ROM: 4K*16-bit Flash program memory**
2. **EEPROM:256 Bytes (TM56E7842 only)**
3. **RAM: 464-bit**
4. **STACK: 8 level**
5. **System clock(Fsys):**
   - Supports 1/2/4/8 crossover options
   - Fast-clock
     - Internal RC fast clock(FIRC): 18.432 MHz ±1% @25℃,V_{CC}=5V (can be calibrated)
   - Slow-clock
     - Internal RC slow clock(SIRC): 50 KHz or 60KHz @VCC=5V
     - External slow clock(SXT): 32.768KHz
6. **Power Saving Operation Mode**
   - FAST Mode: Fast-clock keeps CPU running. Slow-clock can be enabled.
   - SLOW Mode: Slow-clock keeps CPU running. Fast-clock can be disabled or enabled.
   - IDLE Mode: CPU stops. Slow clock enabled. T2 and WKT can be stopped or enabled
   - STOP Mode: CPU stops. All clocks stop.
7. **I/O port: up to 22 programmable I/O pins**
   - Open drain output
   - CMOS push-pull output
   - Schmitt trigger input with pull-up resistor options
8. **Timer**
   - Timer0
     - 8-bit timer with auto-reload/count/interrupt/pause/reset functions
     - Clock source: (Fsys/2 or external pin (PA2) or SXT/16) divided by 1 to 32768
   - Timer1
     - 8-bit timer with auto-reload/interrupt/pause functions
     - Clock source: (Fsys/2) divided by 1 to 256
     - Overflow and Toggle out
   - T2
     - 15-bit timer with interrupt/reset function
     - Clock source: Slow clock or Fsys/128, with 4 division options
     - Purpose: idle mode wake-up timer or simple 15-bit time base
9. **Interrupt**
   - Timer0 / Timer1 / T2 / WKT / LVD Interrupt

- ADC / TK / UART Interrupt
- All GPIOs support pin level change interrupts (except PD4 and PD5).

10. Wake-up timer (WKT)

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

11. Watchdog Timer (WDT)

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

12. 6 groups of 16-bit PWM

- PWM clock is system clock (Fsys), FIRC/256, FIRC (18.432 MHz), or FIRC*2 (36.864 MHz)
- Shared cycle, cycle adjustable
- Independent duty cycle, duty cycle adjustable
- PWM0 supports complementary output (PWM0P, PWM0N), with adjustable dead

13. TKTouch Key

- 13 channels
- 3-bit TK reference clock capacitance adjustment
- 8-bit touch key clock frequency selection (fixed frequency or automatic change)
- 14-bit TK scan length adjustment
- Interrupt/wake up CPU when key is pressed

14. UART

- 7/8/9 bits mode TX/RX selectable
- Supported Baud-Rate range from 9600bps to 115200 bps with proper selected oscillation frequency and baud rate clock divide
- Automatic parity generation and detection
- Detects Overrun, Frame Error and Parity Error

15. 14-bit ADC converter

- 21 input channels and 2 internal reference voltages (VBG and 1/4VCC)
- ADC reference voltage: VCC, VBG
- Four voltage options for VBG (1.18V/2V/2.5V/3V), which are determined by the VBGSEL register. 1.18V and 2.5V offer the highest accuracy.

16. Reset

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

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

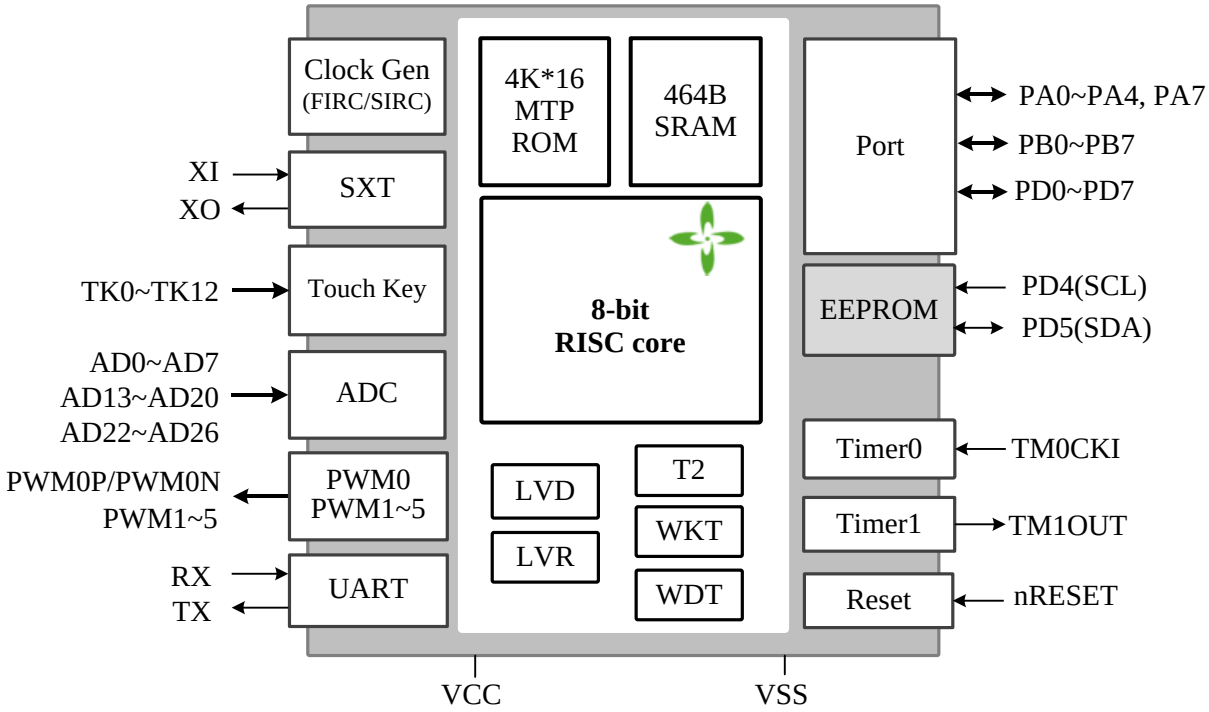
- 16-Level Low Voltage Reset: 1.61V ~ 3.49 V
- 15-Level Low Voltage Detection: 1.87V ~3.49V, 4.3V

18. Operating Voltage : 2.2V~5.5V @ F_{SYS} = 18.432MHz, 25°C

- Users must switch the program ROM read signal delay control register RDCTL to “4ns” to increase the minimum operating voltage.
- When powering up, VCC must exceed POR 1.6V and the selected LVR level. Please refer to the “Electrical Characteristics” chapter to avoid entering the ROM dead zone.

19. **Operating Temperature Range : -40°C to + 105°C**
20. **Integrated 16-bit Cyclic Redundancy Check (CRC)**
21. **Table Read Instruction: 16-bit ROM data lookup table**
22. **Instruction set: 43 Instructions**
23. **Instruction Execution Time**
    - 2 system clocks (Fsys) per instruction except branch
24. **Package Types**
    - SSOP-24
    - SOP-20
    - SOP-16
25. **Emulation tools/chips/interfaces**
    - HT-LINK ICE
    - EV8278
    - PSDA/PSCL use PA0/PA1 port

SYSTEM BLOCK DIAGRAM



TM56M7842 / TM56E7842

PIN ASSIGNMENT DIAGRAM

Please refer to "Appendix: Pin Assignment Diagram" (left-click the link to access the page).

PIN DESCRIPTIONS

Name	In/Out	Pin Description
nRESET	I	External low-level effective reset
VCC, VSS	P	Power supply input pin and ground
PA7, PA4–PA0	I/O	Bit-programmable I/O port for Schmitt trigger input, CMOS push-pull output, or open-drain output. Pull-up resistors can be assigned by software.
PB7-PB0	I/O	Bit-programmable I/O port for Schmitt trigger input, CMOS push-pull output, or open-drain output. Pull-up resistors can be assigned by software.
PD7-PD6 PD3-PD0	I/O	Bit-programmable I/O ports for Schmitt trigger inputs, CMOS push-pull outputs, or open-drain outputs. Pull-up resistors can be assigned by software.
PD5, PD4	I/O	Bit-programmable I/O ports for Schmitt trigger inputs, CMOS push-pull outputs, or open-drain outputs. Pull-up resistors can be assigned by software. PD4 and PD5 of the TM56E7842 are EEPROM communication pins; pull-up resistors can be assigned by software. PD4 is the clock pin (SCL), and PD5 is the data pin (SDA). Communication control is handled by the S/W switch.
PWM0P/PWM0N	O	PWM0 Complementary output
PWM1/PWM2/ PWM3/PWM4/ PWM5	O	PWM1/PWM2/PWM3/PWM4/PWM5 output
PWM0PM	O	The PWM0PSEL register determines whether to change the PWM0P output from PA7 to PB0. *This output function is not supported on the emulation chip.
PWM5M	O	The PWM5SEL register determines whether to change the PWM5 output from PA7 to PB1. *This output function is not supported on the emulation chip.
AD0~AD7/ AD13~AD20/ AD22~AD26	I	ADC input pin
TK0_0~TK0_12	I	Key module 0 (TKM0) input
TM0CKI	I	Timer0 input pin in counter mode
TCOUT	O	Fsys/2 clock output
TM1OUT	O	Timer1 overflow switch output
XI, XO	-	Connection for quartz crystal resonator for system clock
RX	I	UART receive pin
TX	O	UART transmit pin

Programming pins:

6-wire programming: VCC / VSS / PA0 / PA1 / PB0 / PA7

5-wire programming: VCC / VSS / PA0 / PA1 / PA7

Pin Summary

1	TM1OUT/PWM0PM/TK0_00/AD0/PB0
2	TCOUT/PWM5M/TK0_01/AD1/PB1
3	TK0_02/AD2/PB2
4	TK0_03/AD3/PB3
5	RX/TK0_04/AD4/PB4
6	TX/TK0_05/AD5/PB5
7	TK0_06/AD6/PB6
8	TK0_07/AD7/PB7
9	VSS
10	VCC
11	AD17/PD4
12	AD18/PD5
13	PA0/AD25/TK0_11/PWM0N
14	PA2/AD26/TK0_12/PWM2/TM0CKI
15	PA1/AD24/TK0_10/PWM1
16	PA3/AD23/TK0_09/PWM3
17	PA4/AD22/TK0_08/PWM4
18	PA7/PWM0P/PWM5/VPP/nRESET
19	PD3/AD16/TX
20	PD2/AD15/RX
21	PD0/AD13/XO
22	PD1/AD14/XI
23	PD7/AD20
24	PD6/AD19

FUNCTION DESCRIPTION

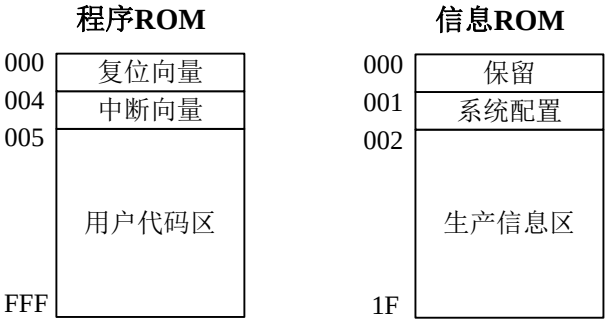
1. CPU Core

1.1 ROM

The ROM can be written to multiple times. The program ROM has a size of 4K × 16 and includes an additional 32 × 16 information ROM.

Under the programmer, when the PROTECT bit in SYSCFG is set to 0, both the program ROM and information ROM can be read and written normally. When the PROTECT bit is set to 1, the program ROM cannot be read, and only the information ROM can be read.

The PROTECT bit can only be cleared after the program ROM has been erased to 0.



106h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RDCTL	–	–	–	–	–	–	RDCTL	
R/W	–	–	–	–	–	–	R/W	R/W
Reset	–	–	–	–	–	–	0	1

191h.1~0 **RDCTL:** Program ROM read signal delay control (this feature is not simulated)
00: Program ROM read signal delay 4ns
01: Program ROM read signal delay 8ns
10: Program ROM read signal delay 12ns
11: Program ROM read signal delay 16ns
*For safety reasons, please change the register under a slow clock.
*Users must switch this register to “4ns” to increase the chip's minimum operating voltage.

1.1.1. Reset Vector (000H)

After reset, system will restart the program counter (PC) at address 000h, all registers will be restored to the default values.

1.1.2. Interrupt Vector (004H)

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

1.1.3. System Configuration Register (SYSCFG))

The production information area is located at the beginning of the information ROM and stores information such as production code, checksum, and adjustment values. The 16-bit system configuration (SYSCFG) is also located here, as shown in the table below. Many system initial conditions are determined by SYSCFG. The 15th bit of SYSCFG is the code protection selection bit. If this bit is set to 1, the data in the program ROM will be protected and cannot be read.

Bit	Description	
15	PROTECT: Code Protection Selection	
	0	Close
	1	Enable
14	XRSTE: External Pin (PA7) Reset Enable	
	0	Close
	1	Enable
13~10	LVR: Low Voltage Reset Voltage Option	
	0000	LVR 1.61V
	0001	LVR 1.74V
	….	….
	1110	LVR 3.37V
	1111	LVR 3.49V
9~8	WDTE: Watchdog Reset Enable	
	0X	WDT disabled
	10	WDT enabled in FAST/SLOW mode, disabled in IDLE/STOP mode
	11	WDT always enabled
7	TKD: Touch Function Option	
	0	Touch function enabled
	1	Touch function disabled
4	PORSEL: POR Power-Saving Function Option	
	0	Disabled
	1	Enabled. POR operates at 1/8 duty cycle (this feature is not simulated)
3	ATDOFF: Automatic Transient Detection	
	0	Enable
	1	Disable

System Configuration Area (SYSCFG)

1.2 RAM and special function registers

The total size of the RAM on this chip is 464 bytes. The Special Function Registers (SFR) and RAM share a 9-bit address for addressing, divided into Banks 0–3 and Bank 4. The addresses F0–FF, 170–17F, and 1F0–1FF in Banks 0–3 all point to the 16-byte addresses 70–7F. Special Function Registers are located in lower positions within Banks 0–3, and some frequently used Special Function Registers, such as STATUS, are designed to be mirrored across multiple Banks to reduce code size and enable faster access. In addition to direct addressing, RAM also supports indirect addressing access.

	BANK0 00~7Fh		BANK1 80h~FFh		BANK2 100h~17Fh		BANK3 180h~1FFh		BANK4 00~7Fh
00h	INDF0	80h	INDF0	100h	INDF0	180h	INDF0	00h	
01h	TM0	81h	OPTION	101h	TM0	181h	OPTION		
02h	PCL	82h	PCL	102h	PCL	182h	PCL		
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS		
04h	FSR	84h	FSR	104h	FSR	184h	FSR		
05h	PAD	85h	PAMODH	105h	TESTREG	185h	DPL		
06h	PBD	86h	PAMODL	106h	RDCTL	186h	DPH		
07h	-	87h	PBMODH	107h	LVRPD	187h	UARTCTL		
08h	PDD	88h	PBMODL	108h	-	188h	UARTSTA		
09h	INDF1	89h	INDF1	109h	INDF1	189h	INDF1		
0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH		
0Bh	INTIE	8Bh	INTIE	10Bh	INTIE	18Bh	INTIE		
0Ch	INTIF	8Ch	-	10Ch	PCH	18Ch	TABR		
0Dh	FSR1	8Dh	PDMODH	10Dh	UBAUD	18Dh	URDATA		
0Eh	INTIE1	8Eh	PDMODL	10Eh	BGTRIM	18Eh	CRCDL		
0Fh	CLKCTL	8Fh	OPTION2	10Fh	IRCF	18Fh	CRCDH		
10h	TM0RLD	90h	PWMOE	110h	-	190H	CRCIN		
11h	TM0CTL	91h	PWMCTL	111h	-	191h	-		
12h	TM1	92h	PWMPRDH	112h	-	192h	-		
13h	TM1RLD	93h	PWMPRDL	113h	-	193h	-		SRAM 128 Bytes
14h	TM1CTL	94h	PWM0DH	114h	-	194h	TKM0TMRL		
15h	T2CTL	95h	PWM0DL	115h	-	195h	TKM0TMRH		
16h	LVCTL	96h	PWM1DH	116h	TKM0DL	196h	-		
17h	ADCDH	97h	PWM1DL	117h	TKM0DH	197h	-		
18h	ADCTL	98h	PWM2DH	118h	ATKUBL	198h			
19h	ADCTL2	99h	PWM2DL	119h	ATKUBH	199h			
1Ah	INTIF1	9Ah	PWM3DH	11Ah	ATKLBL	19Ah	TKM0REFC		
1Bh	-	9Bh	PWM3DL	11Bh	ATKLBH	19Bh	SFR19B		
1Ch	PAWKE	9Ch	PWM4DH	11Ch	TKMCON0	19Ch	ATKCON		
1Dh	PBWKE	9Dh	PWM4DL	11Dh	TKMCON1	19Dh	TKMCHS		
1Eh	-	9Eh	PWM5DH	11Eh	TKMCTL0	19Eh	-		
1Fh	PDWKE	9Fh	PWM5DL	11Fh	TKMCTL1	19Fh	-		
20h 6Fh	SRAM 80 Bytes	A0h EFh	SRAM 80 Bytes	120h 16Fh	SRAM 80 Bytes	1A0h 1EFh	SRAM 80 Bytes		
70h 7Fh	SRAM 16 Bytes	F0h FFh	accesses 70h~7Fh	170h 17Fh	accesses 70h~7Fh	1F0h 1FFh	accesses 70h~7Fh	7Fh	

Special Function Registers (SFR) and RAM

1.2.1 Addressing Mode

Direct addressing of BANK0~3:

When directly addressing special function registers (SFR) and RAM, RP1 and RP0 represent the most significant bit and second most significant bit of the address, respectively, which together with the 7-bit address supported by the direct addressing instruction constitute a 9-bit effective address. In other words, the two register bits RP1 and RP0 determine which BANK to point to.

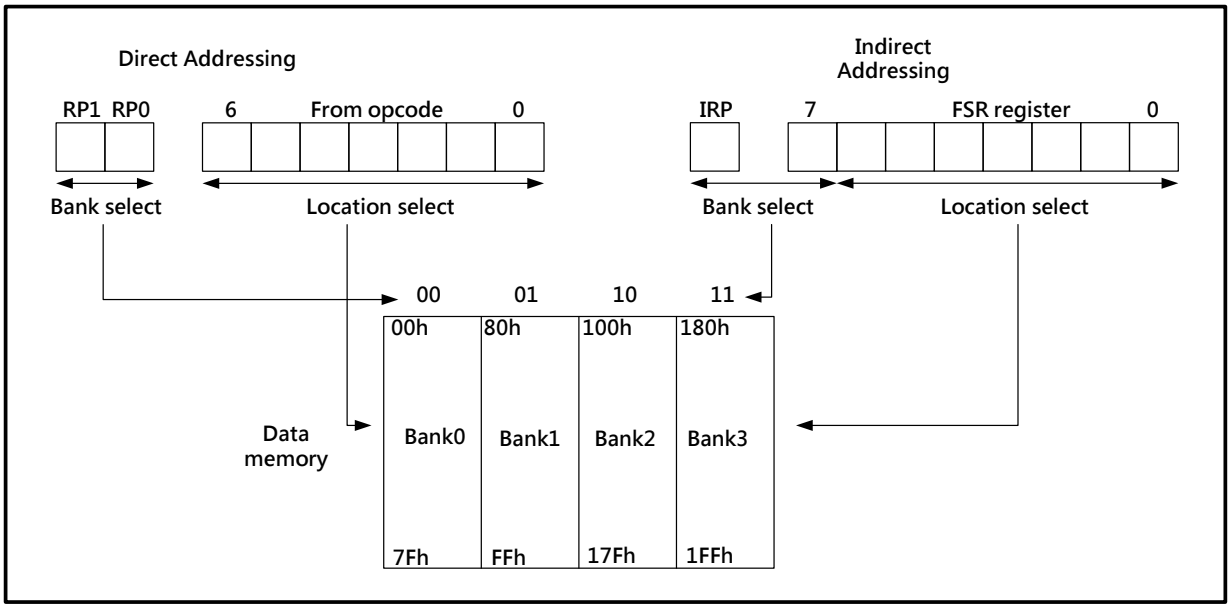
RP1,RP0 (03h.6~5)	BANK
00	0
01	1
10	2
11	3

When users use the new instruction set, keeping RP0=RP1=0 allows them to ignore the BANK position of the register and save code size. By replacing the "F" in the old instruction with "X," the old instruction can be easily replaced with the new instruction.

BC**F**	TM0IE	➔	BC**X**	TM0IE
DEC**F**	CNT, 1	➔	DEC**X**	CNT,1
INC**F**SZ	RAM25, 0	➔	INC**X**SZ	RAM25, 0
MOVW**F**	PAMODL	➔	MOVW**X**	PAMODL
RL**F**	RAMA0, 0	➔	RL**X**	RAMA0, 0
SWAP**F**	ADCTL, 0	➔	SWAP**X**	ADCTL, 0

Indirect addressing between BANK0 and BANK3:

When RAM is accessed via indirect addressing, the IRP register represents the most significant bit of the address. Together with the FSR register, the IRP register forms a 9-bit effective address. Indirect addressing is achieved by using the FSR register and the INDF register. The FSR represents the pointer and stores the address. The INDF is not a physical register; any instruction using INDF actually accesses the memory pointed to by the FSR.



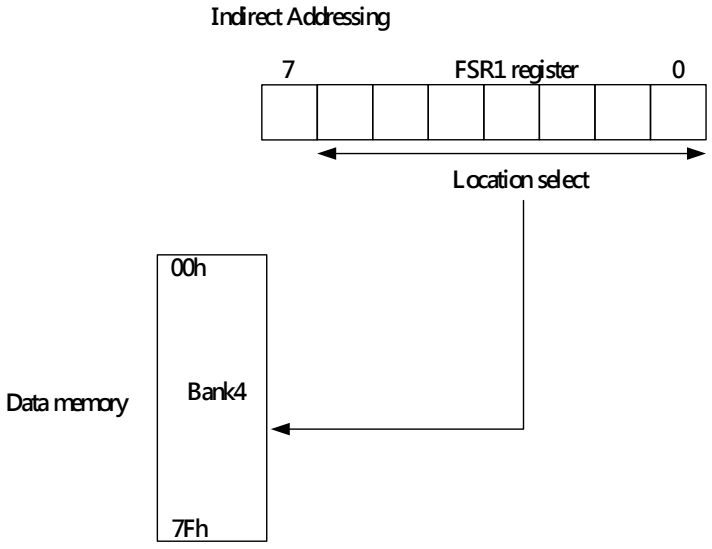
BANK0/1/2/3 Direct/indirect addressing

BANK4 Indirect Addressing:

RAM can achieve indirect addressing by using the FSR1 register and the INDF1 register. FSR1 represents the pointer and stores the address. INDF1 is not a physical register; any instruction that uses INDF1 actually accesses the memory pointed to by FSR1.

Note 1: ADDWXC, SUBWXB, SUBXW, and SUBXWB do not support access to FSR1 and INDF1.

Note 2: C language compilation does not support memory operations in RAM Bank4.



BANK4 Indirect Addressing

Example: Use direct addressing (RP0=RP1=0) to read/write registers for bank0/1/2/3.

```
TM1             equ     12h     ;SFR in Bank0
PWM1DH          equ     96h     ;SFR in Bank1
TKMCTL1         equ     11Fh    ;SFR in Bank2
TKM0TMRL        equ     194h    ;SFR in Bank3
RAM20           equ     20h     ;RAM in Bank0
RAMA0           equ     A0h     ;RAM in Bank1
RAM120          equ     120h    ;RAM in Bank2
RAM1A0          equ     1A0h    ;RAM in Bank3

MOVXW   TM1             ; read TM1          (Bank0) to W
MOVXW   PWM1DH          ; read PWM1PRD   (Bank1) to W
MOVXW   TKMCTL1         ; read TKMCTL1    (Bank2) to W
MOVXW   TKM0TMRL        ; read TKM0TMRL (Bank4) to W

MOVLW   16h
MOVWX   RAM20           ; W=16h Write to RAM[0x20]
MOVWX   RAMA0           ; W=16h Write to RAM[0xA0]
MOVWX   RAM120          ; W=16h Write to RAM[0x120]
MOVWX   RAM1A0          ; W=16h Write to RAM[0x1A0]
.
MOVLW   037H            ; W=37H
MOVWX   LVRPD           ; force LVR+POR close
```

◇ Example: Use indirect addressing (RP0=RP1=0, IRP=1) to read/write registers for bank2/3.

```
BSX     IRP             ; IRP=1  =>Bank2/3
MOVLW   1Fh             ; W=1Fh
MOVWX   FSR             ; FSR = W =10h
MOVXW   INDF            ; read SFR TKMCTL1 (11Fh) to W
.
BSX     IRP             ; IRP=1 =>Bank2/3
MOVLW   1Fh             ; W=1Fh
MOVWX   FSR             ; FSR = W =1Fh
MOVLW   37h             ; W=37h
MOVWX   INDF            ; TKMCTL1 (11FH) = W = 37h
```

◇ Example: Use indirect addressing (RP0=RP1=0, IRP=0) to read/write registers using bank0/1.

```
BCX     IRP             ; IRP=0  =>Bank0/1
MOVLW   10h             ; W=10h
MOVWX   FSR             ; FSR = W =10h
MOVXW   INDF            ; read SFR TM0RLD (10h) to W
.
BCX     IRP             ; IRP=0 =>Bank0/1
MOVLW   10h             ; W=10h
MOVWX   FSR             ; FSR = W =10h
MOVLW   37h             ; W=37h
MOVWX   INDF            ; TM0RLD (10h) = W = 37h
```

◇ Example: Use indirect addressing to read/write registers using bank4.

```
; Write data to BANK4 memory
MOVLW   20h             ; W=20h
MOVWX   FSR1            ; FSR1 = W =20h
MOVLW   5Ah             ; W=5AH
MOVWX   INDF1           ; Write data 5AH to BANK4 memory @20h

; Write data to BANK5 memory
MOVLW   A7h             ; W=A7h
MOVWX   FSR1            ; FSR1 = W =A7h
MOVLW   69h             ; W=69h
MOVWX   INDF1           ; Write data 69h to BANK5 memory @A7h

; Read data from BANK4 memory
MOVLW   20h             ; W=20h
MOVWX   FSR1            ; FSR1 = W =20h
MOVXW   INDF1           ; Write data BANK4 data @20h to W
```

00h/80h/100h/180h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF	INDF							
R/W	R/W							
Reset	0							

00h/80h/100h/180h.7~0 **INDF:** Used for indirect addressing between bank0 and bank3.
It is not a physical register; any instruction that uses INDF actually accesses the memory pointed to by FSR.

04h/84h/104h/184h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSR	FSR							
R/W	R/W							
Reset	0							

04h/84h/104h/184h.7~0 **FSR:** Used for indirect addressing between bank0 and bank3.
FSR stands for pointer, which stores addresses. Any instruction that uses INDF actually accesses the memory pointed to by FSR.

09h/89h/109h/189h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF1	INDF1							
R/W	R/W							
Reset	0							

09h/89h/109h/189h.7~0 **INDF1:** Used for indirect addressing of bank 4.
This is not a physical register; any instruction that uses INDF1 actually accesses the memory pointed to by FSR1.

0Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSR1	FSR1							
R/W	R/W							
Reset	0							

0Dh.7~0 **FSR1:** Used for indirect addressing of bank 4.
FSR1 represents a pointer that stores an address. Any instruction that uses INDF1 actually accesses the memory pointed to by FSR1.

1.3 Program Counter and Stack

1.3.1. Program Counter (PC)

The program counter is 12 bits wide and is used to address a 4Kx16 program ROM.

When executing a program instruction, the program counter contains the address of the next program instruction to be executed. The program counter value typically increments by 1 continuously unless a reset, interrupt, call, jump, or return instruction is encountered.

The initial reset vector (000h) and interrupt vector (004h) are used for program counter initialization and interrupt events. For return (RET/RETI/RETLW) instructions, the program counter retrieves its contents from the top of the stack. For call (CALL) and jump (GOTO) instructions, the lower 11 bits of the program counter are obtained from the instruction code, and the most significant bit of the program counter is obtained from the PCLATH.3 register. Therefore, before executing a CALL/GOTO instruction, if the target address is greater than 2K, the PCLATH.3 register must be set; otherwise, the PCLATH.3 register remains at 0.

For users who use assembly code, the chip also provides new LCALL/LGOTO instructions to replace the CALL/GOTO instructions. When using the new instructions, users do not need to worry about the destination address; they only need to keep PCLATH.3 at 0. The instruction replacements are as follows.

CALL	TABLE	➔	LCALL	TABLE
GOTO	TABLE	➔	LGOTO	TABLE

Users can read PC[11:8] (i.e., the upper 4 bits of the program counter) via the PCH register.

Users can read and write PC[7:0] (i.e., the lower byte of the program counter) via the PCL register.

When the CPU executes any instruction that modifies the PCL register, PC[11:8] is defaulted by the PCLATH register. For users working with assembly code, this default behavior can be altered. After writing 1Ch to the PCH register, when the CPU executes any instruction that modifies the PCL, PC[11:8] remains unchanged, facilitating lookup table operations for users.

1.3.2. STACK

The stack is 12 bits wide and 8 levels deep, used to store the addresses of program instructions. When a call (CALL/LCALL) instruction is executed or an interrupt event occurs, the instructions are pushed onto the stack in order. Following the last-in, first-out (LIFO) principle, when a return (RET/RETI/RETlW) instruction is executed, the instructions are popped off the stack in order.

002h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCL	PCL							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

002h.7~0 **PCL:** PC[7:0], i.e., the lower byte of the program counter.

00Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCLATH	GPR				PCLATH			
R/W	R/W				R/W			
Reset	0	0	0	0	0	0	0	0

00Ah.3~0 **PCLATH:** PC[11:8] (i.e., the upper 4 bits of the program counter) is written to the buffer register. When the CPU executes any instruction that modifies the PCL register, PC[11:8] is provided by the PCLATH register by default. This function can be disabled by the PCH register.

10Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCH	PCH							
R/W	W				R/W			
Reset	0	0	0	0	0	0	0	0

10Ch.7~0 **PCH: (**read**)**
(Only available to users of assembly language; not applicable to users of C language)
After writing 1Ch to this register, when the CPU executes any instruction that modifies the PCL, PC[11:8] remains unchanged.
After writing other values to this register, when the CPU executes any instruction that modifies the PCL, PC[11:8] is provided by the PCLATH register.

10Ch.3~0 **PCH: (**write**)**
Read PC[11:8], i.e., the upper 4 bits of the program counter.

1.4 ALU and Work (W) Registers

The ALU is 8 bits wide and can perform addition, subtraction, shifting, and logical operations. In two-operand instructions, one operand is typically the W register, which is an 8-bit non-addressable register used for ALU operations. The other operand is either a file register or a direct constant. In a single-operand instruction, the operand is either the W register or a file register. Depending on the instruction being executed, the ALU may affect the values of the carry (C), carry-done (DC), and zero (Z) flags in the status register. The C and DC flags also serve as the borrow and half-borrow flags for subtraction, respectively.

*The borrow state is the opposite of the borrow value. The half-borrow state is the opposite of the half-borrow value.

1.5 Status Register (STATUS)

This register contains the arithmetic status and reset status of the ALU. Like other registers, the STATUS register can be the destination of any instruction. If the STATUS register is the destination of an instruction that affects the Z bit, DC bit, or C bit, then write operations to these three bits will be disabled. These bits will be set or cleared according to device logic. Therefore, it is recommended to use only the BCX, BSX, and MOVWX instructions to modify the STATUS register, as these instructions do not affect these bits.

<table>
<tr><td>STATUS</td><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td>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>Reset</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>Bit</td><td colspan="8">Description</td></tr>
<tr><td>7</td><td colspan="8">IRP: The register is located in the Bank selection bit (used for indirect addressing).
0 = Bank 0,1 (00h - FFh)
1 = Bank 2,3 (100h - 1FFh)</td></tr>
<tr><td>6~5</td><td colspan="8">RP1:RP0: Register located in the Bank selection bit (used for direct addressing)
00 = Bank 0 (000h - 07Fh)
01 = Bank 1 (080h - 0FFh)
10 = Bank 2 (100h - 17Fh)
11 = Bank 3 (180h - 1FFh)</td></tr>
<tr><td>4</td><td colspan="8">TO: Timeout flag
0: After power-on reset or CLRWDT/SLEEP instruction
1: WDT timeout</td></tr>
<tr><td>3</td><td colspan="8">PD: Power loss flag
0: After power-on reset or CLRWDT instruction
1: After executing SLEEP instruction</td></tr>
<tr><td>2</td><td colspan="8">Z: Zero flag
0: The result of the logical operation is not zero.
1: The result of the logical operation is zero.</td></tr>
<tr><td rowspan="3">1</td><td colspan="8">DC: Decimal 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: Carry in lower half byte</td><td colspan="4">0: Low half-byte has borrow position
1: No borrow position</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: Carry</td><td colspan="4">0: Borrowed position
1: No borrowed position</td></tr>
</table>

◇ Example: Write immediate data to the status register STATUS register

```
MOVLW    00h
MOVWX    STATUS        ; Clear the STATUS register
```

◇ Example: Bit addressing, setting and clearing the STATUS register.

```
BSX      STATUS, 0     ; Setup C=1.
BCX      STATUS, 0     ; Setup C=0.
```

◇ Example: Determine the C flag using the BTXSS instruction.

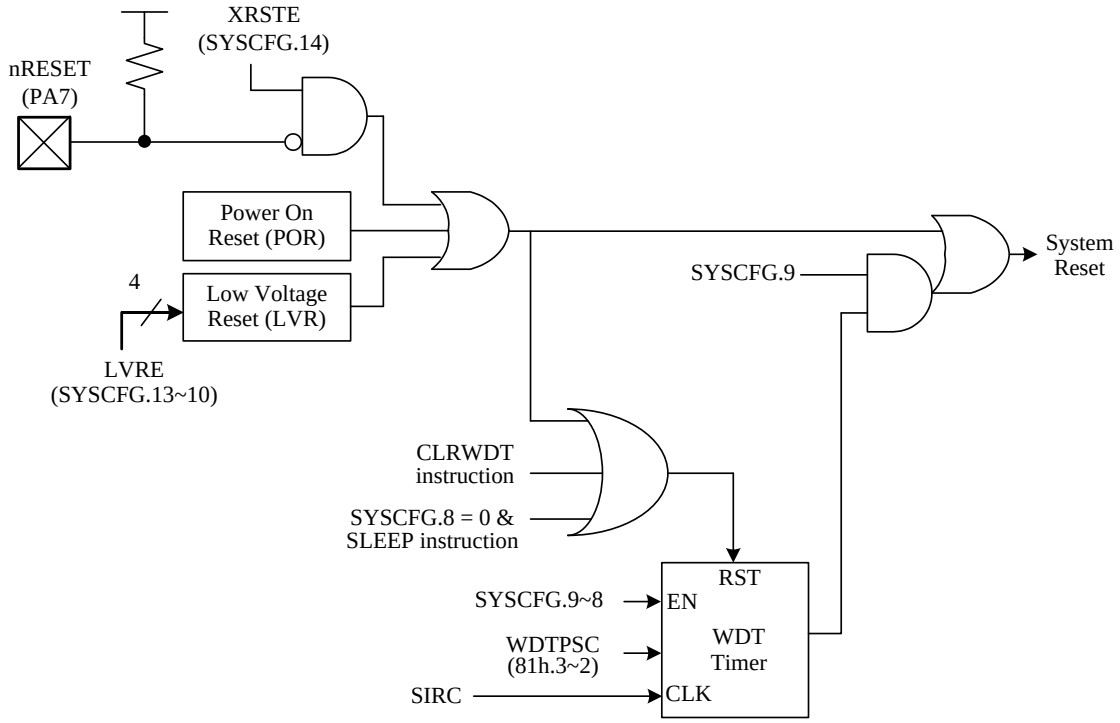
```
BTXSS    STATUS, 0     ; Check the carry flag
LGOTO    LABEL_1       ; If C=0, go to label_1
LGOTO    LABEL_2       ; If C=1, go to label_2
```

2. Reset

The chip can be reset in four ways.

- Power-on reset (POR)
- Low voltage reset (LVR)
- External pin reset (XRST)
- Watchdog Reset (WDTR)

After a reset, the program counter (PC) is cleared, the system starts running from reset vector 000h, and the registers return to their default values. The TO and PD flags at status register (STATUS) are indicate system reset status.



2.1 Power on Reset (POR)

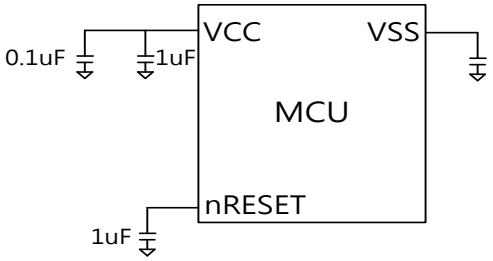
After Power-On-Reset, all system and peripheral control registers are then set to their default hardware Reset values.

2.2 Low Voltage Reset (LVR) (LVR)

The function of low voltage reset is to perform static reset when the power supply voltage is below the threshold level. 16 threshold levels can be selected. The voltage threshold for the low voltage reset function is defined by SYSCFG.

2.3 External Pin Reset (XRST)

The external reset pin is valid for low, which must be longer than 2 SIRC (Slow Internal RC) clock cycles to be detected by the chip. The external pin reset feature can be disabled or enabled by configuring SYSCFG.



2.4 Watchdog Timer Reset (WDT)

The watchdog cycle can be selected by the WDTPSC register. The watchdog timer clock is provided by SIRC (slow internal RC), and the watchdog timer value is cleared by other resets or by the CLRWDT instruction. The watchdog overflow reset function can be disabled or enabled by setting SYSCFG.

◇ Example: Clear the watchdog timer using the CLRWDT instruction.

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

81h/181h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	HWAUTO	INT0EDG	INT1EGE	FREQL	WDTPSC		WKTPSC	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	1	1	1	1

81h.4 **FREQL:** SIRC frequency selection
0: 60KHz
1: 50KHz

81h.3~2 **WDTPSC:** Watchdog cycle
FREQL=1:
00: 188ms@3V, 169ms@5V
01: 371ms@3V, 333ms@5V
10: 749ms@3V, 668ms@5V
11: 1492ms@3V, 1355ms@5V
FREQL=0:
00: 147ms@3V, 135ms@5V
01: 293ms@3V, 272ms@5V
10: 580ms@3V, 529ms@5V
11: 1180ms@3V, 1058ms@5V

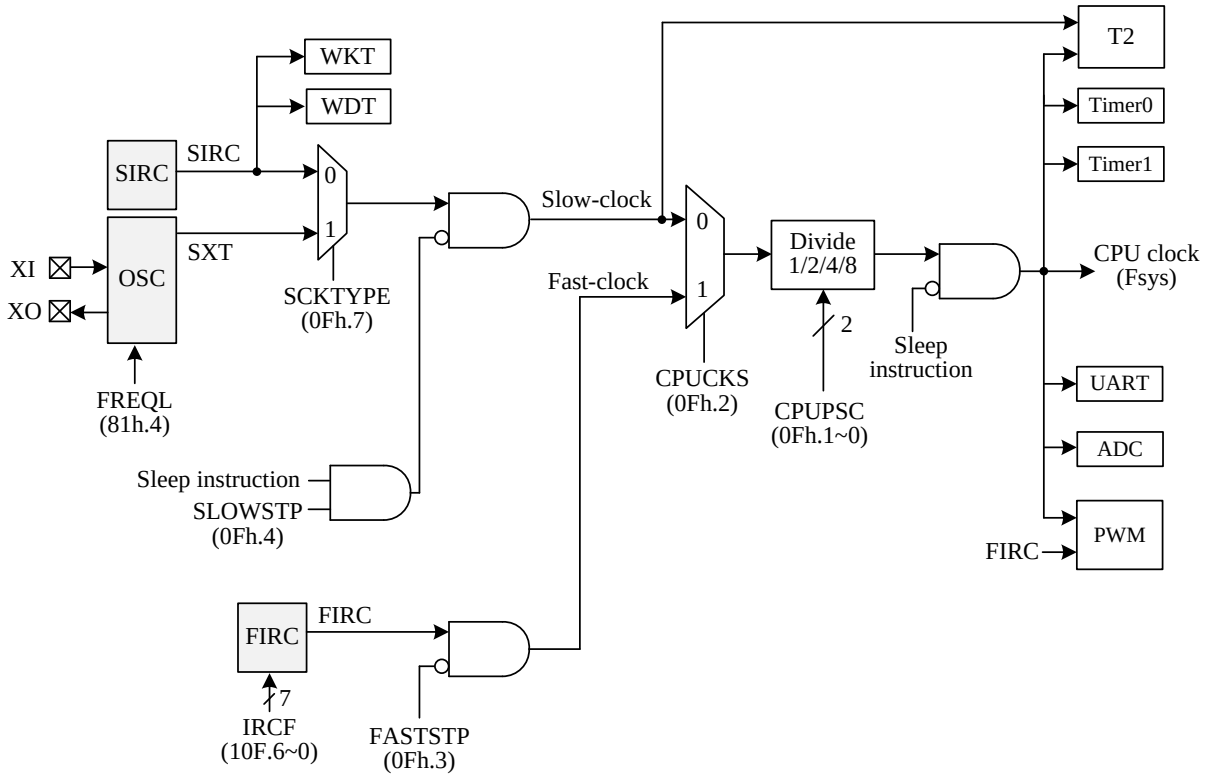
3. Clock Circuitry and Operation Mode

3.1 System Clock

The device features a dual-clock system design. There are two clock sources: the slow clock (Slow-clock) and the fast clock (Fast-clock). The fast clock is the FIRC (Fast Internal RC), and its frequency can be adjusted via the IRCF register. The slow clock can be selected via the SCKTYPE register to use either SIRC (Slow Internal RC) or SXT (Slow External Crystal Oscillator, 32 kHz). If FREQL (81h.4) is set to 0, the SIRC frequency can be switched from 50 kHz to 60 kHz. After a system reset, the CPU clock operates in slow mode with the slow clock set to SIRC.

The CLKCTL register controls the operation of the system clock. The hardware automatically prevents software from making abnormal settings to this register. It is recommended that users use a bitwise write method to modify the CLKCTL register.

The Timer0 and Timer1 modules are driven by Fsys. The T2 module can be driven by either the slow clock or Fsys/128 by setting the T2CKS register. The PWM module can be driven by Fsys, FIRC/256, FIRC (18.4 MHz), or FIRC*2 (36.8 MHz) by setting the PWMCKS register.



Clock Scheme Block Diagram

Fast Mode (FAST):

The CPU clock (Fsys) is the fast clock.

Slow Mode (SLOW):

The CPU clock (Fsys) is the slow clock.

The fast clock can be set to stop or run via the FASTSTP register.

After device reset, this mode is entered with the slow clock set to “SIRC”.

Idle Mode (IDLE):

The CPU clock (Fsys) is stopped. The slow clock remains active.

If SLOWSTP (0Fh.4) is 0 or WKTIE (0Bh.3) is 1 or WDTE (SYSCFG.9~8) is 3, the chip enters idle mode after executing the SLEEP instruction.

If T2CKS is 0, T2 can be used to wake the device from Idle mode.

If WKTIE is 1, WKT can be used to wake the device from Idle mode.

The touch key can be used to wake the device from Idle mode.

The pin interrupt (PXIF) can be used to wake the device from Idle mode.

Stop Mode (STOP):

The CPU clock (Fsys) stops. All clocks in the device stop.

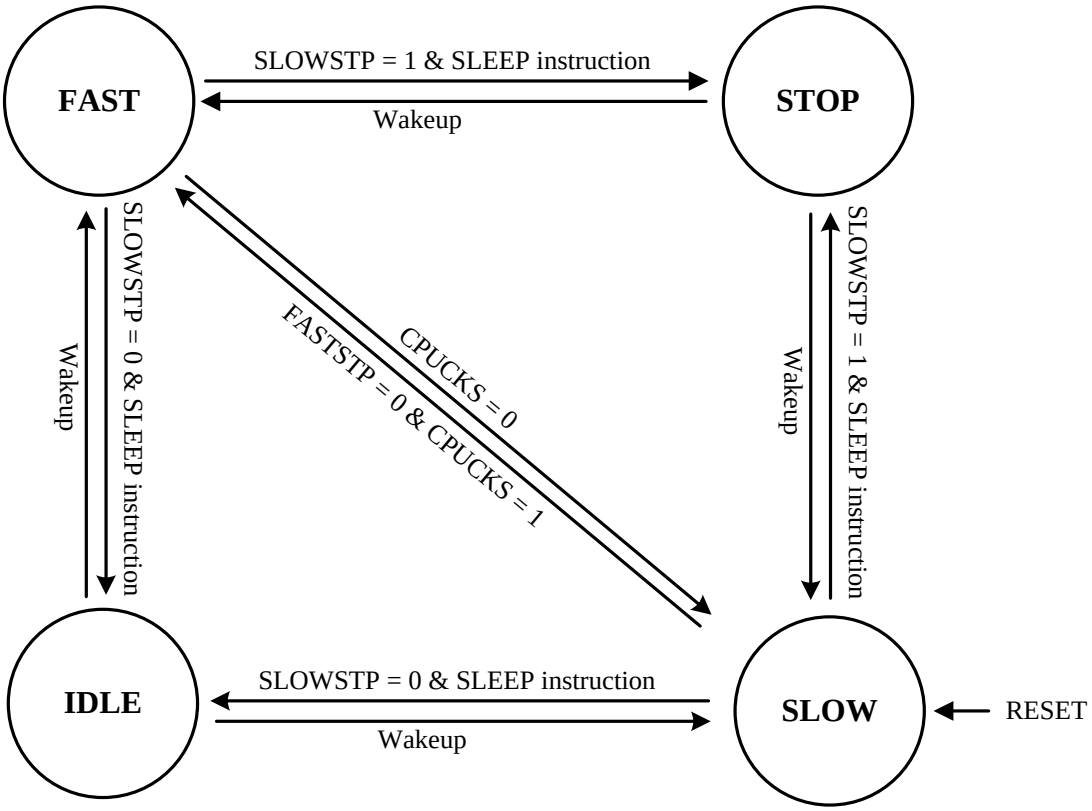
If SLOWSTP (0Fh.4) is 1, WKTIE (0Bh.3) is 0, and WDTE (SYSCFG.9~8) is not 3, the chip will enter stop mode after executing the SLEEP instruction. If both the LVRSAV register and the LVDSAV register are set to 1, the lowest power consumption can be achieved.

The touch key can be used to wake the device from Stop mode.

The pin interrupt (PXIF) can be used to wake the device from Stop mode.

3.2 Dual System Clock Modes Transition

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



*Wakeup refers to wake-up events such as IO interrupts, WKT interrupts, or T2 interrupts.

Clock system switching

Mode	Fsys	Fast-clock	Slow-clock	Timer0 Timer1	T2	Wakeup event (WKT)	Wakeup event
FAST	Fast-clock	Run	Run	Run	Run	Set by WKTIE	-
SLOW	Slow-clock	Set by FASTSTP	Run	Run	Run	Set by WKTIE	-
IDLE	-	Stop	Run	Stop	Set by T2CKS	Set by WKTIE	IO/WKT/T2
STOP	-	Stop	Stop	Stop	Stop	Stop	IO

Clock Mode and Power Consumption Control

- **Switch from FAST mode to SLOW mode**

  Perform the following steps in order:

  Enable Slow-clock (SLOWSTP=0)

  Switch to Slow-clock (CPUCKS=0)

  Stop Fast-clock (FASTSTP=1)

◇ Example: Switch from FAST mode to SLOW mode.

```
BCX     SLOWSTP     ; enable Slow-clock
NOP
BCX     CPUCKS      ; Fsys=Slow-clock
BSX     FASTSTP     ; close Fast-clock
```

- **Switch from SLOW mode to FAST mode**

Perform the following steps in order:

Enable Fast-clock (FASTSTP=0)

Switch to Fast-clock (CPUCKS=1)

◇ Example: Switch from SLOW mode to FAST mode.

```
BCX     FASTSTP     ; enable Fast-clock.
NOP
BSX     CPUCKS      ; Fsys=Fast-clock
```

- **Enter IDLE mode from FAST/SLOW mode**

Perform the following steps:

Execute the sleep command (SLEEP)

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

```
BCX     SLOWSTP     ; enable Slow-clock
SLEEP               ; Enter IDLE mode
```

- **Enter STOP mode from FAST/SLOW mode**

Perform the following steps:

Stop slow clock (SLOWSTP=1)

Stop WKT (WKTIE=0)

Execute sleep command (SLEEP)

◇ Example: Switch FAST/SLOW mode to STOP mode

```
BSX     SLOWSTP     ; close Slow-clock.
MOVLW   0000 0000B  ; close WKT
MOVWX   INTIE
SLEEP               ; Enter STOP mode
```

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

0Fh.7 **SCKTYPE:** Slow clock type. This bit can only be changed in FAST mode (CPUCKS=1).
0: SIRC
1: SXT, PD0 and PD1 are crystal oscillator pins.

0Fh.4 **SLOWSTP**: turn off the slow clock
0: Slow clock continues to run after the SLEEP instruction.
1: If no other peripheral function blocks use the slow clock, the slow clock stops running after the SLEEP instruction.

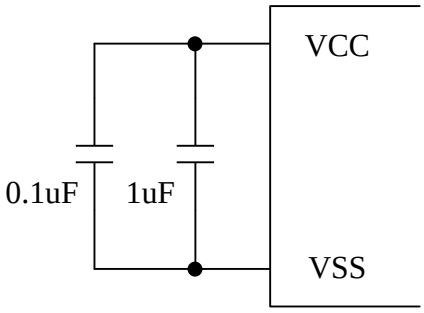
0Fh.3 **FASTSTP**: Turn off the fast clock.
0: System clock is slow, fast clock remains running
1: System clock is slow, fast clock is turned off

0Fh.2 **CPUCKS**: System clock source selection
0: Slow clock
1: Fast clock

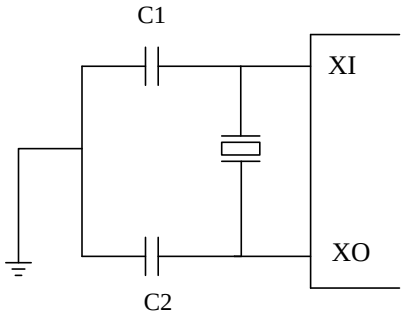
0Fh.1~0 **CPUPSC**: System clock source prescaler. System clock source
00: Divide by 8 01: Divide by 4
10: Divide by 2 11: Divide by 1

3.3 External capacitors

The system clock can operate in three different oscillation modes: FIRC, SIRC, and SXT. Since power supply noise reduces the performance of the clock oscillator, placing 1 uF and 0.1 uF power bypass capacitors close to the VCC/VSS pins can improve the stability of the clock and the entire system.



Power supply bypass capacitor



Crystal oscillator and matching capacitor

3.4 Calculating the ratio of slow clock to fast clock

Allow users to calculate the current ratio of slow clock to fast clock. Given the fast clock frequency, the slow clock rate can be calculated. The sampling time is fixed at 1 slow clock cycle, during which the Timer0 counter is occupied, and the count value can be obtained from register TM0.

The formula is as follows:

Slow clock period = TM0 value × 2 × fast clock period

For example, if the fast clock period is known to be 54.25 ns, and the TM0 value obtained after sampling is 205, the current slow clock period can be calculated as 205 × 2 × 54.25 = 22,242 ns.

Example program:

```
;----- The system clock must be set to the fast clock -----
    bcx     faststp
    bsx     cpucks

    bcx     tm0stp  ; Keep the Timer0 pause control bit at 0

    bsx     cfen    ; During the period when cfen is 1, the Timer0 clock source switches
from the system clock to the internal fast clock
                    ; During the period when cfen is 1, user input to Timer0 is not accepted

    bsx     tm0clr  ; Timer0 value reset to zero
    bcx     tm0clr  ;

    bsx     cfst    ; Start sampling

cfwait:
    btxsc   cfst    ;
    lgoto   cfwait  ; Waiting for sampling

    movxw   tm0     ; Read the Timer0 count value after completing
```

19Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SFR19B	CFEN	CFST	TM0CLR	ULEDMASK	TXSEL	RXSEL	PWM0PSEL	PWM5SEL
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

19Bh.7 **CFEN**: Capture frequency function enabled.
When CFEN=1, the hardware switches the clock source of the Timer0 counter to the fast clock.

19Bh.6 **CFST**: Capture frequency function start flag.
CFST=1 starts sampling, and the hardware automatically clears this bit when sampling is complete.

19Bh.5 **TM0CLR:** Timer0 reset.
Resets Timer0 to zero.

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

01h.7~0 **TM0:** Timer0 data

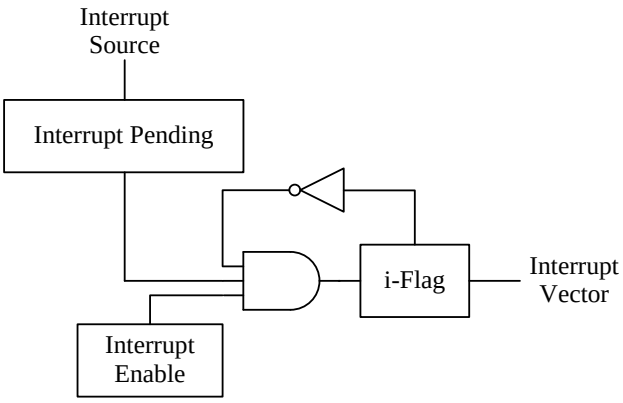
11h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0CTL	T0ISRC	TM0STP	TM0EDG	TM0CKS	TM0PSC			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

11h.6 **TM0STP:** Timer0 counter pause
0: Timer0 running
1: Timer0 paused

4. Interrupt

This chip has 1 level, 1 vector, and 9 interrupt sources. Each interrupt source has its own interrupt enable bit in the register. An interrupt event will set its own suspend flag, regardless of whether its enable control bit is 0 or 1.

If the corresponding register interrupt enable bit (INTIE/INTIE1) is set, it will trigger a CPU service interrupt. The CPU accepts the interrupt at the end of the current instruction cycle. Simultaneously, an "LCALL 004" instruction is inserted into the CPU, and the i-Flag is set to prevent recursive interrupt nesting. The i-Flag is cleared in the instruction following the "RETI" instruction. In other words, at least one instruction in the main program must be executed before the service suspend interrupt. After the interrupt event is triggered, the F/W must clear the interrupt event flag on its own.



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

INTIE.7 **ADCIE:** ADC interrupt enable
0: Disabled
1: Enabled

INTIE.6 **T2IE:** T2 Interrupt Enable
0: Disabled
1: Enabled

INTIE.5 **TM1IE:** Timer1 interrupt enable
0: Disable
1: Enable

INTIE.4 **TM0IE:** Timer0 interrupt enable
0: Disable
1: Enable

INTIE.3 **WKTIE:** Wake-up timer interrupt enable and wake-up timer enable
0: Disabled
1: Enabled

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

0Ch.7 **ADCIF:** ADC interrupt event suspend flag
Set by H/W after ADC conversion is complete. Writing 0 to this bit clears the flag.

0Ch.6 **T2IF:** T2 Interrupt Event Suspend Flag
Set by H/W when T2 overflows. Writing 0 to this bit clears the flag.

0Ch.5 **TM1IF:** Timer1 interrupt event suspend flag
Set by H/W when Timer1 overflows. Writing 0 to this bit clears the flag.

0Ch.4 **TM0IF:** Timer0 interrupt event suspension flag
Set by H/W when Timer0 overflows. Writing 0 to this bit clears the flag.

0Ch.3 **WKTIF:** Timer0 interrupt event suspension flag
Set by H/W when Timer0 overflows. Writing 0 to this bit clears the flag.

0Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE1	LVDIE	-	-	-	TKIE	-	UARTIE	PXIE
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

0Eh.7 **LVDIE:** LVD interrupt enable
0: Disabled
1: Enabled

0Eh.3 **TKIE:** Touch key interrupt enable
0: Disabled
1: Enabled

0Eh.1 **UARTIE:** UART TX/RX Full Interrupt Enabled
0: Disabled
1: Enabled

0Eh.0 **PXIE:** Pin change wake-up interrupt enable
0: Disabled
1: Enabled

1Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF1	LVDIF	-	-	TKM0IF	TKIF	-	UARTIF	PXIF
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

1Ah.7 **LVDIF:** Low voltage detection interrupt suspend flag
This bit is set by H/W. Writing 0 to this bit with S/W will clear this flag.

1Ah.4 **TKM0IF:** Touch key module 0 interrupt suspend flag
This is set by H/W after the TK0 conversion is complete.
S/W Writing 0 to this bit, writing 0 to TKIF, and writing 1 to TKM0SOC will all clear this flag.

1Ah.3 **TKIF:** Touch key interrupt hang flag
This is set by H/W after ATK or TKM0 conversion is complete.
S/W Writing 0 to this bit or writing 1 to TKM0SOC will clear all touch key interrupt flags.

1Ah.1 **UARTIF:** UART interrupt suspend flag
When TX/RX transmission is complete, this bit is set by H/W. Writing 0 to this bit clears the flag.

1Ah.0 **PXIF:** Pin Change Interrupt Suspend Flag
This bit is set by H/W, and the corresponding pin change will clear the flag by writing 0 to this bit.

113h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CFLG	-	-	TXD1F	TXD0F	RCD1OVF	RCD1F	RCD0OVF	RCD0F
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	0	0	0	0

113h.5 **TXD1F:** I2C data transmission register 1 flag
When I2CTXD1 data transmission is complete, this bit is set by H/W. Writing 0 to this bit will clear the flag.

113h.4 **TXD0F:** I2C data transmission register 0 flag
When I2CTXD0 data transmission is complete, this bit is set by H/W. Writing 0 to this bit clears the flag.

113h.3 **RCD1OVF:** Receive data register 1 overflow from I2C
This bit is set by H/W when receiving I2CRCD1 overflow. Writing 0 to this bit will clear this flag.

113h.2 **RCD1F:** Receive data register 1 flag from I2C
When data reception to I2CRCD1 is complete, this bit is set by H/W. Writing 0 to this bit clears the flag.

113h.1 **RCD0OVF:** Receive data register 0 overflow from I2C
This bit is set by H/W when receiving I2CRCD0 overflow. Writing 0 to this bit will clear this flag.

113h.0 **RCD0F:** Receive data register 0 flag from I2C
When data reception to I2CRCD0 is complete, this bit is set by H/W. Writing 0 to this bit clears the flag.

5. I/O Port

5.1 PA0-4, PA7, PB0-7, PD0-7

The chip has multiple pin modes, which are defined by registers PAMODx, PBMODx, and PDMODx. Pin data is defined by registers PAD, PBD, and PDD. Its functions are shown in the table below.

Pin Mode	Pin Data	Description	Digital Output Switch	Digital Input Switch
Mode 0	0	Low output	ON	OFF
	1	Input with pull-up resistor	OFF	ON
Mode 1	0	Low output	ON	OFF
	1	High impedance input	OFF	ON
Mode 2	0	CMOS low output	ON	OFF
	1	CMOS high output	ON	OFF
Mode 3	X	High impedance, for analog signals:	OFF	OFF

GPIO Function Table

Mode 1 disables digital output and enables digital input, while Mode 3 disables both digital output and digital input. Both Mode 1 and Mode 3 can be used for analog signals, but Mode 1, which enables digital input, may consume more power when used for analog signals. It is recommended to use pin mode 3 for analog signals such as ADC/VBGO.

All general-purpose I/O (GPIO) pins default to pin mode 1 with pin data set to 1, i.e., high-impedance state, to prevent voltage conflicts between the pins and external circuits after power-on. For low-power applications, all digital pins (including unused or unrouted pins) should avoid being set to high-impedance state.

Reading pin data (PxD) has different meanings under different instructions. In “Read-Modify-Write” instructions, the CPU actually reads the output data register. In other instructions, the CPU reads the pin state.

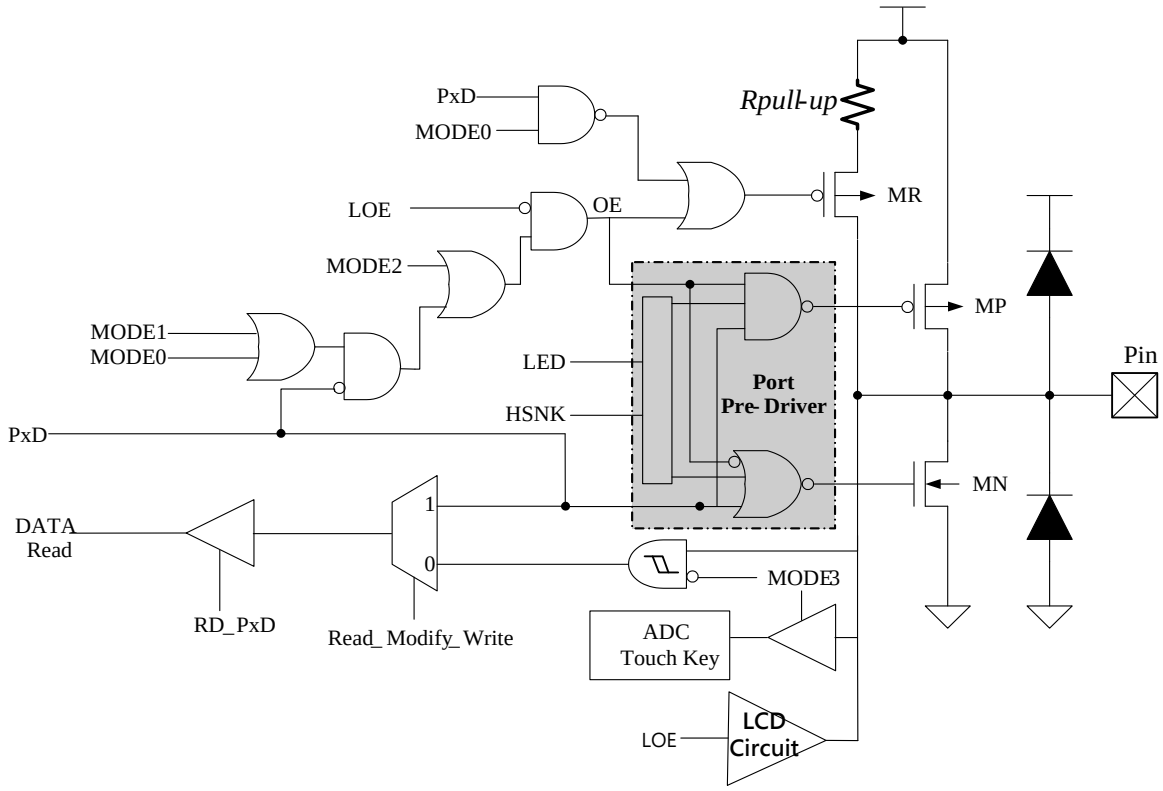
In addition to I/O port functions, each pin has one or more alternative functions, such as ADC and touch keys.

Pin Name	ADC	Touch Key	PWM	Other
PA0	AD25	TK0_11	PWM0N	
PA1	AD24	TK0_10	PWM1	
PA2	AD26	TK0_12	PWM2	TM0CKI
PA3	AD23	TK0_9	PWM3	
PA4	AD22	TK0_8	PWM4	
PA7	-	-	PWM0P/ PWM5	nRESET
PB0	AD0	TK0_0/ATK	PWM0P	TM1OUT
PB1	AD1	TK0_1	PWM5	TCOUT
PB2	AD2	TK0_2		
PB3	AD3	TK0_3		
PB4	AD4	TK0_4		RX
PB5	AD5	TK0_5		TX
PB6	AD6	TK0_6		
PB7	AD7	TK0_7		
PD0	AD13			XI
PD1	AD14			XO
PD2	AD15			RX
PD3	AD16			TX
PD4	AD17			
PD5	AD18			
PD6	AD19			
PD7	AD20			

GPIO Optional features menu

Other Functions	Pin Mode	Pin Data	Description	Other Necessary Register Settings
TM0CKI	0	1	Input with pull-up resistor	TM0CTL
	1	1	Input high impedance state	
RX	0	1	Input with pull-up resistor	UARTCTL
	1	1	Input high impedance state	
TX	2	X	CMOS output	UARTCTL
TK0_0~TK0_12	2	0	The touch key keeps the CMOS output low to resist interference, and the hardware automatically switches to high resistance mode during scanning.	TKM0CHS
AD0~AD7 AD13~20 AD22~26	3	X	High impedance state, used for analog signals.	ADCHS
TCOUT	2	X	CMOS output	OPTION2
TM1OUT	2	X	CMOS output	OPTION2
PWM0N PWM0P PWM1/2/3/4/5	2	X	CMOS output	PWMOE SFR19B
XI, XO	0	1	Input with pull-up resistor	CLKCTL

GPIO Register settings for optional functions



GPIO Pin structure (except PA7)

05h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAD	PAD7			PAD4	PAD3	PAD2	PAD1	PAD0
R/W	R/W			R/W	R/W	R/W	R/W	R/W
Reset	1			1	1	1	1	1

05h.7~0 **PAD:** PA7~PA0 Data

06h	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

06h.7~0 **PBD:** PB7~PB0 Data

08h	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	1	1	1	1

08h.7~0 **PDD:** PD7~PD0 Data

85h	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	1					0	1

85h.7~6 **PA7MOD:** PA7 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

85h.1~0 **PA4MOD:** PA4 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

86h	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

86h.7~6 **PA3MOD:** PA3 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

86h.5~4 **PA2MOD:** PA2 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

86h.3~2 **PA1MOD:** PA1 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

86h.1~0 **PA0MOD:** PA0 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

87h	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

87h.7~6 **PB7MOD:** PB7 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

87h.5~4 **PB6MOD:** PB6 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

87h.3~2 **PB5MOD:** PB5 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2

11: Mode3

87h.1~0 **PB4MOD:** PB4 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

88h	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

88h.7~6 **PB3MOD:** PB3 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

88h.5~4 **PB2MOD:** PB2 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

88h.3~2 **PB1MOD:** PB1 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

88h.1~0 **PB0MOD:** PB0 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

8Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDMODH	PD7MOD		PD6MOD		PD5MOD		PD4MOD	
R/W	R/W		R/W		R/W		R/W	
Reset	0	1	0	1	0	1	0	1

8Dh.7~6 **PD7MOD:** PD7 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

8Dh.5~4 **PD6MOD:** PD6 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

8Dh.3~2 **PD5MOD:** PD5 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

8Dh.1~0 **PD4MOD:** PD4 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

8Eh	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

8Eh.7~6 **PD3MOD:** PD3 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

8Eh.5~4 **PD2MOD:** PD2 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

8Eh.3~2 **PD1MOD:** PD1 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

8Eh.1~0 **PD0MOD:** PD0 Pin Mode Control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

5.2 Pin Level Change Interrupt Wake-up

Except for PD4 and PD5, all I/O pins on the chip support pin level change interrupt wake-up functionality (including rising edge and falling edge triggering).

When a pin level change occurs, if the corresponding PxWKE (x=A, B, D) register is set to 1 and PXIE is set to 1, this feature will pull PXIF high, wake the device from IDLE/STOP mode, and trigger the interrupt routine.

When a pin level change occurs, if the corresponding pin's PxWKE (x=A,B,D) register is set to 1 and PXIE is 0, this feature will pull up PXIF and wake the device from IDLE/STOP mode, but will not trigger the interrupt routine.

◇ Example: Set port B to wake up when the pin level changes.

```
ORG 00h                         ; reset vector
    LGOTO       START           ;

ORG 004h                        ; interrupt vector
    LGOTO       INT             ;
START:
    MOVLW       00000000B
    MOVWX       PBMODH          ; Set pin mode
    MOVWX       PBMODL

    MOVLW       11111111B
    MOVWX       PBWKE;          ; Set PB as pin level change wake-up
    BSX         PXIE            ; Pin level change interrupt enable
MAIN:
    ….
    SLEEP
    ….
    LGOTO       MAIN
INT:
    MOVLW       11111110B
    MOVWX       INTIF1          ; Clear PXIF
    …..
    RET
```

1Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PAWKE	PAWKE7			PAWKE4	PAWKE3	PAWKE2	PAWKE1	PAWKE0
R/W	R/W			R/W	R/W	R/W	R/W	R/W
Reset	0			0	0	0	0	0

1Ch.7~0 **PAWKE:** PA Pin wake-up enable

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

1Dh.7~0 **PBWKE:** PB Pin wake-up enable

1Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDWKE	PDWKE							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

1Fh.7~0 **PDWKE:** PD Pin wake-up enable (Not included PD5,PD4)

0Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE1	LVDIE	-			TKIE	-	UARTIE	PXIE
R/W	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

0Eh.0 **PXIE:** Pin wake-up interrupt enable
0: Disabled
1: Enabled

1Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF1	LVDIF	-	-	TKM0IF	TKIF	-	UARTIF	PXIF
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

1Ah.0 **PXIF:** Pin Interrupt Flag
This bit is set by H/W. Writing 0 to this bit will clear the flag.

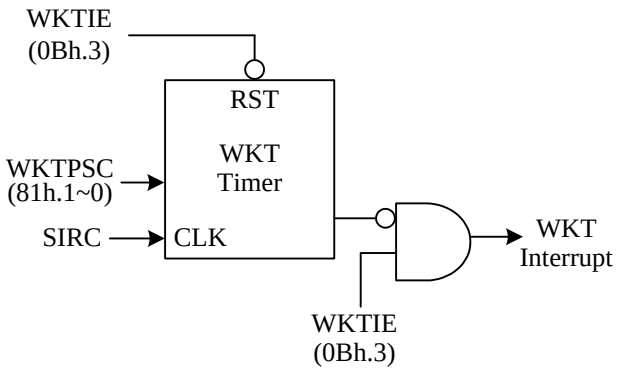
6. Peripheral Functional Block

6.1 Wakeup Timer (WKT)

The wake-up timer uses SIRC (Slow Internal RC) as its clock, and the WKT overflow period can be adjusted via the WKTPSC register. If the WKTIE register is set to 1, the system will enter the WKT interrupt routine when WKT overflows.

In addition to enabling the interrupt function, the WKTIE register also serves as a function switch for the wake-up timer.

Users cannot read or write the wake-up timer value.



WDT and WKT Block Diagram

◇ Example:

```
MOVLW   000101 10 B     ;
MOVWX   OPTION          ; Select WKT period
MOVLW   1111 0 111B     ; Use byte operations to clear the WKT interrupt flag.
MOVWX   INTIF           ; Do not use bit manipulation “BCX WKTIF” to clear the interrupt flag.
MOVLW   0000 1 000B     ; Enable WKT interrupt function
MOVWX   INTIE
```

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

0Ch.3 **WKTIF:** Wake-up timer interrupt event suspend flag
When the wake-up timer times out, this bit is set by H/W. Writing 0 to this bit clears the flag.

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

0Bh.3 **WKTIE:** Wake-up timer interrupt enable
0: Disable
1: Enable

81h/181h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	HWAUTO	INT0EDG	INT1EGE	FREQL	WDTPSC		WKTPSC	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	1	1	1	1	1

81h.4 **FREQL:** SIRC Frequency Selection
0: 60KHz
1: 50KHz

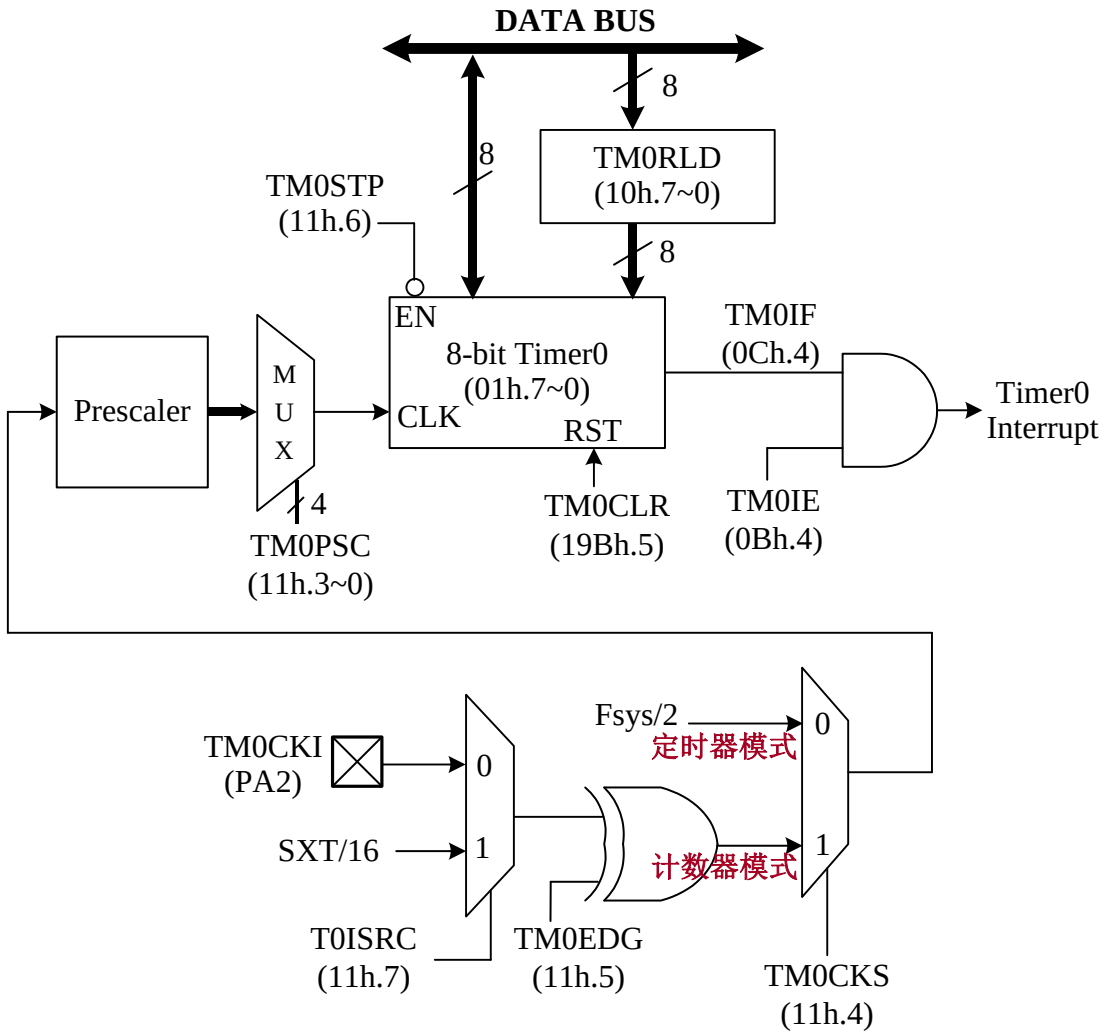
81h.1~0 **WKTPSC:** WKT cycle
FREQL=1:
00: 22ms@3V, 20ms@5V
01: 44ms@3V, 40ms@5V
10: 89ms@3V, 80ms@5V
11: 177ms@3V, 160ms@5V
FREQL=0:
00: 18ms@3V, 16ms@5V
01: 35ms@3V, 32ms@5V
10: 70ms@3V, 63ms@5V
11: 139ms@3V, 125ms@5V

6.2 Timer0

Timer0 is an 8-bit timer whose clock rate is selected by the TM0PSC register. If TM0IE is set to 1, it will generate a Timer0 interrupt routine when an overflow occurs. The Timer0 period can be adjusted via the TM0RELOAD register. The TM0STP register can be used to pause/resume Timer0 counting.

When the TM0CKS register is set to 1, Timer0 switches from timer mode to counter mode.

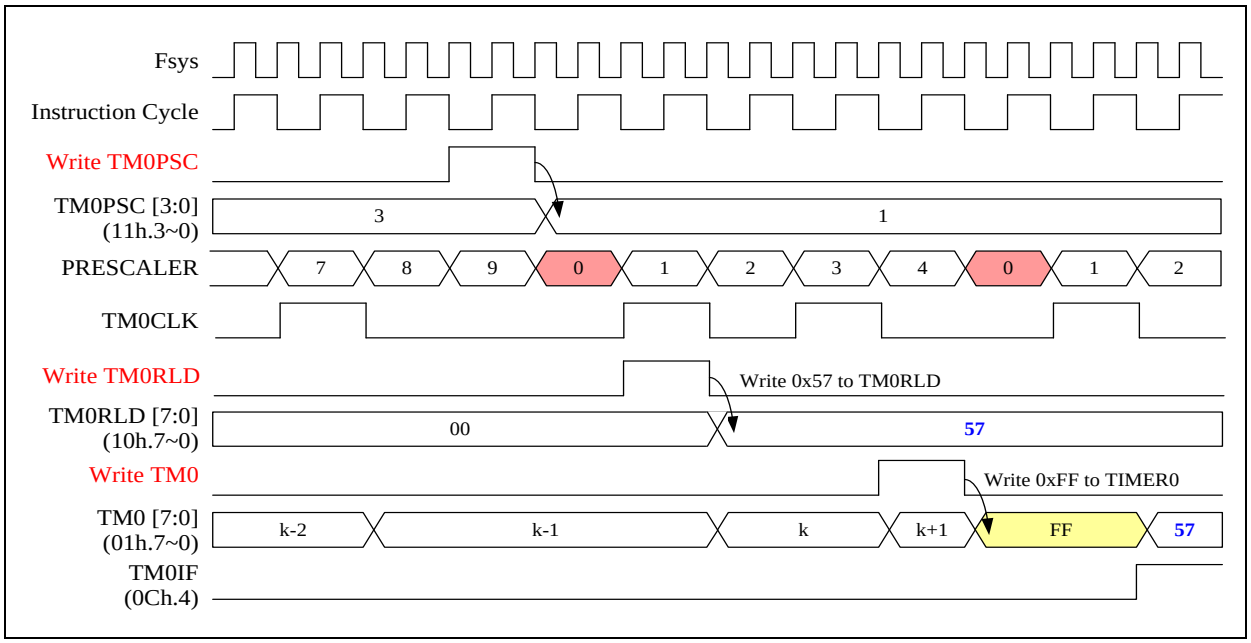
In counter mode, it is recommended to keep the TM0PSC register at 0. The TM0EDG register determines whether the counter mode uses rising edge or falling edge triggering, and the T0ISRC register determines the count source as either the PA2 pin input or SXT/16.



Timer0 Block Diagram

	定时器Mode Fsys/2	计数器Mode TM0CKI (PA2 pin)	计数器Mode SXT/16
TM0CKS	0	1	1
T0ISRC	x	0	1

Timer0 Clock Source and Mode Control



Timer0 in timer mode (TM0CKS=0)

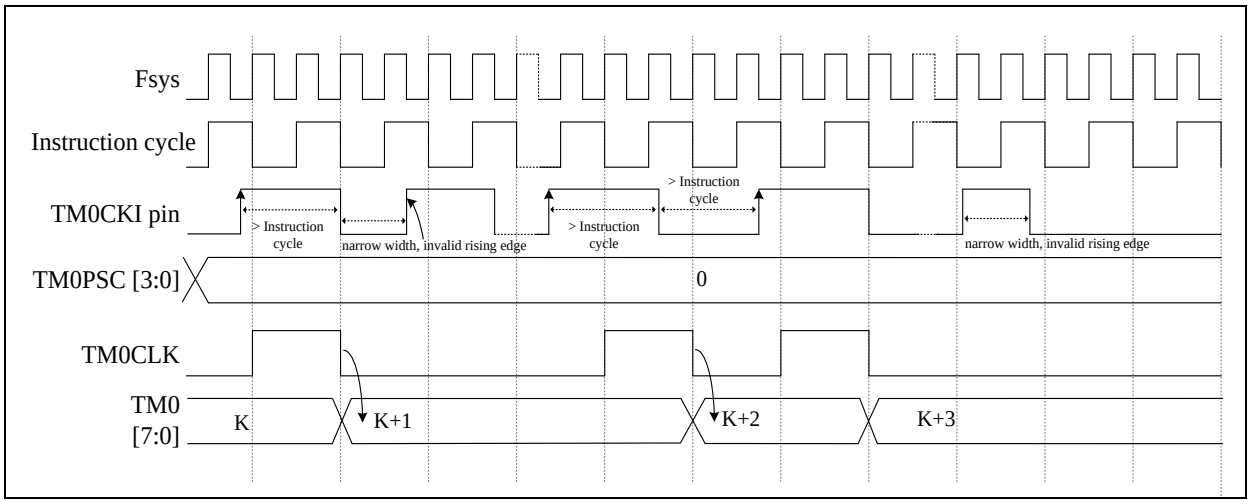
The formula for calculating the Timer0 interrupt time is as follows:

Timer0 interrupt interval period time = Fsys / 2 / TM0PSC / (256-TM0)

◇ Example: TM0 operates in timer mode.

```
; Set TM0 clock source and divider
        MOVLW       0000 0101B      ;
        MOVWX       TM0CTL          ; Timer0 clock is the clock source (Fsys/2) divided by 32.
; Pause TM0 counting
        BSX         TM0STP          ; Pause TM0 counting
        MOVLW       156             ;
        MOVWX       TM0             ;
        MOVLW       156             ;
        MOVWX       TM0RLD          ;

; Enable the TM0 timer and interrupt function.
        MOVLW       1110 1111B      ; Clear the Timer0 interrupt flag through byte operations.
        MOVWX       INTIF           ;
        MOVLW       0001 0000B      ; Enable Timer0 interrupt function
        MOVWX       INTIE           ;
        BCX         TM0STP          ; Start TM0 counting
```



Timer0 in counter mode (TM0CKS=1, T0ISRC=0, TM0EDG=0)

◇ Example: TM0 operates in counter mode.

```
MOVLW   00110000B   ; Pause TM0 counting and set to counter mode
MOVWX   TM0CTL      ; The counter count source is the PA2 pin, and it is set to trigger on the
falling edge.
MOVLW   00H
MOVWX   TM0         ;
BCX     TM0STP      ; Start TM0 counting
```

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

01h **TM0:** Timer0 data

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

0Bh.4 **TM0IE:** Timer0 interrupt enabled
0: Disabled
1: Enabled

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

0Ch.4 **TM0IF:** Timer0 interrupt event suspend flag
When Timer0 overflows, this bit is set by H/W. Writing 0 to this bit clears this flag.

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

10h **TM0RLD:** Timer0 reloads data

11h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0CTL	T0ISRC	TM0STP	TM0EDG	TM0CKS	TM0PSC			
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

11h.7 **T0ISRC:** Timer0 counting mode clock source selection
0: TM0CKI pin (PA2)
1: SXT/16

11h.6 **TM0STP:** Timer0 counter pause
0: Timer0 running
1: Timer0 paused

11h.5 **TM0EDG:** Timer0 prescaler count edge of TM0CKI pin
0: indicates rising edge
1: indicates falling edge

11h.4 **TM0CKS**: Timer0 clock source selection
0: Fsys/2
1: TM0CKI (PA2 pin) or SXT/16

11h.3~0 **TM0PSC**: Timer0 frequency division. *When TM0CKS=1, it is recommended that TM0PSC be kept at the default value of 0000.
Timer0 clock is the clock source (Fsys/2 or TM0CKI or SXT/16) divided by

0000: /1	0001: /2	0010: /4	0011: /8
0100: /16	0101: /32	0110: /64	0111: /128
1000: /256	1001: /512	1010: /1024	1011: /2048
1100: /4096	1101: /8192	1110: /16384	1111: /32768

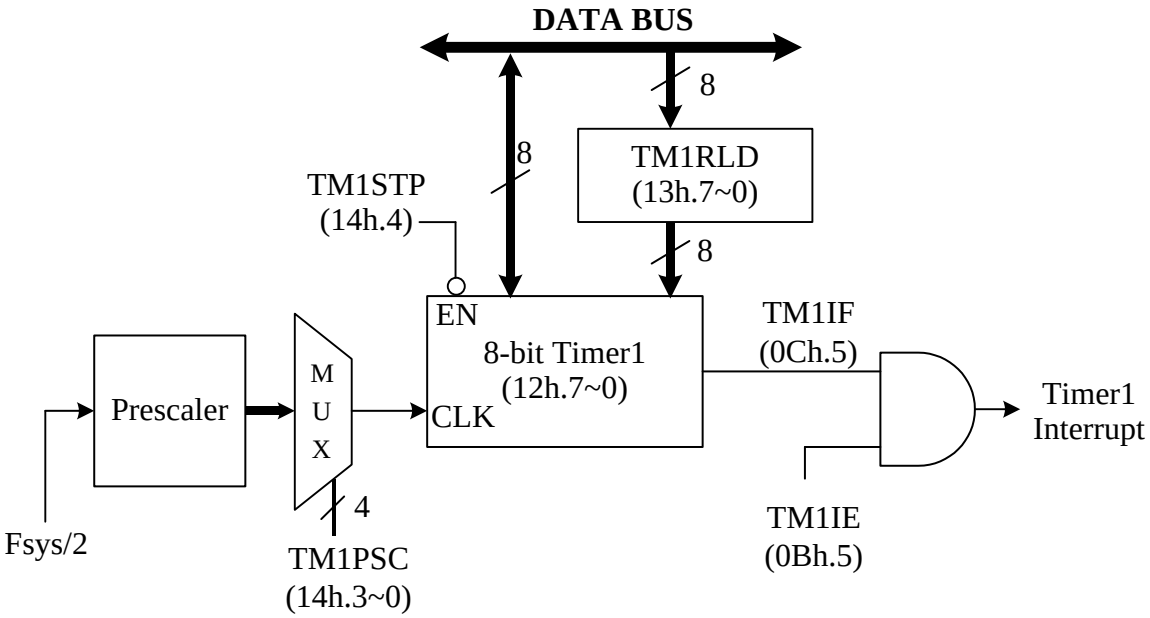
19Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SFR19B	CFEN	CFST	TM0CLR	ULEDMASK	TXSEL	RXSEL	PWM0PSEL	PWM5SEL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

19Bh.5 **TM0CLR:** Timer0 reset
Reset Timer0 to zero.

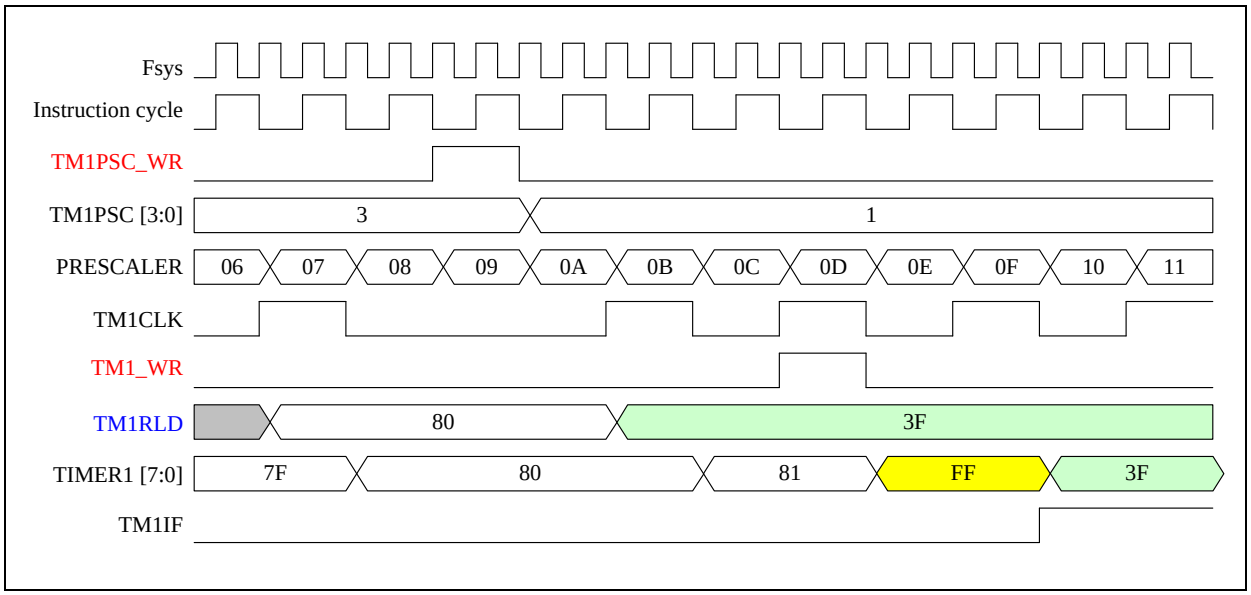
6.3 Timer1

Timer1 is an 8-bit timer whose clock rate is selected by the TM1PSC register. If TM1IE is set to 1, it will generate a Timer1 interrupt routine when an overflow occurs. The Timer1 period can be adjusted via the TM1RELOAD register. The TM1STP register can be used to pause/resume Timer1 counting.

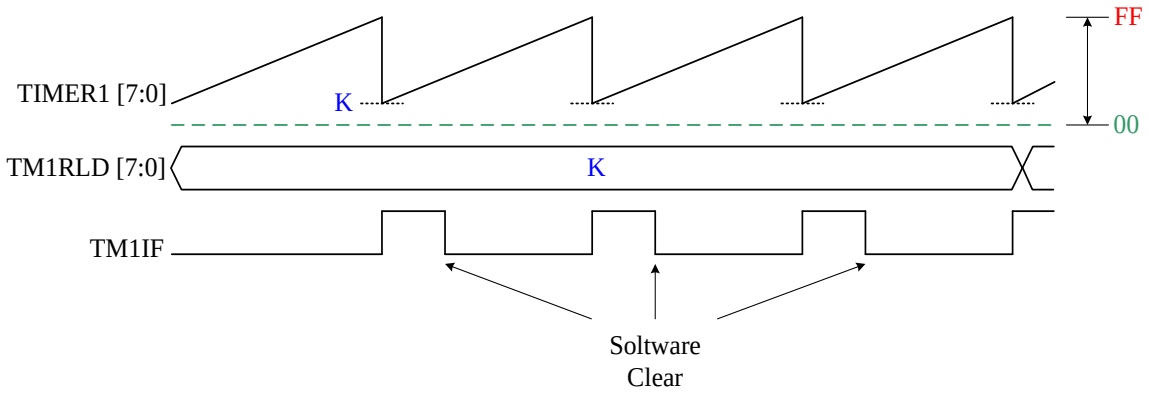
Timer1 can output the overflow flag signal to PB0 (TM1OUT).



Timer1 Block Diagram



Timer1 Sequence Diagram



Timer1 Reload diagram

◇ 例:

```
MOVLW     0001 0001B     ; Pause TM1 counting
MOVWX     TM1CTL         ; Set Timer1 clock to Fsys/4.
BSX       TM1OE          ; Enable Timer1 output pin (PB0)
MOVLW     F0H
MOVWX     TM1            ; Set Timer1 period
MOVLW     110 11111B     ; Clear TM1 interrupt flag by byte operation
MOVWX     INTIF          ;
MOVLW     001 00000B     ; Enable the TM1 interrupt function.
MOVWX     INTIE          ;
BCX       TM1STP         ; Start TM1 counting
```

Timer0 cycle calculation:
Given that Fsys = 4 MHz, TM1PSC = 1, and TM1RLD = 0xF0, then the TM1 clock source = Fsys/4 = 1 MHz.
The TM1 cycle is 1 MHz * (0xFF – 0xF0) = 1 us * 16 = 16 us, and the TM1OUT output cycle is 16 * 2 = 32 us.

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

0Bh.5 **TM1IE:** Timer1 interrupt enable
0: Disable
1: Enable

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

0Ch.5 **TM1IF:** Timer1 interrupt event suspend flag
Set by H/W when Timer1 overflows. Writing 0 to this bit clears the flag.

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

12h **TM1:** Timer1 count data

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

13h.7~0 **TM1RLD:** Timer1 reloads data

14h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1CTL	–	–	–	TM1STP	TM1PSC			
R/W	–	–	–	R/W	W	W	W	W
Reset	–	–	–	0	0	0	0	0

14h.4 **TM1STP:** Pause Timer1
0: Timer1 running
1: Timer1 paused

14h.3~0 **TM1PSC:** Timer1 frequency division. The Timer1 clock is the clock source (Fsys/2) divided by
0000: Fsys/1 0001: Fsys/2 0010: Fsys/4 0011: Fsys/8
0100: Fsys/16 0101: Fsys/32 0110: Fsys/64 0111: Fsys/128
1xxx: Fsys/256

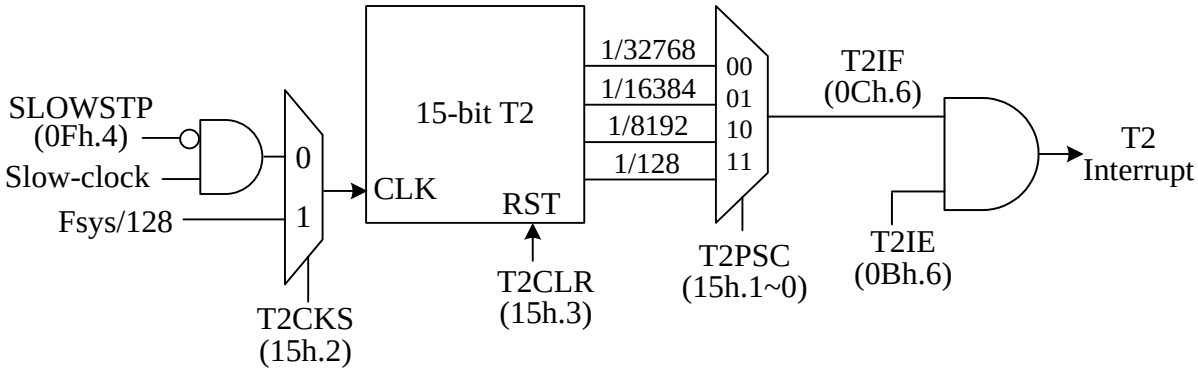
08Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION2	TCOE	TM1OE	PWMCKS					-
R/W	R/W	R/W	R/W	R/W				-
Reset	0	0	0	0				-

08F.6 **TM1OE:** Enable Timer1 overflow to switch output to PB0 pin (TM1OUT)

6.4 T2

T2 is a 15-bit timer. Users cannot read or write the value of the T2 timer. The T2CKS register determines whether the T2 clock source comes from Fsys/128 or the slow clock, and the clock rate used is selected by the T2PSC register. If T2IE is set to 1, it will generate the T2 interrupt routine when an overflow occurs. Setting the T2CLR register can reset T2.

To keep T2 running after executing the SLEEP instruction, the T2CKS register must be set to 0 and the SLOWSTP register must be set to 0.



T2 Block Diagram

◇ Example:

```
MOVLW     0000<u>1111</u>B    ; Stop T2
MOVWX     T2CTL          ; The T2 clock is (Fsys/128) divided by 16384.

MOVLW     1<u>0</u>111111B    ; Clear the T2 interrupt flag through byte operations.
MOVWX     INTIF          ;
MOVLW     0<u>1</u>000000B    ; Enable T2 interrupt function.
MOVWX     INTIE          ;
BCX       T2CLR          ; Start T2
```

T2 calculation:

If Fsys is 4.608MHz, then T2 clock = (4.608MHz/128)/16384 = 2.197Hz

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

0Bh.6 **T2IE:** T2 interrupt enable
0: Disabled
1: Enabled

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

0Ch.6 **T2IF:** T2 interrupt event flag
When T2 overflows, this bit is set by H/W. Writing 0 to this bit clears the flag.

15h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CTL					T2CLR	T2CKS	T2PSC	
R/W					R/W	R/W	R/W	R/W
Reset					1	0	0	0

15h.3 **T2CLR:** Reset the T2 counter.
Reset Timer0 to zero.
15h.2 **T2CKS:** T2 clock source selection.
0: Slow clock; 1: Fsys/128
15h.1~0 **T2PSC:** T2 frequency division. T2 clock is the clock source (slow clock or Fsys/128) divided by
00: 32768 01: 16384 10: 8192 11: 128

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

0Fh.4 **SLOWSTP**: Disable slow clock
0: Slow clock continues to run after SLEEP instruction
1: Slow clock stops running after SLEEP instruction if no other peripheral function blocks use slow clock

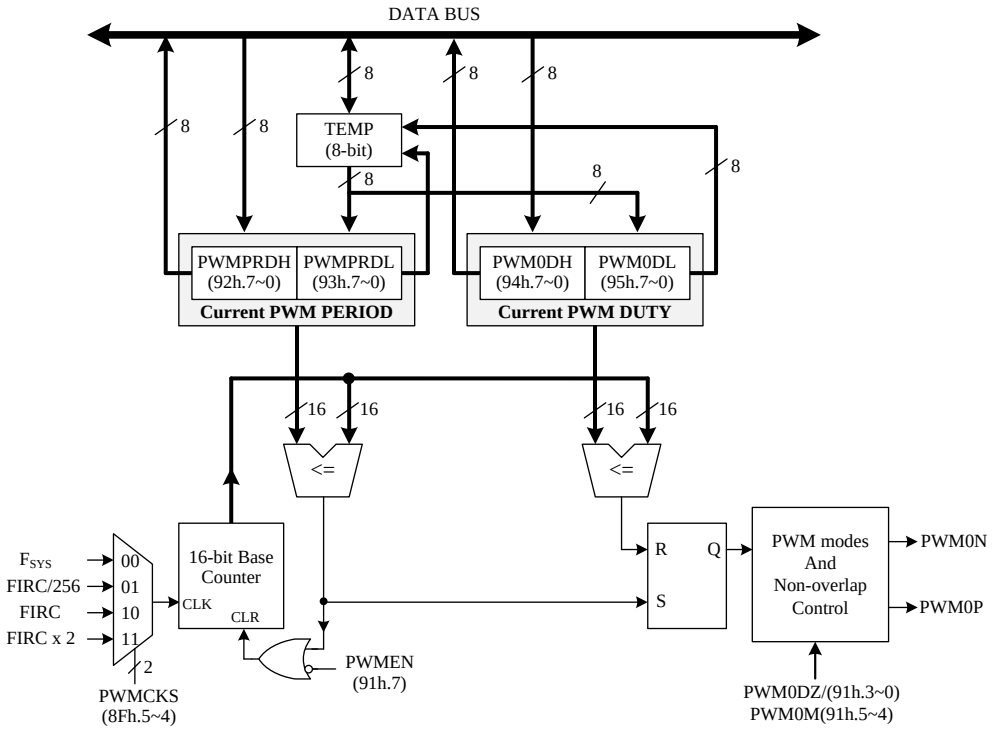
6.5 PWM

The chip features six 16-bit PWM modules, designated PWM0 through PWM5. Each PWM module (PWM0 to PWM5) has its own independent 16-bit duty cycle register and shares the same set of 16-bit period registers. PWM can generate various frequency waveforms based on PWMCLK, with a duty cycle resolution of 65,536. The frequency can be selected as Fsys, FIRC, FIRC*2, or FIRC/256, determined by PWMCKS.

The 16-bit PWM period PWMPRD and PWM duty cycle PWM0D to PWM5D registers all have a low-byte and high-byte structure. Writing to these register pairs must be done in a specific order. **That is, the low-byte register must be written first, followed by the high-byte register.** For example, the period value is written to the PWMPRDL register (93h), followed by the PWMPRDH register (92h). When the high-byte data is written to the register, the PWMPRD register is immediately updated to the new value.

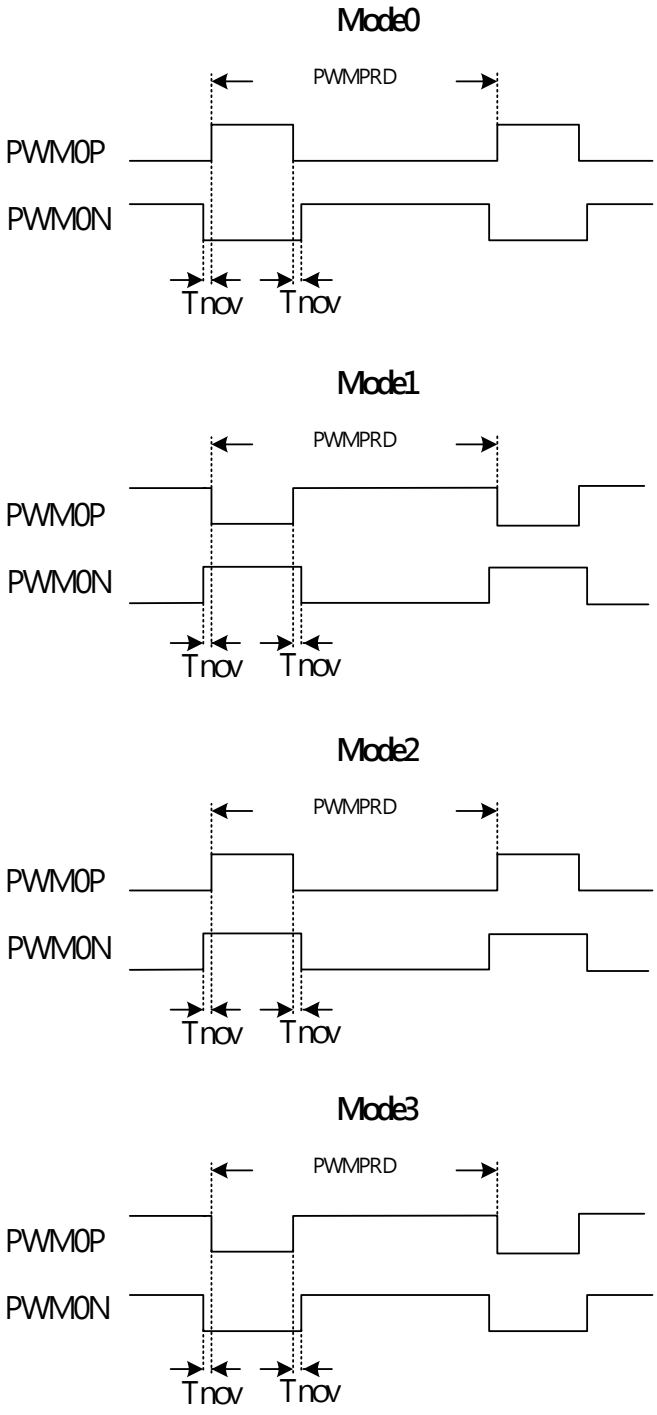
If PWMEN is cleared, PWM0–5 will be cleared and stopped; otherwise, PWM0–5 will continue to run. The PWM0 structure is shown below. The PWM0 duty cycle can be changed by writing to PWM0DL and PWM0DH. The PWM0 period can be set by writing the period value to the PWMPRDL and PWMPRDH registers. There is a digital comparator that compares the PWM counter with PWMPRD. If the PWM counter is greater than PWMPRD, the PWM counter is cleared, and the PWM output signal is set to a high level. When the 16-bit base counter matches the PWM duty cycle registers PWM0D to PWM5D, the PWM output signals PWM0 to PWM5 are reset to a low level.

Only PWM0 has dead-time control (PWM0DZ), which is divided into PWM0P and PWM0N outputs. The remaining PWM1–5 do not have dead-time control. The PWM1–5 outputs are PWM1O–PWM5O.



PWM Block Diagram (using PWM0 as an example)

As shown in the figure below, PWM0 has four output modes (PWM0OM). The edges of the PWM pulses can be separated by 16 different dead time intervals, i.e., non-overlapping clock intervals (Tnov). The width of Tnov can be selected within the range of 0 to 15 PWM clock cycles. The default output mode is Mode 0.



PWM0 Dead Zone and Output Mode

◇ Example: [CPU running in fast mode, Fsys=FIRC 18.432Mhz]

```
MOVLW     00000000B     ;
MOVWX     OPTION2       ; PWMCKS=00B, PWM clock source = Fsys = FIRC 18.432 MHz
                        ; Tpwmclk=(1/18.432M)
MOVLW     20h
MOVWX     PWMPRDL       ; Set PWM cycle low byte = 20h
                        ; The low byte data is stored in TEMP instead of the PWMPRD register.
```

```
MOVLW     03h
MOVWX     PWMPRDH     ; Set PWM cycle high byte = 03h
                      ; PWMPRD = 0320h (TEMP data is also saved to PWMPRD)
MOVLW     90h
MOVWX     PWM0DL      ; Set PWM0 cycle low byte = 90h
                      ; The low byte data is stored in TEMP, not in the PWM0D register.
MOVLW     01h         ;
MOVWX     PWM0DH      ; Set PWM0 cycle high byte = 01h
                      ; PWM0D = 0190h (TEMP data is also saved to PWM0D)
MOVLW     10010100B   ; Set PWMEN=1, PWM0OM=10B=Mode 2,
MOVWX     PWMCTL      ; PMW0DZ=0100B= 4*Tpwmclk
MOVLW     10xxxx10B   ;
MOVWX     PAMODL      ; Set PA7 and PA0 as CMOS outputs.
MOVLW     11000000B
MOVWX     PWMOE       ; Enable PWM0P output to PA7, enable PWM0N output to PA0
```

08Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION2	TCOE	TM1OE	PWMCKS					-
R/W	R/W	R/W	R/W	R/W				-
Reset	0	0	0	0				-

08Fh.7 **TCOE:** TCOUT allows output.
0: Not enabled.
1: Enabled, output to PB1.

08Fh.6 **TM1OE:** Timer1 overflow switch enables output.
0: Disabled.
1: Enabled, output to PB0.

08Fh.5~4 **PWMCKS:** PWM clock source
00: Fsys 01:FIRC/256 10:FIRC 11:FIRC*2

90h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMOE	PWM0POE	PWM0NOE	PWM5OE	PWM4OE	PWM3OE	PWM2OE	PWM1OE	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-
Reset	0	0	0	0	0	0	0	-

90h.7 **PWM0POE:** PWM0P allows output
0: Disabled
1: Enabled, PWM0P outputs to PA7 or PB0

90h.6 **PWM0NOE:** PWM0N allows output
0: Disabled
1: Enabled, PWM0N outputs to PA0

90h.5 **PWM5OE:** PWM5 allows output
0: Disabled
1: Enabled, PWM5 outputs to PA7 or PB1; For PA7, the output priority of PWM0P is higher than that of PWM5

90h.4 **PWM4OE:** PWM4 allows output
0: Disabled
1: Enabled, PWM4 outputs to PA4

90h.3 **PWM3OE:** PWM3 allows output
0: Disabled
1: Enabled, PWM3 output to PA3

90h.2 **PWM2OE:** PWM2 allows output
0: Disabled
1: Enabled, PWM2 output to PA2

90h.1 **PWM1OE:** PWM1 allows output
0: Disabled
1: Enabled, PWM1 output to PA1

91h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMCTL	PWMEN	-	PWM0OM		PWM0DZ			
R/W	R/W	-	R/W		R/W			
Reset	0	-	0	0	0	0	0	0

91h.7 **PWMEN:** PWM clock enable
0: Disabled
1: Enabled

91h.5~4 **PWM0OM:** PWM0 output mode
00: PWM0 Mode 0
01: PWM0 Mode 1
10: PWM0 Mode 2
11: PWM0 Mode 3

91h.3~0 **PWM0DZ:** PWM0 Dead Zone (Non-Overlapping) Control
0000: No non-overlapping
0001: Non-overlapping width is 1 PWM clock cycle
0010: Non-overlapping width is 2 PWM clock cycles
……
1111: Non-overlapping width is 16 PWM clock cycles

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

92h.7~0 **PWMPRDH:** PWM cycle data high byte

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

93h.7~0 **PWMPRDL:** PWM cycle data low byte

94h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DH	PWM0DH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

94h.7~0 **PWM0DH:** PWM0 duty cycle data high byte

95h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DL	PWM0DL							
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

95h.7~0 **PWM0DL:** PWM0 duty cycle data low byte

96h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1DH	PWM1DH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

96h.7~0 **PWM1DH:** PWM1 duty cycle data high byte

97h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1DL	PWM1DL							
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

97h.7~0 **PWM1DL:** PWM1 duty cycle data low byte

98h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2DH	PWM2DH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

98h.7~0 **PWM2DH:** PWM2 duty cycle data high byte

99h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2DL	PWM2DL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

99h.7~0 **PWM2DL:** PWM2 duty cycle data low byte

9Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM3DH	PWM3DH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

9Ah.7~0 **PWM3DH:** PWM3 duty cycle data high byte

9Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM3DL	PWM3DL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

9Bh.7~0 **PWM3DL:** PWM3 duty cycle data low byte

9Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM4DH	PWM4DH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

9Ch.7~0 **PWM4DH:** PWM4 duty cycle data high byte

9Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM4DL	PWM4DL							
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

9Dh.7~0 **PWM4DL:** PWM4 duty cycle data low byte

9Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM5DH	PWM5DH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

9Eh.7~0 **PWM5DH:** PWM5 duty cycle data high byte

9Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM5DL	PWM5DL							
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

9Fh.7~0 **PWM5DL:** PWM5 duty cycle data low byte

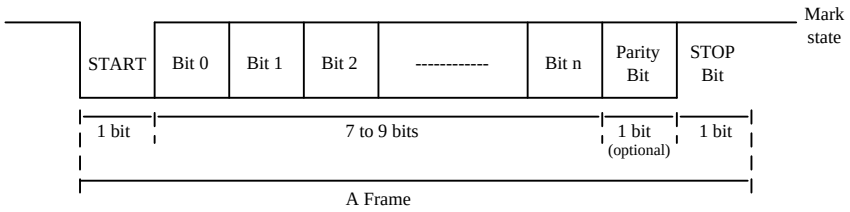
19Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SFR19B	CFEN	CFST	TM0CLR	ULEDMASK	TXSEL	RXSEL	PWM0PSEL	PWM5SEL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

19Bh.1 **PWM0PSEL:** PWM0P pin selection.
0: PA7 as PWM0P output
1: PB0 as PWM0P output

19Bh.0 **PWM5SEL:** PWM5 pin selection.
0: PA7 as PWM5 output
1: PB1 as PWM5 output

6.6 UART

The chip's built-in standard UART (Universal Asynchronous Receiver/Transmitter) is responsible for serial communication between devices. The transmitting device converts parallel data into serial data and sends the bit stream data via the TX line. The receiving device receives the data in serial form via the RX line, converts it back into parallel data, and stores it in a register for the microcontroller to read. The TX/RX module also handles data synchronization, parity bit generation and detection, frame errors, overflow errors, and interrupt generation. The UART data format is shown in Figure 6.6.1.



UART Data Format

The transmission line typically remains in the 1 mark state (logic 1) until transmission or reception begins and transitions to the 0 space state (logic 0). The first bit transmitted is the start bit, which is 1 bit wide (1 bps). This is followed by the data bit stream. The least significant bit (LSB) is sent first. The number of bits transmitted or received is defined by the UMODE control bit, which can be set to 7-bit, 8-bit, or 9-bit mode. The parity bit follows the data bits and can be optionally sent, but in 9-bit mode transmission, the parity bit is not sent.

UART Mode

UART can operate in three modes: Mode 0 (7-bit data), Mode 1 (8-bit data), and Mode 2 (9-bit data). The data format for Mode 0 and Mode 1 can be transmitted and received with parity bits based on the PRE bit. If PRE = 1, the TX data format will append a parity bit after the data bits; otherwise, TX will only transmit the data bits. Mode 2 does not support the transmission or reception of parity bits. It is important to note that the UART settings between two communicating devices must be identical so that the RX section can verify the data format and parity in the same way as the TX section. The parity selection bit is the even parity bit in the UARTS control register. If even = 1, even parity is used; otherwise, odd parity is used. Each UART transmission ends with a STOP bit. If the STOP bit is not received by the RX, the RX will set the FMERR bit in the UARTS control register to indicate that the transmission may not have been correctly received/transmitted, and it must be resent or handled by the firmware. Figure 6.6.2 shows the data formats for the three UART modes.

	UMODE[1:0]	PRE	S 1 2 3 4 5 6 7 8 9 10
UART Mode 0	0 0	0	7-bit data STOP
	0 0	1	7-bit data Pty STOP
UART Mode 1	0 1	0	8-bit data STOP
	0 1	1	8-bit data Pty STOP
UART Mode 2	1 1	x	9-bit data STOP

UART mode

UART system module

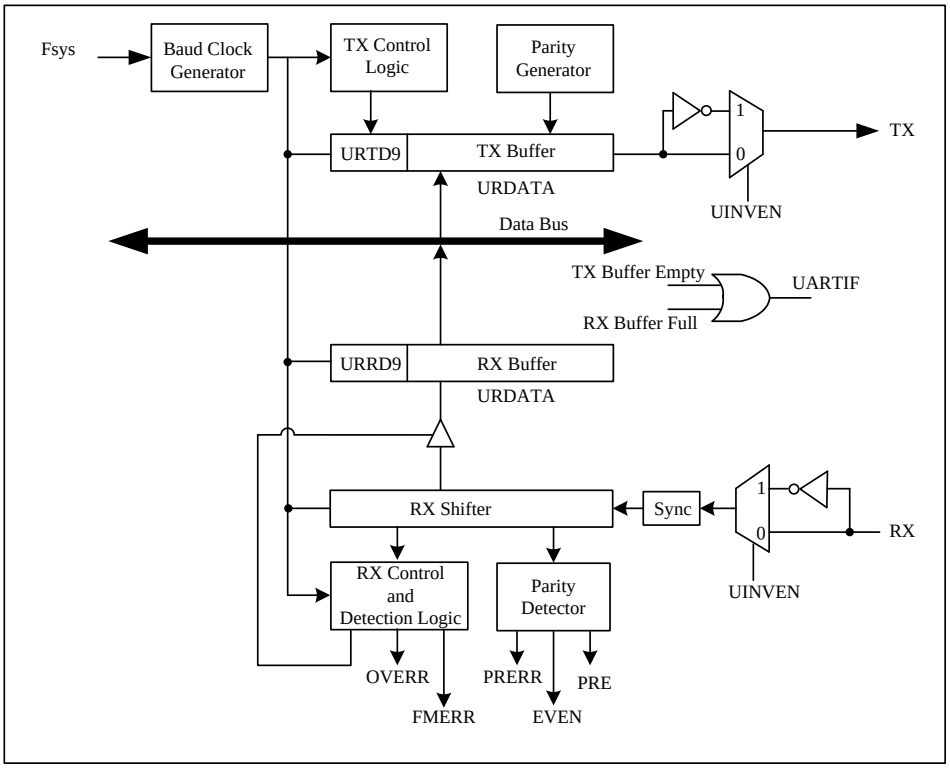
The figure below shows the system module of the UART built into the chip. The baud clock generator divides Fsys to generate the appropriate frequency for the UART baud rate. The divisor is the UBAUD register. Each data stream is sampled using 16 baud clocks, which are generated by the baud clock generator. The desired baud rate can be obtained by setting the UBAUD register.

$$\text{Baud Rate} = \frac{\text{Fsys}}{\text{32*UBAUD}}$$

The baud rate clock is sent to the TX and RX control logic and the TX/RX shift buffer. Once URDATA is written, the TX control logic begins sending data to the TX pin (PB5). If the UART mode is mode 2, URTD9 is sent. When the transmission is complete, the “TX buffer Empty” signal is raised. If UARTIE is set, a UART interrupt is generated.

When UINVENis set to 1, UINVEN will invert the output. When UINVEN is set to 1, the RX input will be inverted.

The UART RX signal is input through the RX pin (PB4). If UINVEN is set, the input is inverted via a synchronization circuit and then timed to the RX shift register. Depending on the selected UART mode, the number of bits to be shifted into the RX shift register is determined, and then processed by the RX control logic, including frame error detection (FMERR), parity error detection (PRERR), and overflow error detection (OVERR). A frame error indicates that the stop bit of the incoming frame was not detected. A parity error indicates that the parity bit of the incoming data does not match the parity calculated by the RX control logic for the data bits. An overflow error occurs when the previous transmission stored in URDATA has not been read before the new incoming transmission is completed. When an overflow error occurs, the old URDATA is lost.



UART block diagram

Fsys (Hz)	Expected baud rate (bps)	Register UBAUD	Actual baud rate (bps)	Frequency deviation (%)	
18432000	2400	240	2400	0	
18432000	4800	120	4800	0	
18432000	9600	60	9600	0	
18432000	14400	40	14400	0	
18432000	19200	30	19200	0	
18432000	28800	20	28800	0	
18432000	38400	15	38400	0	
18432000	57600	10	57600	0	
18432000	76800	8	72000	6.25%	Don't use
18432000	76800	7	82285.7	7.14%	Don't use
18432000	115200	5	115200	0	
18432000	192000	3	192000	0	

UART Commonly used baud rates

◇ Example:

[CPU running at FAST mode , Fsys=FIRC 18.432 MHz]

UART Baud Rate = 115200 bps, RX pin (PB4), TX pin (PB5).
Transmit 91H data to TX (PB5)

```
BCX     FASTSTP     ;
BSX     CPUCKS      ; Fsys=18.432 MHz

MOVLW   1001 0101B  ; RX (PB4) Pin Mode=1, TX (PB5) Pin Mode=2
MOVWX   PAMODL;

MOVLW   00000101B
```

```
MOVWX   UBAUD       ; 115200 bps (@Fsys=18.432MHz)

BSX     UARTIE      ; Enable UART interrupt
MOVLW   00101010B   ;
MOVWX   UARTCTL     ; 8bit mode, UTXE=1, URXE=0, UARTE=1, UINVEN=0
MOVLW   10010001B   ;
MOVWX   URDATA      ; TX Buffer=91h, then H/W sends it to the TX pin via the TX shifter.

BTXSS   UTBE
LGOTO   $-1         ; Check the TX buffer until the buffer is empty (transmission complete).
……
…...
```

0Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE1	LVDIE	-	-	-	TKIE	-	UARTIE	PXIE
R/W	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

0Eh.1 **UARTIE:** UART TX/RX completion interrupt enable
0: Disabled
1: Enabled

1Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF1	LVDIF	-	-	TKM0IF	TKIF	-	UARTIF	PXIF
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

1Ah.1 **UARTIF:** UART Interrupt Wait Flag
When TX/RX transmission is complete, this bit is set to 1 by H/W; writing 0 to this bit clears the flag.

10Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UBAUD	UBAUD							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

10Dh **UBAUD:** UART baud rate divider

187h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UARTCTL	URTD9	UMODE		URINTTF	UTXE	URXE	UARTE	UINVEN
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

187h.7 **URTD9:** The 9th bit of UART transmission
187h.6~5 **UMODE:** UART Mode Selection
00: 7 digits
01: 8 digits
10: 9 digits
11: Reserved
187h.4 **URINTTF:** UART TX transmission complete flag
When UART TX transmission is complete, this bit is set to 1 by H/W; clearing UARTIF also clears this flag.
187h.3 **UTXE:** UART transmission enabled
0: Disabled
1: Enabled
187h.2 **URXE:** UART receive enable
0: Disabled

1: Enabled

187h.1 **UARTE**: UART function enabled
0: Disabled
1: Enabled

187h.0 **UINVEN**: UART TX/RX output/input reversal enable
0: Disabled
1: Enabled

188h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
UARTSTA	UTBE	URRD9	EVEN	PRE	PREERR	OVERR	FMERR	URBF
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R
Reset	0	0	0	0	0	0	0	0

188h.7 **UTBE:** UART transmission status
0: Transmitting TX
1: TX buffer is empty

188h.6 **URRD9**: UART received the 9th bit

188h.5 **EVEN:** UART parity
0: Even number
1: Odd number

188h.4 **PRE:** UART Parity Check Addition
0: Disabled
1: Enabled

188h.3 **PRERR**: UART parity error
Set to 1 when a parity error occurs, cleared to 0 by F/W

188h.2 **OVERR**: UART overflow error
Set to 1 when an overflow error occurs, cleared to 0 by F/W

188h.1 **FMERR:** UART frame error
Set to 1 when a frame error occurs, cleared to 0 by F/W

188h.0 **URBF:** UART buffer status
Set to 1 when 1 byte (or 9 bits) is received.
F/W reading URDATA will automatically clear to 0, then receive subsequent data without overflow errors.

18Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
URDATA	URDATA							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

18Dh **URDATA:** Data transmission/reception buffer
Write: Transmission buffer
Read: Reception buffer

19Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SFR19B	CFEN	CFST	TM0CLR	ULEDMASK	TXSEL	RXSEL	PWM0PSEL	PWM5SEL
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

19Bh.3 **TXSEL:** TX pin selection.
0: PB5 as TX output
1: PD3 as TX output

19Bh.2 **RXSEL:** RX pin selection.
0: PB4 as RX output
1: PD2 as RX output

6.7 Analog-to-Digital Converter (ADC)

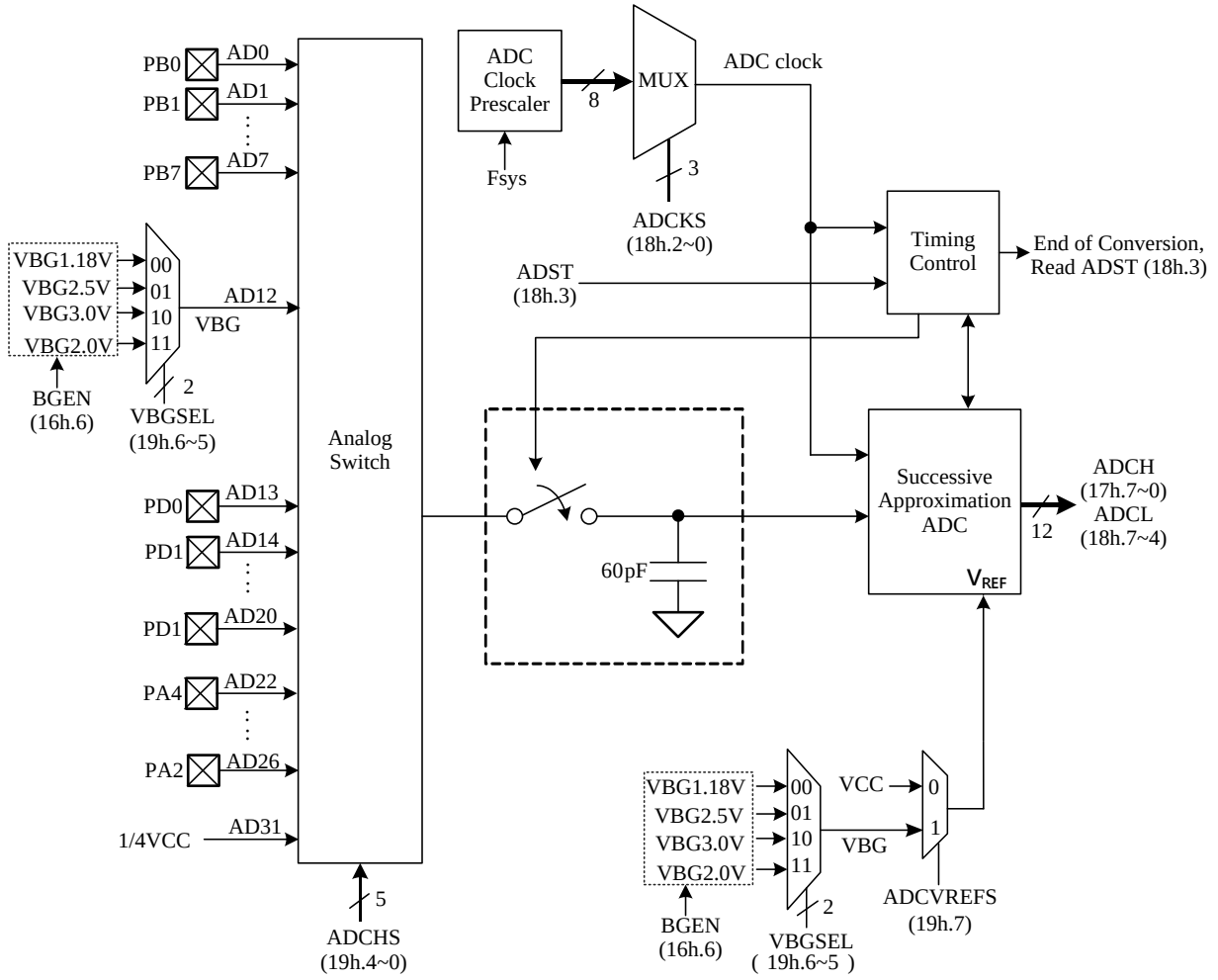
The 12-bit ADC consists of a 21-channel analog input multiplexer, control registers, successive approximation registers, and output data registers. Users need to set ADCKS (18h.2~0) to select the appropriate ADC clock frequency and enable ADC conversion by setting the ADST (18h.3) control bit. After conversion is complete, the ADST bit is automatically cleared by the hardware. Users can poll this bit to determine the conversion status.

When a pin is used as an ADC input, the corresponding PxMODx register must be set to “Pin Mode 3.” This setting disables the pin's logic input path to conserve power.

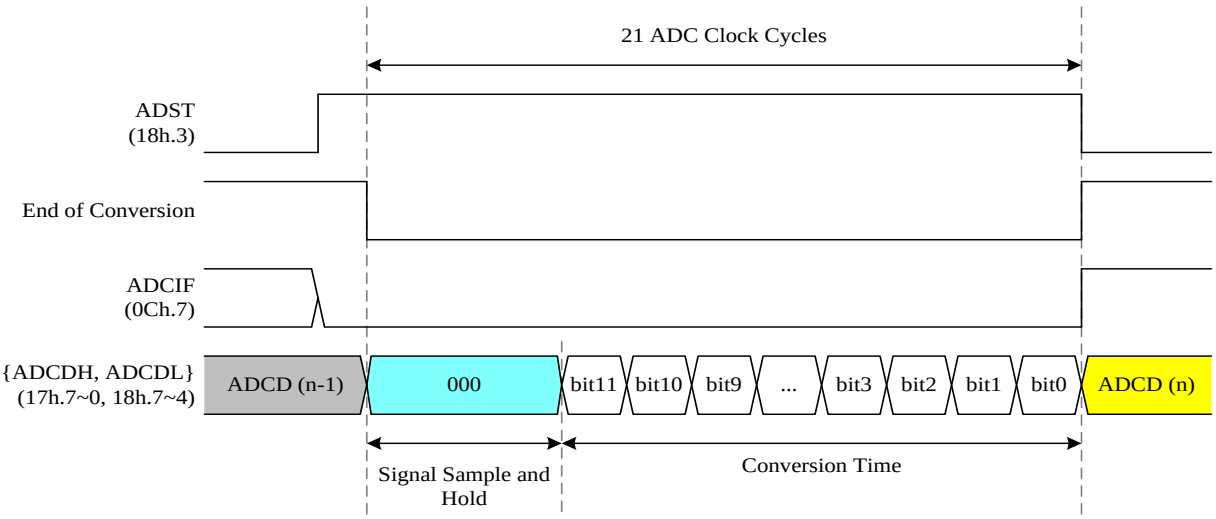
The ADC module's reference voltage is selected via the ADCVREFS register and VBGSEL register.

Users must configure the ADCHS register to select the ADC input channel. In addition to external pins, AD12 uses the internal bandgap voltage source VBG, and AD31 uses the internal voltage source 1/4VCC.

The internal bandgap voltage source VBG is enabled by the BGEN register. When the ADC uses the internal bandgap voltage source VBG, the BGEN register must be enabled. To save power, please disable the BGEN register when no longer in use.



ADCArchitecture Diagram



◇ Example:

[CPU running at FAST mode , Fsys=FIRC 18.432 MHz]

ADC clock frequency =FIRC/32, ADC channel=AD15 (PD2).

```
        MOVLW   00000111B       ; Fsys=18.432 MHz
        MOVWX   CLKCTL          ;

        MOVLW   01110101B       ; PD2(AD15) Set to pin mode 3
        MOVWX   PDMODL          ;

        MOVLW   00000011B       ; Fsys=18.432 MHz
        MOVWX   ADCTL           ; 18h.2~0 (ADCKS) =ADC clock =Fsys/32=576KHz

        MOVLW   01110011B       ; 16h.6=1, enable VBG;
        MOVWX   LVCTL           ;

        MOVLW   1 00 0 0111B    ; ADC Reference voltage source=VBG1.18V
        MOVWX   ADCTL2          ; Set to AD15 (PD2 pin).

       BSX      ADST            ; 18h.3 (ADST) , ADC Start conversion signal

WAIT_ADC:
        BTXSC   ADST            ; Waiting for ADC conversion to complete.
        LGOTO   WAIT_ADC

        MOVXW   ADCDH           ; 17h.7~0, read ADC result [11:4] to W
        …
        MOVXW   ADCTL           ; 18h.7~4, read ADC results [3:0] to W
        …
```

16h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LVCTL	LVDF	BGEN	LVRSAV	LVDSAV	LVDS			
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	1	1	1	0	0	1	0

16h.6 **BGEN:** Bandgap voltage VBG enable
When the ADC uses the internal bandgap voltage source VBG, the BG EN register must be enabled. To save power consumption, disable the BGEN register when use is complete.
0: Bandgap voltage disabled
1: Bandgap voltage enabled in FAST/SLOW mode, automatically disabl ed in STOP/IDLE mode

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

17h.7~0 **ADCDH:** ADC output value (ADCQ [11:4])

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

18h.7~4 **ADCDL:** ADC output value (ADCQ [3:0])
18h.3 **ADST:** ADC Start Flag
When the user sets ADST=1, the ADC starts conversion. After conversion is complete, the H/W automatically clears this flag.
18h.2~0 **ADCKS:** ADC clock frequency selection:
000: Fsys/256 100: Fsys/16
001: Fsys/128 101: Fsys/8
010: Fsys/64 110: Fsys/4
011: Fsys/32 111: Fsys/2

19h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCTL2	ADCVREFS	VBGSEL		ADCHS				
R/W	R/W	R/W		R/W				
Reset	0	0			0	0	0	0

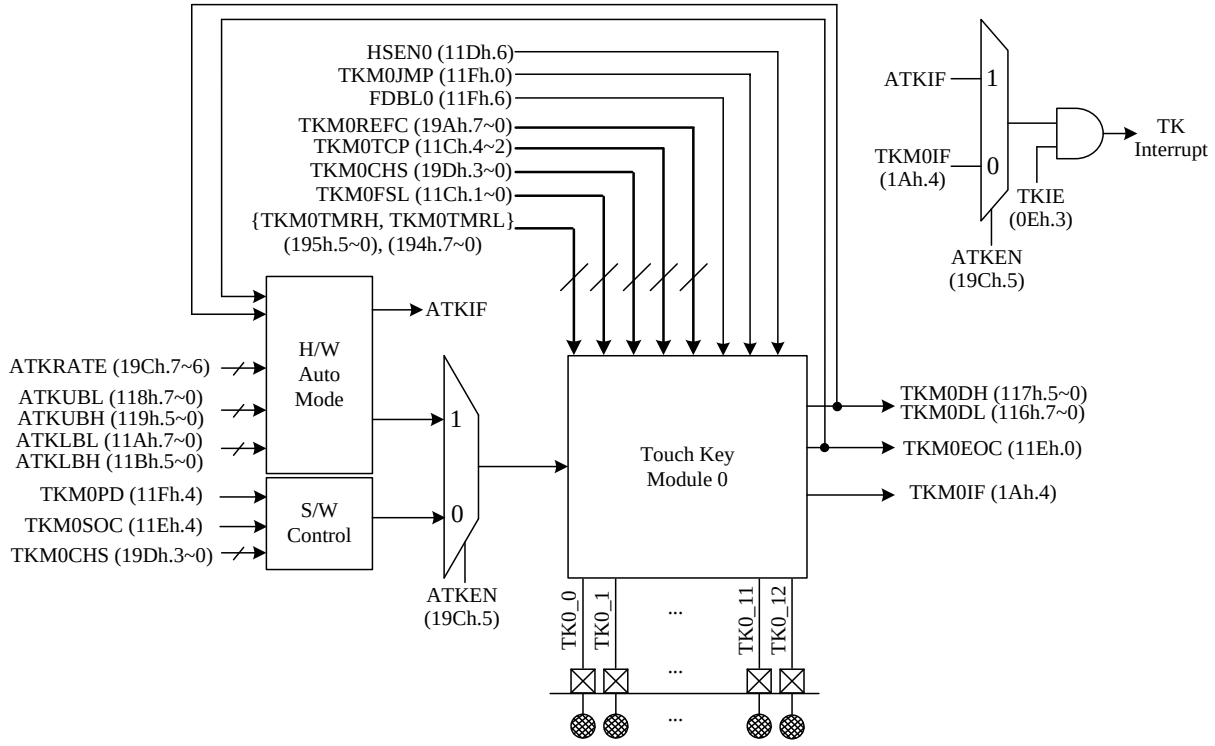
19h.7 **ADCVREFS:** ADC reference voltage selection
0: VCC
1: VBG
19h.6~5 **VBGSEL:** VBG voltage selection
00: 1.18V
01: 2.5V
10: 3.0V
11: 2.0V
19h.4~0 **ADCHS:** ADC channel selection

00000: AD0 (PB0)	01000: Retain	10000: AD16 (PD3)	11000: AD24 (PA1)
00001: AD1 (PB1)	01001: Retain	10001: AD17 (PD4)	11001: AD25 (PA0)
00010: AD2 (PB2)	01010: Retain	10010: AD18 (PD5)	11010: AD26 (PA2)

00011: AD3 (PB3)	01011: Retain	10011: AD19 (PD6)	11011: Retain
00100: AD4 (PB4)	01100: AD12 (VBGO)	10100: AD20 (PD7)	11100: Retain
00101: AD5 (PB5)	01101: AD13 (PD0)	10101: Retain	11101: Retain
00110: AD6 (PB6)	01110: AD14 (PD1)	10110: AD22 (PA4)	11110: Retain
00111: AD7 (PB7)	01111: AD15 (PD2)	10111: AD23 (PA3)	11111: 1/4 VCC

6.8 Touch buttons

The touch key provides a simple, straightforward, and reliable method for detecting finger touches. In most applications, it does not require any external components. The device supports 13-channel touch key detection in S/W manual mode and 1 channel in H/W automatic mode (ATK), located at CH0 (PB0) in touch key mode 0.



Touch module 0 with ATK structure

To use the touch keys, users must press the table to set the pin mode correctly. Setting idle touch keys to pin mode 2 and output 0 can reduce mutual interference between adjacent keys.

Touch Key Pin Mode Settings	TK0_0~TK0_12
Pin is a touch key, idle	Mode 0, 1, 2, 3
Pin is a touch key, S/W scan	Mode 2, Output 0
Pin is a touch key, H/W auto scan	TK0_0 Mode 2, Output 0

(ATK)	

6.8.1 Touch button software manual detection mode

Touch key module 0 channel 0 (TK0_0, PB0) can also be used as H/W automatic mode (ATK). There are two oscillators in the touch key module: reference clock (RCKx) and touch clock (TCKx). They are connected to the reference counter and data counter, respectively. The frequency of RCKx can be adjusted by setting TKMxREFC. The reference counter is used to control the conversion time. TKMxTCP is the touch key clock frequency selection (available only when TKMxJMP = 0). When TKMxJMP = 1, the touch key clock frequency changes automatically.

From the start of the touch key conversion to the end, setting TKMxTMR requires 0 to 4096 RCK oscillation cycles. After the conversion is complete, the user can retrieve the TK data (TKMxDH, TKMxDL) from the data counter. The TK data is influenced by finger touch. When the finger touch slows down the TCK, the value of the TK Data is less than when there is no finger touch. Depending on the TKMxDATA, the user can check whether it has been touched. Appropriate TKMxTMR and TKMxREFC settings can adjust the TK data to suit the system board environment. To obtain the best TKMxREFC settings, users can try different TKMxREFC values and find one that makes the TK data and TKMxTMR as close as possible. On the other hand, users can adjust the overall operating frequency of the TK system (including TCKx and RCKx) by setting TKMxFSL (frequency selection).

Setting the control signal TKM0HSEN (11D.6) can improve the performance of touch key module 0.

When scanning begins, users assign TKMxPD=0, then set the TKMxSOC bit to initiate touch key conversion. Upon conversion completion, the TKMxSOC bit is automatically cleared. However, if SYSCLK is too slow, the H/W may fail to clear TKMxSOC due to clock sampling rate issues. TKMxEOC=0 indicates that conversion is in progress. TKMxEOC=1 indicates that the conversion is complete, and the touch key count result is stored in the 14-bit TK data counters TKMxDH and TKMxDL.

TKMxIF is activated when TKMxPD changes from 1 to 0, so after the user clears TKMxPD, TKMxIF must also be cleared.

TKM0IF	TKIF	Status
0	0	Idle
1	1	TK module 0 conversion complete
0	1	ATK conversion complete

Touch Key Interrupt Flag Description

◇ Example: Touch key channel = TK0_2 (PB2)

```
.ORG 000h
        LGOTO           START
```

```
.ORG 004h
INT:
            BTXSC       TKIF        ; Verify TKIF
            LCALL       INT_TK
            RETI
INT_TK:
            BTXSC       TKM0IF      ; Verify TKM0IF
            LCALL       INT_TKM0
            LCALL       INT_ATK     ; TKIF=1 and TKM0IF=0, indicating ATKIF
            RET
INT_TKM0:
            MOVLW       11101111B   ; Clear TKM0IF
            MOVWX       TKMFLG
            MOVXW       TKM0DH      ; read TK0 DATA[13:8] to W register
            …
            MOVXW       TKM0DL      ; read TK0 DATA[7:0] to W register
            …
            RET
INT_ATK:
            MOVLW       11110111B   ; Clear TKIF
            MOVWX       TKMFLG
            …
            RET
START:
            …
SET_MODE:
            MOVLW       xx10xxxxB   ; PBMODL[5:4] = 10b
            MOVWX       PBMODL      ; Set PB2 as Mode 2 for touch key input.
            BCX         PBD,2       ; Clear PB2 for CMOS low output
TK_INIT:
            MOVLW       000 001 10B ;
            MOVWX       TKMCON0     ; Set TKM0TCP=1, TKM0FSL=2

            MOVLW       01 xxxxxxB  ;
            MOVWX       TKMCON1     ; Set TKM0HSEN=1, TKM0 high sensitivity
            MOVLW       00h
            MOVWX       TKM0TMRH
            MOVLW       64H
            MOVWX       TKM0TMRL    ; TKM0TMR=64H


            MOVLW       4DH
            MOVWX       TKM0REFC    ; TKM0REFC=64H

            MOVLW       0000 0010B  ; Set TKM0CHS=2 (TK0_2, PB2)
            MOVWX       TKMCHS      ;

            MOVLW       00000000B
            MOVWX       TKMCTL1     ; TKM0PD=0, enable TKM0
            MOVLW       11110111B
            MOVWX       INTIF1      ; Clear TKIF
            BSX         TKIE        ; Enable TKIE, enable TK interrupt

TK_START:
            BSX         TKM0SOC     ; Start TKM0 conversion
            …
            …
```

6.8.2 Hardware automatic touch button detection mode

The touch keys have an H/W automatic mode. This feature can operate in fast/slow/idle modes, thereby reducing S/W workload and lowering chip power consumption. To use this feature, users need to set ATKEN=1 and TKM0PD=0 to enable H/W control of the TKM0 module. If ATKEN is set to "1," the ATK interval timer will generate an overflow flag after timing out, triggering the touch key H/W automatic mode to start. This allows the H/W to fully control the touch key module. Then, the H/W automatically detects the TK data count of TK0_0 at a rate of 20/25/40/80 ms according to ATKRATE (F19C.7–6).

As shown below. If the TK data count of the key is less than the specified lower comparison limit {ATKLBH, ATKLBL}, the H/W will generate an interrupt flag TKIF (1Ah.3). If TKIE (0Eh.3) is set to 1, it will generate an automatic touch key interrupt and wake up the CPU. After the TK interrupt, the user can switch the TK module to S/W manual mode and identify/confirm the key touch event.

◇ Example: Using ATK hardware automatic touch key detection module, touch key channel = TK0_0 (PB0)

```
.ORG  000h
              LGOTO          START
.ORG  004h
INT:
              BTXSC          TKIF           ; Verify TKIF
              LCALL          INT_TK
              RETI
INT_TK:
              BTXSC          TKM0IF         ; Verify TKM0IF
              LCALL          INT_TKM0
              LCALL          INT_ATK        ; TKIF=1, TKM0IF=0 indicates ATKIF
              RET
INT_TKM0:
              MOVLW          11101111B      ; Clear TKM0IF
              MOVWX          TKMFLG
              …
              RET
INT_ATK:
              MOVLW          11110111B      ; ClearTKIF
              MOVWX          TKMFLG
              …
              RET

START:
                …..
SET_MODE:
              MOVLW          xxxxxx10B      ; PBMODL[1:0] = 10b
              MOVWX          PBMODL         ; Set PB0 as Mode 2 for Touch Key input
              BCX            PBD,2          ; Clear PB2 to make CMOS output low level

TK_INIT:
              MOVLW          xxx 001 10B    ;
              MOVWX          TKMCON0        ; TKM0TCP=1, TKM0FSL=2
              MOVLW          x1xxxxxxB      ;
```

```
        MOVWX   TKMCON1     ; Set sensitivity

        MOVLW   04h
        MOVWX   TKM0TMRH
        MOVLW   64H
        MOVWX   TKM0TMRL    ; TKM0TMR=464H
        MOVLW   00h

        MOVLW   4DH
        MOVWX   TKM0REFC    ; TKM0REFC=64H
        MOVLW   00100000B   ; TKM0PD=0,EnableTKM0 activity
        MOVWX   TKMCTL1     ; TKM0JMP=0, TKM0
                            ; Clock mode fixed (TKM0TCP)
        MOVLW   11110111B
        MOVWX   INTIF1      ; ClearTKIF
        BSX     TKIE        ; EnableTKIE,  EnableTKInterrupt
        MOVLW   20H     ;
        MOVWX   ATKLBL      ; Set ATKLBL= 20H
        MOVLW   04H     ;
        MOVWX   ATKLBH      ; Set ATKLBH=04H; ATKLB = 0420H
        MOVLW   01100000B   ; ATKRATE=01B ATK scan Rate=25ms
ATK_START:
        MOVWX   ATKCON      ; ATKEN=1, Enable ATK hardware automatic touch
    function
        SLEEP               ; Sleep command, wait for ATK Interrupt wake-up
        NOP
```

0Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIE1	LVDIE	EEPIE			TKIE	I2CIE	UARTIE	PXIE
R/W	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

0Eh.3 **TKIE:** Touch Key Interrupt Enable
0: Close
1: Enable

1Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTIF1	LVDIF	-	-	TKM0IF	TKIF	-	UARTIF	PXIF
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

1Ah.4 **TKM0IF:** Touch Key Module 0 Interrupt Suspend Flag
This is set by H/W after the TK0 conversion is complete. S/W Writing 0 to this bit, writing 0 to TKIF, or writing 1 to TKM0SOC will clear this flag.

1Ah.3 **TKIF:** Touch KeyInterrupt Suspend Flag
This is set by H/W after ATK or TK0 conversion is complete.Writing 0 to this bit or writing 1 to TKM0SOC will clear all Touch Key Interrupt flags.

116h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKM0DL	TKM0DL							
R/W	R	R	R	R	R	R	R	R
Reset	-	-	-	-	-	-	-	-

116h.7~0 **TKM0DL:**Touch Key Module 0 data [7:0]

117h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

TKM0DH	-	-	TKM0DH					
R/W	-	-	R	R	R	R	R	R
Reset	-	-	-	-	-	-	-	-

117h.5~0 **TKM0DH:**Touch Key Module 0 data [13:8]

118h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATKUBL	ATKUBL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	-	-	-	-	-	-	-

118h.7~0 **ATKUBL:** ATKTouch Keylimit [7:0]

119h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATKUBH	-	-	ATKUBH					
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	-	-	-	-	-	-

119h.5~0 **ATKUBH:** ATKTouch Keylimit [13:8]

11Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATKLBL	ATKLBL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	-	-	-	-	-	-	-

11Ah.7~0 **ATKLBL:** ATKTouch Keylower limit [7:0]

11Bh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATKLBH	-	-	ATKLBH					
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	-	-	-	-	-	-

11Bh.5~0 **ATKLBH:** ATKTouch Keylower limit [13:8]

11Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKMCON0				TKM0TCP			TKM0FSL	
R/W				R/W	R/W	R/W	R/W	R/W
Reset				0	0	1	0	0

11Ch.4~2 **TKM0TCP:** Touch Key Module 0 Clock Frequency Selection (only effective when TKM0JMP=0)

11Ch.1~0 **TKM0FSL:** Touch key module 0 RCK0/TCK0 Frequency selection;

00:slowest

…

11:fastest

11Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKMCON1		TKM0HSEN	-	-	-	-	-	-
R/W		R/W	-	-	-	-	-	-
Reset		0	-	-	-	-	-	-

11Dh.6 **TKM0HSEN:** Touch Key Module 0 Sensitivity

0: Normal

1: Higher sensitivity

11Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKMCTL0	-	-	-	TKM0SOC	-	-	-	TKM0EOC
R/W	-	-	-	R/W	-	-	-	R/W
Reset	-	-	-	0	-	-	-	-

11Eh.4 **TKM0SOC:** Start Touch Key Module 0 conversion.
Set 1 to start Touch Key Module 0 conversion. If SYSCLK is fast enough, this bit will be H/WClear at the end of the conversion. S/W can also write 0 to clear this flag.

11Eh.0 **TKM0EOC:** Touch conversion flag Module 0 end key
TKM0EOC may have a 3uS delay after TKM0SOC=1, so F/W must wait long enough before polling the flag.
0: indicates conversion in progress
1: conversion complete

11Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKMCTL1		FDBL0	-	TKM0PD				TKM0JMP
R/W		R/W		R/W				R/W
Reset		0		1				0

11Fh.6 **FDBL0:** Touch Key Module 0 Counting speed doubled.
0: No doubling
1: Doubling

11Fh.4 **TKM0PD:** Touch key module 0 power off
0: Touch key module 0 running
1: Touch key module 0 off

11Fh.0 **TKM0JMP:** Touch Key Module 0 Clock Mode
0: Fixed Frequency (refer to TKM0TCP)
1: Auto Switching

194h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKM0TMRL	TKM0TMRL							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

194h.7~0 **TKM0TMRL** Touch Key Module 0 Reference Counter [7~0]

195h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKM0TMRH	-	-	TKM0TMRH					
R/W	-	-	R/W					
Reset	-	-	0	0	0	0	0	0

195h.3~0 **TKM0TMRH**: Touch Key Module 0 Reference Counter [13~8]

19Ah	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKM0REFC	TKM0REFC							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

19Ah.7~0 **TKM0REFC** Touch Key Module 0 Reference clock capacitor selection

19Ch	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ATKCON	ATKRATE		ATKEN					
R/W	R/W	R/W		-				
Reset	0	0		-				

19Ch.7~6 **ATKRATE**: Touch Key automatic scanning interval time
00: 20ms

01: 25ms
10: 40ms
11: 80ms

19Ch.5 **ATKEN**: Touch Key Automatic Scan Mode Enable

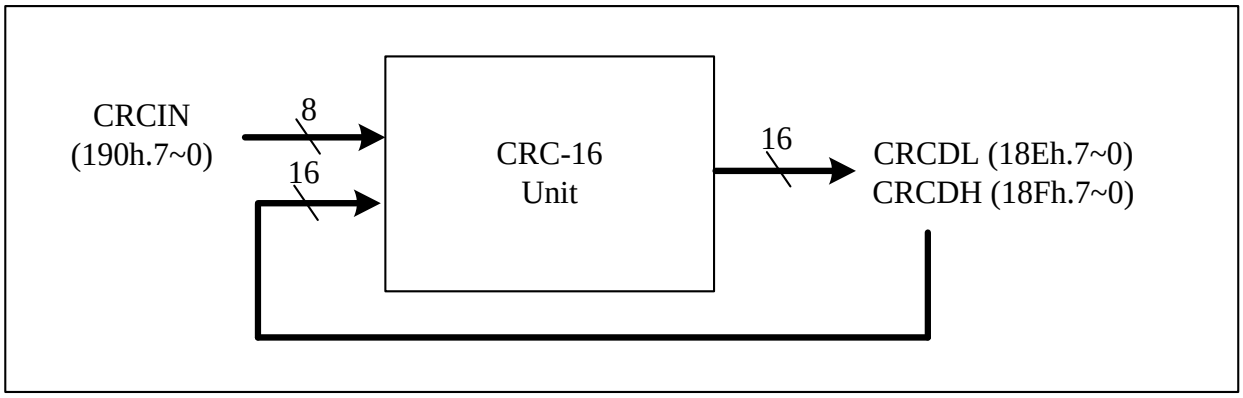
19Dh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TKMCHS					TKM0CHS			
R/W					R/W			
Reset					0	0	0	0

19Dh.3~0 **TKM0CHS**: Touch Key Module 0 Channel selection

0000: TK0_0 (PB0)
0001: TK0_1 (PB1)
0010: TK0_2 (PB2)
0011: TK0_3 (PB3)
0100: TK0_4 (PB4)
0101: TK0_5 (PB5)
0110: TK0_6 (PB6)
0111: TK0_7 (PB7)
1000: TK0_8 (PA4)
1001: TK0_9 (PA3)
1010: TK0_10 (PA1)
1011: TK0_11 (PA0)
1100: TK0_12 (PA2)

6.9 Cyclic Redundancy Check (CRC)

The chip supports an integrated 16-bit cyclic redundancy check (CRC) function. The CRC calculation unit is an error detection technology test algorithm used to verify the correctness of data transmission or storage. CRC calculation takes an 8-bit data stream or data block as input and generates a 16-bit output remainder. The data stream is calculated using the same generating polynomial.



CRC16 Block Diagram

The CRC generator provides 16-bit CRC result calculations based on the CRC-16-IBM polynomial. In this CRC generator, only one polynomial is available for numerical calculations. It does not support 16-bit CRC calculations based on any other polynomials. Each write operation to the CRCINRegister creates a combination of the previous CRC values stored in the CRCDH and CRCDLRegister. The calculation will require one MCU instruction cycle.

CRC-16-IBM (Modbus) polynomial representation: $X^{16} + X^{15} + X^2 + 1$

18Eh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCDL	CRCDL							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

18Eh.7~0 **CRCDL**: 16 bit CRCVerify data bit 7~0

18Fh	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCDH	CRCDH							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

18Fh.7~0 **CRCDH:** 16 bit CRC Verify data bit 15~8

190h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CRCIN	CRCIN							
R/W	W							
Reset	-	-	-	-	-	-	-	-

190h.7~0 **CRCIN:** CRC data input, write to this register to start CRC calculation

6.10 EEPROM (TM56E7842 only)

The TM56E7842 contains a 256-byte EEPROM chip, controlled by PD4 and PD5 as communication pins. Built-in pull-up resistors eliminate the need for external circuitry. The following usage notes apply:

PD4 is the clock communication pin (SCL), and PD5 is the data communication pin (SDA).

Control pins must be set to Open Drain with built-in pull-up resistors. The corresponding SFR pin should be set to Mode 1.

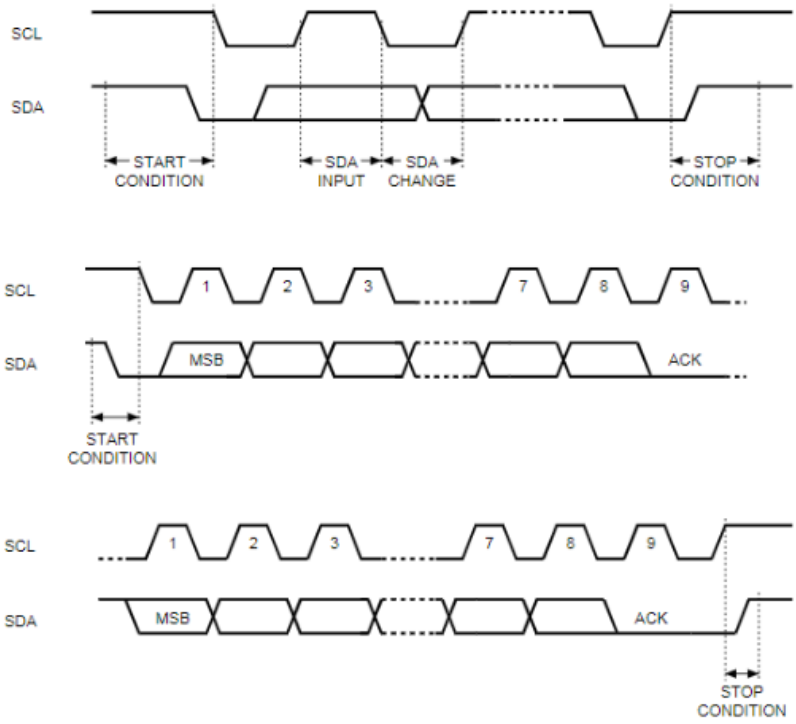
Device Select Code:

	Device				Chip Enable			RW
Bit	b7	b6	b5	b4	b3	b2	b1	b0
Device Selection	1	0	1	0	0	0	0	R/W

Operating Modes:

Mode	RW	Bytes	Initial sequence
Current Address Read	‘1’	1	START,Device Select, RW=’1’
Random Address Read	‘0’	1	START,Device Select, RW=’0’,Address
	‘1’		reSTART,Device Select RW=’1’
Sequential Read	‘1’	1 to 256	Similar to Current or Random Mode
Byte Write	‘0’	1	START,Device Select ,RW=’0’
Page Write	‘0’	8	START,Device Select ,RW=’0’

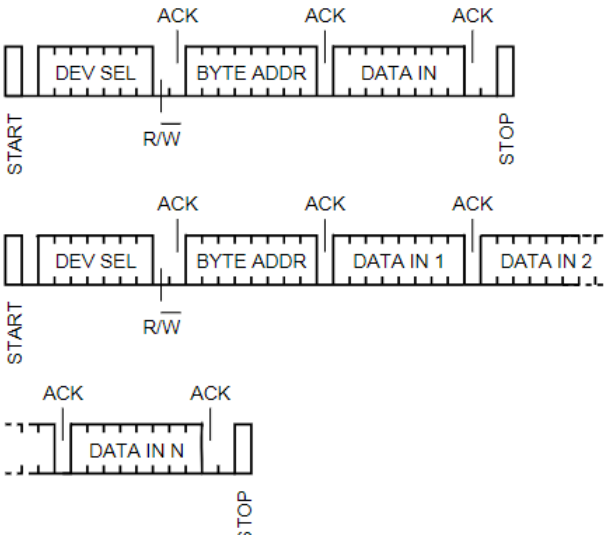
Communication Protocol:



Write mode sequence:

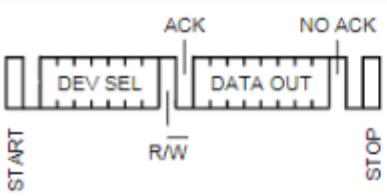
Byte Write

Page Write

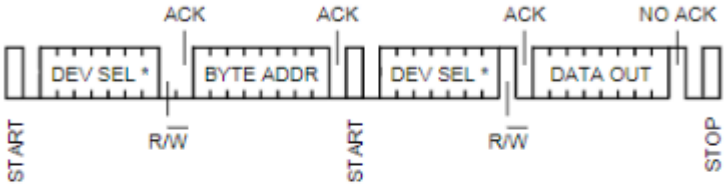


Read mode sequence:

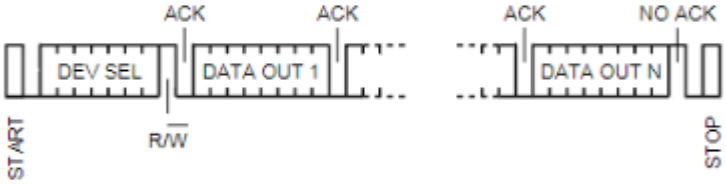
Byte Read



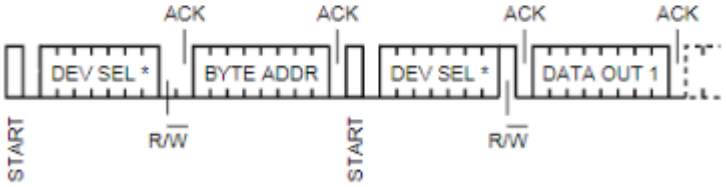
RANDOM ADDRESS READ

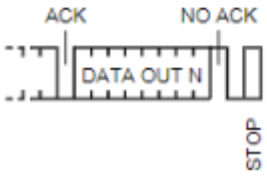


SEQUENTIAL CURRENT READ



SEQUENTIAL RANDOM READ





6.11 EV8278 Emulation

The following are explanations of the EV8278's unsupported simulation functions and usage precautions:

1. Unable to simulate PWM0P output at PB0.
2. Unable to simulate PWM5 output at PB1.
3. Unable to simulate UART TXD/RXD mapping to PD3/PD2.
4. Unable to simulate the function in section 3.4 that calculates the ratio of slow to fast clock.
5. Unable to simulate the TM56E7842 PD4/PD5 connection to EEPROM memory.
6. During simulation, operations on the Bank4 area can only be executed at full speed (Free run).
7. C language development does not support the use of Bank4 area operations.
8. The EV8278 provides SOP16/SOP20/SSOP24 pin simulation chips.
9. For SOP16/SOP20, please refer to the pin diagram description section for the remaining pins, except for the functions mentioned above that cannot be simulated.
10. The pin order of Pins 11~18 and Ports D7~0 of SSOP24 differs from the mass-produced package, including the multiplexed function pins XI, XO, ADC, TX, and RX. Please refer to Figure 6-11-1 below for detailed pinout information. Strikethrough indicates functions that EV cannot simulate, and red text indicates pins with different ordering.

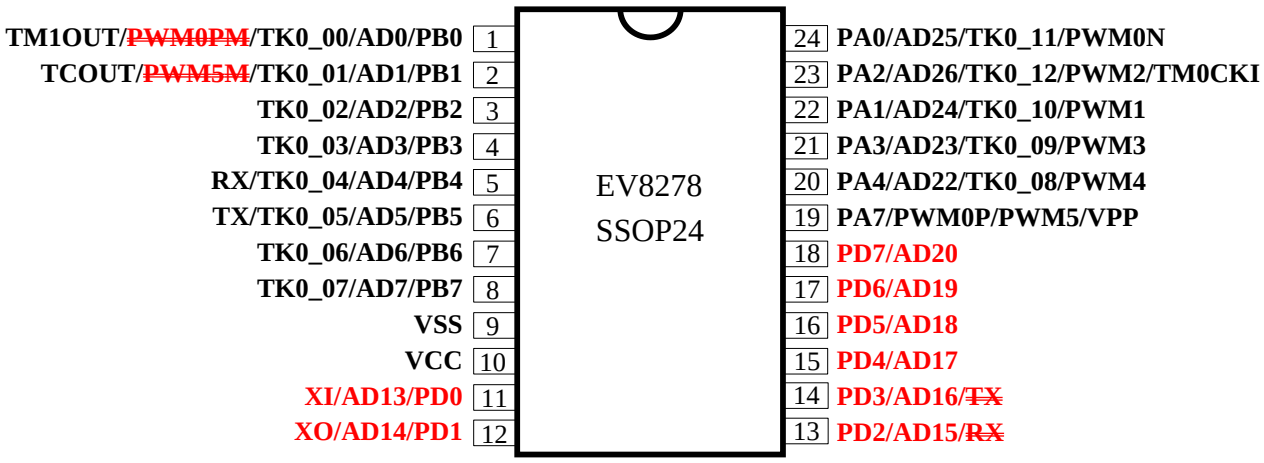


图 6-11-1 EV8278 SSOP24

MEMORY MAP

Name	Address	R/W	Rst	Description
INDF (00h/80h/100h/180h)				**Function related to: RAM**
INDF	00.7~0	R/W	-	Used for indirect addressing between bank0 and bank3. Not a physical register; any instruction using INDF actually accesses the memory pointed to by FSR.
TM0 (01h/101h)				**Function related to: Timer0**
TM0	01.7~0	R/W	0	Timer0 Count data
PCL (02h/82h/102h/182h)				**Function related to: program counter**
PCL	02.7~0	R/W	0	PC[7:0]，the low byte of the program counter.
STATUS (03h/83h/103h/183h)				**Function related to: STATUS**
IRP	03.7	R/W	0	Register Bank Selection (for indirect addressing)
RP1	03.6	R/W	0	Register Bank Selection 1 for direct addressing
RP0	03.5	R/W	0	Register Bank Selection 0 for direct addressing
TO	03.4	R	0	Watchdog timeout flag. Watchdog timeout setting. Cleared by power-on reset, ‘SLEEP’ or ‘CLRWDT’ instruction.
PD	03.3	R	0	Power loss flag. Set by the ‘SLEEP’ command. Cleared by power-on reset or the ‘CLRWDT’ command.
Z	03.2	R/W	0	Zero flag
DC	03.1	R/W	0	Decimal carry flag
C	03.0	R/W	0	Carry flag
FSR (04h/84h/104h/184h)				**Function related to: RAM**
FSR	04.7~0	R/W	-	Used for indirect addressing between Bank0 and Bank3. FSR stands for pointer, which stores addresses. Any instruction that uses INDF actually accesses the memory pointed to by FSR.
PAD (05h)				**Function related to: Port A**
PAD	05.7~0	R	-	Port A pin or “data register” status
		W	FF	Port A output data register
PBD (06h)				**Function related to: Port B**
PBD	06.7~0	R	-	Port B pin or “data register” status
		W	FF	Port B output data register
PDD (08h)				**Function related to: Port D**
PDD	08.7~0	R	-	Port D pin or “data register” status
		W	FF	Port D output data register
INDF1 (09h/89h/109h/189h)				**Function related to:RAM**
INDF1	09.7~0	R/W	0	Used for Bank4 indirect addressing. Not a physical register; any instruction using INDF1 actually accesses the memory pointed to by FSR1.
PCLATH (0Ah/8Ah/10Ah/18Ah)				**Function related to: program counter**
PCLATH	0A.3~0	R/W	0	PC[11:8] (i.e., the upper 4 bits of the program counter) is written to the buffer register. When the CPU executes any instruction that modifies the PCLRegister, PC[11:8] is provided by the PCLATHRegister by default. This function can be closed by the PCHRegister.。
INTIE (0Bh/8Bh/10Bh/18Bh)				**Function related to: Interrupt Enable**

Name	Address	R/W	Rst	Description
ADCIE	0B.7	R/W	0	ADC Interrupt Enable 0: Close 1: Enable
T2IE	0B.6	R/W	0	T2 Interrupt Enable 0: Close 1: Enable
TM1IE	0B.5	R/W	0	Timer1 Interrupt Enable 0: Close 1: Enable
TM0IE	0B.4	R/W	0	Timer0 Interrupt Enable 0: Close 1: Enable
WKTIE	0B.3	R/W	0	Wake-up timer interrupt enable, which is also wake-up timer enable. 0: Close 1: Enable
	0B.2	R/W	0	Retain
	0B.1	R/W	0	Retain
	0B.0	R/W	0	Retain
INTIF (0Ch)				**Function related to: Interrupt flags**
ADCIF	0C.7	R	-	ADC Interrupt Flag, set by H/W after ADC conversion is complete
		W	0	Write 0: Clear the flag; Write 1: No action
T2IF	0C.6	R	-	T2Interrupt Event Suspend Flag, set by H/W when T2 overflows
		W	0	Write 0: Clear the flag; write 1: No action
TM1IF	0C.5	R	-	Timer1 Interrupt Event Pending Flag, set by H/W when Timer1 overflows
		W	0	Write 0: Clear the flag; write 1: No action
TM0IF	0C.4	R	-	Timer0 Interrupt Event Pending Flag, set by H/W when Timer0 overflows
		W	0	Write 0: Clear the flag; write 1: No action
WKTIF	0C.3	R	-	WKT Interrupt event suspend flag, set by H/W when WKT times out
		W	0	Write 0: Clear the flag; write 1: No action
	0C.2	R/W	0	Retain
	0C.1	R/W	0	Retain
	0C.0	R/W	0	Retain
FSR1 (0Dh)				**Function related to: RAM**
FSR1	0D.7~0	R/W	-	Used for Bank4 indirect addressing. FSR1 represents a pointer that stores an address. Any instruction that uses INDF1 actually accesses the memory pointed to by FSR1.
INTIE1 (0Eh)				**Function related to:Interrupt Enable 组 1**
LVDIE	0E.7	R/W	0	LVDInterrupt Enable 0:Disabled 1:Enable
	0E.6	R/W	0	Retain
TKIE	0E.3	R/W	0	TKInterrupt Enable 0:Disabled 1:Enable
	0E.2	R/W	0	Retain
UARTIE	0E.1	R/W	0	UART TX/RX Interrupt Enable 0:Disabled 1:Enable
PXIE	0E.0	R/W	0	Pin level change Interrupt Enable 0:Disabled 1:Enable
CLKCTL (0Fh)				**Function related to: Fsys**

Name	Address	R/W	Rst	Description
SCKTYPE	0F.7	R/W	0	Slow clock type selection 0: SIRC 1: SXT
	0F.6	R/W	0	Retain
SLOWSTP	0F.4	R/W	0	Close Slow Clock 0: Slow clock continues running after SLEEP instruction. 1: If no other peripheral function blocks use the slow clock, the slow clock stops running after the SLEEP instruction.
FASTSTP	0F.3	R/W	1	Close Fast Clock 0: System clock is slow, fast clock remains running 1: System clock is slow, fast clock closes
CPUCKS	0F.2	R/W	0	System clock source selection 0: Slow clock 1: Fast clock
CPUPSC	0F.1~0	R/W	11	System clock source pre-divider, system clock source 00: Divide by 8 01: Divide by 4 10: Divide by 2 11: Divide by 1
TM0RLD (10h)				**Related functions：TM0**
TM0RLD	10.7~0	R/W	0	Timer0 reloads data
TM0CTL (11h)				**Related functions：TM0**
T0ISRC	11.7	R/W	0	Timer0 counting mode clock source selection 0: TM0CKI (PA2) 1: SXT/16
TM0STP	11.6	R/W	0	PauseTimer0 0: Timer0 running 1: Timer0 Pause
TM0EDG	11.5	R/W	0	TM0CKI (PA2) Edge Selection 0: Rising Edge 1: Falling Edge
TM0CKS	11.4	R/W	0	Timer0 clock source selection 0: Fsys/2 1: TM0CKI (PA2) or SXT/16
TM0PSC	11.3~0	R/W	0	Timer0 frequency selection。 When TM0CKS=1, it is recommended to keep TM0PSC at the default value of 0000. The Timer 0 clock is the clock source. (Fsys/2 or TM0CKI or SXT/16) divided by 0000: /1 0101: /32 1010: /1024 1111: /32768 0001: /2 0110: /64 1011: /2048 0010: /4 0111: /128 1100: /4096 0011: /8 1000: /256 1101: /8192 0100: /16 1001: /512 1110: /16384
TM1 (12h)				**Function related to: Timer1**
TM1	12.7~0	R/W	0	Timer1 Count data
TM1RLD (13h)				**Function related to: Timer1**
TM1RLD	13.7~0	R/W	0	Timer1 reloads data
TM1CTL (14h)				**Function related to: Timer1**
TM1STP	14.4	R/W	0	Pause Timer1 0: Timer1 running 1: Timer1 Pause
TM1PSC	14.3~0	R/W	0	Timer1pre-divider 。Timer1The clock is the clock source.(Fsys/2)divide by 0000: Fsys/1 0001: Fsys/2

Name	Address	R/W	Rst	Description
				0010: Fsys/4 0011: Fsys/8 0100: Fsys/16 0101: Fsys/32 0110: Fsys/64 0111: Fsys/128 1xxx: Fsys/256
T2CTL (15h)				**Function related to: T2**
T2CLR	15.3	R/W	1	Reset the T2 counter. Reset Timer0 to zero.
T2CKS	15.2	R/W	0	T2 Clock Source Selection 0: Slow Clock 1: Fsys/128
T2PSC	15.1~0	R/W	0	T2 Clock Source Selection 0: Slow Clock 1: Fsys/128
LVCTL (16h)				**Related function：LVR/LVD**
LVDF	16.7	R	-	Low voltage detection mark, when Vcc≦LVD, set by H/W
BGEN	16.6	R/W	1	Bandgap voltage VBGEnable When the ADC uses the internal bandgap voltage source VBG, the BGENRegister must be enabled. To save power consumption, close the BGENRegister when finished using it. 0: Bandgap voltage disabled 1: Bandgap voltage enabled in FAST/SLOW mode and automatically disabled in IDLE/STOP mode
LVRSAV	16.5	R/W	1	POR/LVR Power Saving Settings 0: Enable POR/LVR in STOP/IDLE mode 1: Disable POR/LVR in STOP/IDLE mode
LVDSAV	16.4	R/W	1	LVD power saving settings 0: Enable LVD in STOP/IDLE mode 1: Disable LVD in STOP/IDLE mode
LVDS	16.3~0	R/W	0010	LVD voltage selection 0000: Disable LVD function 0001: 4.30V 0010: 1.87V 0011: 2.00V … 1110: 3.37V 1111: 3.49V
ADCDH (17h)				**Function related to: ADC**
ADCDH	17.7~0	R	-	ADCoutput data bit 11~4
ADCTL (18h)				**Function related to: ADC**
ADCDL	18.7~4	R	-	ADCoutput data bit 3~0
ADST	18.3	R/W	0	ADC starts flag. When the user sets ADST=1, the ADC starts conversion. After conversion is complete, the H/W automatically clears this flag.
ADCKS	18.2~0	R/W	0	ADC clock frequency selection: 000: Fsys/256 100: Fsys/16 001: Fsys/128 101: Fsys/8 010: Fsys/64 110: Fsys/4 011: Fsys/32 111: Fsys/2
ADCTL2 (19h)				**Function related to: ADC**
ADCVREFS	19.7	R/W	0	ADC VREF setting,

Name	Address	R/W	Rst	Description
				0: VCC, 1: VBG
VBGSEL	19.6~5	R/W	0	VBG selection setting, 00: 1.18V, 01: 2.5V, 10: 3.0V, 11: 2.0V
ADCHS	19.4~0	R/W	1D	ADC Channel selection 00000: ADC0 (PB0) 00001: ADC1 (PB1) 00010: ADC2 (PB2) 00011: ADC3 (PB3) 00100: ADC4 (PB4) 00101: ADC5 (PB5) 00110: ADC6 (PB6) 00111: ADC7 (PB7) 01000: Retain 01001: Retain 01010: Retain 01011: Retain 01100: ADC12(VBG) 01101: ADC13(PD0) 01110: ADC14(PD1) 01111: ADC15(PD2) 10000: ADC16(PD3) 10001: ADC17(PD4) 10010: ADC18(PD5) 10011: ADC19(PD6) 10100: ADC20(PD7) 10101: Retain 10110: ADC22(PA4) 10111: ADC23(PA3) 11000: ADC24(PA1) 11001: ADC25(PA0) 11010: ADC26(PA2) 11011: Retain 11111: ADC31(1/4 VCC) others: Retain
INTIF1 (1Ah)				**Function related to:Interrupt flag**
LVDIF	1A.7	R/W	0	Low voltage detection interrupt suspend flag. When low voltage occurs, H/W sets this interrupt flag. S/W writing 0 clears this flag; S/W writing 1 has no effect.
	1A.6	R/W	0	Retain
	1A.5	R/W	0	Retain
TKM0IF	1A.4	R/W	0	Touch Key Module 0 Interrupt Suspend Flag This is set by H/W after TK0 conversion is complete. S/W writing 0 to this bit, writing 0 to TKIF, or writing 1 to TKM0SOC will clear this flag. S/W writing 1 will have no effect.
TKIF	1A.3	R/W	0	Touch KeyInterrupt Suspend Flag This is set by H/W after ATK or TK0 conversion is complete. Writing 0 to this bit or writing 1 to TKMxSOC will clear all Touch KeyInterrupt flags. Writing 1 to S/W will have no effect.
	1A.2	R/W	0	Retain
UARTIF	1A.1	R/W	0	When TX/RX transmission is complete, the UARTInterrupt flag set by H/W; F/W writes 0 to clear this flag.
PXIF	1A.0	R/W	0	PortA/B/D pin level change Interrupt flag. F/W writes 0 to clear this flag.

Name	Address	R/W	Rst	Description
PAWKE (1Ch)				**Function related to: Port A**
PAWKE	1C.7~0	R/W	0	PA pin wake-up enable 0:Close 1:Enable
PBWKE (1Dh)				**Function related to: Port B**
PBWKE	1D.7~0	R/W	0	PB pin wake-up enable 0:Close 1:Enable
PDWKE (1Fh)				**Function related to: Port D**
PDWKE	1F.7~0	R/W	0	PD pin wake-up enable(Not included PD5,PD4) 0:Close 1:Enable
User Data Memory				
RAM	20~6F	R/W	-	RAM Bank0 area (80 bytes)
RAM	70~7F	R/W	-	RAM public area (16 bytes)
OPTION (81h/181h)				
HWAUTO	81.7	R/W	0	Enable the automatic save function when entering/exiting subroutines. When entering/exiting interrupt subroutines, the hardware automatically saves/restores WREG, FSR, TABR, PCLATH, DPL, DPH, and STATUS (excluding TO and PD). 0: Disable 1: Enable
-	81.6	R/W	0	Retain
-	81.5	R/W	0	Retain
FREQL	81.4	R/W	1	SIRC Frequency Selection 0: 60KHz 1: 50KHz
WDTPSC	81.3~2	R/W	11	WDT cycle selection: FREQL=1: 00: 188ms@3V, 169ms@5V 01: 371ms@3V, 333ms@5V 10: 749ms@3V, 668ms@5V 11: 1492ms@3V, 1355ms@5V FREQL=0: 00: 147ms@3V, 135ms@5V 01: 293ms@3V, 272ms@5V 10: 580ms@3V, 529ms@5V 11: 1180ms@3V, 1058ms@5V
WKTPSC	81.1~0	R/W	11	WKT cycle selection: FREQL=1: 00: 22ms@3V, 20ms@5V 01: 44ms@3V, 40ms@5V 10: 89ms@3V, 80ms@5V 11: 177ms@3V, 160ms@5V FREQL=0: 00: 18ms@3V, 16ms@5V 01: 35ms@3V, 32ms@5V 10: 70ms@3V, 63ms@5V 11: 139ms@3V, 125ms@5V
PAMODH (85h)				**Function related to: PortA**

Name	Address	R/W	Rst	Description
PA7MOD	85.7~6	R/W	01	PA7、PA4 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PA4MOD	85.1~0	R/W	01	
PAMODL (86h)				**Function related to: PortA**
PA3MOD	86.7~6	R/W	01	PA3~PA0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PA2MOD	86.5~4	R/W	01	
PA1MOD	86.3~2	R/W	01	
PA0MOD	86.1~0	R/W	01	
PBMODH (87h)				**Function related to: PortB**
PB7MOD	87.7~6	R/W	01	PB7~PB4 I/O mode control 00: Mode0 01:Mode1 10: Mode2 11: Mode3
PB6MOD	87.5~4	R/W	01	
PB5MOD	87.3~2	R/W	01	
PB4MOD	87.1~0	R/W	01	
PBMODL (88h)				**Function related to: PortB**
PB3MOD	88.7~6	R/W	01	PB3~PB0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PB2MOD	88.5~4	R/W	01	
PB1MOD	88.3~2	R/W	01	
PB0MOD	88.1~0	R/W	01	
PDMODH (8Dh)				**Function related to:Port D**
PD7MOD	8D.7~6	R/W	01	PD7~PD4 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PD6MOD	8D.5~4	R/W	01	
PD5MOD	8D.3~2	R/W	01	
PD4MOD	8D.1~0	R/W	01	
PDMODL (8Eh)				**Function related to:Port D**
PD3MOD	8E.7~6	R/W	01	PD3~PD0 I/O mode control 00: Mode0 01: Mode1 10: Mode2 11: Mode3
PD2MOD	8E.5~4	R/W	01	
PD1MOD	8E.3~2	R/W	01	
PD0MOD	8E.1~0	R/W	01	
OPTION2 (8Fh)				
TCOE	8F.7	R/W	0	TCOUT Output Switch 0: Disabled. 1: Enabled, output to PB1.
TM1OE	8F.6	R/W	0	Timer1 overflow switches output switch 0: Disabled. 1: Enabled, output to PB0
PWMCKS	8F.5~4	R/W	0	PWM clock source selection 00: Fsys 01:FIRC/256 10:FIRC 11:FIRC*2
-	8F.2	R/W	0	Retain
-	8F.1	R/W	0	Retain
-	8F.0	R/W	0	Retain
PWMOE (90h)				**Related function： PWM0/PWM1**
PWM0POE	90.7	R/W	0	PWM0P output Enable to PA7 or PB0 0:Disabled 1:Enable
PWM0NOE	90.6	R/W	1	PWM0N output Enable to PA0 0:Disabled

Name	Address	R/W	Rst	Description
				1:Enable
PWM5OE	90.5	R/W	1	PWM5 output Enable to PA7 or PB1 0: Disabled 1: Enable (for PA7, PWM0P's output priority is higher than PWM5)
PWM4OE	90.4	R/W	1	PWM4 output Enable to PA4 0:Disabled 1:Enable
PWM3OE	90.3	R/W	1	PWM3 output Enable to PA3 0:Disabled 1:Enable
PWM2OE	90.2	R/W	0	PWM2 output Enable to PA2 0:Disabled 1:Enable
PWM1OE	90.1	R/W	0	PWM1 output Enable to PA1 0:Disabled 1:Enable
PWMCTL (91h)				**Related function：PWM**
PWMEN	91.7	R/W	0	PWM Clock Enable 0: Disabled 1: Enable
PWM0OM	91.5~4	R/W	0	PWM0 Output Mode 00 ~ 11: Mode 0 ~ Mode 3
PWM0DZ	91.3~0	R/W	0	PWM0 dead zone width selection 0000~1111: 0~ 14,16 *Tpwmclk
PWMPRDH (92h)				**Related function：PWM**
PWMPRDH	92.7~0	R/W	0	PWM duty cycle high byte
PWMPRDL (93h)				**Related function：PWM**
PWMPRDL	93.7~0	R/W	0	PWM duty cycle low byte
PWM0DH (94h)				**Related function：PWM**
PWM0DH	94.7~0	R/W	0	PWM0 duty cycle high byte
PWM0DL (95h)				**Related function：PWM**
PWM0DL	95.7~0	R/W	0	PWM0 duty cycle low byte
PWM1DH (96h)				**Related function：PWM**
PWM1DH	96.7~0	R/W	0	PWM1duty cycle high byte
PWM1DL (97h)				**Related function：PWM**
PWM1DL	97.7~0	R/W	0	PWM1 duty cycle low byte
PWM2DH (98h)				**Related function：PWM**
PWM2DH	98.7~0	R/W	0	PWM2duty cycle high byte
PWM2DL (99h)				**Related function：PWM**
PWM2DL	99.7~0	R/W	0	PWM2 duty cycle low byte
PWM3DH (9Ah)				**Related function：PWM**
PWM3DH	9A.7~0	R/W	0	PWM3duty cycle high byte
PWM3DL (9Bh)				**Related function：PWM**
PWM3DL	9B.7~0	R/W	0	PWM3 duty cycle low byte
PWM4DH (9Ch)				**Related function：PWM**
PWM4DH	9C.7~0	R/W	0	PWM4duty cycle high byte
PWM4DL (9Dh)				**Related function：PWM**

Name	Address	R/W	Rst	Description
PWM4DL	9D.7~0	R/W	0	PWM4 duty cycle low byte
PWM5DH (9Eh)				**Related function：PWM**
PWM5DH	9E.7~0	R/W	0	PWM5duty cycle high byte
PWM5DL (9Fh)				**Related function：PWM**
PWM5DL	9F.7~0	R/W	0	PWM5 duty cycle low byte
User Data Memory				
RAM	A0~EF	R/W	-	RAM bank1 area (80 bytes)
TESTREG (105h)				
TESTREG	105.7~0	R/W	00	Test bit, keep 00
RDCTL (106h)				**Related function：ROM Read**
RDCTL	106.1~0	W	1	Program ROM read signal delay control 00: Program ROM read signal delay 4ns 01: Program ROM read signal delay 8ns 10: Program ROM read signal delay 12ns 11: Program ROM read signal delay 16ns For safety reasons, please change the register when the clock is slow. Users must switch this register to “4ns” to improve the performance of the minimum operating voltage. This feature is not simulated.
LVRPD (107h)				**Related function： LVR**
LVRPD	107	W	0	LVR power Close control Write 0x37 to force Close LVR+POR Write 0x38 to force Close LVR, POR remains Enable Write 0x39 to force Close POR, LVR remains Enable Writing other values to this register will Enable LVR+POR
PORPDF	107.1	R	0	0: When Register107h is written with other values 1: When Register.107h is written with 0x37 or 0x39;
LVRPDF	107.0	R	0	0: When Register107h is written with other values 1: When Register107h is written with 0x37 or 0x38;
PCH (10Ch)				**Related function：program counter**
PCH	10C.3~0	R	0	Read PC[11:8], i.e., the upper 4 bits of the program counter.
		W	0	(Only available to users of assembly language; not applicable to users of C language.) After writing 1Ch to this register, when the CPU executes any instruction that modifies PCL, PC[11:8] remains unchanged. After writing other values to this register, when the CPU executes any instruction that modifies the PCL, PC[11:8] is provided by the PCLATH register.
UBAUD (10Dh)				**Related function：UART**
UBAUD	10D	R/W	0	UART baud rate divider
BGTRIM (10Eh)				**Related function：bandgap voltage VBG**
BGTRIM	10E.4~0	R/W		VBG voltage adjustment 00h: Minimum voltage... 1Fh: Maximum voltage
TKM0DL (116h)				**Related function：Touch Key**
TKM0DL	116.7~0	R	-	Touch Key Module 0 data LSB [7~0]
TKM0DH (117h)				**Related function：Touch Key**

Name	Address	R/W	Rst	Description
TKM0DH	117.5~0	R	-	Touch Key Module 0 data MSB [13~8]
ATKUBL (118h)				**Related function：Touch Key**
ATKUBL	118.7~0	R	-	ATKTouch Key Module limit[7~0]
ATKUBH (119h)				**Related function：Touch Key**
ATKUBH	119.5~0	R	-	ATKTouch Key Module limit[13~8]
ATKLBL (11Ah)				**Related function：Touch Key**
ATKLBL	11A.7~0	R	-	ATKTouch Key Module lower limit[7~0]
ATKLBH (11Bh)				**Related function：Touch Key**
ATKLBH	11B.5~0	R	-	ATKTouch Key Module lower limit[13~8]
TKMCON0 (11Ch)				**Related function：Touch Key**
-	11C.7~5	R/W	001	Retain
TKM0TCP	11C.4~2	R/W	001	Touch Key Module 0 Clock Frequency Selection (only effective when TKM0JMP=0) 000: Slow …, 111: Fastest
TKM0FSL	11C.1~0	R/W	0	Touch Key Module RCK0/TCK0 Frequency Selection 00: Slow, …, 11: Fastest
TKMCON1 (11Dh)				**Related function：Touch Key**
-	11D.7	R/W	0	Retain
TKM0HSEN	11D.6	R/W	0	Touch Key Module 0 Sensitivity 0: Low sensitivity 1: High sensitivity
TKMCTL0 (11Eh)				**Related function：Touch Key**
-	11E.5	R/W	0	Retain
TKM0SOC	11E.4	R/W	0	Touch Key module 0 starts conversion, HW clears it when conversion is complete
-	11E.1	R	-	Retain
TKM0EOC	11E.0	R	-	Touch Key module 0 conversion complete
TKMCTL1 (11Fh)				**Related function：Touch Key**
-	11F.7	R/W	0	Retain
FDBL0	11F.6	R/W	1	Touch Key Module 0 Count Doubling 0: No doubling 1: Doubling
-	11F.5	R/W	0	Retain
TKM0PD	11F.4	R/W	1	Touch Key Module 0 Close 0: Touch KeyEnable 1: Touch KeyClose
-	11F.3~2	R/W	0	Retain
-	11F.1	R/W	0	Retain
TKM0JMP	11F.0	R/W	0	Touch Key Module 0 Clock Mode 0: Fixed Frequency 1: Automatic Frequency Switching
User Data Memory				
RAM	120~16F	R/W	-	RAM Bank2 area (80 bytes)
DPL (185h)				**Function related to:Table read**

Name	Address	R/W	Rst	Description
DPL	185.7~0	R/W	0	Read low address, low byte of data ROM pointer (DPTR)
DPH (186h)				**Function related to:Table read**
DPH	186.4~0	R/W	0	Read high address, high byte of data ROM pointer (DPTR)
UARTCTL (187h)				**Function related to:UART**
URTD9	187.7	R/W	0	The 9th bit of UART transmission
UMODE	187.6~5	R/W	0	UART Mode Selection 0: 7 bits 1: 8 bits 2: 9 bits 3: Retain
URINTTF	187.4	R	0	When UART TX transmission is complete, these bits are set to 1 by H/W; clearing UARTIF also clears this flag.
UTXE	187.3	R/W	0	UART transmission enabled 0:Disabled 1:Enable
URXE	187.2	R/W	0	UART Receive Enable 0:Disabled 1:Enable
UARTE	187.1	R/W	0	UART Function Enable 0:Disabled 1:Enable
UINVEN	187.0	R/W	0	UART TX/RXoutput/Input reversal Enable 0:Disabled 1:Enable
UARTSTA (188h)				**Function related to:UART**
UTBE	188.7	R	0	0: Send TX 1: TX buffer is empty;
URRD9	188.6	R	0	Received 9 bits
EVEN	188.5	R/W	0	UART parity 0: Odd 1: Even
PRE	188.4	R/W	0	UART parity check 0: Close 1: Enable,
PRERR	188.3	R/W	0	Set to 1 when a parity error occurs, and clear it by writing 0 to F/W.
OVERR	188.2	R/W	0	Set to 1 when an overflow error occurs, and clear it by writing 0 to F/W.
FMERR	188.1	R/W	0	Set to 1 when a frame error occurs, and clear it by writing 0 to F/W.
URBF	188.0	R	0	Buffer full flag. Set to 1 when 8 bits or 9 bits are received. F/W reads UARTDAT Register and clears it to 0.
TABR (18Ch)				**Function related to: Table Read**
TABR	18C.7~0	R/W	0	TABRRegister is designed for reading and writing ROM. Writing 1 to TABR is equivalent to the instruction TABRL. Writing 2 to TABR is equivalent to the instruction TABRH. Reading TABR returns the ROM read value. Assembly users can use TABRRegister or the instructions TABRL/TABRH. C language users can only use TABRRegister.
URDATA (18Dh)				**Function related to: UART**

Name	Address	R/W	Rst	Description
URDATA	18D.7~0	R/W	0	UART TX/RX data buffer
		R	0	Move bits from receive buffer
		W	0	to transmitter
CRCDL (18Eh)				**Function related to: CRC16**
CRCDL	18E.7~0	R/W	0	CRC16 data 7~0
CRCDH(18Fh)				**Function related to: CRC16**
CRCDH	18F.7~0	R/W	0	CRC16 data 15~8
CRCIN(190h)				**Function related to: CRC16**
CRCIN	190.7~0	W	0	CRC Input data
TKM0TMRL (194h)				**Function related to: Touch Key**
TKM0TMRL	194.7~0	R/W	FF	Touch Key Module 0 Reference Counter LSB [7~0]
TKM0TMRH (195h)				**Function related to: Touch Key**
TKM0TMRH	195.5~0	R/W	0	Touch Key Module 0 Reference Counter MSB [13~8]
TKM0REFC (19Ah)				**Function related to: Touch Key**
TKM0REFC	19A.7~0	R/W	-	Touch Key Module 0 Reference Clock Capacitor Selection
SFR19B (19Bh)				
CFEN	19B.7	R/W	0	Capture frequency function enabled. When CFEN=1, the hardware switches the Timer0 clock source to the fast clock.
CFST	19B.6	R/W	0	Capture frequency function start flag. When CFST=1, sampling begins, and the hardware automatically clears this bit when sampling is complete.
TM0CLR	19B.5	R/W	0	Timer0 reset。
ULEDMASK	19B.4	R/W	0	UART TX masking for cascaded LED applications. When ULEDMASK = 1: (1) During the UART stop bit and idle period, the TX pin is fixed at output 0. (2) The 16-pre-divider is not used as the UART clock source.
TXSEL	19B.3	R/W	0	UART TX Pin selection. 0：PB5 as UART TX 1：PD3 as UART TX
RXSEL	19B.2	R/W	0	UART RX Pin selection. 0：PB4 as UART TX 1：PD2 as UART TX
PWM0PSEL	19B.1	R/W	0	PWM0P Pin selection. 0：PA7 as PWM0P output 1：PB0 as PWM0P output
PWM5SEL	19B.0	R/W	0	PWM5 Pin selection. 0：PA7 as PWM5 output 1：PB1 as PWM5 output
ATKCON (19Ch)				**Function related to: Touch Key**
ATKRATE	19C.7~6	R/W	0	ATK scan rate (provided by SIRC) 00: 20ms 01: 25ms 10: 40ms 11: 80ms
ATKEN	19C.5	R/W	0	ATK Enable (TKM0 CH0 : PB0)
TKMCHS0 (19Dh)				**Function related to: Touch Key**
-	19D.7~4	R/W	0	Retain
TKM0CHS	19D.3~0	R/W	0	Touch Key Module 0 Channel Selection 00: TK0_0 01:TK0_1 02:TK0_2 03:TK0_3 04:TK0_4 05: TK0_5 06:TK0_6 05:TK0_7 08:TK0_8 09:TK0_9

Name	Address	R/W	Rst	Description
				10:TK0_11 11:TK0_11 12:TK0_12
User Data Memory				
RAM	1A0~1EF	R/W	-	RAM Bank 3 area (80 bytes)
Indirect Addressing User Data Memory				
RAM	00~ 7F	R/W	-	Indirectly address RAM Bank4 area (128 bytes); use **INDF1/FSR1**

INSTRUCTION SET

Each instruction is a 16-bit word, consisting of an opcode (used to specify the instruction type) and one or more operands (used to further specify the operation of the instruction). The following table lists the instructions categorized into byte-oriented, bit-oriented, and immediate-oriented operations.

For byte-oriented instructions, “f” represents the address specifier, and “d” represents the destination specifier. The address specifier is used to specify which address in program memory the instruction will use. The destination specifier specifies where the operation result will be placed. If “d” is “0,” the result is stored in the WRegister. If ‘d’ is “1,” the result is placed at the address specified by the instruction.

For bit-oriented instructions, “b” represents the bit field indicator, which selects the number of bits affected by the operation, while ‘f’ represents the address indicator. For immediate value operations, “k” represents the immediate value or constant value.

Abbreviations	Description
f	File Register Address
b	Bits Address
k	Immediate Number. Constant Number or Label
d	Target Selection Field, 0: Work Register, 1: File Register
W	Work Register
Z	Zero Flag
C	Bits Flag or Borrow Bits Flag
DC	Decimal Bits Flag or Decimal Borrow Bits Flag
PC	Program Counter
TOS	Top Level Stack
GIE	General Interrupt Enable Flag (i-Flag)
[]	Selection Field
( )	Contents
.	Bits Field
B	Before
A	After
←	Allocation Direction

Mnemonic		Op Code	Cycle	Flag Affect	Description
Byte-Oriented File Register Instruction					
ADDWX	f, d	ff00 0111 dfff ffff	1	C, DC, Z	W and “f” added together
ADDWXC	f, d	ff01 0111 dfff ffff	1	C, DC, Z	W and “f” added together with C
ANDWX	f, d	ff00 0101 dfff ffff	1	Z	W and “f” ANDed
CLRX	f	ff00 0001 1fff ffff	1	Z	Clear “f”
CLRW		0000 0001 0100 0000	1	Z	Clear W
COMX	f, d	ff00 1001 dfff ffff	1	Z	Invert “f”
DECX	f, d	ff00 0011 dfff ffff	1	Z	Decrement “f” by 1
DECXSZ	f, d	ff00 1011 dfff ffff	1 or 2	-	Decrement “f” by 1; skip if zero
INCX	f, d	ff00 1010 dfff ffff	1	Z	Increment "f"
INCXSZ	f, d	ff00 1111 dfff ffff	1 or 2	-	Increment "f", skip if zero
IORWX	f, d	ff00 0100 dfff ffff	1	Z	OR W with "f"
MOVX	f,d	ff00 1000 dfff ffff	1	Z	Move "f"
MOVXW	f	ff00 1000 0fff ffff	1	Z	Move "f" to W
MOVWX	f	ff00 0000 1fff ffff	1	-	Move W to "f"
RLX	f, d	ff00 1101 dfff ffff	1	C	Rotate left "f" through carry
RRX	f, d	ff00 1100 dfff ffff	1	C	Rotate right "f" through carry
SUBWX	f, d	ff00 0010 dfff ffff	1	C, DC, Z	Subtract W from "f"
SUBWXB	f, d	ff01 0100 dfff ffff	1	C, DC, Z	"f" subtract W and $\overline{C}$
SUBXW	f, d	ff01 0010 dfff ffff	1	C, DC, Z	W subtract "f"
SUBXWB	f, d	ff01 0011 dfff ffff	1	C, DC, Z	W subtract "f" and $\overline{C}$
SWAPX	f, d	ff00 1110 dfff ffff	1	-	Swap the high and low half-bytes of “f”
TSTX	f	ff00 1000 1fff ffff	1	Z	Check whether “f” is 0
XORWX	f, d	ff00 0110 dfff ffff	1	Z	W and “f” are different or
Bit-Oriented File Register Instruction					
BCX	f, b	ff11 00bb bfff ffff	1	-	Clear the “b” bits of ‘f”
BSX	f, b	ff11 01bb bfff ffff	1	-	Set the “b” bits of “f”
BTXSC	f, b	ff11 10bb bfff ffff	1 or 2	-	Skip if the “b” bits of “f” are 0
BTXSS	f, b	ff11 11bb bfff ffff	1 or 2	-	Skip if the “b” bits of “f” are 1
Literal and Control Instruction					
ADDLW	k	0001 1100 kkkk kkkk	1	C, DC, Z	Add Literal "k" and W
ANDLW	k	0001 1011 kkkk kkkk	1	Z	AND Literal "k" with W
LCALL	k	kk10 0kkk kkkk kkkk	2	-	Call subroutine “k”
CLRWDT		0001 1110 0000 0100	1	TO, PD	Clear watchdog timer
LGOTO	k	kk10 1kkk kkkk kkkk	2	-	Jump to branch “k”
IORLW	k	0001 1010 kkkk kkkk	1	Z	OR Literal "k" with W
MOVLW	k	0001 1001 kkkk kkkK	1	-	Move Literal "k" to W
NOP		0000 0000 0000 0000	1	-	No operation
RET		0000 0000 0100 0000	2	-	Return from subroutine
RETI		0000 0000 0110 0000	2	-	Return from interrupt
RETLW	k	0001 1000 kkkk kkkk	2	-	Return with Literal in W
SLEEP		0001 1110 0000 0011	1	TO, PD	Go into Power-down mode, Clock oscillation stops
SUBLW	k	0001 1111 kkkk kkkk	1	C, DC, Z	Subtract W from literal
TABRH		0000 0000 0101 1000	2	-	Lookup ROM high data to W
TABRL		0000 0000 0101 0000	2	-	Lookup ROM low data to W
XORLW	k	0001 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	0001 1100 kkkk kkkk	
Description	将 W Register的内容和 8 bits立即数 'k' 相加，将结果放入WRegister中。	
Cycle	1	
Example	ADDLW 0x15	B : W =0x10 A : W =0x25

ADDWX	Add W and "f"	
Syntax	ADDWX f [,d]	
Operands	f : 000h ~ 1FFh, d : 0, 1	
Operation	(destination) ← (W) + (f)	
Status Affected	C, DC, Z	
OP-Code	ff00 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	ADDWX FSR, 0	B : W =0x17, FSR =0xC2 A : W =0xD9, FSR =0xC2

ADDWXC	Add W and "f"	
Syntax	ADDWXC f [,d]	
Operands	f : 000h ~ 1FFh, d : 0, 1	
Operation	(destination) ← (W) + (f) + C	
Status Affected	C, DC, Z	
OP-Code	ff01 0111 dfff ffff	
Description	Add the contents of the W register with register 'f' and C. 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	ADDWXC FSR, 0	B : W =0x17, FSR =0xC2, C=0 A : W =0xD9, FSR =0xC2, C=0
Example	ADDWXC FSR, 0	B : W =0x17, FSR =0xC2, C=1 A : W =0xDA, FSR =0xC2, C=0

ANDLW	Logical AND Literal "k" with W	
Syntax	ANDLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← (W) AND k	
Status Affected	Z	
OP-Code	0001 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

ANDWX	AND W with “f”	
Syntax	ANDWX f [,d]	
Operands	f : 000h ~ 1FFh, d : 0, 1	
Operation	(destination) ← (W) AND (f)	
Status Affected	Z	
OP-Code	ff00 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	ANDWX FSR, 1	B : W =0x17, FSR =0xC2 A : W =0x17, FSR =0x02

BCX	**Clear "b" bit of "f"**	
Syntax	BCX f [,b]	
Operands	f : 000h ~ 1FFh, b : 0 ~ 7	
Operation	(f.b) ← 0	
Status Affected	-	
OP-Code	ff11 00bb bfff ffff	
Description	Bit 'b' in register 'f' is cleared.	
Cycle	1	
Example	BCX FLAG_REG, 7	B : FLAG_REG =0xC7 A : FLAG_REG =0x47

BSX	**Set "b" bit of "f"**	
Syntax	BSX f [,b]	
Operands	f : 000h ~ 1FFh, b : 0 ~ 7	
Operation	(f.b) ← 1	
Status Affected	-	
OP-Code	ff11 01bb bfff ffff	
Description	Bit 'b' in register 'f' is set.	
Cycle	1	
Example	BSX FLAG_REG, 7	B : FLAG_REG =0x0A A : FLAG_REG =0x8A

BTXSC	**Test "b" bit of "f", skip if clear(0)**	
Syntax	BTXSC f [,b]	
Operands	f : 00h ~ 1FFh, b : 0 ~ 7	
Operation	Skip next instruction if (f.b) =0	
Status Affected	-	
OP-Code	ff11 10bb bfff ffff	
Description	If bit 'b' in register 'f' is 1, then the next instruction is executed. If bit 'b' in register 'f' is 0, then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.	
Cycle	1 or 2	
Example	LABEL1 BTXSC FLAG, 1 TRUE LGOTO SUB1 FALSE ...	B : PC =LABEL1 A : if FLAG.1 =0, PC =FALSE if FLAG.1 =1, PC =TRUE

BTXSS	**Test "b" bit of "f", skip if set(1)**	
Syntax	BTXSS f [,b]	
Operands	f : 000h ~ 1FFh, b : 0 ~ 7	
Operation	Skip next instruction if (f.b) =1	
Status Affected	-	
OP-Code	ff11 11bb bfff ffff	
Description	If bit 'b' in register 'f' is 0, then the next instruction is executed. If bit 'b' in register 'f' is 1, then the next instruction is discarded, and a NOP is executed instead, making this a 2nd cycle instruction.	
Cycle	1 or 2	
Example	LABEL1 BTXSS FLAG, 1 TRUE LGOTO SUB1 FALSE ...	B : PC =LABEL1 A : if FLAG.1 =0, PC =TRUE if FLAG.1 =1, PC =FALSE

CLRX	**Clear "f"**	
Syntax	CLRX f	
Operands	f : 000h ~ 1FFh	
Operation	(f) ← 00h, Z ← 1	
Status Affected	Z	
OP-Code	ff00 0001 1fff ffff	
Description	The contents of register 'f' are cleared and the Z bit is set.	
Cycle	1	
Example	CLRX FLAG_REG	B : FLAG_REG =0x5A A : FLAG_REG =0x00, Z =1

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

CLRWDT	**Clear Watchdog Timer**	
Syntax	CLRWDT	
Operands	-	
Operation	WDT Timer ← 00h	
Status Affected	TO, PD	
OP-Code	0001 1110 0000 0100	
Description	CLRWDT instruction clears the Watchdog Timer	
Cycle	1	
Example	CLRWDT	B : WDT counter =? A : WDT counter =0x00

COMX	**Complement "f"**	
Syntax	COMX f [,d]	
Operands	f : 000h ~ 1FFh, d : 0, 1	
Operation	(destination) ← ($\bar{f}$)	
Status Affected	Z	
OP-Code	ff00 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	COMX REG1, 0	B : REG1 =0x13 A : REG1 =0x13, W =0xEC

DECXSZ	**Decrement "f", Skip if 0**

DECX	**Decrement "f"**	
Syntax	DECX f [,d]	
Operands	f : 000h ~ 1FFh, d : 0, 1	
Operation	(destination) ← (f) - 1	
Status Affected	Z	
OP-Code	ff00 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	DECX CNT, 1	B : CNT =0x01, Z =0 A : CNT =0x00, Z =1
Syntax	DECXSZ f [,d]	
Operands	f : 000h ~ 1FFh, d : 0, 1	
Operation	(destination) ← (f) - 1, skip next instruction if result is 0	
Status Affected	-	
OP-Code	ff00 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 DECXSZ CNT, 1 LGOTO LOOP CONTINUE	B : PC =LABEL1 A : CNT =CNT – 1 if CNT =0, PC =CONTINUE if CNT ≠0, PC =LABEL1 + 1

INCX	**Increment "f"**	
Syntax	INCX f [,d]	
Operands	f : 000h ~ 1FFh	
Operation	(destination) ← (f) + 1	
Status Affected	Z	
OP-Code	ff00 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	INCX CNT, 1	B : CNT =0xFF, Z =0 A : CNT =0x00, Z =1

INCXSZ	**Increment "f", Skip if 0**
Syntax	INCXSZ f [,d]
Operands	f : 00h ~ 1FFh, d : 0, 1
Operation	(destination) ← (f) + 1, skip next instruction if result is 0
Status Affected	-
OP-Code	ff00 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 INCXSZ CNT, 1 LGOTO 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	0001 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

IORWX	**Inclusive OR W with "f"**
Syntax	IORWF f [,d]
Operands	f : 00h ~ 1FFh, d : 0, 1
Operation	(destination) ← (W) OR k
Status Affected	Z
OP-Code	ff00 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	IORWX RESULT, 0 B : RESULT =0x13, W =0x91 A : RESULT =0x13, W =0x93, Z =0

LCALL	**Call subroutine "k"**
Syntax	LCALL k
Operands	k : 000h ~ 1FFFh
Operation	Operation: TOS ← (PC) + 1, PC.10~0 ← k
Status Affected	-
OP-Code	kk10 0kkk kkkk kkkk
Description	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The 13-bit immediate address is loaded into PC bits <12:0>. The upper bits of PC are loaded from PCLATH. CALL is a two-cycle instruction.
Cycle	2
Example	LABEL1 LCALL SUB1 B : PC =LABEL1 A : PC =SUB1, TOS =LABEL1 + 1

LGOTO	Unconditional Branch	
Syntax	LGOTO k	
Operands	k : 0000h ~ 1FFFh	
Operation	PC.12~0 ← k	
Status Affected	-	
OP-Code	kk10 1kkk kkkk kkkk	
Description	LGOTO is an unconditional branch. The 13-bit immediate value is loaded into PC bits <12:0>.The upper bits of PC are loaded from PCLATH. LGOTO is a two-cycle instruction.	
Cycle	2	
Example	LABEL1 LGOTO SUB1	B : PC =LABEL1 A : PC =SUB1

MOVX	Move f	
Syntax	MOVX f [,d]	
Operands	f : 000h ~ 1FFh	
Operation	(destination) ← (f)	
Status Affected	Z	
OP-Code	ff00 1000 dfff ffff	
Description	The contents of register 'f' are moved to a destination dependent upon the status of d. If d=0, destination is W register. If d =1, the destination is file register f itself. d=1 is useful to test a file register, since status flag Z is affected.	
Cycle	1	
Example	MOVX FSR,0	B : FSR =0xC2, W =? A : FSR =0xC2, W =0xC2

MOVXW	Move "f" to W	
Syntax	MOVXW f	
Operands	f : 000h ~ 1FFh	
Operation	(W) ← (f)	
Status Affected	Z	
OP-Code	ff00 1000 0fff ffff	
Description	The contents of register 'f' are moved to W register.	
Cycle	1	
Example	MOVXW 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	0001 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

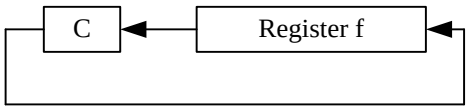
MOVWX	**Move W to "f"**	
Syntax	MOVWX f	
Operands	f : 000h ~ 1FFh	
Operation	(f) ← (W)	
Status Affected	-	
OP-Code	ff00 0000 1fff ffff	
Description	Move data from W register to register 'f'.	
Cycle	1	
Example	MOVWX REG1	B : REG1 =0xFF, W =0x4F A : REG1 =0x4F, W =0x4F

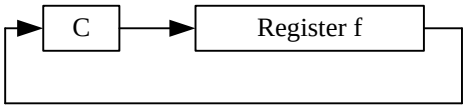
NOP	**No Operation**	
Syntax	NOP	
Operands	-	
Operation	No Operation	
Status Affected	-	
OP-Code	0000 0000 0000 0000	
Description	No Operation	
Cycle	1	
Example	NOP	-

RET	**Return from Subroutine**	
Syntax	RET	
Operands	-	
Operation	PC ← TOS	
Status Affected	-	
OP-Code	0000 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	0000 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	0001 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	LCALL TABLE : TABLE ADDWX PCL, 1 RETLW k1 RETLW k2 : RETLW kn B : W =0x07 A : W =value of k8

RLX	**Rotate Left "f" through Carry**
Syntax	RLX f [,d]
Operands	f : 000h ~ 1FFh, d : 0, 1
Operation	
Status Affected	C
OP-Code	ff00 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	RLX REG1, 0 B : REG1 =1110 0110, C =0 A : REG1 =1110 0110 W =1100 1100, C =1

RRX	**Rotate Right "f" through Carry**
Syntax	RRX f [,d]
Operands	f : 000h ~ 1FFh, d : 0, 1
Operation	
Status Affected	C
OP-Code	ff00 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	RRX 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	001 1110 0000 0011	
Description	Go into Power-down mode with the oscillator stops.	
Cycle	1	
Example	SLEEP	-

SUBLW	Subtract W from Literal	
Syntax	SUBLW k	
Operands	k : 00h ~ FFh	
Operation	(W) ← k – (W)	
Status Affected	C, DC, Z	
OP-Code	0001 1111 kkkk kkkk	
Description	The W register is subtracted (2's complement method) from the eight-bit literal "k". The result is placed in the W register.	
Cycle	1	
Example	SUBLW 0x25	B : W =0x15 A : W =0x10

SUBWX	Subtract W from "f"	
Syntax	SUBWX f [,d]	
Operands	f : 00h ~ 1FFh, d : 0, 1	
Operation	(destination) ← (f) – (W)	
Status Affected	C, DC, Z	
OP-Code	ff00 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	SUBWX REG1, 1	B : REG1 =0x03, W =0x02, C =?, Z =? A : REG1 =0x01, W =0x02, C =1, Z =0
	SUBWX REG1, 1	B : REG1 =0x02, W =0x02, C =?, Z =? A : REG1 =0x00, W =0x02, C =1, Z =1
	SUBWX REG1, 1	B : REG1 =0x01, W =0x02, C =?, Z =? A : REG1 =0xFF, W =0x02, C =0, Z =0

SUBWXB — Subtract W and $\overline{C}$ from "f"

Syntax	SUBWXB f [,d]
Operands	f : 000h ~ 1FFh, d : 0, 1
Operation	(destination) ← (f) – (W) – $\overline{C}$
Status Affected	C, DC, Z
OP-Code	ff01 0100 dfff ffff
Description	Subtract (2's complement method) W register and $\overline{C}$ 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	SUBWXB REG1, 1 — B : REG1 =0x19, W =0x0D, C =1, Z =? A : REG1 =0x0C,W =0x0D, C =1, Z =0
	SUBWXB REG1, 1 — B : REG1 =0x1B, W =0x1A, C =0 Z =? A : REG1 =0x00, W =0x1A, C =1, Z =1
	SUBWXB REG1, 1 — B : REG1 =0x03, W =0x0E, C =1, Z =? A : REG1 =0xF5, W =0x0E, C =0, Z =0

SUBXW — Subtract "f" from W

Syntax	SUBXW f [,d]
Operands	f : 000h ~ 1FFh, d : 0, 1
Operation	(destination) ←(W) – (f)
Status Affected	C, DC, Z
OP-Code	ff01 0010 dfff ffff
Description	Subtract (2's complement method) register 'f' from W. 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	SUBXW REG1, 1 — B : REG1 =0x02, W =0x03, C =?, Z =? A : REG1 =0x01, W =0x03, C =1, Z =0
	SUBXW REG1, 1 — B : REG1 =0x02, W =0x02, C =?, Z =? A : REG1 =0x00, W =0x02, C =1, Z =1
	SUBXW REG1, 1 — B : REG1 =0x02, W =0x01, C =?, Z =? A : REG1 =0xFF, W =0x02, C =0, Z =0

SUBXWB — Subtract "f" and $\overline{C}$ from W

Syntax	SUBXWB f [,d]
Operands	f : 000h ~ 1FFh, d : 0, 1
Operation	(destination) ←(W) – (f) – $\overline{C}$
Status Affected	C, DC, Z
OP-Code	ff01 0011 dfff ffff
Description	Subtract (2's complement method) register 'f' and $\overline{C}$ from W. 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	SUBXWB REG1, 1 — B : REG1 =0x03, W =0x02, C =1, Z =? A : REG1 =0xFF, W =0x02, C =0, Z =0
	SUBXWB REG1, 1 — B : REG1 =0x02, W =0x05, C =1, Z =? A : REG1 =0x03, W =0x05, C =1, Z =0
	SUBXWB REG1, 1 — B : REG1 =0x01, W =0x02, C =0, Z =? A : REG1 =0x00, W =0x02, C =1, Z =1

SWAPX	Swap Nibbles in "f"
Syntax	SWAPX f [,d]
Operands	f : 00h ~ 1FFh, d : 0, 1
Operation	(destination,7~4) ← (f.3~0), (destination.3~0) ← (f.7~4)
Status Affected	-
OP-Code	ff00 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	SWAPX 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	0000 0000 0101 1000
Description	The W and TABR register is loaded with high byte of ROM[DPTR]. This is a two-cycle instruction.
Cycle	2
Example	MOVLW (TAB1&0xFF) MOVWX DPL ;Where DPL is register MOVLW (TBA1>>8)&0xFF MOVWX DPH ;Where DPH is register TABRL ;W =0x89, TABR=0x89 TABRH ;W =0x37, TABR=0x37 ORG 0234H TAB1: DT 0x3789, 0x2277 ;ROM data 16 bits

TABRL	Return DPTR low byte to W
Syntax	TABRL
Operands	-
Operation	(W) ← ROM[DPTR] low byte content, Where DPTR = {DPH[max:8], FSR[7:0]}
Status Affected	-
OP-Code	0000 0000 0101 0000
Description	The W and TABR register is loaded with low byte of ROM[DPTR]. This is a two-cycle instruction.
Cycle	2
Example	MOVLW (TAB1&0xFF) MOVWX DPL ;Where DPL register MOVLW (TBA1>>8)&0xFF MOVWX DPH ;Where DPH register TABRL ;W =0x89, TABR=0x89 TABRH ;W =0x37, TABR=0x37 ORG 0234H TAB1: DT 0x3789, 0x2277 ;ROM data 16 bits

TSTX	Test if "f" is zero	
Syntax	TSTX f	
Operands	f : 00h ~ 1FFh	
Operation	Set Z flag if (f) is 0	
Status Affected	Z	
OP-Code	ff00 1000 1fff ffff	
Description	If the content of register 'f' is 0, Zero flag is set to 1.	
Cycle	1	
Example	TSTX 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	0001 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

XORWX	Exclusive OR W with "f"	
Syntax	XORWX f [,d]	
Operands	f : 00h ~ 1FFh, d : 0, 1	
Operation	(destination) ← (W) XOR (f)	
Status Affected	Z	
OP-Code	ff00 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	XORWX REG, 1	B : REG =0xAF, W =0xB5 A : REG =0x1A, W =0xB5

ELECTRICAL CHARACTERISTICS

1. Absolute Maximum Ratings (T_A =25℃)

Parameter	Rating	Unit
Supply voltage	V_{SS} -0.3 to V_{SS} +5.5	V
Input voltage	V_{SS} -0.3 to V_{CC} +0.3	
Output voltage	V_{SS} -0.3 to V_{CC} +0.3	
Output current high per 1 PIN	-25	mA
Output current high per all PIN	-80	
Output current low per 1 PIN	+30	
Output current low per all PIN	+150	
Maximum operating voltage	5.5	V
Operating temperature	-40 to +105	℃
Storage temperature	-65 to +150	

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

Parameter	Sym	Conditions		Min	Typ	Max	Unit
Working voltage	V_{cc}	Fsys=18.432Mhz (RDCTL=4ns)		2.2		5.5	
		Fsys= 9.216Mhz (RDCTL=4ns)		1.7		5.5	
High input voltage	V_{IH}	All inputs	V_{CC} = 3~5V	0.6Vcc	–	Vcc	V
Low input voltage	V_{IL}	All inputs	V_{CC} = 3~5V	Vss	–	0.2Vcc	V
I/O port Pull current	I_{OH}	PA0~PA6 PB0~PB7 PD0~PD7	V_{CC} = 5V,V_{OH} = 4.5V		12	-	mA
			V_{CC} = 5V,V_{OH} = 3.5V		30	-	
			V_{CC} = 3V,V_{OH} = 2.7V		5	-	
			V_{CC} = 3V,V_{OH} = 2.1V		13	-	
I/O port Injection current	I_{OL}	PB1~PB7 PD4~PD7	V_{CC} = 5V,V_{OL} = 0.5V		46	-	mA
			V_{CC} = 5V,V_{OL} = 1.5V		99	-	
			V_{CC} = 3V,V_{OL} = 0.3V		20	-	
			V_{CC} = 3V,V_{OL} = 0.9V		46	-	
		PA0~PA4 PB0 PD0~PD3	V_{CC} = 5V,V_{OL} = 0.5V		39	-	mA
			V_{CC} = 5V,V_{OL} = 1.5V		89	-	
			V_{CC} = 3V,V_{OL} = 0.3V		18	-	
			V_{CC} = 3V,V_{OL} = 0.9V		42	-	
		PA7	V_{CC} = 5V,V_{OL} = 0.5V		36	-	mA
			V_{CC} = 5V,V_{OL} = 1.5V		82	-	
			V_{CC} = 3V,V_{OL} = 0.3V		16	-	
			V_{CC} = 3V,V_{OL} = 0.9V		38	-	
Input leakage current (pin high)	I_{ILH}	All inputs	V_{IN} = V_{CC}	–	–	1	uA
Input leakage current (pin low)	I_{ILL}	All inputs	V_{IN} = 0V	–	–	–1	uA
Working current (no load)		Fast Mode FIRC 18.4 MHz	V_{CC} = 5V	-	5.0	-	mA
			V_{CC} = 3V	-	2.9	-	
		Fast Mode FIRC 9.2 MHz	V_{CC} = 5V	-	3.5	-	
			V_{CC} = 3V	-	2.1	-	
		Fast Mode FIRC 4.6 MHz	V_{CC} = 5V	-	2.4	-	
			V_{CC} = 3V	-	1.6	-	
		Fast mode FIRC 2.3 MHz	V_{CC} = 5V	-	1.8	-	
			V_{CC} = 3V	-	1.2	-	

		Slow mode SXT 32KHz FIRC stopped POR/LVR enabled	V_{CC} = 5V	-	230	-	uA
			V_{CC} = 3V	-	200	-	
		Slow mode SXT 32KHzz FIRC Stop POR/LVR Off	V_{CC} = 5V	-	36	-	
			V_{CC} = 3V	-	13	-	
		Slow mode SIRC 50KHz FIRC Stop POR/LVR On	V_{CC} = 5V	-	220	-	
			V_{CC} = 3V	-	200	-	
		Slow mode SIRC 50KHz FIRC Stop POR/LVR Off	V_{CC} = 5V	-	20	-	
			V_{CC} = 3V	-	12	-	
		Idle Mode SIRC 50 KHz POR/LVR Off	V_{CC} = 5V	-	7	-	uA
			V_{CC} = 3V	-	2	-	
		Stop Mode POR/LVR Off	V_{CC} = 5V	-	-	1	uA
			V_{CC} = 3V	-	-	1	
pull-up resistor	R_{UP}	V_{IN} = 0 V PA0~PA4 PB0~PB7 PD0~PD3 PD6~PD7	V_{CC} = 5V	-	32	-	KΩ
			V_{CC} = 3V	-	53	-	
		V_{IN} = 0 V PD4~PD5	V_{CC} = 5V	-	6.4	-	KΩ
			V_{CC} = 3V	-	9.9	-	
		V_{IN} = 0 V PA7	V_{CC} = 5V	-	35	-	KΩ
			V_{CC} = 3V	-	37	-	

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

Parameter	Condition	Min	Typ	Max	Unit
FIRC Frequency	25℃, V_{CC} =5.0 V	-1%	18.432	+1%	MHz
	-40℃ ~ 105℃, V_{CC} = 5.0 V	-1.5%	18.432	+1.5%	
	-40℃ ~ 105℃, V_{CC} = 3.0 ~ 5.0V	-4.5%	18.432	+1.5%	

Parameter	Condition	Min	Typ	Max	Unit
SIRC Frequency	25℃, V_{CC} =3.0 V, FREQL=1	-	45	-	KHz
	25℃, V_{CC} =5.0 V, FREQL=1	-	50	-	
	25℃, V_{CC} =3.0 V, FREQL=0	-	57	-	
	25℃, V_{CC} =5.0 V, FREQL=0	-	60	-	

4. EEPROM Characteristics (T_A = 25°C)

Parameter	Conditions	Min	Typ	Max	Unit
Write voltage	-40℃~85℃	2.0	5.0	5.5	V
Clock frequency	VCC=2.0V			100	KHz
	VCC=3.0V			400	
Write cycle time	VCC=2.0~5.5V			5	ms
Write endurance	VCC=2.0V~5.0V, 25℃	1KK			cycles

Note: These parameters are eigenvalues, not 100% measured values.

5. Reset Timing Characteristics (T_A = 25°C)

Parameter	Conditions	Min	Typ	Max	Unit
RESET Input Low width	Input V_{CC} = 5 V ±10 %	-	30	–	μs
WDT time	V_{CC} = 3 V, WDTPSC = 11, FREQL=1	-	1492	-	ms
	V_{CC} = 5 V, WDTPSC = 11, FREQL=1	-	1355	-	
	V_{CC} = 3 V, WDTPSC = 11, FREQL=0	-	1180	-	
	V_{CC} = 5 V, WDTPSC = 11, FREQL=0	-	1058	-	
WKT time	V_{CC} = 3 V, WKTPSC = 11, FREQL=1	-	177	-	ms
	V_{CC} = 5 V, WKTPSC = 11, FREQL=1	-	160	-	
	V_{CC} = 3 V, WKTPSC = 11, FREQL=0		139		
	V_{CC} = 5 V, WKTPSC = 11, FREQL=0		125		
CPU start up time	V_{CC} = 5 V	-	39	-	ms
	V_{CC} = 3 V		43		

6. LVR Circuit Characteristics (T_A = 25℃)

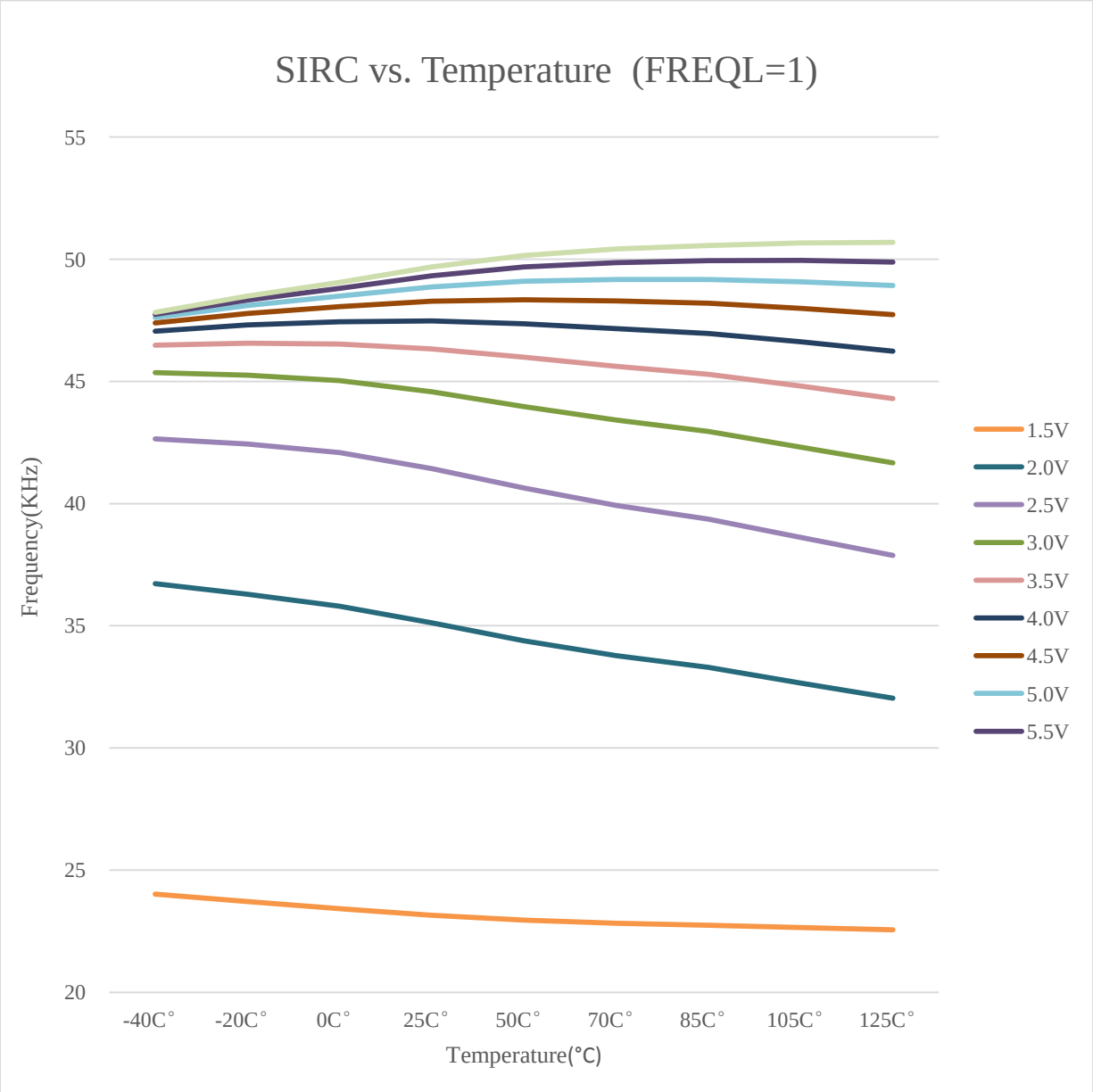
Parameter	Sym	Conditions	Min	Typ	Max	Unit
LVR Reference Voltage	LVR_{th}	T_A = 25℃	–	3.49	–	V
			–	3.37	–	
			–	3.25	–	
			–	3.12	–	
			–	2.99	–	
			–	2.86	–	
			–	2.73	–	
			–	2.61	–	
			–	2.49	–	
			–	2.37	–	
			–	2.25	–	
			–	2.13	–	
			–	2.00	–	
			–	1.87	–	
			–	1.74	–	
			–	1.61	–	

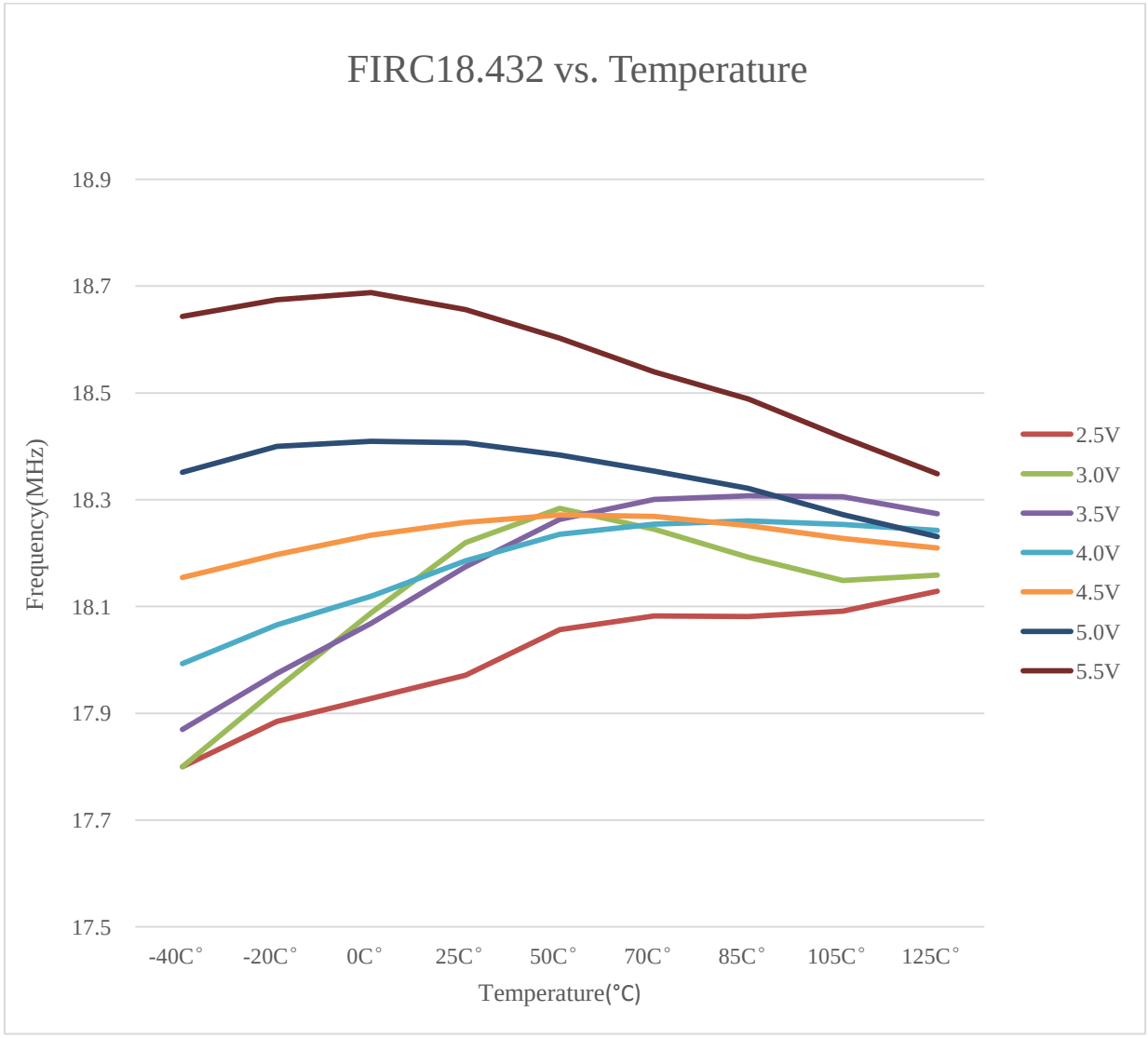
LVD Reference Voltage	LVD_{th}	T_A = 25℃	–	3.49	–	V
			–	3.37	–	
			–	3.25	–	
			–	3.12	–	
			–	2.99	–	
			–	2.86	–	
			–	2.73	–	
			–	2.61	–	
			–	2.49	–	
			–	2.37	–	
			–	2.25	–	
			–	2.13	–	
			–	2.00	–	
			–	1.87	–	
			–	4.30	–	
			–	-	–	
LVR Hysteresis Voltage	V_{HYS_LVR}	T_A = 25℃	–	±0.1	–	V
Low Voltage	T_{LVR}	T_A = 25℃	100	–	–	μs

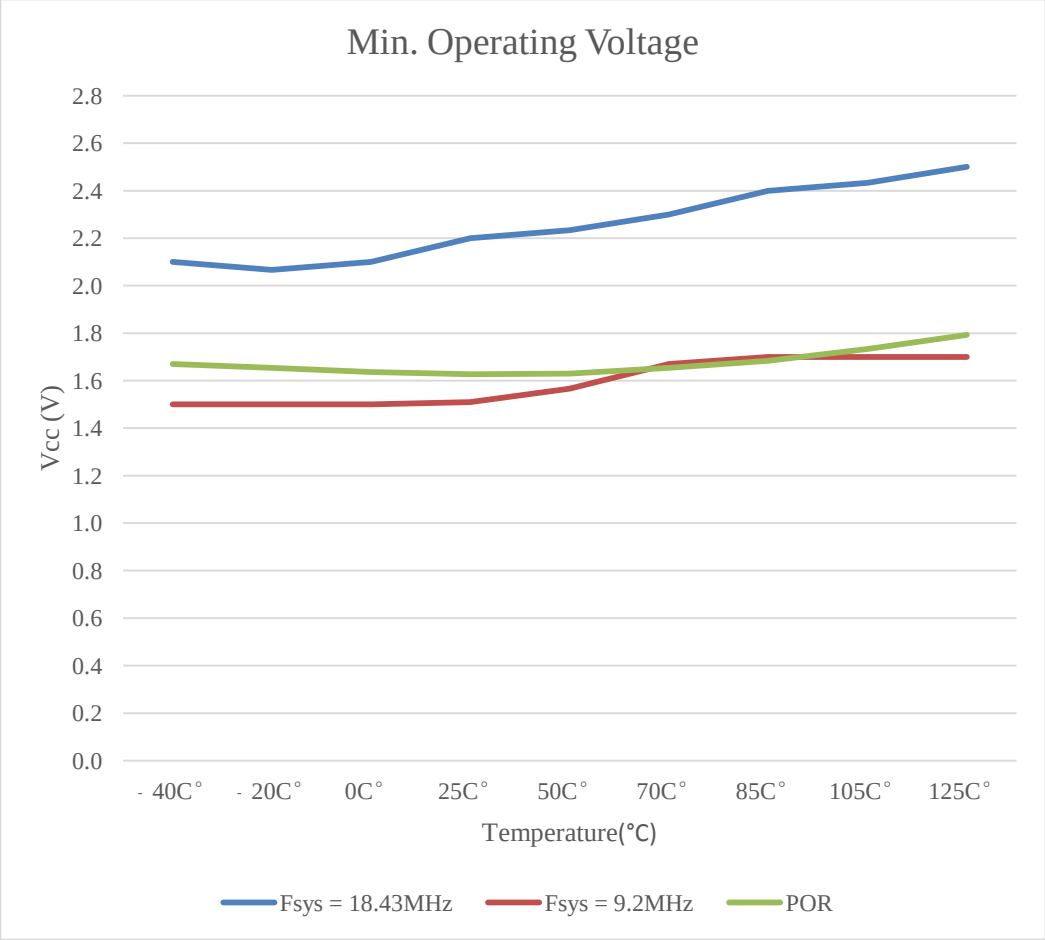
7. ADC Electrical Characteristics (T_A = 25℃, VCC = 3.0V to 5.5V, VSS = 0V)

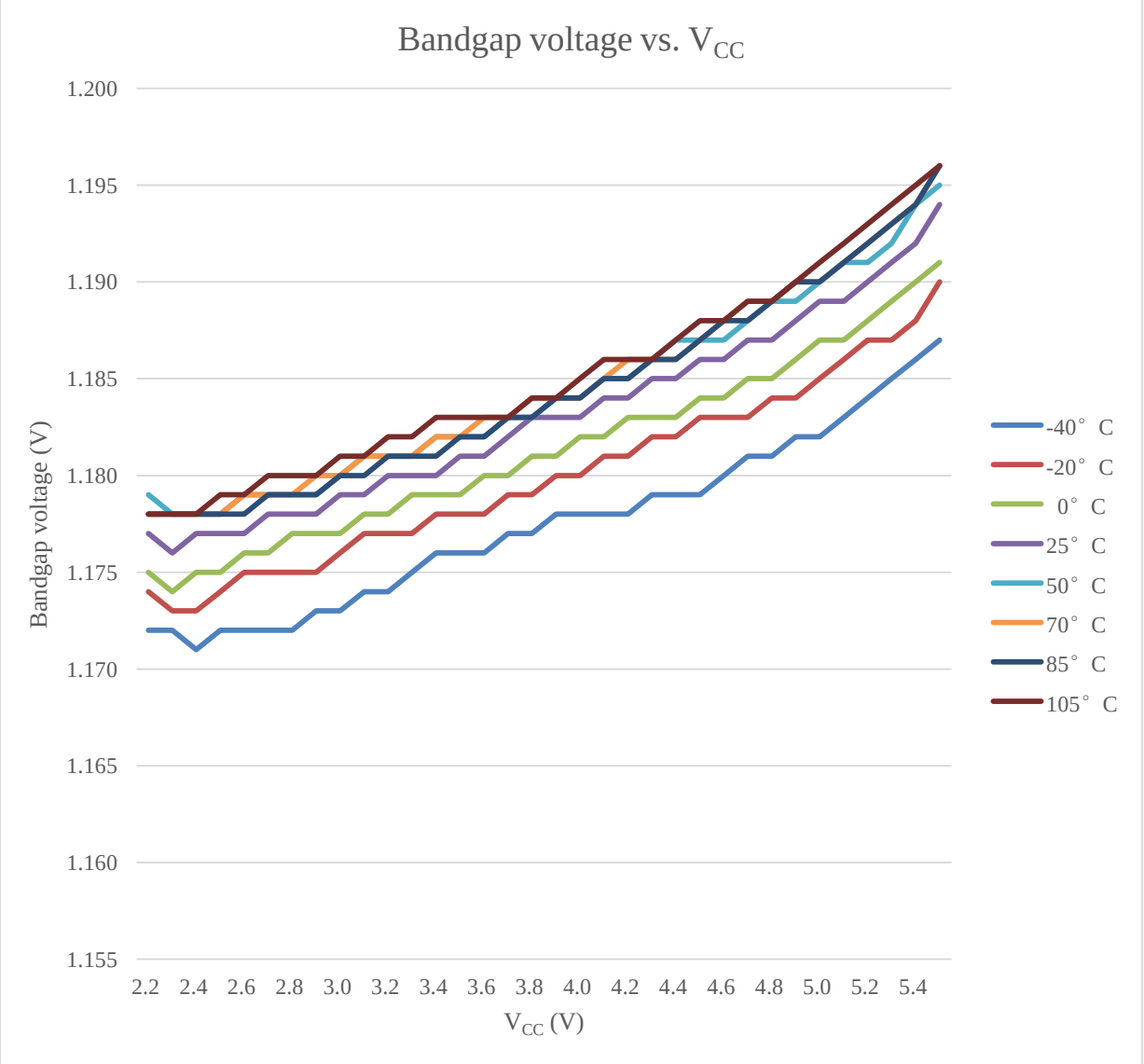
Parameter	Conditions	Min	Typ	Max	Units
Total accuracy	V_{CC} = 5V, V_{SS} = 0V, F_{ADC} = 1 MHz	–	±3	±13	LSB
Integral nonlinear error		–	±3.2	±15	
Differential nonlinear error		–	±1	±4	
Maximum input clock frequency (F_{ADC})	Source impedance (Rs<10K ohm)	–	–	2	MHz
	Source impedance (Rs<20K ohm)	–	–	1	
	Source impedance (Rs<50K ohm)	–	–	0.5	
	Source is VBG (ADCHS=01110b)	–	–	2	
Conversion time	F_{ADC} = 1 MHz	–	21	–	μs
Bandgap voltage reference (VBG)	-20℃~105℃, V_{CC} = 3.0~5.0V	-2%	1.18	+1.5%	V
ADC reference voltage (V_{REF}) (ADVREFS=01b)	-20℃~105℃, V_{CC} = 3.0~5.0V	-2.5%	2.5	+2%	
V_{CC}/4 reference voltage	25℃, V_{CC} = 3.0~5.0V	-1%	0.25*Vcc	1%	
Input voltage	–	V_{SS}	–	Vcc	

8. Characteristic Graphs




FIRC18.432 vs. Temperature
18.9
18.7
18.5
18.3
18.1
17.9
17.7
17.5
Frequency(MHz)
-40C°
-20C°
0C°
25C°
50C°
70C°
85C°
105C°
125C°
Temperature(℃)
2.5V
3.0V
3.5V
4.0V
4.5V
5.0V
5.5V


Min. Operating Voltage
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Vcc (V)
- 40C°
- 20C°
0C°
25C°
50C°
70C°
85C°
105C°
125C°
Temperature(℃)
Fsys = 18.43MHz
Fsys = 9.2MHz
POR

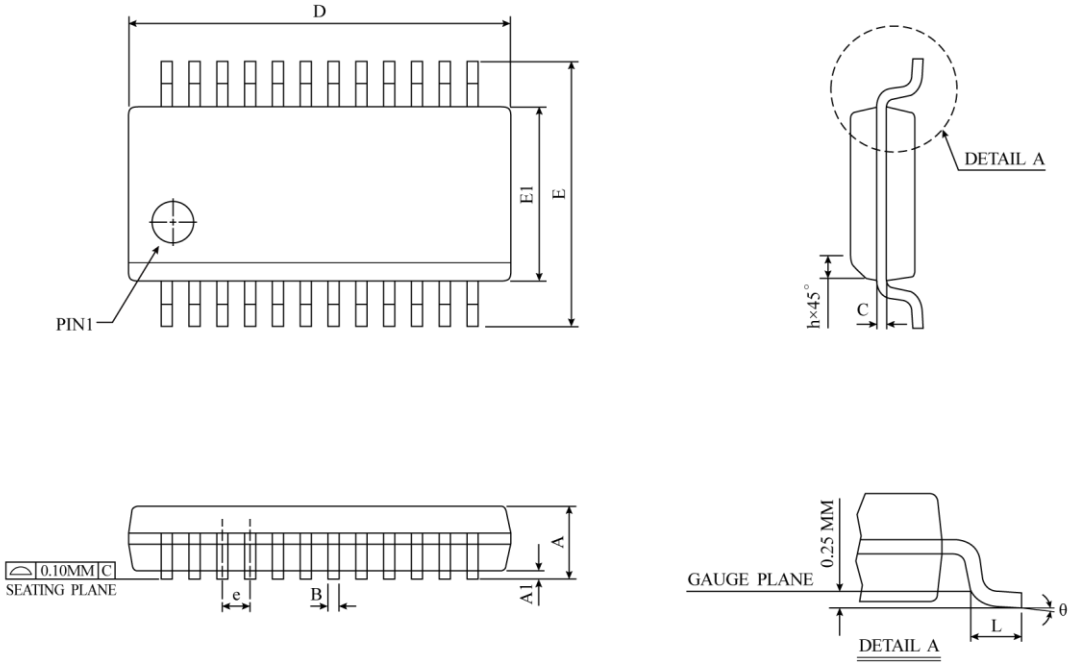

Bandgap voltage vs. V_{CC}
1.200
1.195
1.190
1.185
1.180
1.175
1.170
1.165
1.160
1.155
Bandgap voltage (V)
2.2 2.4 2.6 2.8 3.0 3.2 3.4 3.6 3.8 4.0 4.2 4.4 4.6 4.8 5.0 5.2 5.4
V_{CC} (V)
-40° C
-20° C
0° C
25° C
50° C
70° C
85° C
105° C

PACKAGING INFORMATION

Please note that the packaging information provided here is for reference only. As this information is updated frequently, users can contact sales staff for the latest packaging information and inventory.

Ordering number	Package
TM56F78424E1	SSOP 24 pin (150mil)
TM56F78423S	SOP 20-pin (300 mil)
TM56F78422S	SOP 16-pin (150 mil)
TM56F78422S1	SOP 16-pin (150 mil)
TM56E78424E1	SSOP 24 pin (150mil)
TM56E78423S	SOP 20-pin (300 mil)
TM56E78422S	SOP 16-pin (150 mil)
TM56E78422S1	SOP 16-pin (150 mil)

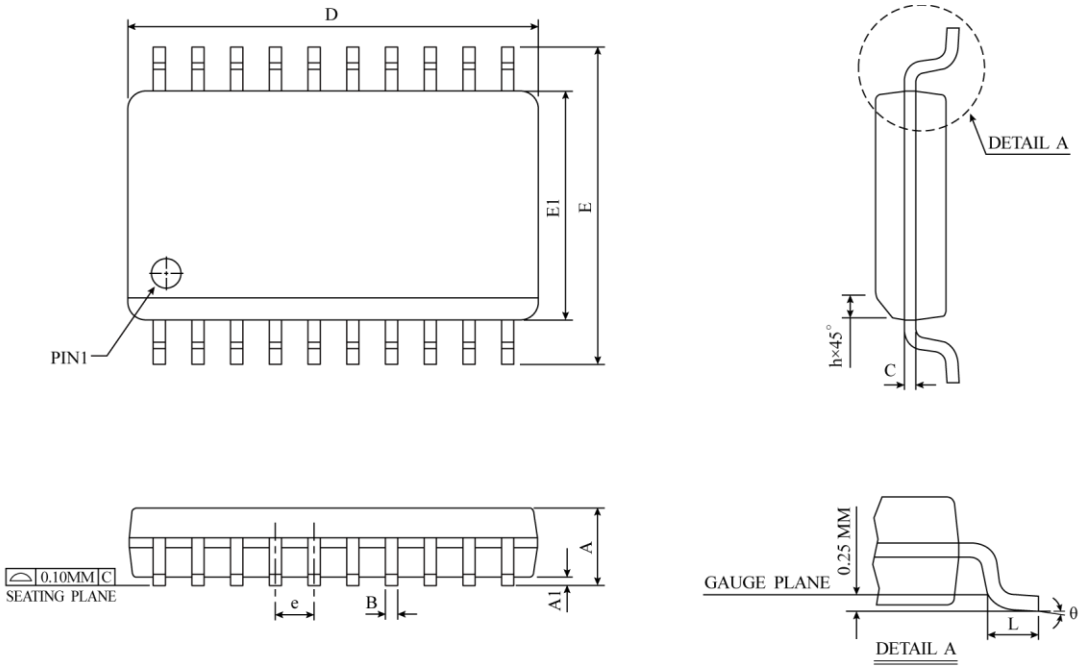
SSOP-24 (150mil) Package Dimension



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.55	1.75	0.053	0.061	0.069
A1	0.10	0.18	0.25	0.004	0.007	0.010
A2	-	-	1.50	-	-	0.059
B	0.20	0.25	0.30	0.008	0.010	0.012
C	0.18	0.22	0.25	0.007	0.009	0.010
D	8.56	8.65	8.74	0.337	0.341	0.344
E	5.79	6.00	6.20	0.228	0.236	0.244
E1	3.81	3.90	3.99	0.150	0.154	0.157
e	0.635 BSC			0.025 BSC		
L	0.41	0.84	1.27	0.016	0.033	0.050
θ	0°	4°	8°	0°	4°	8°
JEDEC	M0-137 (AE)					

Ⓑ * NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD PROTRUSIONS OR GAT BURRS.
MOLD PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED 0.006 INCH PER SIDE.

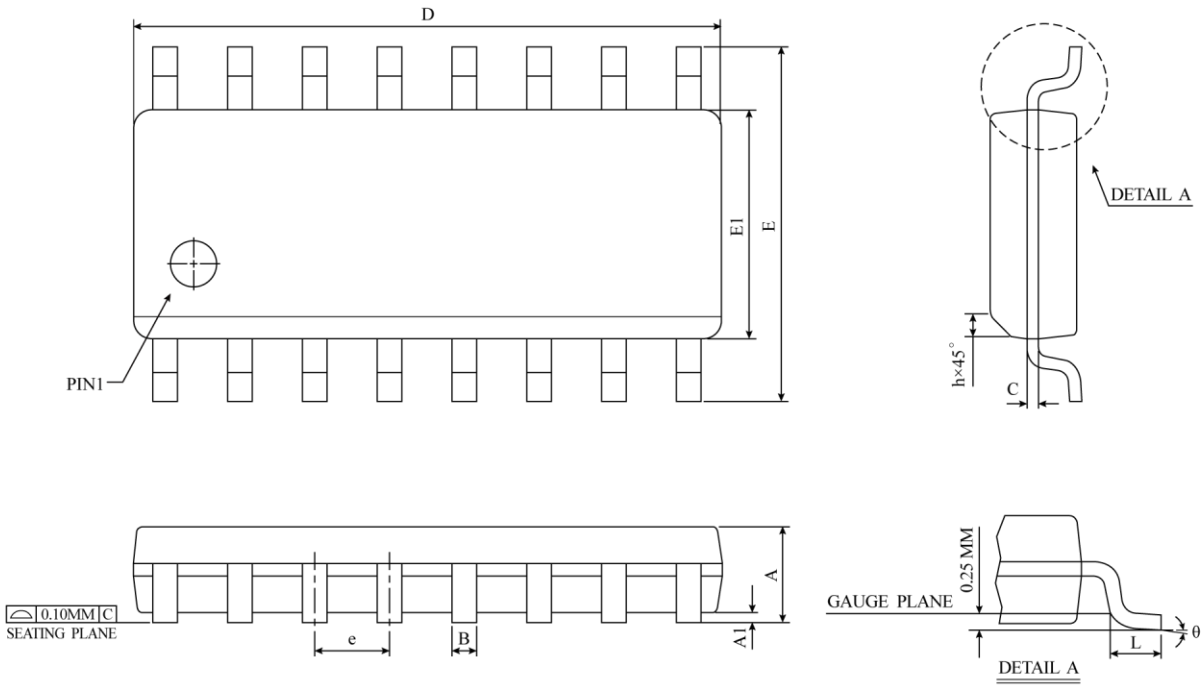
SOP-20 (300 mil) Package Dimension



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

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

SOP-16 (150 mil) Package Dimension

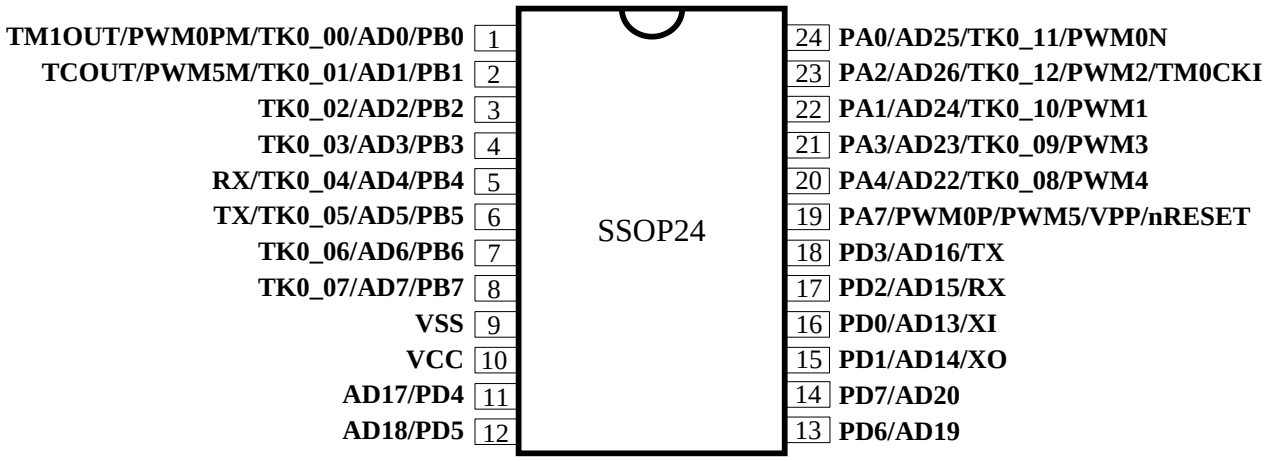


SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	1.35	1.55	1.75	0.0532	0.0610	0.0688
A1	0.10	0.18	0.25	0.0040	0.0069	0.0098
B	0.33	0.42	0.51	0.0130	0.0165	0.0200
C	0.19	0.22	0.25	0.0075	0.0087	0.0098
D	9.80	9.90	10.00	0.3859	0.3898	0.3937
E	5.80	6.00	6.20	0.2284	0.2362	0.2440
E1	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)					

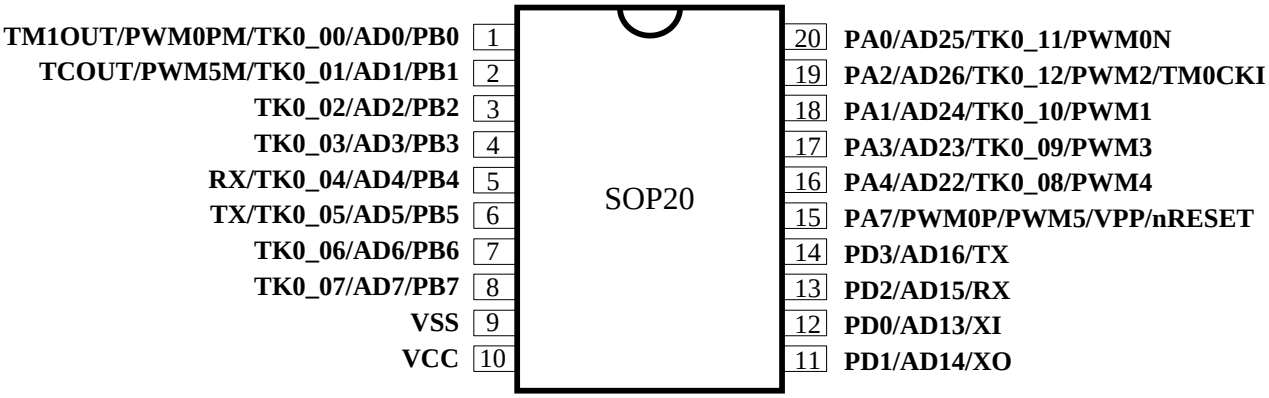
* NOTES : DIMENSION " D " DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS. MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED 0.15 MM (0.006 INCH) PER SIDE.

Appendix: Pin Assignment Diagram

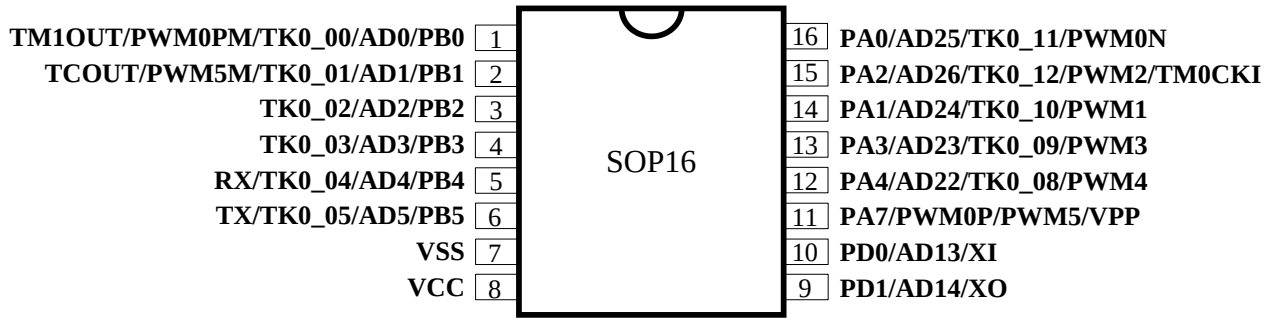
TM56M78424E1:



TM56M78423S/E78423S



TM56M78422S/E78422S



TM56M78422S1/E78422S1