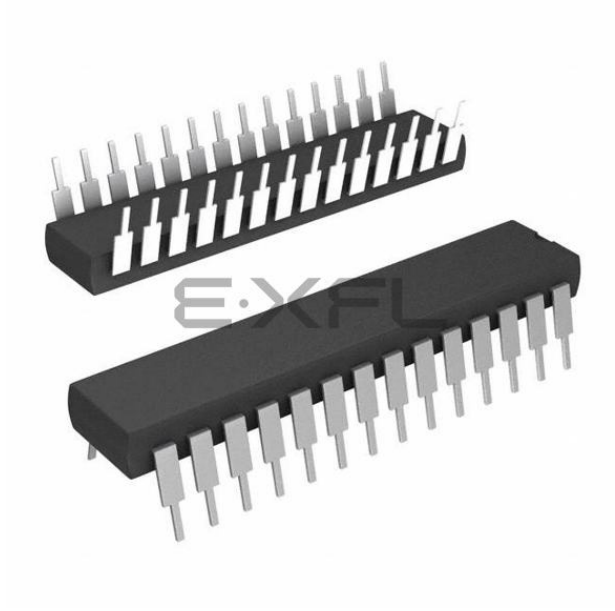


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	ECANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f25k80-i-sp

Table of Contents

2.6 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 3.0 "Oscillator Configurations"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Layout suggestions are shown in Figure 2-4. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins, and other signals in close proximity to the oscillator, are benign (i.e., free of high frequencies, short rise and fall times, and other similar noise).

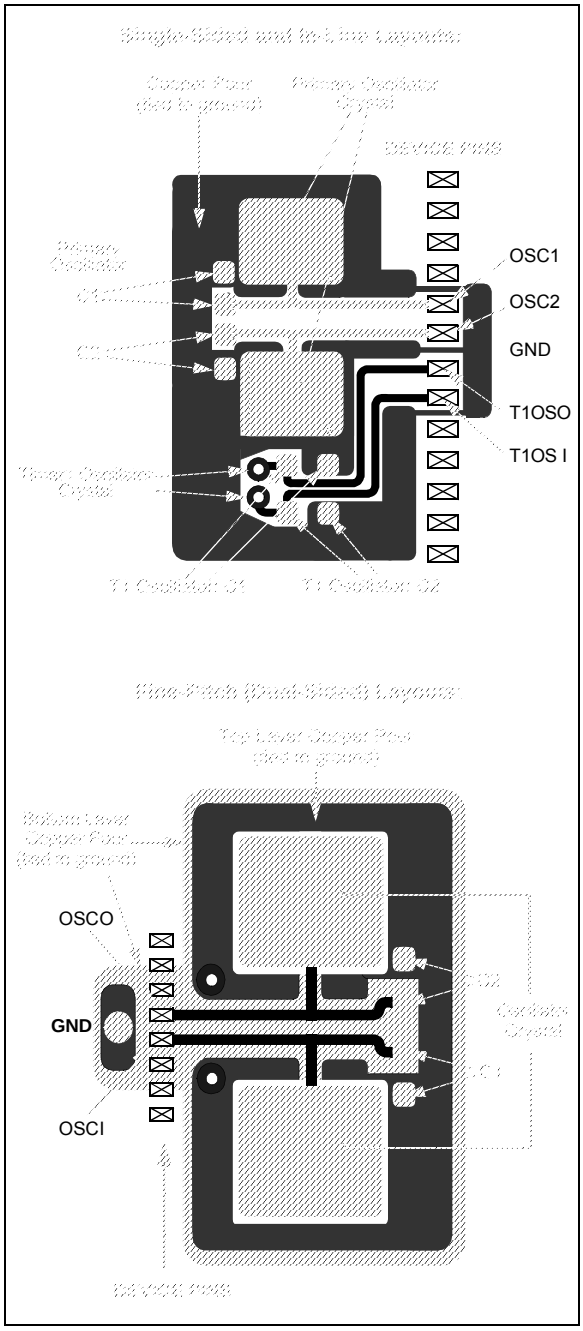
For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the corporate web site (www.microchip.com):

- *AN826,* "*Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices"*
- *AN849, "Basic PICmicro® Oscillator Design"*
- *AN943, "Practical PICmicro® Oscillator Analysis and Design"*
- *AN949, "Making Your Oscillator Work"*

2.7 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 kΩ to 10 kΩ resistor to VSS on unused pins and drive the output to logic low.

FIGURE 2-5: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT



REGISTER 5-1: RCON: RESET CONTROL REGISTER

R/W-0	R/W-1(1)	R/W-1	R/W-1	R-1	R-1	R/W-0(2)	R/W-0
IPEN	SBOREN	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **IPEN:** Interrupt Priority Enable bit
1 = Enables priority levels on interrupts
0 = Disables priority levels on interrupts (PIC16CXXX Compatibility mode)

bit 6 **SBOREN:** BOR Software Enable bit(1)
If BOREN<1:0> = 01:
1 = BOR is enabled
0 = BOR is disabled
If BOREN<1:0> = 00, 10 or 11:
Bit is disabled and reads as '0'.

bit 5 **$\overline{CM}$:** Configuration Mismatch Flag bit
1 = A Configuration Mismatch Reset has not occurred.
0 = A Configuration Mismatch Reset has occurred (must be set in software once the Reset occurs)

bit 4 **$\overline{RI}$:** RESET Instruction Flag bit
1 = The RESET instruction was not executed (set by firmware only)
0 = The RESET instruction was executed causing a device Reset (must be set in software after a Brown-out Reset occurs)

bit 3 **$\overline{TO}$:** Watchdog Time-out Flag bit
1 = Set by power-up, CLRWDT instruction or SLEEP instruction
0 = A WDT time-out has occurred

bit 2 **$\overline{PD}$:** Power-down Detection Flag bit
1 = Set by power-up or by the CLRWDT instruction
0 = Set by execution of the SLEEP instruction

bit 1 **$\overline{POR}$:** Power-on Reset Status bit(2)
1 = A Power-on Reset has not occurred (set by firmware only)
0 = A Power-on Reset has occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
1 = A Brown-out Reset has not occurred (set by firmware only)
0 = A Brown-out Reset has occurred (must be set in software after a Brown-out Reset occurs)

Note 1: If SBOREN is enabled, its Reset state is '1'; otherwise, it is '0'.
2: The actual Reset value of $\overline{POR}$ is determined by the type of device Reset. See the notes following this register and **Section 5.7 "Reset State of Registers"** for additional information.

Note 1: It is recommended that the $\overline{POR}$ bit be set after a Power-on Reset has been detected so that subsequent Power-on Resets may be detected.
2: Brown-out Reset is said to have occurred when $\overline{BOR}$ is '0' and $\overline{POR}$ is '1' (assuming that $\overline{POR}$ was set to '1' by software immediately after a Power-on Reset).

TABLE 5-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices			Power-on Reset, Brown-out Reset	MCLR Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
BAUDCON2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	01x0 0-00	01x0 0-00	uuuu u-uu
IPR4	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 -111	1111 -111	uuuu -uuu
PIR4	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 -000	0000 -000	uuuu -uuu
PIE4	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 -000	0000 -000	uuuu -uuu
CVRCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
CMSTAT	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xx-- ----	xx-- ----	uu-- ----
TMR3H	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
T3GCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0x00	0000 0x00	uuuu u-uu
SPBRG1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
RCREG1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
TXREG1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
TXSTA1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0010	0000 0010	uuuu uuuu
RCSTA1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 000x	0000 000x	uuuu uuuu
T1GCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0x00	0000 0x00	uuuu u-uu
PR4	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 1111	1111 1111	uuuu uuuu
HLVDCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
BAUDCON1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	01x0 0-00	01x0 0-00	uuuu u-uu
RCSTA2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 000x	0000 000x	uuuu uuuu
IPR3	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	--11 111-	--11 111-	--uu uuu-
PIR3	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	--00 000-	--x0 xxx-	--uu uuu-
PIE3	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	--00 000-	0000 0000	uuuu uuuu
IPR2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1--- 1111	1--- 111x	u--- uuuu
PIR2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0--- 0000	0--- 000x	u--- uuuu(1)
PIE2	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0--- 0000	0--- 0000	u--- uuuu
IPR1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	1111 1111	1111 1111	uuuu uuuu
	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	-111 1111	-111 1111	-uuu uuuu
PIR1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu(1)
	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	-000 0000	-000 0000	-uuu uuuu
PIE1	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	-000 0000	-000 0000	-uuu uuuu
PSTR1CON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	00-0 0001	xx-x xxxx	—
OSCTUNE	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0000 0000	0000 0000	uuuu uuuu
REFOCON	PIC18F2XK80	PIC18F4XK80	PIC18F6XK80	0-00 0000	0-00 0000	u-uu uuuu

Legend: u = unchanged; x = unknown; - = unimplemented bit, read as '0'; q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 5-3 for Reset value for specific conditions.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read as '0'.

TABLE 6-2: PIC18F66K80 FAMILY REGISTER FILE SUMMARY (CONTINUED)

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR on page
FCFh	TMR1H	Timer1 Register High Byte								89
FCEh	TMR1L	Timer1 Register Low Bytes								89
FCDh	T1CON	TMR1CS1	TMR1CS0	T1CKPS1	T1CKPS0	SOSCEN	$\overline{\text{T1SYNC}}$	RD16	TMR1ON	89
FCCh	TMR2	Timer2 Register								89
FCBh	PR2	Timer2 Period Register								89
FCAh	T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	89
FC9h	SSPBUF	MSSP Receive Buffer/Transmit Register								89
FC8h	SSPADD	MSSP Address Register (I^2C™ Slave Mode), MSSP Baud Rate Reload Register (I^2C Master Mode)								89
FC8h	SSPMSK	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	89
FC7h	SSPSTAT	SMP	CKE	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF	89
FC6h	SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	89
FC5h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	89
FC4h	ADRESH	A/D Result Register High Byte								89
FC3h	ADRESL	A/D Result Register Low Byte								89
FC2h	ADCON0	—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	89
FC1h	ADCON1	TRIGSEL1	TRIGSEL0	VCFG1	VCFG0	VNCFG	CHSN2	CHSN1	CHSN0	89
FC0h	ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	89
FBFh	ECCP1AS	ECCP1ASE	ECCP1AS2	ECCP1AS1	ECCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0	89
FBEh	ECCP1DEL	P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0	89
FBDh	CCPR1H	Capture/Compare/PWM Register 1 High Byte								89
FBCh	CCPR1L	Capture/Compare/PWM Register 1 Low Byte								89
FBBh	CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	89
FBAh	TXSTA2	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	89
FB9h	BAUDCON2	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	89
FB8h	IPR4	TMR4IP	EEIP	CMP2IP	CMP1IP	—	CCP5IP	CCP4IP	CCP3IP	89
FB7h	PIR4	TMR4IF	EEIF	CMP2IF	CMP1IF	—	CCP5IF	CCP4IF	CCP3IF	89
FB6h	PIE4	TMR4IE	EEIE	CMP2IE	CMP1IE	—	CCP5IE	CCP4IE	CCP3IE	89
FB5h	CVRCON	CVREN	CVROE	CVRSS	CVR4	CVR3	CVR2	CVR1	CVR0	89
FB4h	CMSTAT	CMP2OUT	CMP1OUT	—	—	—	—	—	—	89
FB3h	TMR3H	Timer3 Register High Byte								89
FB2h	TMR3L	Timer3 Register Low Bytes								89
FB1h	T3CON	TMR3CS1	TMR3CS0	T3CKPS1	T3CKPS0	SOSCEN	$\overline{\text{T3SYNC}}$	RD16	TMR3ON	89
FB0h	T3GCON	TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{\text{T3DONE}}$	T3GVAL	T3GSS1	T3GSS0	89
FAFh	SPBRG1	EUSART1 Baud Rate Generator Register Low Byte								89
FAEh	RCREG1	EUSART1 Receive Register								89
FADh	TXREG1	EUSART1 Transmit Register								89
FACh	TXSTA1	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	89
FABh	RCSTA1	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	89
FAAh	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{T1DONE}}$	T1GVAL	T1GSS1	T1GSS0	89
FA9h	PR4	Timer4 Period Register								89
FA8h	HLVDCON	VDIRMAG	BGVST	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0	89
FA7h	BAUDCON1	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	89
FA6h	RCSTA2	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	89
FA5h	IPR3	—	—	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP	—	89
FA4h	PIR3	—	—	RC2IF	TX2IF	CTMUIF	CCP2IF	CCP1IF	—	89
FA3h	PIE3	—	—	RC2IE	TX2IE	CTMUIE	CCP2IE	CCP1IE	—	89
FA2h	IPR2	OSCFIP	—	—	—	BCLIP	HLVDIP	TMR3IP	TMR3GIP	89
FA1h	PIR2	OSCFIF	—	—	—	BCLIF	HLVDIF	TMR3IF	TMR3GIF	89
FA0h	PIE2	OSCFIE	—	—	—	BCLIE	HLVDIE	TMR3IE	TMR3GIE	89

EXAMPLE 8-1: DATA EEPROM READ

```
        MOVLW   DATA_EE_ADDRH      ;
        MOVWF   EEADRH             ; Upper bits of Data Memory Address to read
        MOVLW   DATA_EE_ADDR       ;
        MOVWF   EEADR              ; Lower bits of Data Memory Address to read
        BCF     EECON1, EEPGD      ; Point to DATA memory
        BCF     EECON1, CFGS       ; Access EEPROM
        BSF     EECON1, RD         ; EEPROM Read
        NOP
        MOVF    EEDATA, W          ; W = EEDATA
```

EXAMPLE 8-2: DATA EEPROM WRITE

```
                MOVLW   DATA_EE_ADDRH      ;
                MOVWF   EEADRH             ; Upper bits of Data Memory Address to write
                MOVLW   DATA_EE_ADDR       ;
                MOVWF   EEADR              ; Lower bits of Data Memory Address to write
                MOVLW   DATA_EE_DATA       ;
                MOVWF   EEDATA             ; Data Memory Value to write
                BCF     EECON1, EEPGD      ; Point to DATA memory
                BCF     EECON1, CFGS       ; Access EEPROM
                BSF     EECON1, WREN       ; Enable writes

                BCF     INTCON, GIE        ; Disable Interrupts
                MOVLW   55h                ;
Required        MOVWF   EECON2             ; Write 55h
Sequence        MOVLW   0AAh               ;
                MOVWF   EECON2             ; Write 0AAh
                BSF     EECON1, WR         ; Set WR bit to begin write
                BTFSC   EECON1, WR         ; Wait for write to complete GOTO $-2
                BSF     INTCON, GIE        ; Enable Interrupts

                                           ; User code execution
                BCF     EECON1, WREN       ; Disable writes on write complete (EEIF set)
```

REGISTER 10-15: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

R/W-1	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
OSCFIP	—	—	—	BCLIP	HLVDIP	TMR3IP	TMR3GIP
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **OSCFIP:** Oscillator Fail Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6-4 **Unimplemented:** Read as ‘0’

bit 3 **BCLIP:** Bus Collision Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **HLVDIP:** High/Low-Voltage Detect Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR3IP:** TMR3 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR3GIP:** TMR3 Gate Interrupt Priority bit
1 = High priority
0 = Low priority

For more details on selecting the optimum C1 and C2 for a given crystal, see the crystal manufacture's applications information. The optimum value depends in part on the amount of parasitic capacitance in the circuit, which is often unknown. For that reason, it is highly recommended that thorough testing and validation of the oscillator be performed after values have been selected.

14.5.1 USING SOSC AS A CLOCK SOURCE

The SOSC oscillator is also available as a clock source in power-managed modes. By setting the clock select bits, SCS<1:0> (OSCCON<1:0>), to '01', the device switches to SEC_RUN mode and both the CPU and peripherals are clocked from the SOSC oscillator. If the IDLEN bit (OSCCON<7>) is cleared and a SLEEP instruction is executed, the device enters SEC_IDLE mode. Additional details are available in **Section 4.0 "Power-Managed Modes"**.

Whenever the SOSC oscillator is providing the clock source, the SOSC System Clock Status flag, SOSCRUN (OSCCON2<6>), is set. This can be used to determine the controller's current clocking mode. It can also indicate the clock source currently being used by the Fail-Safe Clock Monitor.

If the Clock Monitor is enabled and the SOSC oscillator fails while providing the clock, polling the SOCSRUN bit will indicate whether the clock is being provided by the SOSC oscillator or another source.

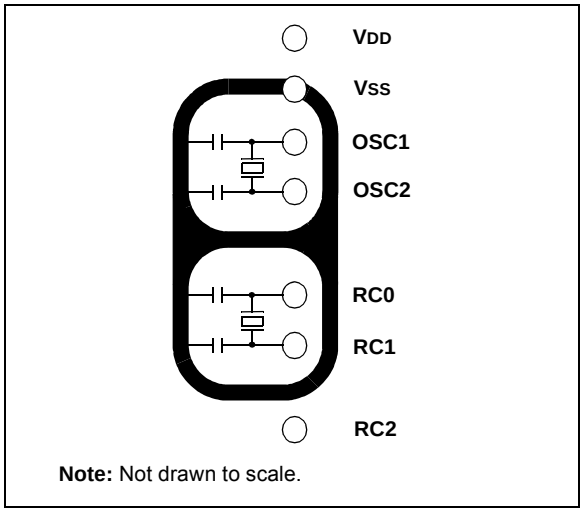
14.5.2 SOSC OSCILLATOR LAYOUT CONSIDERATIONS

The SOSC oscillator circuit draws very little power during operation. Due to the low-power nature of the oscillator, it may also be sensitive to rapidly changing signals in close proximity. This is especially true when the oscillator is configured for extremely Low-Power mode, SOSCSEL<1:0> (CONFIG1L<4:3>) = 01.

The oscillator circuit, displayed in Figure 14-2, should be located as close as possible to the microcontroller. There should be no circuits passing within the oscillator circuit boundaries other than VSS or VDD.

If a high-speed circuit must be located near the oscillator, it may help to have a grounded guard ring around the oscillator circuit. The guard, as displayed in Figure 14-3, could be used on a single-sided PCB or in addition to a ground plane. (Examples of a high-speed circuit include the ECCP1 pin, in Output Compare or PWM mode, or the primary oscillator, using the OSC2 pin.)

FIGURE 14-3: OSCILLATOR CIRCUIT WITH GROUNDED GUARD RING



Note: Not drawn to scale.

In the Low Drive Level mode, SOSCSEL<1:0> = 01, it is critical that RC2 I/O pin signals be kept away from the oscillator circuit. Configuring RC2 as a digital output, and toggling it, can potentially disturb the oscillator circuit, even with a relatively good PCB layout. If possible, either leave RC2 unused or use it as an input pin with a slew rate limited signal source. If RC2 must be used as a digital output, it may be necessary to use the Higher Drive Level Oscillator mode (SOSCSEL<1:0> = 11) with many PCB layouts.

Even in the Higher Drive Level mode, careful layout procedures should still be followed when designing the oscillator circuit.

In addition to dV/dt induced noise considerations, it is important to ensure that the circuit board is clean. Even a very small amount of conductive, soldering flux residue can cause PCB leakage currents that can overwhelm the oscillator circuit.

14.6 Timer1 Interrupt

The TMR1 register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The Timer1 interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit, TMR1IF (PIR1<0>). This interrupt can be enabled or disabled by setting or clearing the Timer1 Interrupt Enable bit, TMR1IE (PIE1<0>).

16.1 Timer3 Gate Control Register

The Timer3 Gate Control register (T3GCON), provided in Register 14-2, is used to control the Timer3 gate.

REGISTER 16-2: T3GCON: TIMER3 GATE CONTROL REGISTER(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-x	R/W-0	R/W-0
TMR3GE	T3GPOL	T3GTM	T3GSPM	T3GGO/$\overline{\text{T3DONE}}$	T3GVAL	T3GSS1	T3GSS0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **TMR3GE:** Timer3 Gate Enable bit

If TMR3ON = 0:
This bit is ignored.

If TMR3ON = 1:
1 = Timer3 counting is controlled by the Timer3 gate function
0 = Timer3 counts regardless of Timer3 gate function

bit 6 **T3GPOL:** Timer3 Gate Polarity bit

1 = Timer3 gate is active-high (Timer3 counts when gate is high)
0 = Timer3 gate is active-low (Timer3 counts when gate is low)

bit 5 **T3GTM:** Timer3 Gate Toggle Mode bit

1 = Timer3 Gate Toggle mode is enabled.
0 = Timer3 Gate Toggle mode is disabled and toggle flip-flop is cleared
Timer3 gate flip-flop toggles on every rising edge.

bit 4 **T3GSPM:** Timerx Gate Single Pulse Mode bit

1 = Timer3 Gate Single Pulse mode is enabled and is controlling Timer3 gate
0 = Timer3 Gate Single Pulse mode is disabled

bit 3 **T3GGO/$\overline{\text{T3DONE}}$:** Timer3 Gate Single Pulse Acquisition Status bit

1 = Timer3 Gate Single Pulse mode acquisition is ready, waiting for an edge
0 = Timer3 Gate Single Pulse mode acquisition has completed or has not been started
This bit is automatically cleared when T3GSPM is cleared.

bit 2 **T3GVAL:** Timer3 Gate Current State bit

Indicates the current state of the Timerx gate that could be provided to TMR3H:TMR3L. Unaffected by Timerx Gate Enable (TMR3GE) bit.

bit 1-0 **T3GSS<1:0>:** Timer3 Gate Source Select bits

11 = Comparator 2 output
10 = Comparator 1 output
01 = TMR4 to match PR4 output
00 = Timer3 gate pin
Watchdog Timer oscillator is turned on if TMR3GE = 1, regardless of the state of TMR3ON.

Note 1: Programming the T3GCON prior to T3CON is recommended.

REGISTER 16-3: OSCCON2: OSCILLATOR CONTROL REGISTER 2

U-0	R-0	U-0	RW-1	R/W-0	U-0	R-x	R/W-0
—	SOSCRUN	—	SOSCDRV(1)	SOSCGO	—	MFIOFS	MFIOSEL
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6 **SOSCRUN:** SOSC Run Status bit
- 1 = System clock comes from a secondary SOSC
- 0 = System clock comes from an oscillator other than SOSC

bit 5 **Unimplemented:** Read as '0'

bit 4 **SOSCDRV:** Secondary Oscillator Drive Control bit(1)
- 1 = High-power SOSC circuit selected
- 0 = Low/high-power select is done via the SOSCSEL<1:0> Configuration bits

bit 3 **SOSCGO:** Oscillator Start Control bit
- 1 = Oscillator is running even if no other sources are requesting it
- 0 = Oscillator is shut off if no other sources are requesting it (When the SOSC is selected to run from a digital clock input, rather than an external crystal, this bit has no effect.)

bit 2 **Unimplemented:** Read as '0'

bit 1 **MFIOFS:** MF-INTOSC Frequency Stable bit
- 1 = MF-INTOSC is stable
- 0 = MF-INTOSC is not stable

bit 0 **MFIOSEL:** MF-INTOSC Select bit
- 1 = MF-INTOSC is used in place of HF-INTOSC frequencies of 500 kHz, 250 kHz and 31.25 kHz
- 0 = MF-INTOSC is not used

Note 1: When SOSC is selected to run from a digital clock input, rather than an external crystal, this bit has no effect.

16.5.2 TIMER3 GATE SOURCE SELECTION

The Timer3 gate source can be selected from one of four different sources. Source selection is controlled by the T3GSS<1:0> bits (T3GCON<1:0>). The polarity for each available source is also selectable and is controlled by the T3GPOL bit (T3GCON<6>).

TABLE 16-2: TIMER3 GATE SOURCES

T3GSS<1:0>	Timer3 Gate Source
00	Timerx Gate Pin
01	TMR4 to Match PR4 (TMR4 increments to match PR4)
10	Comparator 1 Output (comparator logic high output)
11	Comparator 2 Output (comparator logic high output)

16.5.2.1 T3G Pin Gate Operation

The T3G pin is one source for Timer3 gate control. It can be used to supply an external source to the Timerx gate circuitry.

16.5.2.2 Timer4 Match Gate Operation

The TMR4 register will increment until it matches the value in the PR4 register. On the very next increment cycle, TMR4 will be reset to 00h. When this Reset occurs, a low-to-high pulse will automatically be generated and internally supplied to the Timerx gate circuitry. The pulse will remain high for one instruction cycle and will return back to a low state until the next match.

Depending on T3GPOL, Timerx increments differently when TMR4 matches PR4. When T3GPOL = 1, Timer3 increments for a single instruction cycle following a TMR4 match with PR4. When T3GPOL = 0, Timer3 increments continuously, except for the cycle following the match, when the gate signal goes from low-to-high.

16.5.2.3 Comparator 1 Output Gate Operation

The output of Comparator 1 can be internally supplied to the Timer3 gate circuitry. After setting up Comparator 1 with the CM1CON register, Timer3 will increment depending on the transitions of the CMP1OUT (CMSTAT<6>) bit.

16.5.2.4 Comparator 2 Output Gate Operation

The output of Comparator 2 can be internally supplied to the Timer3 gate circuitry. After setting up Comparator 2 with the CM2CON register, Timer3 will increment depending on the transitions of the CMP2OUT (CMSTAT<7>) bit.

16.5.3 TIMER3 GATE TOGGLE MODE

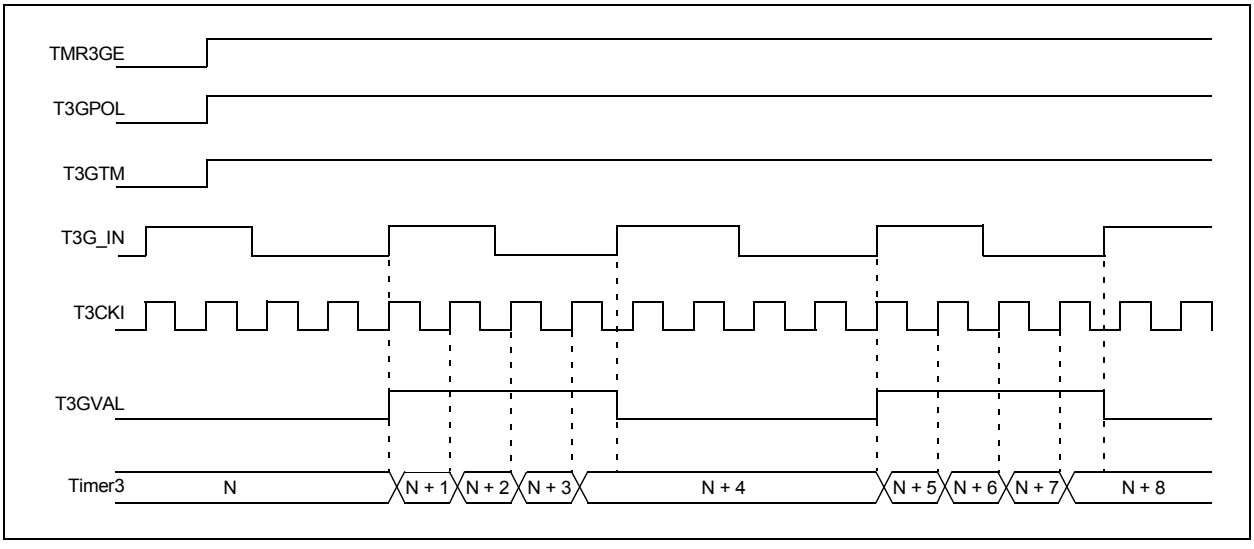
When Timer3 Gate Toggle mode is enabled, it is possible to measure the full cycle length of a Timer3 gate signal, as opposed to the duration of a single level pulse.

The Timer3 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. (For timing details, see Figure 16-3.)

The T3GVAL bit will indicate when the Toggled mode is active and the timer is counting.

Timer3 Gate Toggle mode is enabled by setting the T3GTM bit (T3GCON<5>). When the T3GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

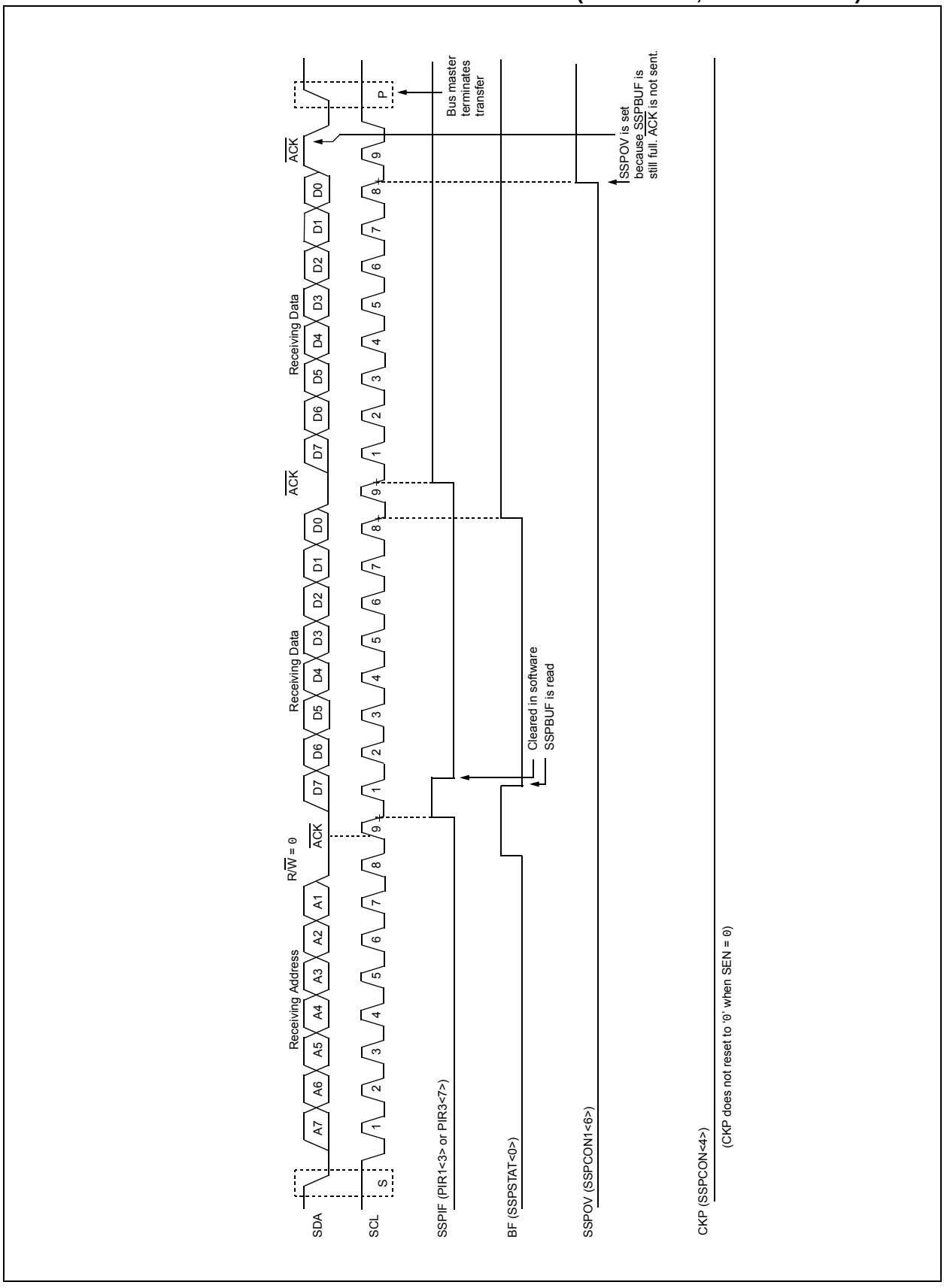
FIGURE 16-3: TIMER3 GATE TOGGLE MODE



EXAMPLE 18-2: CURRENT CALIBRATION ROUTINE

```
#include "p18cxxx.h"

#define COUNT 500                       //@ 8MHz = 125uS.
#define DELAY for(i=0;i<COUNT;i++)
#define RCAL .027                       //R value is 4200000 (4.2M)
                                        //scaled so that result is in
                                        //1/100th of uA
#define ADSCALE 1023                    //for unsigned conversion 10 sig bits
#define ADREF 3.3                       //Vdd connected to A/D Vr+

int main(void)
{
    int i;
    int j = 0; //index for loop
    unsigned int Vread = 0;
    double VTot = 0;
    float Vavg=0, Vcal=0, CTMUISrc = 0;  //float values stored for calcs

//assume CTMU and A/D have been setup correctly
//see Example 25-1 for CTMU & A/D setup
setup();

CTMUCONHbits.CTMUEN = 1;                //Enable the CTMU
    for(j=0;j<10;j++)
    {
        CTMUCONHbits.IDISSEN = 1;       //drain charge on the circuit
        DELAY;                          //wait 125us
        CTMUCONHbits.IDISSEN = 0;       //end drain of circuit

        CTMUCONLbits.EDG1STAT = 1;      //Begin charging the circuit
                                        //using CTMU current source
        DELAY;                          //wait for 125us
        CTMUCONLbits.EDG1STAT = 0;      //Stop charging circuit

        PIR1bits.ADIF = 0;              //make sure A/D Int not set
        ADCON0bits.GO=1;                //and begin A/D conv.
        while(!PIR1bits.ADIF);          //Wait for A/D convert complete

        Vread = ADRES;                  //Get the value from the A/D
        PIR1bits.ADIF = 0;              //Clear A/D Interrupt Flag
        VTot += Vread;                  //Add the reading to the total
    }

    Vavg = (float)(VTot/10.000);        //Average of 10 readings
    Vcal = (float)(Vavg/ADSCALE*ADREF);
    CTMUISrc = Vcal/RCAL;               //CTMUISrc is in 1/100ths of uA

}
```

FIGURE 21-8: I²C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 7-BIT ADDRESS)



21.4.7 BAUD RATE

In I^2C Master mode, the Baud Rate Generator (BRG) reload value is placed in the 8 bits of the SSPADD register (Figure 21-19). When a write occurs to SSPBUF, the Baud Rate Generator will automatically begin counting. The BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TCY) on the Q2 and Q4 clocks. In I^2C Master mode, the BRG is reloaded automatically.

Once the given operation is complete (i.e., transmission of the last data bit is followed by $\overline{ACK}$), the internal clock will automatically stop counting and the SCL pin will remain in its last state.

Table 21-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD. The SSPADD BRG value of 00h is not supported.

FIGURE 21-19: BAUD RATE GENERATOR BLOCK DIAGRAM

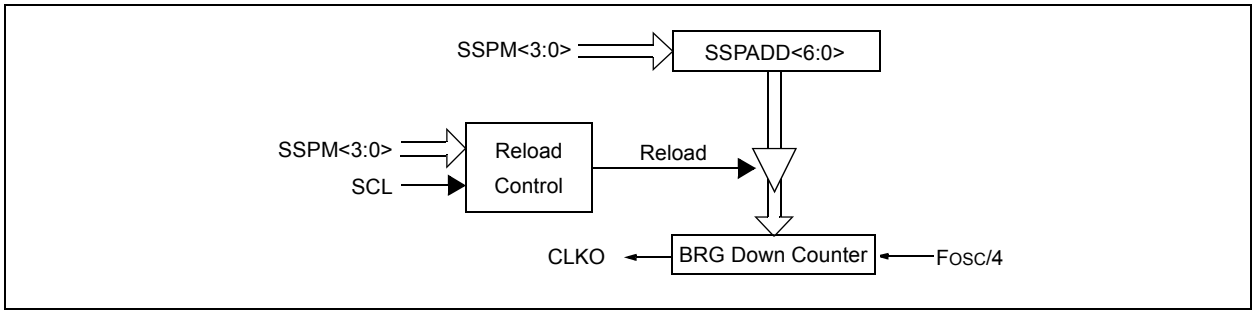


TABLE 21-3: I^2C™ CLOCK RATE w/BRG

FOSC	FCY	FCY * 2	BRG Value	FSCL (2 Rollovers of BRG)
40 MHz	10 MHz	20 MHz	18h	400 kHz(1)
40 MHz	10 MHz	20 MHz	1Fh	312.5 kHz
40 MHz	10 MHz	20 MHz	63h	100 kHz
16 MHz	4 MHz	8 MHz	09h	400 kHz(1)
16 MHz	4 MHz	8 MHz	0Ch	308 kHz
16 MHz	4 MHz	8 MHz	27h	100 kHz
4 MHz	1 MHz	2 MHz	02h	333 kHz(1)
4 MHz	1 MHz	2 MHz	09h	100 kHz
16 MHz(2)	4 MHz	8 MHz	03h	1 MHz(1)

Note 1: The I^2C interface does not conform to the 400 kHz I^2C specification (which applies to rates greater than 100 kHz) in all details, but may be used with care where higher rates are required by the application.

2: A minimum 16-MHz FOSC is required for 1 MHz I^2C.

REGISTER 24-1: CMxCON: COMPARATOR CONTROL x REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
CON	COE	CPOL	EVPOL1	EVPOL0	CREF	CCH1	CCH0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **CON:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled

bit 6 **COE:** Comparator Output Enable bit
1 = Comparator output is present on the CxOUT pin
0 = Comparator output is internal only

bit 5 **CPOL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

bit 4-3 **EVPOL<1:0>:** Interrupt Polarity Select bits
11 = Interrupt generation on any change of the output(1)
10 = Interrupt generation only on high-to-low transition of the output
01 = Interrupt generation only on low-to-high transition of the output
00 = Interrupt generation is disabled

bit 2 **CREF:** Comparator Reference Select bit (non-inverting input)
1 = Non-inverting input connects to internal CV_{REF} voltage
0 = Non-inverting input connects to CxINA pin

bit 1-0 **CCH<1:0>:** Comparator Channel Select bits
11 = Inverting input of comparator connects to V_{BG}
10 = Inverting input of comparator connects to C2INB pin(2)
01 = Inverting input of comparator connects to CxINC pin
00 = Inverting input of comparator connects to C1INB pin(2)

Note 1: The CMPxIF is automatically set any time this mode is selected and must be cleared by the application after the initial configuration.

2: Comparator 1 uses C2INB as an input to the inverting terminal. Comparator 2 uses C1INB as an input to the inverted terminal.

REGISTER 27-45: RXFCONn: RECEIVE FILTER CONTROL REGISTER 'n' [0 ≤ n ≤ 1](1)

RXFCON0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	RXF7EN	RXF6EN	RXF5EN	RXF4EN	RXF3EN	RXF2EN	RXF1EN	RXF0EN

RXFCON1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	RXF15EN	RXF14EN	RXF13EN	RXF12EN	RXF11EN	RXF10EN	RXF9EN	RXF8EN
	bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-0 **RXF<7:0>EN:** Receive Filter n Enable bits
0 = Filter is disabled
1 = Filter is enabled

Note 1: This register is available in Mode 1 and 2 only.

Note: Register 27-46 through Register 27-51 are writable in Configuration mode only.

REGISTER 27-46: SDFLC: STANDARD DATA BYTES FILTER LENGTH COUNT REGISTER(1)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	FLC4	FLC3	FLC2	FLC1	FLC0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **FLC<4:0>:** Filter Length Count bits

Mode 0:

Not used; forced to '00000'.

00000-10010 = 0 18 bits are available for standard data byte filter. Actual number of bits used depends on the DLC<3:0> bits (RXBnDLC<3:0> or BnDLC<3:0> if configured as RX buffer) of the message being received.

If DLC<3:0> = 0000 No bits will be compared with incoming data bits.

If DLC<3:0> = 0001 Up to 8 data bits of RXFnEID<7:0>, as determined by FLC<2:0>, will be compared with the corresponding number of data bits of the incoming message.

If DLC<3:0> = 0010 Up to 16 data bits of RXFnEID<15:0>, as determined by FLC<3:0>, will be compared with the corresponding number of data bits of the incoming message.

If DLC<3:0> = 0011 Up to 18 data bits of RXFnEID<17:0>, as determined by FLC<4:0>, will be compared with the corresponding number of data bits of the incoming message.

Note 1: This register is available in Mode 1 and 2 only.

27.4.3 MODE 2 – ENHANCED FIFO MODE

In Mode 2, two or more receive buffers are used to form the receive FIFO (first in, first out) buffer. There is no one-to-one relationship between the receive buffer and acceptance filter registers. Any filter that is enabled and linked to any FIFO receive buffer can generate acceptance and cause FIFO to be updated.

FIFO length is user-programmable, from 2-8 buffers deep. FIFO length is determined by the very first programmable buffer that is configured as a transmit buffer. For example, if Buffer 2 (B2) is programmed as a transmit buffer, FIFO consists of RXB0, RXB1, B0 and B1, creating a FIFO length of 4. If all programmable buffers are configured as receive buffers, FIFO will have the maximum length of 8.

The following is the list of resources available in Mode 2:

- Three transmit buffers: TXB0, TXB1 and TXB2
- Two receive buffers: RXB0 and RXB1
- Six buffers programmable as TX or RX; receive buffers form FIFO: B0-B5
- Automatic RTR handling on B0-B5
- Sixteen acceptance filters: RXF0-RXF15
- Two dedicated acceptance mask registers; RXF15 programmable as third mask: RXM0-RXM1, RXF15
- Programmable data filter on standard identifier messages: SDFLC, useful for DeviceNet protocol

27.5 CAN Message Buffers

27.5.1 DEDICATED TRANSMIT BUFFERS

The PIC18F66K80 family devices implement three dedicated transmit buffers – TXB0, TXB1 and TXB2. Each of these buffers occupies 14 bytes of SRAM and are mapped into the SFR memory map. These are the only transmit buffers available in Mode 0. Mode 1 and 2 may access these and other additional buffers.

Each transmit buffer contains one Control register (TXBnCON), four Identifier registers (TXBnSIDL, TXBnSIDH, TXBnEIDL, TXBnEIDH), one Data Length Count register (TXBnDLC) and eight Data Byte registers (TXBnDm).

27.5.2 DEDICATED RECEIVE BUFFERS

The PIC18F66K80 family devices implement two dedicated receive buffers: RXB0 and RXB1. Each of these buffers occupies 14 bytes of SRAM and are mapped into SFR memory map. These are the only receive buffers available in Mode 0. Mode 1 and 2 may access these and other additional buffers.

Each receive buffer contains one Control register (RXBnCON), four Identifier registers (RXBnSIDL, RXBnSIDH, RXBnEIDL, RXBnEIDH), one Data Length Count register (RXBnDLC) and eight Data Byte registers (RXBnDm).

There is also a separate Message Assembly Buffer (MAB) which acts as an additional receive buffer. MAB is always committed to receiving the next message from the bus and is not directly accessible to user firmware. The MAB assembles all incoming messages one by one. A message is transferred to appropriate receive buffers only if the corresponding acceptance filter criteria is met.

27.5.3 PROGRAMMABLE TRANSMIT/ RECEIVE BUFFERS

The ECAN module implements six new buffers: B0-B5. These buffers are individually programmable as either transmit or receive buffers. These buffers are available only in Mode 1 and 2. As with dedicated transmit and receive buffers, each of these programmable buffers occupies 14 bytes of SRAM and are mapped into SFR memory map.

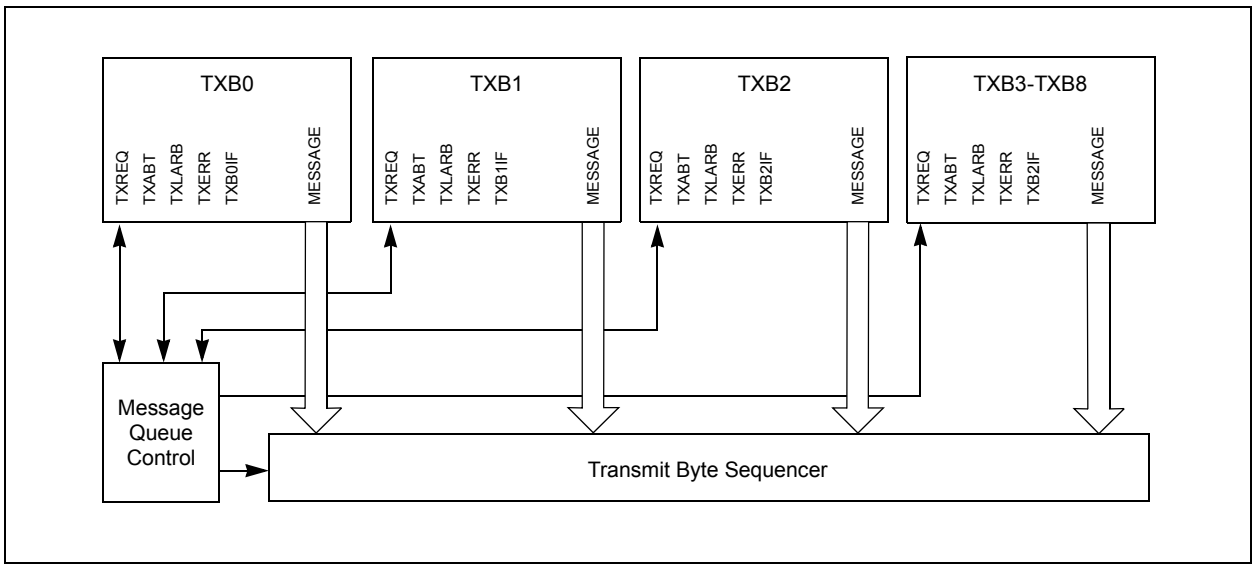
Each buffer contains one Control register (BnCON), four Identifier registers (BnSIDL, BnSIDH, BnEIDL, BnEIDH), one Data Length Count register (BnDLC) and eight Data Byte registers (BnDm). Each of these registers contains two sets of control bits. Depending on whether the buffer is configured as transmit or receive, one would use the corresponding control bit set. By default, all buffers are configured as receive buffers. Each buffer can be individually configured as a transmit or receive buffer by setting the corresponding TXENn bit in the BSEL0 register.

When configured as transmit buffers, user firmware may access transmit buffers in any order similar to accessing dedicated transmit buffers. In receive configuration with Mode 1 enabled, user firmware may also access receive buffers in any order required. But in Mode 2, all receive buffers are combined to form a single FIFO. Actual FIFO length is programmable by user firmware. Access to FIFO must be done through the FIFO Pointer bits (FP<4:0>) in the CANCON register. It must be noted that there is no hardware protection against out of order FIFO reads.

27.6.3 TRANSMIT PRIORITY

Transmit priority is a prioritization within the PIC18F66K80 family devices of the pending transmittable messages. This is independent from, and not related to, any prioritization implicit in the message arbitration scheme built into the CAN protocol. Prior to sending the Start-of-Frame (SOF), the priority of all buffers that are queued for transmission is compared. The transmit buffer with the highest priority will be sent first. If two buffers have the same priority setting, the buffer with the highest buffer number will be sent first. There are four levels of transmit priority. If the TXP bits for a particular message buffer are set to '11', that buffer has the highest possible priority. If the TXP bits for a particular message buffer are set to '00', that buffer has the lowest possible priority.

FIGURE 27-2: TRANSMIT BUFFERS



31.6.3 TIMING DIAGRAMS AND SPECIFICATIONS

FIGURE 31-4: EXTERNAL CLOCK TIMING

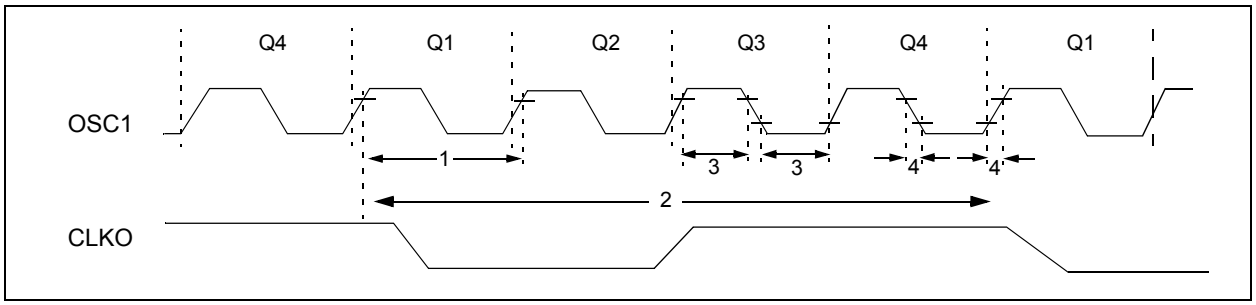


TABLE 31-6: EXTERNAL CLOCK TIMING REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
1A	FOSC	External CLKIN Frequency(1)	DC	64	MHz	EC, ECIO Oscillator mode
		Oscillator Frequency(1)	DC	4	MHz	RC Oscillator mode
			0.1	4	MHz	XT Oscillator mode
			4	16	MHz	HS Oscillator mode
			4	16	MHz	HS + PLL Oscillator mode
			5	33	kHz	LP Oscillator mode
1	TOSC	External CLKIN Period(1)	15.6	—	ns	EC, ECIO Oscillator mode
		Oscillator Period(1)	250	—	ns	RC Oscillator mode
			250	10,000	ns	XT Oscillator mode
			40	250	ns	HS Oscillator mode
			62.5	250	ns	HS + PLL Oscillator mode
			5	200	µs	LP Oscillator mode
2	TCY	Instruction Cycle Time(1)	62.5	—	ns	TCY = 4/FOSC
3	TOSL, TOSH	External Clock in (OSC1) High or Low Time	30	—	ns	XT Oscillator mode
			2.5	—	µs	LP Oscillator mode
			10	—	ns	HS Oscillator mode
4	TOSR, TOSF	External Clock in (OSC1) Rise or Fall Time	—	20	ns	XT Oscillator mode
			—	50	ns	LP Oscillator mode
			—	7.5	ns	HS Oscillator mode

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time base period for all configurations except PLL. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min.” values with an external clock applied to the OSC1/CLKIN pin. When an external clock input is used, the “max.” cycle time limit is “DC” (no clock) for all devices.