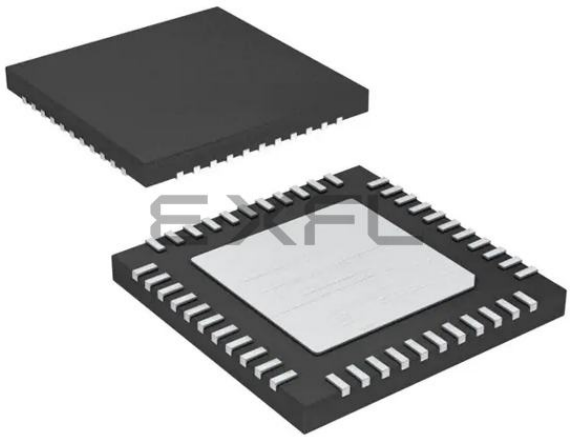


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

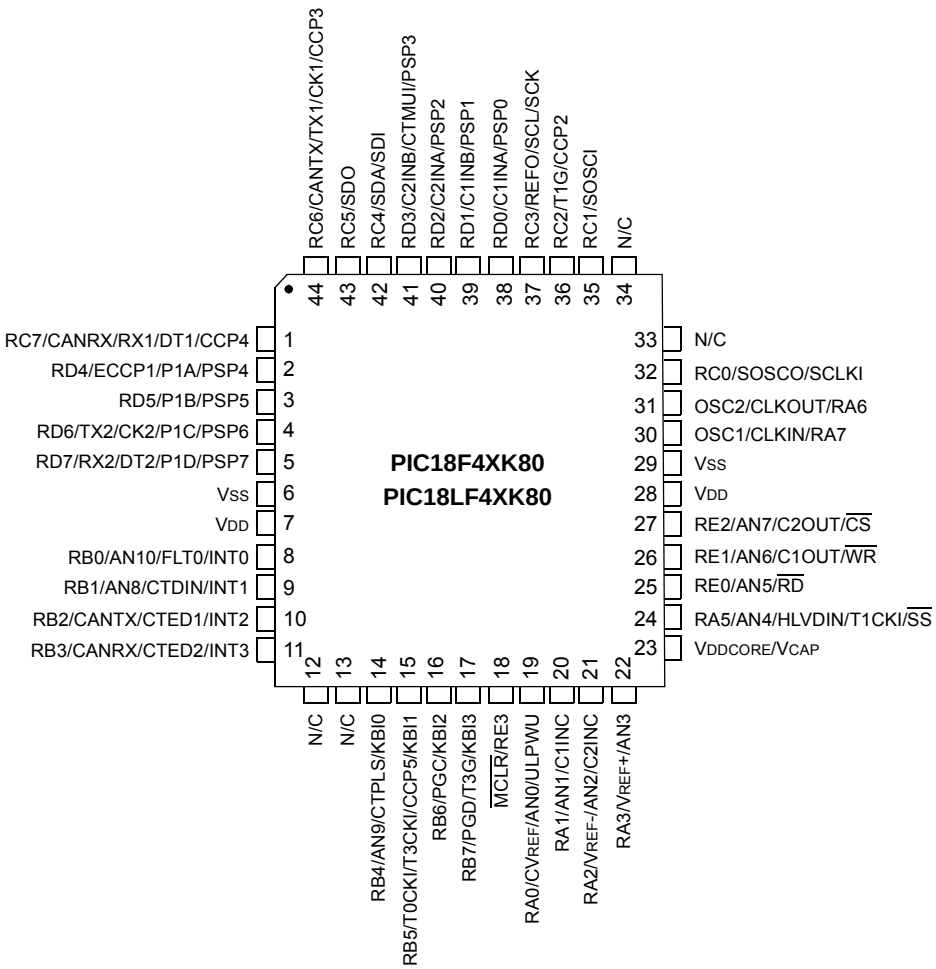
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	35
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf46k80-i-ml

Pin Diagrams (Continued)

44-Pin TQFP



3.3 Clock Sources and Oscillator Switching

Essentially, PIC18F66K80 family devices have these independent clock sources:

- Primary oscillators
- Secondary oscillators
- Internal oscillator

The **primary oscillators** can be thought of as the main device oscillators. These are any external oscillators connected to the OSC1 and OSC2 pins, and include the External Crystal and Resonator modes and the External Clock modes. If selected by the FOSC<3:0> Configuration bits (CONFIG1H<3:0>), the internal oscillator block may be considered a primary oscillator. The internal oscillator block can be one of the following:

- 31 kHz LF-INTOSC source
- 31 kHz to 500 kHz MF-INTOSC source
- 31 kHz to 16 MHz HF-INTOSC source

The particular mode is defined by the FOSCx Configuration bits. The details of these modes are covered in **Section 3.5 "External Oscillator Modes"**.

The **secondary oscillators** are external clock sources that are not connected to the OSC1 or OSC2 pin. These sources may continue to operate, even after the controller is placed in a power-managed mode. PIC18F66K80 family devices offer the SOSC (Timer1/3/5/7) oscillator as a secondary oscillator source.

The SOSC can be enabled from any peripheral that requests it. The SOSC can be enabled several ways by doing one of the following:

- The SOSC is selected as the source by either of the odd timers, which is done by each respective SOSCEN bit (TxCON<3>)
- The SOSC is selected as the CPU clock source by the SCSx bits (OSCCON<1:0>)
- The SOSCGO bit is set (OSCCON2<3>)

The SOSCGO bit is used to warm up the SOSC so that it is ready before any peripheral requests it.

The secondary oscillator has three Run modes. The SOSCSEL<1:0> bits (CONFIG1L<4:3>) decide the SOSC mode of operation:

- 11 = High-Power SOSC Circuit
- 10 = Digital (SCLKI) mode
- 11 = Low-Power SOSC Circuit

If a secondary oscillator is not desired and digital I/O on port pins, RC0 and RC1, is needed, the SOSCSELx bits must be set to Digital mode.

In addition to being a primary clock source in some circumstances, the **internal oscillator** is available as a power-managed mode clock source. The LF-INTOSC source is also used as the clock source for several special features, such as the WDT and Fail-Safe Clock Monitor. The internal oscillator block is discussed in more detail in **Section 3.6 "Internal Oscillator Block"**.

The PIC18F66K80 family includes features that allow the device clock source to be switched from the main oscillator, chosen by device configuration, to one of the alternate clock sources. When an alternate clock source is enabled, various power-managed operating modes are available.

3.3.1 OSC1/OSC2 OSCILLATOR

The OSC1/OSC2 oscillator block is used to provide the oscillator modes and frequency ranges:

Mode	Design Operating Frequency
LP	31.25-100 kHz
XT	100 kHz to 4 MHz
HS	4 MHz to 25 MHz
EC	0 to 64 MHz (external clock)
EXTRC	0 to 4 MHz (external RC)

The crystal-based oscillators (XT, HS and LP) have a built-in start-up time. The operation of the EC and EXTRC clocks is immediate.

3.3.2 CLOCK SOURCE SELECTION

The System Clock Select bits, SCS<1:0> (OSCCON<1:0>), select the clock source. The available clock sources are the primary clock defined by the FOSC<3:0> Configuration bits, the secondary clock (SOSC oscillator) and the internal oscillator. The clock source changes after one or more of the bits is written to, following a brief clock transition interval.

The OSTS (OSCCON<3>) and SOSCRUN (OSCCON2<6>) bits indicate which clock source is currently providing the device clock. The OSTS bit indicates that the Oscillator Start-up Timer (OST) has timed out and the primary clock is providing the device clock in primary clock modes. The SOSCRUN bit indicates when the SOSC oscillator (from Timer1/3/5/7) is providing the device clock in secondary clock modes. In power-managed modes, only one of these bits will be set at any time. If neither of these bits is set, the INTOSC is providing the clock or the internal oscillator has just started and is not yet stable.

The IDLEN bit (OSCCON<7>) determines if the device goes into Sleep mode or one of the Idle modes when the SLEEP instruction is executed.

TABLE 4-4: EXIT DELAY ON WAKE-UP BY RESET FROM SLEEP MODE OR ANY IDLE MODE (BY CLOCK SOURCES)

Power-Managed Mode	Clock Source[5]	Exit Delay	Clock Ready Status Bits
PRI_IDLE mode	LP, XT, HS	T_{CSD}[1]	OSTS
	HSPLL		
	EC, RC		
	HF-INTOSC[2]		HFIOFS
	MF-INTOSC[2]		MFIOFS
	LF-INTOSC		None
SEC_IDLE mode	SOSC	T_{CSD}[1]	SOSCRUN
RC_IDLE mode	HF-INTOSC[2]	T_{CSD}[1]	HFIOFS
	MF-INTOSC[2]		MFIOFS
	LF-INTOSC		None
Sleep mode	LP, XT, HS	T_{OST}[3]	OSTS
	HSPLL	$T_{OST} + t_{rc}$[3]	
	EC, RC	T_{CSD}[1]	
	HF-INTOSC[2]	T_{IOBST}[4]	HFIOFS
	MF-INTOSC[2]		MFIOFS
	LF-INTOSC		None

Note 1: T_{CSD} (Parameter 38, Table 31-11) is a required delay when waking from Sleep and all Idle modes, and runs concurrently with any other required delays (see **Section 4.4 “Idle Modes”**).

2: Includes postscaler derived frequencies. On Reset, INTOSC defaults to HF-INTOSC at 8 MHz.

3: T_{OST} is the Oscillator Start-up Timer (Parameter 32, Table 31-11). T_{RC} is the PLL Lock-out Timer (Parameter F12, Table 31-7); it is also designated as T_{PLL}.

4: Execution continues during T_{IOBST} (Parameter 39, Table 31-11), the INTOSC stabilization period.

5: The clock source is dependent upon the settings of the SCSx (OSCCON<1:0>), IRCFx (OSCCON<6:4>) and FOSCx (CONFIG1H<3:0>) bits.

EXAMPLE 7-3: WRITING TO FLASH PROGRAM MEMORY

```
                MOVLW   SIZE_OF_BLOCK          ; number of bytes in erase block
                MOVWF   COUNTER
                MOVLW   BUFFER_ADDR_HIGH       ; point to buffer
                MOVWF   FSR0H
                MOVLW   BUFFER_ADDR_LOW
                MOVWF   FSR0L
                MOVLW   CODE_ADDR_UPPER        ; Load TBLPTR with the base
                MOVWF   TBLPTRU                ; address of the memory block
                MOVLW   CODE_ADDR_HIGH
                MOVWF   TBLPTRH
                MOVLW   CODE_ADDR_LOW
                MOVWF   TBLPTRL
READ_BLOCK
                TBLRD*+                        ; read into TABLAT, and inc
                MOVF    TABLAT, W              ; get data
                MOVWF   POSTINC0               ; store data
                DECFSZ  COUNTER                ; done?
                BRA     READ_BLOCK             ; repeat
MODIFY_WORD
                MOVLW   DATA_ADDR_HIGH         ; point to buffer
                MOVWF   FSR0H
                MOVLW   DATA_ADDR_LOW
                MOVWF   FSR0L
                MOVLW   NEW_DATA_LOW           ; update buffer word
                MOVWF   POSTINC0
                MOVLW   NEW_DATA_HIGH
                MOVWF   INDF0
ERASE_BLOCK
                MOVLW   CODE_ADDR_UPPER        ; load TBLPTR with the base
                MOVWF   TBLPTRU                ; address of the memory block
                MOVLW   CODE_ADDR_HIGH
                MOVWF   TBLPTRH
                MOVLW   CODE_ADDR_LOW
                MOVWF   TBLPTRL
                BSF     EECON1, EEPGD          ; point to Flash program memory
                BCF     EECON1, CFGS           ; access Flash program memory
                BSF     EECON1, WREN           ; enable write to memory
                BSF     EECON1, FREE           ; enable Row Erase operation
                BCF     INTCON, GIE            ; disable interrupts
                MOVLW   55h
Required        MOVWF   EECON2                 ; write 55h
Sequence        MOVLW   0AAh
                MOVWF   EECON2                 ; write 0AAh
                BSF     EECON1, WR             ; start erase (CPU stall)
                BSF     INTCON, GIE            ; re-enable interrupts
                TBLRD*-                        ; dummy read decrement
                MOVLW   BUFFER_ADDR_HIGH       ; point to buffer
                MOVWF   FSR0H
                MOVLW   BUFFER_ADDR_LOW
                MOVWF   FSR0L
WRITE_BUFFER_BACK
                MOVLW   SIZE_OF_BLOCK          ; number of bytes in holding register
                MOVWF   COUNTER
WRITE_BYTE_TO_HREGS
                MOVFF   POSTINC0, WREG         ; get low byte of buffer data
                MOVWF   TABLAT                 ; present data to table latch
                TBLWT+*                        ; write data, perform a short write
                                               ; to internal TBLWT holding register.
                DECFSZ  COUNTER                ; loop until buffers are full
                BRA     WRITE_BYTE_TO_HREGS
```

TABLE 11-7: PORTD FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RD0/C1INA/ PSP0	RD0	0	O	DIG	LATD<0> data output.
		1	I	ST	PORTD<0> data input.
	C1INA	1	I	ANA	Comparator 1 Input A.
	PSP0	x	I/O	ST	Parallel Slave Port data.
RD1/C1INB/ PSP1	RD1(1)	0	O	DIG	LATD<1> data output.
		1	I	ST	PORTD<1> data input.
	C1INB(1)	1	I	ANA	Comparator 1 Input B.
	PSP1(1)	x	I/O	ST	Parallel Slave Port data.
RD2/C2INA/ PSP2	RD2	0	O	DIG	LATD<2> data output.
		1	I	ST	PORTD<2> data input.
	C2INA	1	I	ANA	Comparator 2 Input A.
	PSP2	x	I/O	ST	Parallel Slave Port data.
RD3/C2INB/ CTMUI/PSP3	RD3	0	O	DIG	LATD<3> data output.
		1	I	ST	PORTD<3> data input.
	C2INB	1	I	ANA	Comparator 2 Input B.
	CTMUI	x	I	—	CTMU pulse generator charger for the C2INB comparator input.
	PSP3	x	I/O	ST	Parallel Slave Port data.
RD4/ECCP1/ P1A/PSP4	RD4	0	O	DIG	LATD<4> data output.
		1	I	ST	PORTD<4> data input.
	ECCP1	0	O	DIG	ECCP1 compare output and ECCP1 PWM output; takes priority over port data.
		1	I	ST	ECCP1 capture input.
	P1A	0	O	DIG	ECCP1 Enhanced PWM output, Channel A. May be configured for tri-state during Enhanced PWM shutdown events; takes priority over port data.
	PSP4	x	I/O	ST	Parallel Slave Port data.
RD5/P1B/PSP5	RD5	0	O	DIG	LATD<5> data output.
		1	I	ST	PORTD<5> data input.
	P1B	0	O	DIG	ECCP1 Enhanced PWM output, Channel B. May be configured for tri-state during Enhanced PWM shutdown events; takes priority over port data.
	PSP5	x	I/O	ST	Parallel Slave Port data.
RD6/TX2/CK2 P1C/PSP6	RD6	0	O	DIG	LATD<6> data output.
		1	I	ST	PORTD<6> data input.
	TX2(1)	0	O	DIG	Asynchronous serial data output (EUSARTx module); takes priority over port data.
	CK2(1)	0	O	DIG	Synchronous serial clock output (EUSARTx module); user must configure as an input.
		1	I	ST	Synchronous serial clock input (EUSARTx module); user must configure as an input.
	P1C	0	O	DIG	ECCP1 Enhanced PWM output, Channel C. May be configured for tri-state during Enhanced PWM.
	PSP6	x	I/O	ST	Parallel Slave Port data.

Legend: O = Output; I = Input; ANA = Analog Signal; DIG = CMOS Output; ST = Schmitt Trigger Buffer Input; x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

Note 1: This is the pin assignment for 40 and 44-pin devices (PIC18F4XK80).

TABLE 11-9: PORTE FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RE5/CANTX	RE5(1)	0	O	DIG	LATE<5> data output.
		1	I	ST	PORTE<5> data input.
	CANTX(1,2)	0	O	DIG	CAN bus TX.
RE6/RX2/DT2	RE6(1)	0	O	DIG	LATE<6> data output.
		1	I	ST	PORTE<6> data input.
	RX2(1)	1	I	ST	Asynchronous serial receive data input (EUSARTx module).
	DT2(1)	1	O	DIG	Synchronous serial data output (EUSARTx module); takes priority over port data.
		1	I	ST	Synchronous serial data input (EUSARTx module); user must configure as an input.
RE7/TX2/CK2	RE7(1)	0	O	DIG	LATE<7> data output.
		1	I	ST	PORTE<7> data input.
	TX2(1)	0	O	DIG	Asynchronous serial data output (EUSARTx module); takes priority over port data.
	CK2(1)	0	O	DIG	Synchronous serial clock output (EUSARTx module); user must configure as an input.
		1	I	ST	Synchronous serial clock input (EUSARTx module); user must configure as an input.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = CMOS Output, ST = Schmitt Trigger Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

Note 1: These bits are unavailable for 40 and 44-pin devices (PIC18F4XK0).

2: This is the alternate pin assignment for CANRX and CANTX on 64-pin devices (PIC18F6XK80) when the CANMX Configuration bit is cleared.

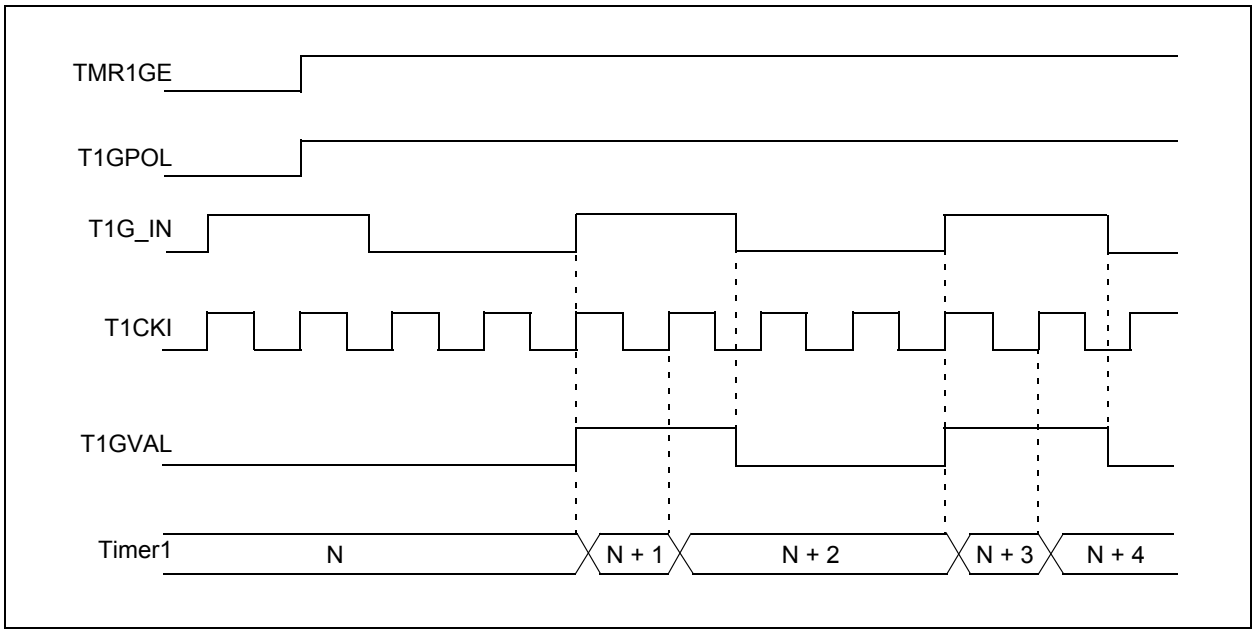
TABLE 11-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTE	RE7(1)	RE6(1)	RE5(1)	RE4(1)	RE3	RE2	RE1	RE0
LATE	LATE7	LATE6	LATE5	LATE4	—	LATE2	LATE1	LATE0
TRISE	TRISE7	TRISE6	TRISE5	TRISE4	—	TRISE2	TRISE1	TRISE0
PADCFG1	RDPU	REPU	RFPU(1)	RGPU(1)	—	—	—	CTMUDS
ANCON0	ANSEL7	ANSEL6	ANSEL5	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0

Legend: Shaded cells are not used by PORTE.

Note 1: These bits are unimplemented on 44-pin devices, read as '0'.

FIGURE 14-4: TIMER1 GATE COUNT ENABLE MODE



14.8.2 TIMER1 GATE SOURCE SELECTION

The Timer1 gate source can be selected from one of four sources. Source selection is controlled by the T1GSSx (T1GCON<1:0>) bits (see Table 14-4).

TABLE 14-4: TIMER1 GATE SOURCES

T1GSS<1:0>	Timer1 Gate Source
00	Timer1 Gate Pin
01	TMR2 to Match PR2 (TMR2 increments to match PR2)
10	Comparator 1 Output (comparator logic high output)
11	Comparator 2 Output (comparator logic high output)

The polarity for each available source is also selectable, controlled by the T1GPOL bit (T1GCON<6>).

14.8.2.1 T1G Pin Gate Operation

The T1G pin is one source for Timer1 gate control. It can be used to supply an external source to the Timer1 gate circuitry.

14.8.2.2 Timer2 Match Gate Operation

The TMR2 register will increment until it matches the value in the PR2 register. On the very next increment cycle, TMR2 will be reset to 00h. When this Reset occurs, a low-to-high pulse will automatically be generated and internally supplied to the Timer1 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 T1GPOL, Timer1 increments differently when TMR2 matches PR2. When T1GPOL = 1, Timer1 increments for a single instruction cycle following a TMR2 match with PR2. When T1GPOL = 0, Timer1 increments continuously except for the cycle following the match when the gate signal goes from low-to-high.

14.8.2.3 Comparator 1 Output Gate Operation

The output of Comparator 1 can be internally supplied to the Timer1 gate circuitry. After setting up Comparator 1 with the CM1CON register, Timer1 will increment depending on the transitions of the CMP1OUT (CMSTAT<6>) bit.

14.8.2.4 Comparator 2 Output Gate Operation

The output of Comparator 2 can be internally supplied to the Timer1 gate circuitry. After setting up Comparator 2 with the CM2CON register, Timer1 will increment depending on the transitions of the CMP2OUT (CMSTAT<7>) bit.

18.1 CTMU Registers

The control registers for the CTMU are:

- CTMUCONH
- CTMUCONL
- CTMUICON

The CTMUCONH and CTMUCONL registers (Register 18-1 and Register 18-2) contain control bits for configuring the CTMU module edge source selection, edge source polarity selection, edge sequencing, A/D trigger, analog circuit capacitor discharge and enables. The CTMUICON register (Register 18-3) has bits for selecting the current source range and current source trim.

REGISTER 18-1: CTMUCONH: CTMU CONTROL HIGH REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
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 **CTMUEN:** CTMU Enable bit
1 = Module is enabled
0 = Module is disabled

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **CTMUSIDL:** Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 4 **TGEN:** Time Generation Enable bit
1 = Enables edge delay generation
0 = Disables edge delay generation

bit 3 **EDGEN:** Edge Enable bit
1 = Edges are not blocked
0 = Edges are blocked

bit 2 **ESGSEQEN:** Edge Sequence Enable bit
1 = Edge 1 event must occur before Edge 2 event can occur
0 = No edge sequence is needed

bit 1 **IDISSEN:** Analog Current Source Control bit
1 = Analog current source output is grounded
0 = Analog current source output is not grounded

bit 0 **CTTRIG:** CTMU Special Event Trigger bit
1 = CTMU Special Event Trigger is enabled
0 = CTMU Special Event Trigger is disabled

18.7 Measuring Temperature with the CTMU

The constant current source provided by the CTMU module can be used for low-cost temperature measurement by exploiting a basic property of common and inexpensive diodes. An on-chip temperature sense diode is provided on A/D Channel 29 to further simplify design and cost.

18.7.1 BASIC PRINCIPAL

We can show that the forward voltage (V_F) of a P-N junction, such as a diode, is an extension of the equation for the junction's thermal voltage:

$$V_F = \frac{kT}{q} \ln\left(1 - \frac{I_F}{I_S}\right)$$

where k is the Boltzmann constant (1.38×10^{-23} J K^{-1}), T is the absolute junction temperature in kelvin, q is the electron charge (1.6×10^{-19} C), I_F is the forward current applied to the diode and I_S is the diode's characteristic saturation current, which varies between devices.

Since k and q are physical constants, and I_S is a constant for the device, this only leaves T and I_F as independent variables. If I_F is held constant, it follows from the equation that V_F will vary as a function of T. As the natural log term of the equation will always be negative, the temperature will be negatively proportional to V_F. In other words, as temperature increases, V_F decreases.

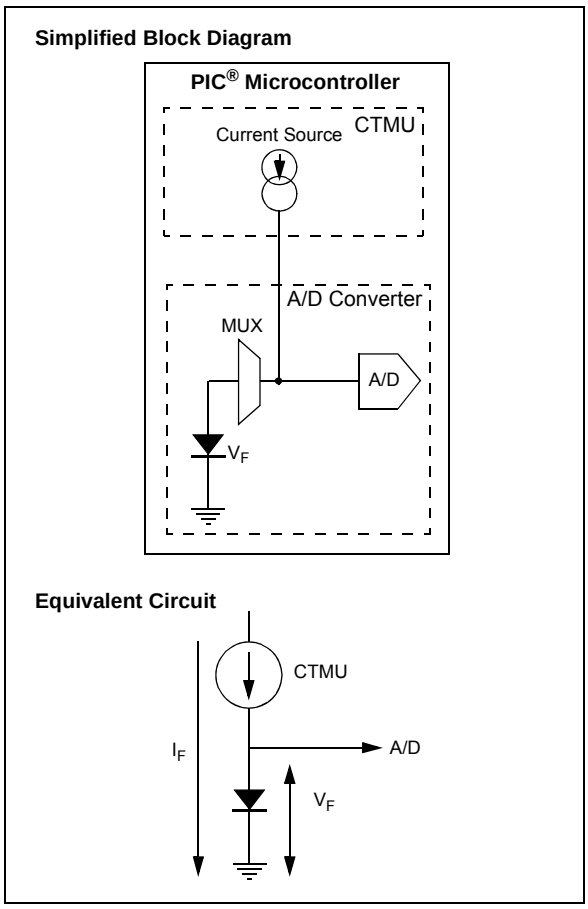
By using the CTMU's current source to provide a constant I_F, it becomes possible to calculate the temperature by measuring the V_F across the diode.

18.7.2 IMPLEMENTATION

To implement this theory, all that is needed is to connect a regular junction diode to one of the microcontroller's A/D pins (Figure 18-2). The A/D channel multiplexer is shared by the CTMU and the A/D.

To perform a measurement, the multiplexer is configured to select the pin connected to the diode. The CTMU current source is then turned on and an A/D conversion is performed on the channel. As shown in the equivalent circuit diagram, the diode is driven by the CTMU at I_F. The resulting V_F across the diode is measured by the A/D. A code snippet is shown in Example 18-5.

FIGURE 18-4: CTMU TEMPERATURE MEASUREMENT CIRCUIT



EXAMPLE 18-5: ROUTINE FOR TEMPERATURE MEASUREMENT USING INTERNAL DIODE

```
// Initialize CTMU
CTMUICON = 0x03;
CTMUCONHbits.CTMUEN = 1;
CTMUCONLbits.EDG1STAT = 1;

// Initialize ADC
ADCON0 = 0x75;                          // Enable ADC and connect to Internal diode
ADCON1 = 0x00;
ADCON2 = 0xBE;                          //Right Justified

ADCON0bits.GO = 1;                      // Start conversion
while(ADCON0bits.G0);
Temp = ADRES;                           // Read ADC results (inversely proportional to temperature)
```

Note: The temperature diode is not calibrated or standardized; the user must calibrate the diode to their application.

20.2.2 TIMER1/2/3/4 MODE SELECTION

The timers that are to be used with the capture feature (Timer1 2, 3 or 4) must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation may not work. The timer to be used with each ECCP module is selected in the CCPTMRS register (Register 20-2).

20.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCP1IE interrupt enable bit clear to avoid false interrupts. The interrupt flag bit, CCP1IF, should also be cleared following any such change in operating mode.

20.2.4 ECCP PRESCALER

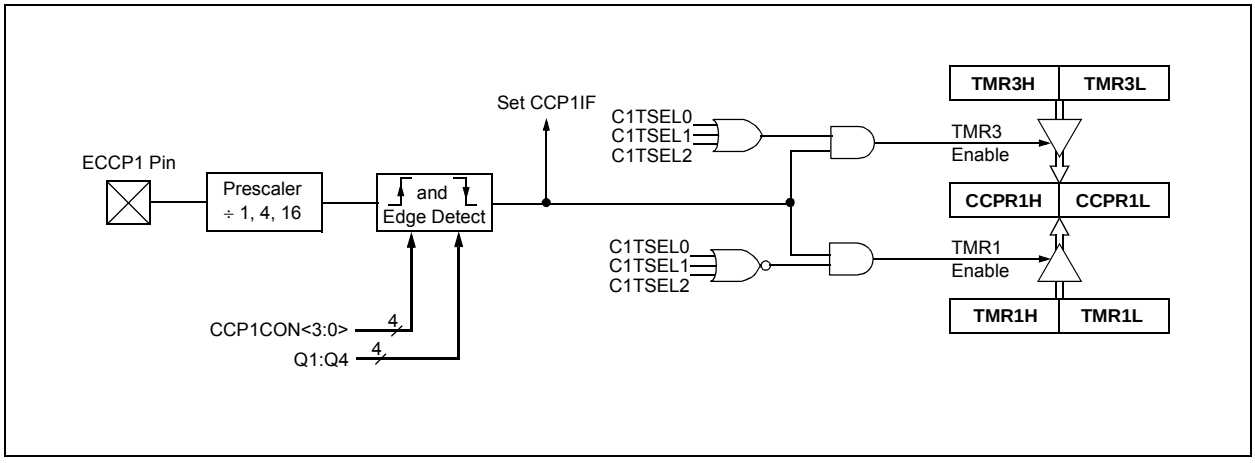
There are four prescaler settings in Capture mode; they are specified as part of the operating mode selected by the mode select bits (CCP1M<3:0>). Whenever the ECCP module is turned off, or Capture mode is disabled, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 20-1 provides the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

EXAMPLE 20-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF   CCP1CON     ; Turn ECCP module off
MOVLW  NEW_CAPT_PS ; Load WREG with the
                   ; new prescaler mode
                   ; value and ECCP ON
MOVWF  CCP1CON     ; Load ECCP1CON with
                   ; this value
```

FIGURE 20-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



21.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

21.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be devices such as serial EEPROMs, shift registers, display drivers and A/D Converters. The MSSP module can operate in either of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C™)
  - Full Master mode
  - Slave mode (with general address call)

The I^2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode with 5-bit and 7-bit address masking (with address masking for both 10-bit and 7-bit addressing)

21.2 Control Registers

The MSSP module has three associated control registers. These include a status register (SSPSTAT) and two control registers (SSPCON1 and SSPCON2). The use of these registers and their individual configuration bits differ significantly depending on whether the MSSP module is operated in SPI or I^2C mode.

Additional details are provided under the individual sections.

21.3 SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

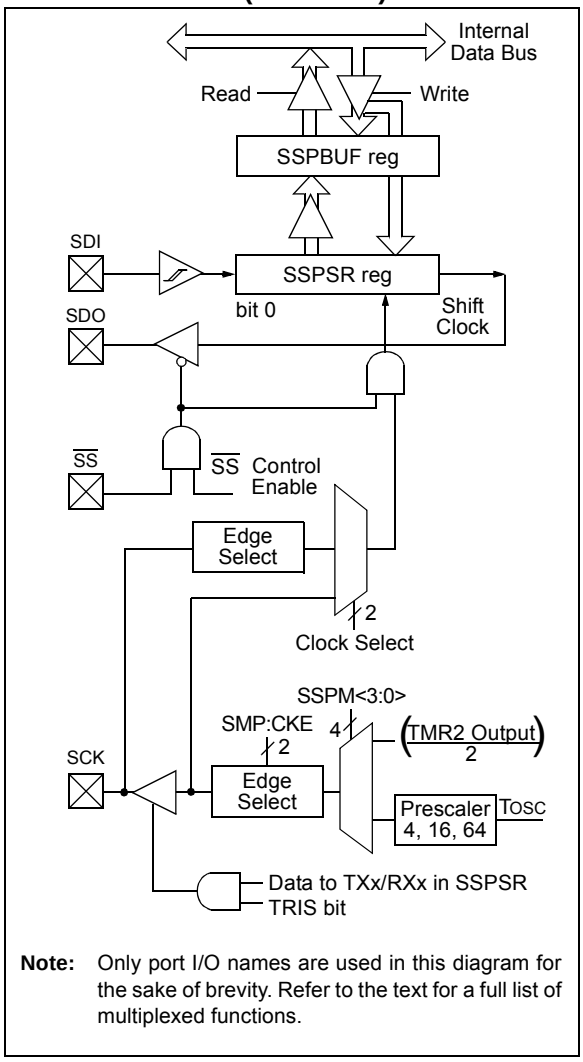
- Serial Data Out (SDO) – RC5/SDO
- Serial Data In (SDI) – RC4/SDA/SDI
- Serial Clock (SCK) – RC3/REF0/SCL/SCK

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select ($\overline{SS}$) – RA5/AN4/C2INB/HLVDIN/T1CKI/$\overline{SS}$/CTMU1

Figure 21-1 shows the block diagram of the MSSP module when operating in SPI mode.

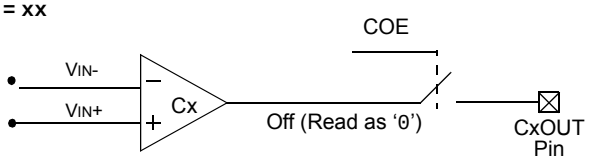
FIGURE 21-1: MSSP BLOCK DIAGRAM (SPI MODE)



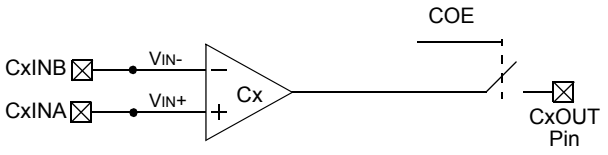
Note: Only port I/O names are used in this diagram for the sake of brevity. Refer to the text for a full list of multiplexed functions.

FIGURE 24-4: COMPARATOR CONFIGURATIONS

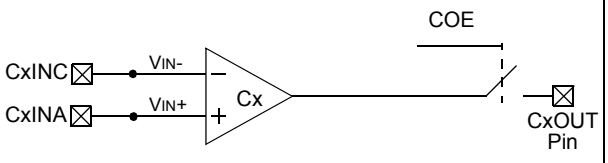
Comparator Off
CON = 0, CREF = x, CCH<1:0> = xx



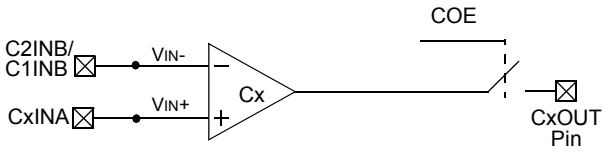
Comparator CxINB > CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 00



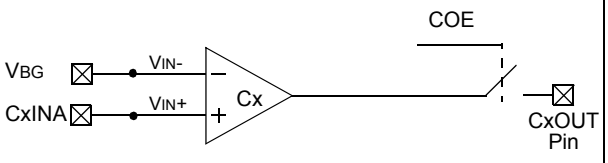
Comparator CxINC > CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 01



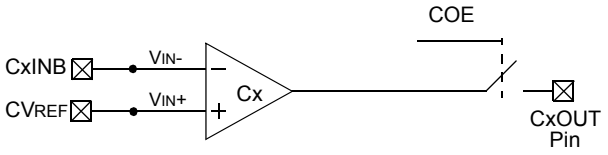
Comparator C2INB/C1INB > CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 10



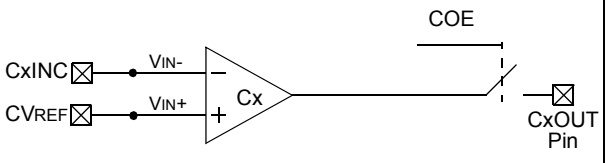
Comparator VBG > CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 11



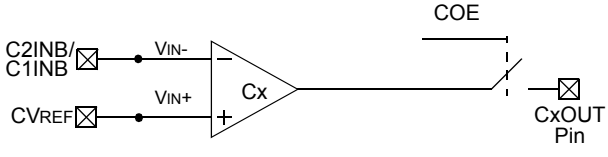
Comparator CxINB > CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 00



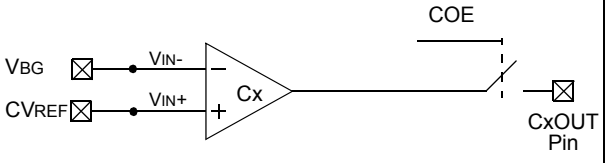
Comparator CxINC > CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 01



Comparator C2INB/C1INB > CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 10



Comparator VBG > CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 11



Note 1: VBG is the Internal Reference Voltage (see Table 31-2).

25.0 COMPARATOR VOLTAGE REFERENCE MODULE

The comparator voltage reference is a 32-tap resistor ladder network that provides a selectable reference voltage. Although its primary purpose is to provide a reference for the analog comparators, it may also be used independently of them.

A block diagram of the module is shown in Figure 25-1. The resistor ladder is segmented to provide a range of CVREF values and has a power-down function to conserve power when the reference is not being used. The module's supply reference can be provided from either device VDD/VSS or an external voltage reference.

25.1 Configuring the Comparator Voltage Reference

The comparator voltage reference module is controlled through the CVRCON register (Register 25-1). The comparator voltage reference provides a range of output voltage with 32 levels.

The CVR<4:0> selection bits (CVRCON<4:0>) offer a range of output voltages. Equation 25-1 shows the how the comparator voltage reference is computed.

EQUATION 25-1:

If CVRSS = 1:

$$CV_{REF} = \left(V_{REF-} + \frac{CVR<4:0>}{32}\right) \bullet (V_{REF+} - V_{REF-})$$

If CVRSS = 0:

$$CV_{REF} = \left(AV_{SS} + \frac{CVR<4:0>}{32}\right) \bullet (AV_{DD} - AV_{SS})$$

The comparator reference supply voltage can come from either VDD and VSS, or the external VREF+ and VREF- that are multiplexed with RA3 and RA2. The voltage source is selected by the CVRSS bit (CVRCON<5>).

The settling time of the comparator voltage reference must be considered when changing the CVREF output (see Table 31-2 in **Section 31.0 "Electrical Characteristics"**).

REGISTER 25-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE	CVRSS	CVR4	CVR3	CVR2	CVR1	CVR0
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 **CVREN:** Comparator Voltage Reference Enable bit
1 = CVREF circuit powered on
0 = CVREF circuit powered down

bit 6 **CVROE:** Comparator VREF Output Enable bit
1 = CVREF voltage level is output on CVREF pin
0 = CVREF voltage level is disconnected from CVREF pin

bit 5 **CVRSS:** Comparator VREF Source Selection bit
1 = Comparator reference source, CVRSRC = VREF+ – VREF-
0 = Comparator reference source, CVRSRC = AVDD – AVSS

bit 4-0 **CVR<4:0>:** Comparator VREF Value Selection 0 ≤ CVR<4:0> ≤ 31 bits
When CVRSS = 1:
CVREF = (VREF-) + (CVR<4:0>/32) • (VREF+ – VREF-)
When CVRSS = 0:
CVREF = (AVSS) + (CVR<4:0>/32) • (AVDD – AVSS)

26.2 HLVD Setup

To set up the HLVD module:

1. Select the desired HLVD trip point by writing the value to the HLVDL<3:0> bits.
2. Set the VDIRMAG bit to detect high voltage (VDIRMAG = 1) or low voltage (VDIRMAG = 0).
3. Enable the HLVD module by setting the HLVDEN bit.
4. Clear the HLVD interrupt flag (PIR2<2>), which may have been set from a previous interrupt.
5. If interrupts are desired, enable the HLVD interrupt by setting the HLVDIE and GIE bits (PIE2<2> and INTCON<7>, respectively).

   An interrupt will not be generated until the IRVST bit is set.

Note:	Before changing any module settings (VDIRMAG, HLVDL<3:0>), first disable the module (HLVDEN = 0), make the changes and re-enable the module. This prevents the generation of false HLVD events.

26.3 Current Consumption

When the module is enabled, the HLVD comparator and voltage divider are enabled and consume static current. The total current consumption, when enabled, is specified in electrical specification Parameter D022B (Table 31-11).

Depending on the application, the HLVD module does not need to operate constantly. To reduce current requirements, the HLVD circuitry may only need to be enabled for short periods where the voltage is checked. After such a check, the module could be disabled.

26.4 HLVD Start-up Time

The internal reference voltage of the HLVD module, specified in electrical specification Parameter 37 (**Section 31.0 “Electrical Characteristics”**), may be used by other internal circuitry, such as the programmable Brown-out Reset. If the HLVD or other circuits using the voltage reference are disabled to lower the device’s current consumption, the reference voltage circuit will require time to become stable before a low or high-voltage condition can be reliably detected. This start-up time, TIRVST, is an interval that is independent of device clock speed. It is specified in electrical specification Parameter 37 (Table 31-11).

The HLVD interrupt flag is not enabled until TIRVST has expired and a stable reference voltage is reached. For this reason, brief excursions beyond the set point may not be detected during this interval (see Figure 26-2 or Figure 26-3).

REGISTER 27-3: ECANCON: ENHANCED CAN CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
MDSEL1(1)	MDSEL0(1)	FIFOWM(2)	EWIN4	EWIN3	EWIN2	EWIN1	EWIN0
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 7-6 **MDSEL<1:0>:** Mode Select bits(1)
00 = Legacy mode (Mode 0, default)
01 = Enhanced Legacy mode (Mode 1)
10 = Enhanced FIFO mode (Mode 2)
11 = Reserved

bit 5 **FIFOWM:** FIFO High Water Mark bit(2)
1 = Will cause FIFO interrupt when one receive buffer remains
0 = Will cause FIFO interrupt when four receive buffers remain(3)

bit 4-0 **EWIN<4:0>:** Enhanced Window Address bits
These bits map the group of 16 banked CAN SFRs into Access Bank addresses, 0F60-0F6Dh. The exact group of registers to map is determined by the binary value of these bits.
Mode 0:
Unimplemented: Read as '0'
Mode 1, 2:
00000 = Acceptance Filters 0, 1, 2 and BRGCON2, 3
00001 = Acceptance Filters 3, 4, 5 and BRGCON1, CIOCON
00010 = Acceptance Filter Masks, Error and Interrupt Control
00011 = Transmit Buffer 0
00100 = Transmit Buffer 1
00101 = Transmit Buffer 2
00110 = Acceptance Filters 6, 7, 8
00111 = Acceptance Filters 9, 10, 11
01000 = Acceptance Filters 12, 13, 14
01001 = Acceptance Filter 15
01010-01110 = Reserved
01111 = RXINT0, RXINT1
10000 = Receive Buffer 0
10001 = Receive Buffer 1
10010 = TX/RX Buffer 0
10011 = TX/RX Buffer 1
10100 = TX/RX Buffer 2
10101 = TX/RX Buffer 3
10110 = TX/RX Buffer 4
10111 = TX/RX Buffer 5
11000-11111 = Reserved

Note 1: These bits can only be changed in Configuration mode. See Register 27-1 to change to Configuration mode.
2: This bit is used in Mode 2 only.
3: If FIFO is configured to contain four or less buffers, then the FIFO interrupt will trigger.

REGISTER 27-13: RXB0CON: RECEIVE BUFFER 0 CONTROL REGISTER (CONTINUED)

bit 2 Mode 0:
RB0DBEN: Receive Buffer 0 Double-Buffer Enable bit
1 = Receive Buffer 0 overflow will write to Receive Buffer 1
0 = No Receive Buffer 0 overflow to Receive Buffer 1
Mode 1, 2:
FILHIT<4:0>: Filter Hit bit 2
This bit combines with other bits to form filter acceptance bits<4:0>.

bit 1 Mode 0:
JTOFF: Jump Table Offset bit (read-only copy of RXB0DBEN)**(2)**
1 = Allows jump table offset between 6 and 7
0 = Allows jump table offset between 1 and 0
Mode 1, 2:
FILHIT<4:0>: Filter Hit bit 1
This bit combines with other bits to form filter acceptance bits<4:0>.

bit 0 Mode 0:
FILHIT0: Filter Hit bit 0
This bit indicates which acceptance filter enabled the message reception into Receive Buffer 0.
1 = Acceptance Filter 1 (RXF1)
0 = Acceptance Filter 0 (RXF0)
Mode 1, 2:
FILHIT<4:0>: Filter Hit bit 0
This bit, in combination with FILHIT<4:1>, indicates which acceptance filter enabled the message reception into this receive buffer.
01111 = Acceptance Filter 15 (RXF15)
01110 = Acceptance Filter 14 (RXF14)
...
00000 = Acceptance Filter 0 (RXF0)

Note 1: This bit is set by the CAN module upon receiving a message and must be cleared by software after the buffer is read. As long as RXFUL is set, no new message will be loaded and the buffer will be considered full. After clearing the RXFUL flag, the PIR5 bit, RXB0IF, can be cleared. If RXB0IF is cleared, but RXFUL is not cleared, then RXB0IF is set again.

2: This bit allows the same filter jump table for both RXB0CON and RXB1CON.

REGISTER 27-16: RXBnSIDL: RECEIVE BUFFER 'n' STANDARD IDENTIFIER REGISTERS, LOW BYTE [0 ≤ n ≤ 1]

R-x	R-x	R-x	R-x	R-x	U-0	R-x	R-x
SID2	SID1	SID0	SRR	EXID	—	EID17	EID16
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 **SID<2:0>:** Standard Identifier bits (if EXID = 0)
Extended Identifier bits, EID<20:18> (if EXID = 1).

bit 4 **SRR:** Substitute Remote Request bit

bit 3 **EXID:** Extended Identifier bit
1 = Received message is an extended data frame, SID<10:0> are EID<28:18>
0 = Received message is a standard data frame

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

bit 1-0 **EID<17:16>:** Extended Identifier bits

REGISTER 27-17: RXBnEIDH: RECEIVE BUFFER 'n' EXTENDED IDENTIFIER REGISTERS, HIGH BYTE [0 ≤ n ≤ 1]

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
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 **EID<15:8>:** Extended Identifier bits

REGISTER 27-18: RXBnEIDL: RECEIVE BUFFER 'n' EXTENDED IDENTIFIER REGISTERS, LOW BYTE [0 ≤ n ≤ 1]

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
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 **EID<7:0>:** Extended Identifier bits

BTFSC **Bit Test File, Skip if Clear**

Syntax: BTFSC f, b {,a}

Operands: $0 \le f \le 255$
$0 \le b \le 7$
$a \in [0,1]$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

1011	bbba	ffff	ffff

Description: If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a two-cycle instruction.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    BTFSC   FLAG, 1, 0
FALSE   :
TRUE    :
```

Before Instruction
PC = address (HERE)
After Instruction
If FLAG<1> = 0;
PC = address (TRUE)
If FLAG<1> = 1;
PC = address (FALSE)

BTFSS **Bit Test File, Skip if Set**

Syntax: BTFSS f, b {,a}

Operands: $0 \le f \le 255$
$0 \le b < 7$
$a \in [0,1]$

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

1010	bbba	ffff	ffff

Description: If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a two-cycle instruction.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

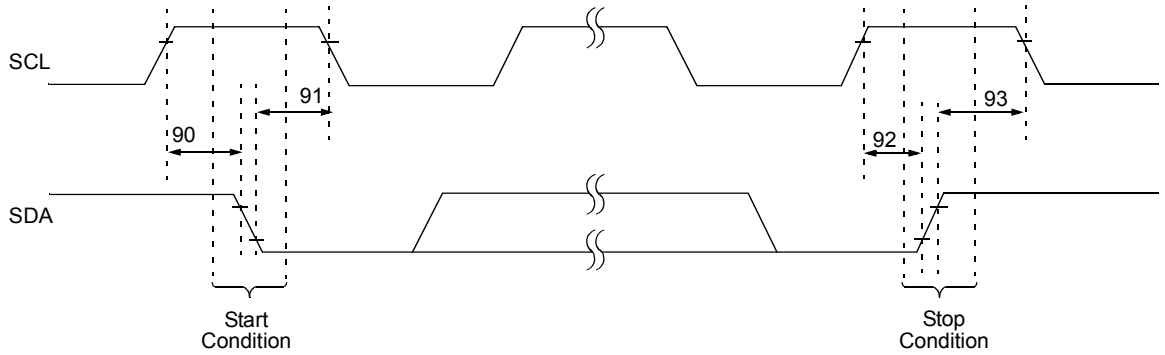
If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    BTFSS   FLAG, 1, 0
FALSE   :
TRUE    :
```

Before Instruction
PC = address (HERE)
After Instruction
If FLAG<1> = 0;
PC = address (FALSE)
If FLAG<1> = 1;
PC = address (TRUE)

FIGURE 31-15: I²C™ BUS START/STOP BITS TIMING



Note: Refer to Figure 31-3 for load conditions.

TABLE 31-19: I²C™ BUS START/STOP BITS REQUIREMENTS (SLAVE MODE)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
90	TSU:STA	Start Condition Setup Time	100 kHz mode	4700	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	600	—		
91	THD:STA	Start Condition Hold Time	100 kHz mode	4000	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	600	—		
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	4700	—	ns	
			400 kHz mode	600	—		
93	THD:STO	Stop Condition Hold Time	100 kHz mode	4000	—	ns	
			400 kHz mode	600	—		

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC18F66K80 Family Literature Number: DS39977F

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?