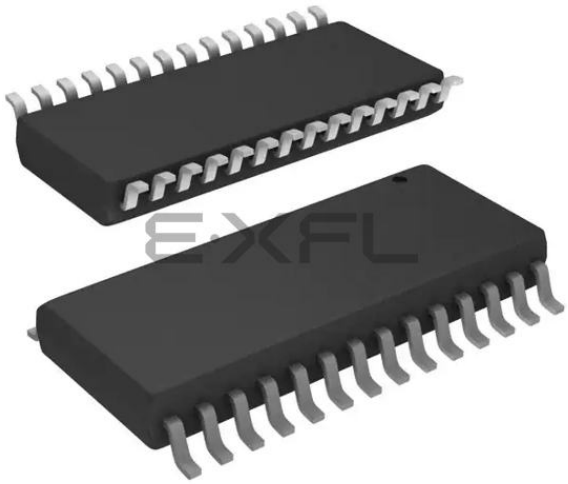


Welcome to E-XFL.COM

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	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 19x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf26k22-i-so

FIGURE 1: 28-PIN PDIP, SOIC, SSOP DIAGRAM

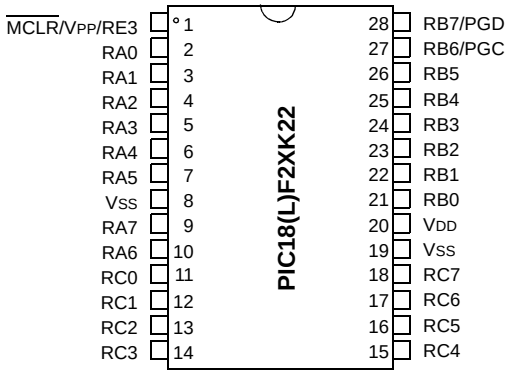
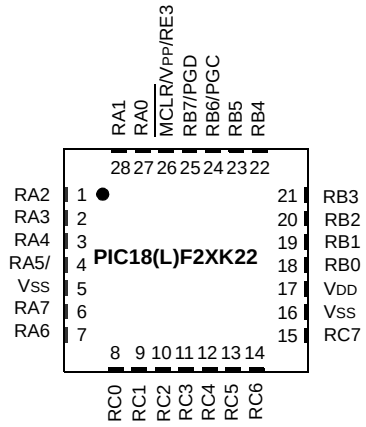


FIGURE 2: 28-PIN QFN, UQFN(1) DIAGRAM



Note 1: The 28-pin UQFN package is available only for PIC18(L)F23K22 and PIC18(L)F24K22.

TABLE 1-3: PIC18(L)F4XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)

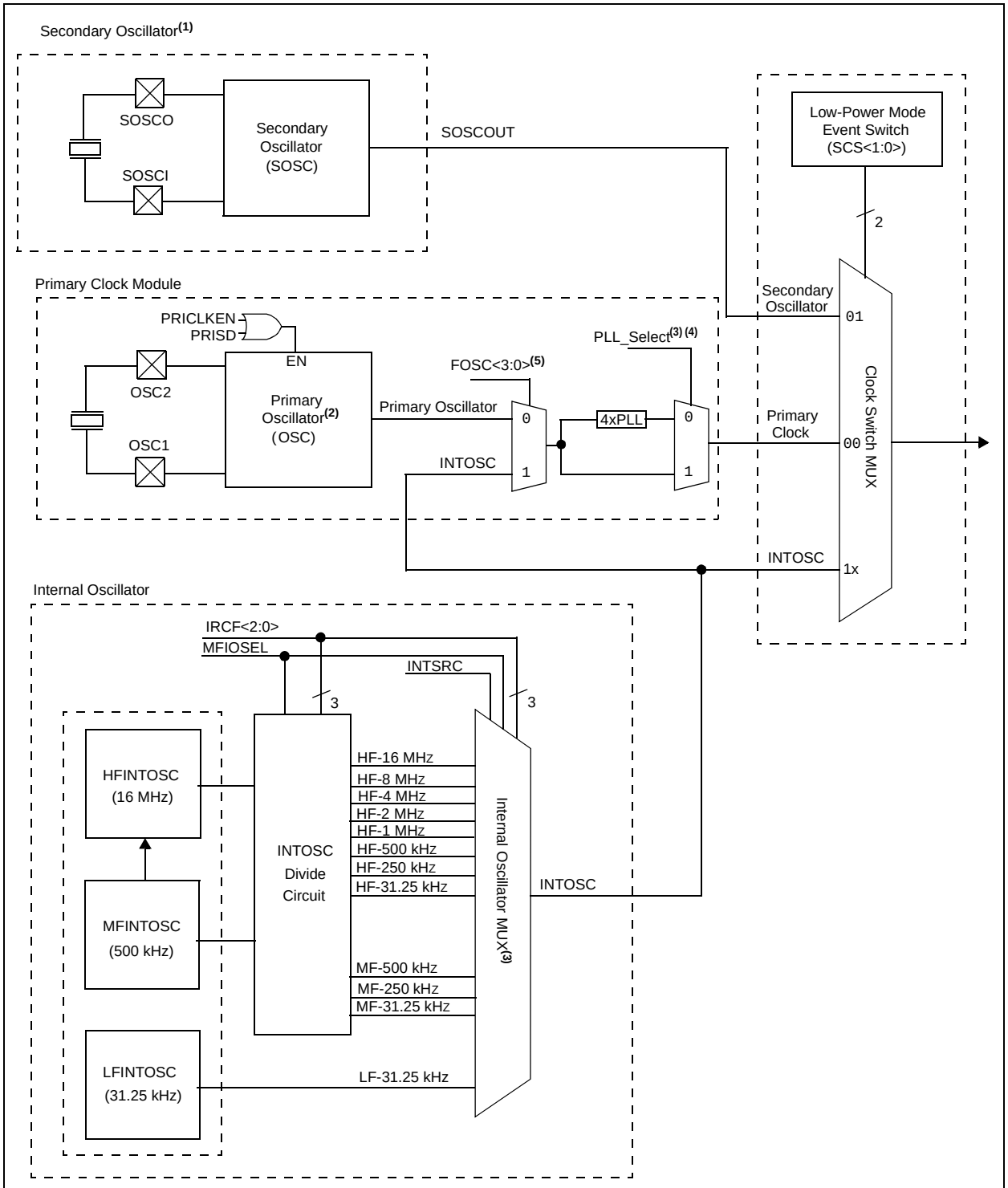
Pin Number				Pin Name	Pin Type	Buffer Type	Description
PDIP	TQFP	QFN	UQFN				
21	40	40	36	RD2/P2B/AN22			
				RD2	I/O	ST	Digital I/O
				P2B(1)	O	CMOS	Enhanced CCP2 PWM output.
				AN22	I	Analog	Analog input 22.
22	41	41	37	RD3/P2C/$\overline{SS2}$/AN23			
				RD3	I/O	ST	Digital I/O.
				P2C	O	CMOS	Enhanced CCP2 PWM output.
				$\overline{SS2}$	I	TTL	SPI slave select input (MSSP).
				AN23	I	Analog	Analog input 23.
27	2	2	2	RD4/P2D/SDO2/AN24			
				RD4	I/O	ST	Digital I/O.
				P2D	O	CMOS	Enhanced CCP2 PWM output.
				SDO2	O	—	SPI data out (MSSP).
				AN24	I	Analog	Analog input 24.
28	3	3	3	RD5/P1B/AN25			
				RD5	I/O	ST	Digital I/O.
				P1B	O	CMOS	Enhanced CCP1 PWM output.
				AN25	I	Analog	Analog input 25.
29	4	4	4	RD6/P1C/TX2/CK2/AN26			
				RD6	I/O	ST	Digital I/O.
				P1C	O	CMOS	Enhanced CCP1 PWM output.
				TX2	O	—	EUSART asynchronous transmit.
				CK2	I/O	ST	EUSART synchronous clock (see related RXx/DTx).
				AN26	I	Analog	Analog input 26.
30	5	5	5	RD7/P1D/RX2/DT2/AN27			
				RD7	I/O	ST	Digital I/O.
				P1D	O	CMOS	Enhanced CCP1 PWM output.
				RX2	I	ST	EUSART asynchronous receive.
				DT2	I/O	ST	EUSART synchronous data (see related TXx/CKx).
				AN27	I	Analog	Analog input 27.
8	25	25	23	RE0/P3A/CCP3/AN5			
				RE0	I/O	ST	Digital I/O.
				P3A(2)	O	CMOS	Enhanced CCP3 PWM output.
				CCP3(2)	I/O	ST	Capture 3 input/Compare 3 output/PWM 3 output.
				AN5	I	Analog	Analog input 5.
9	26	26	24	RE1/P3B/AN6			
				RE1	I/O	ST	Digital I/O.
				P3B	O	CMOS	Enhanced CCP3 PWM output.
				AN6	I	Analog	Analog input 6.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power.

Note 1: Default pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.

2: Alternate pin assignment for P2B, T3CKI, CCP3/P3A and CCP2/P2A when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

FIGURE 2-1: SIMPLIFIED OSCILLATOR SYSTEM BLOCK DIAGRAM



Note 1: Details in Figure 2-4.

2: Details in Figure 2-2.

3: Details in Figure 2-3.

4: Details in Table 2-1.

5: The Primary Oscillator MUX uses the INTOSC branch when FOSC<3:0> = 100x.

7.6 Operation During Code-Protect

Data EEPROM memory has its own code-protect bits in Configuration Words. External read and write operations are disabled if code protection is enabled.

The microcontroller itself can both read and write to the internal data EEPROM, regardless of the state of the code-protect Configuration bit. Refer to **Section 24.0 "Special Features of the CPU"** for additional information.

7.7 Protection Against Spurious Write

There are conditions when the user may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been implemented. On power-up, the WREN bit is cleared. In addition, writes to the EEPROM are blocked during the Power-up Timer period (TPWRT).The write initiate sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch or software malfunction.

7.8 Using the Data EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). When variables in one section change frequently, while variables in another section do not change, it is possible to exceed the total number of write cycles to the EEPROM without exceeding the total number of write cycles to a single byte. Refer to the Data EEPROM Memory parameters in **Section 27.0 "Electrical Specifications"** for write cycle limits. If this is the case, then an array refresh must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

A simple data EEPROM refresh routine is shown in Example 7-3.

Note:	If data EEPROM is only used to store constants and/or data that changes rarely, an array refresh is likely not required. See specification.

EXAMPLE 7-3: DATA EEPROM REFRESH ROUTINE

```
        CLRF    EEADR               ; Start at address 0
        CLRF    EEADRH              ; if > 256 bytes EEPROM
        BCF     EECON1, CFGS        ; Set for memory
        BCF     EECON1, EEPGD       ; Set for Data EEPROM
        BCF     INTCON, GIE         ; Disable interrupts
        BSF     EECON1, WREN        ; Enable writes
Loop                                ; Loop to refresh array
        BSF     EECON1, RD          ; Read current address
        MOVLW   55h                 ;
        MOVWF   EECON2              ; Write 55h
        MOVLW   0AAh                ;
        MOVWF   EECON2              ; Write 0AAh
        BSF     EECON1, WR          ; Set WR bit to begin write
        BTFSC   EECON1, WR          ; Wait for write to complete
        BRA     $-2
        INCFSZ  EEADR, F            ; Increment address
        BRA     LOOP                ; Not zero, do it again
        INCFSZ  EEADRH, F           ; if > 256 bytes, Increment address
        BRA     LOOP                ; if > 256 bytes, Not zero, do it again

        BCF     EECON1, WREN        ; Disable writes
        BSF     INTCON, GIE         ; Enable interrupts
```

TABLE 10-5: PORTB I/O SUMMARY (CONTINUED)

Pin	Function	TRIS Setting	ANSEL Setting	Pin Type	Buffer Type	Description
RB6/KBI2/PGC	RB6	0	—	O	DIG	LATB<6> data output; not affected by analog input.
		1	—	I	TTL	PORTB<6> data input; disabled when analog input enabled.
	IOC2	1	—	I	TTL	Interrupt-on-change pin.
	TX2(3)	1	—	O	DIG	EUSART asynchronous transmit data output.
	CK2(3)	1	—	O	DIG	EUSART synchronous serial clock output.
		1	—	I	ST	EUSART synchronous serial clock input.
	PGC	x	—	I	ST	In-Circuit Debugger and ICSP™ programming clock input.
RB7/KBI3/PGD	RB7	0	—	O	DIG	LATB<7> data output; not affected by analog input.
		1	—	I	TTL	PORTB<7> data input; disabled when analog input enabled.
	IOC3	1	—	I	TTL	Interrupt-on-change pin.
	RX2(2), (3)	1	—	I	ST	EUSART asynchronous receive data input.
	DT2(2), (3)	1	—	O	DIG	EUSART synchronous serial data output.
		1	—	I	ST	EUSART synchronous serial data input.
	PGD	x	—	O	DIG	In-Circuit Debugger and ICSP™ programming data output.
		x	—	I	ST	In-Circuit Debugger and ICSP™ programming data input.

Legend: AN = Analog input or output; TTL = TTL compatible input; HV = High Voltage; OD = Open Drain; XTAL = Crystal; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I^2C = Schmitt Trigger input with I^2C.

Note 1: Default pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are set.

2: Alternate pin assignment for P2B, T3CKI, CCP3 and CCP2 when Configuration bits PB2MX, T3CMX, CCP3MX and CCP2MX are clear.

3: Function on PORTD and PORTE for PIC18(L)F4XK22 devices.

REGISTER 10-4: ANSELB – PORTB ANALOG SELECT REGISTER

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
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 **Unimplemented:** Read as ‘0’

bit 5-0 **ANSB<5:0>:** RB<5:0> Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

REGISTER 10-5: ANSELC – PORTC ANALOG SELECT REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0
ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—
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 7-2 **ANSC<7:2>:** RC<7:2> Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

bit 1-0 **Unimplemented:** Read as ‘0’

REGISTER 10-6: ANSELD – PORTD ANALOG SELECT REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0
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 **ANSD<7:0>:** RD<7:0> Analog Select bit
1 = Digital input buffer disabled
0 = Digital input buffer enabled

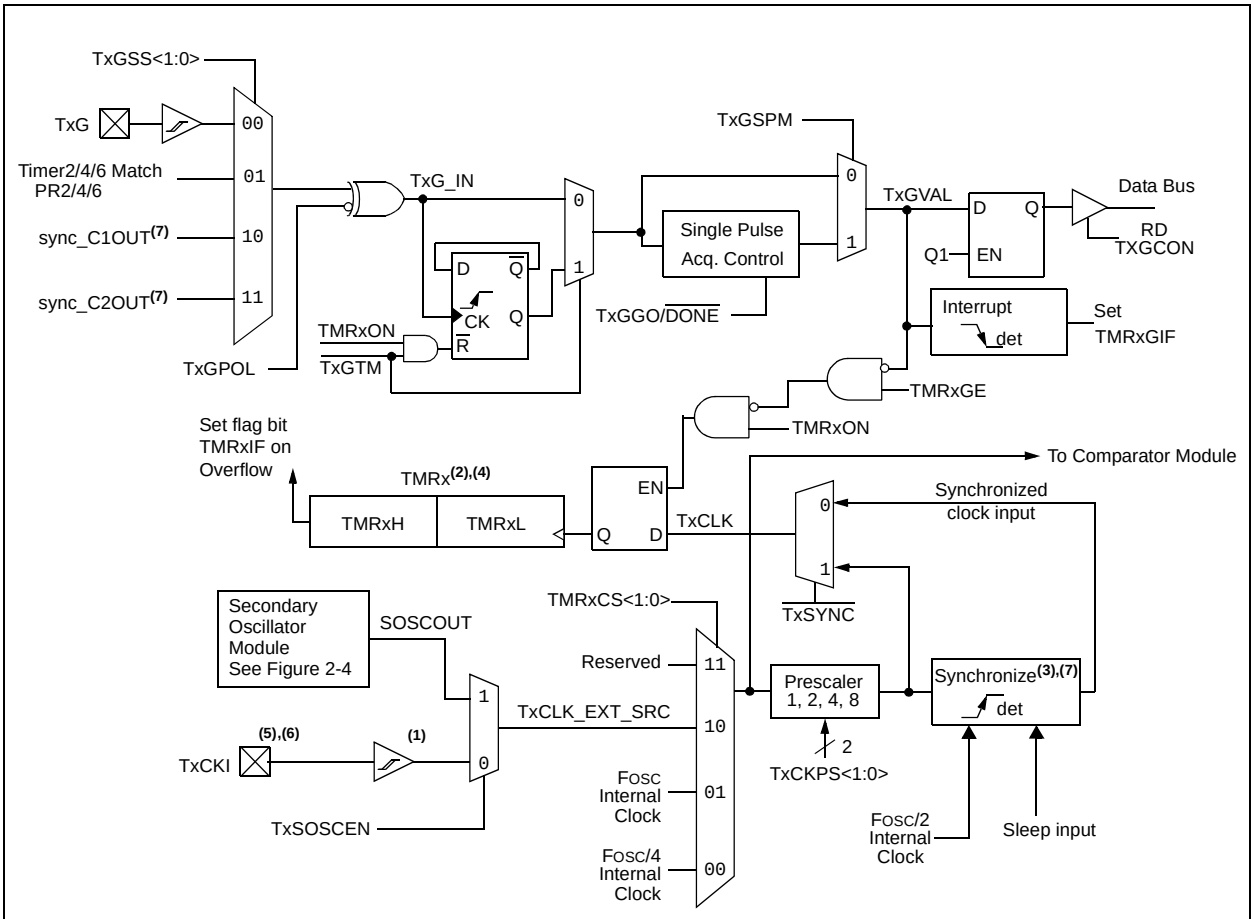
12.0 TIMER1/3/5 MODULE WITH GATE CONTROL

The Timer1/3/5 module is a 16-bit timer/counter with the following features:

- 16-bit timer/counter register pair (TMRxH:TMRxL)
- Programmable internal or external clock source
- 2-bit prescaler
- Dedicated Secondary 32 kHz oscillator circuit
- Optionally synchronized comparator out
- Multiple Timer1/3/5 gate (count enable) sources
- Interrupt on overflow
- Wake-up on overflow (external clock, Asynchronous mode only)
- 16-Bit Read/Write Operation
- Time base for the Capture/Compare function
- Special Event Trigger (with CCP/ECCP)
- Selectable Gate Source Polarity
- Gate Toggle mode
- Gate Single-pulse mode
- Gate Value Status
- Gate Event Interrupt

Figure 12-1 is a block diagram of the Timer1/3/5 module.

FIGURE 12-1: TIMER1/3/5 BLOCK DIAGRAM



Note 1: ST Buffer is high speed type when using TxCKI.

2: Timer1/3/5 register increments on rising edge.

3: Synchronize does not operate while in Sleep.

4: See Figure 12-2 for 16-Bit Read/Write Mode Block Diagram.

5: T1CKI is not available when the secondary oscillator is enabled. (SOSCGO = 1 or TXSOSCEN = 1)

6: T3CKI is not available when the secondary oscillator is enabled, unless T3CMX = 1.

7: Synchronized comparator output should not be used in conjunction with synchronized TxCKI.

13.1 Timer2/4/6 Operation

The clock input to the Timer2/4/6 module is the system instruction clock (FOSC/4).

TMRx increments from 00h on each clock edge.

A 4-bit counter/prescaler on the clock input allows direct input, divide-by-4 and divide-by-16 prescale options. These options are selected by the prescaler control bits, TxCKPS<1:0> of the TxCON register. The value of TMRx is compared to that of the Period register, PRx, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMRx to 00h on the next cycle and drives the output counter/postscaler (see **Section 13.2 "Timer2/4/6 Interrupt"**).

The TMRx and PRx registers are both directly readable and writable. The TMRx register is cleared on any device Reset, whereas the PRx register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMRx register
- a write to the TxCON register
- Power-on Reset (POR)
- Brown-out Reset (BOR)
- $\overline{\text{MCLR}}$ Reset
- Watchdog Timer (WDT) Reset
- Stack Overflow Reset
- Stack Underflow Reset
- RESET Instruction

Note:	TMRx is not cleared when TxCON is written.

13.2 Timer2/4/6 Interrupt

Timer2/4/6 can also generate an optional device interrupt. The Timer2/4/6 output signal (TMRx-to-PRx match) provides the input for the 4-bit counter/postscaler. This counter generates the TMRx match interrupt flag which is latched in TMRxIF of the PIR1/PIR5 registers. The interrupt is enabled by setting the TMRx Match Interrupt Enable bit, TMRxIE of the PIE1/PIE5 registers. Interrupt Priority is selected with the TMRxIP bit in the IPR1/IPR5 registers.

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, TxOUTPS<3:0>, of the TxCON register.

13.3 Timer2/4/6 Output

The unscaled output of TMRx is available primarily to the CCP modules, where it is used as a time base for operations in PWM mode. The timer to be used with a specific CCP module is selected using the CxTSEL<1:0> bits in the CCPTMRS0 and CCPTMRS1 registers.

Timer2 can be optionally used as the shift clock source for the MSSPx modules operating in SPI mode by setting SSPM<3:0> = 0011 in the SSPxCON1 register. Additional information is provided in **Section 15.0 "Master Synchronous Serial Port (MSSP1 and MSSP2) Module"**.

13.4 Timer2/4/6 Operation During Sleep

The Timer2/4/6 timers cannot be operated while the processor is in Sleep mode. The contents of the TMRx and PRx registers will remain unchanged while the processor is in Sleep mode.

13.5 Peripheral Module Disable

When a peripheral module is not used or inactive, the module can be disabled by setting the Module Disable bit in the PMD registers. This will reduce power consumption to an absolute minimum. Setting the PMD bits holds the module in Reset and disconnects the module's clock source. The Module Disable bits for Timer2 (TMR2MD), Timer4 (TMR4MD) and Timer6 (TMR6MD) are in the PMD0 Register. See **Section 3.0 "Power-Managed Modes"** for more information.

15.6.8 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN, of the SSPxCON2 register. When this bit is set, the SCLx pin is pulled low and the contents of the Acknowledge data bit are presented on the SDAx pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCLx pin is deasserted (pulled high). When the SCLx pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCLx pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSPx module then goes into Idle mode (Figure 15-30).

15.6.8.1 WCOL Status Flag

If the user writes the SSPxBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write does not occur).

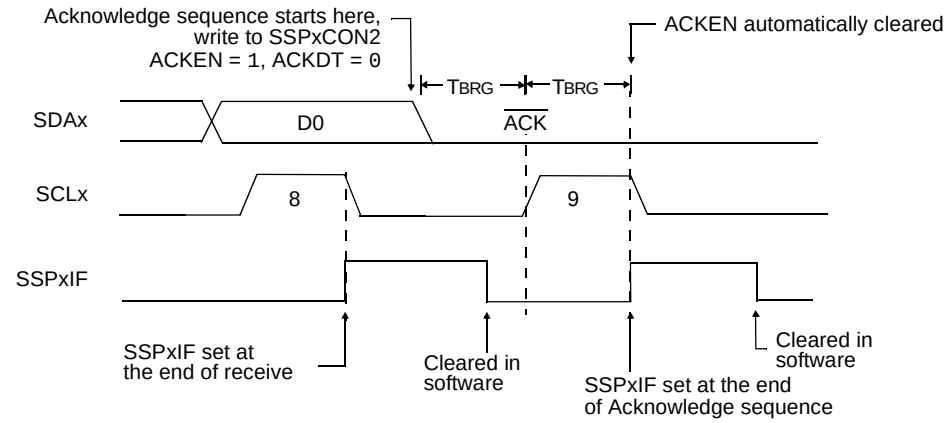
15.6.9 STOP CONDITION TIMING

A Stop bit is asserted on the SDAx pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN, of the SSPxCON2 register. At the end of a receive/transmit, the SCLx line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDAx line low. When the SDAx line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCLx pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDAx pin will be deasserted. When the SDAx pin is sampled high while SCLx is high, the P bit of the SSPxSTAT register is set. A TBRG later, the PEN bit is cleared and the SSPxIF bit is set (Figure 15-31).

15.6.9.1 WCOL Status Flag

If the user writes the SSPxBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write does not occur).

FIGURE 15-30: ACKNOWLEDGE SEQUENCE WAVEFORM



Note: TBRG = one Baud Rate Generator period.

16.4.4 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TXSTAx register. The Break character transmission is then initiated by a write to the TXREGx. The value of data written to TXREGx will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TXSTAx register indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 16-9 for the timing of the Break character sequence.

16.4.4.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to enable the Break sequence.
3. Load the TXREGx with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREGx to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TXREGx becomes empty, as indicated by the TXxIF, the next data byte can be written to TXREGx.

16.4.5 RECEIVING A BREAK CHARACTER

The Enhanced EUSART module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RCSTAx register and the Received data as indicated by RCREGx. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

A Break character has been received when;

- RCxIF bit is set
- FERR bit is set
- RCREGx = 00h

The second method uses the Auto-Wake-up feature described in **Section 16.4.3 "Auto-Wake-up on Break"**. By enabling this feature, the EUSART will sample the next two transitions on RXx/DTx, cause an RCxIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUDCONx register before placing the EUSART in Sleep mode.

FIGURE 16-9: SEND BREAK CHARACTER SEQUENCE

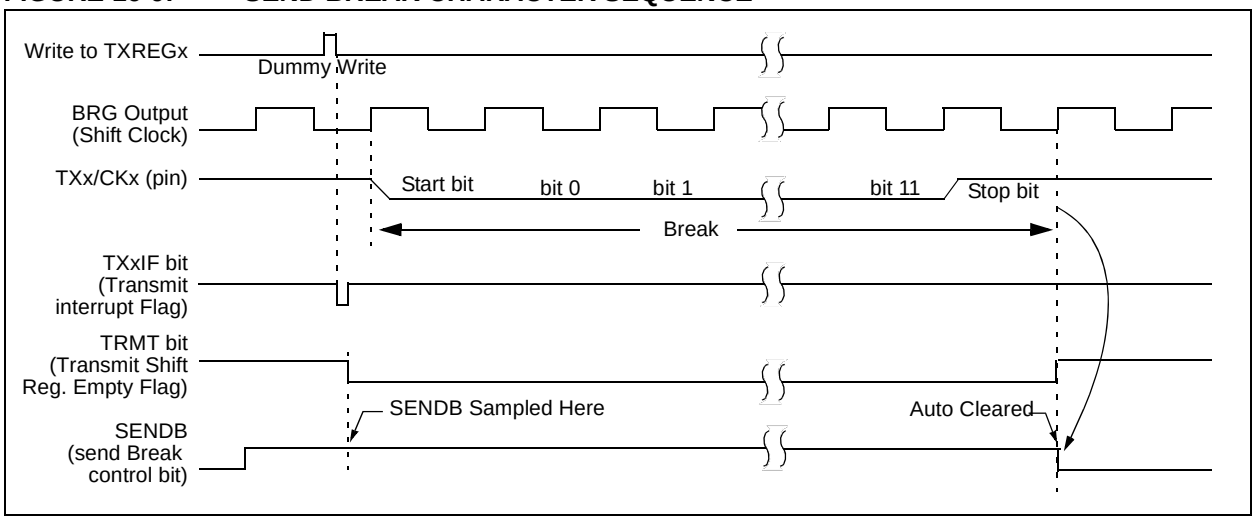


TABLE 17-2: REGISTERS ASSOCIATED WITH A/D OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ADCON0	—	CHS<4:0>					GO/$\overline{\text{DONE}}$	ADON	295
ADCON1	TRIGSEL	—	—	—	PVCFG<1:0>		NVCFG<1:0>		296
ADCON2	ADFM	—	ACQT<2:0>			ADCS<2:0>			297
ADRESH	A/D Result, High Byte								298
ADRESL	A/D Result, Low Byte								298
ANSELA	—	—	ANSA5	—	ANSA3	ANSA2	ANSA1	ANSA0	149
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	150
ANSELC	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	—	—	150
ANSELD(1)	ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0	150
ANSELE(1)	—	—	—	—	—	ANSE2	ANSE1	ANSE0	151
CCP5CON	—	—	DC5B<1:0>		CCP5M<3:0>				198
CTMUCONH	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	323
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	121
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	CTMUIP	TMR5GIP	TMR3GIP	TMR1GIP	123
IPR4	—	—	—	—	—	CCP5IP	CCP4IP	CCP3IP	124
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	117
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	CTMUIE	TMR5GIE	TMR3GIE	TMR1GIE	119
PIE4	—	—	—	—	—	CCP5IE	CCP4IE	CCP3IE	120
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	112
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	CTMUIF	TMR5GIF	TMR3GIF	TMR1GIF	114
PIR4	—	—	—	—	—	CCP5IF	CCP4IF	CCP3IF	115
PMD1	MSSP2MD	MSSP1MD	—	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	53
PMD2	—	—	—	—	CTMUMD	CMP2MD	CMP1MD	ADCMD	54
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	151
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	151
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	151
TRISD(1)	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	151
TRISE	WPUE3	—	—	—	—	TRISE2(1)	TRISE1(1)	TRISE0(1)	151

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used by this module.

Note 1: Available on PIC18(L)F4XK22 devices.

TABLE 17-3: CONFIGURATION REGISTERS ASSOCIATED WITH THE ADC MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CONFIG3H	MCLRE	—	P2BMX	T3CMX	HFOFST	CCP3MX	PBADEN	CCP2MX	348

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used by the ADC module.

19.1 CTMU Operation

The CTMU works by using a fixed current source to charge a circuit. The type of circuit depends on the type of measurement being made. In the case of charge measurement, the current is fixed, and the amount of time the current is applied to the circuit is fixed. The amount of voltage read by the A/D is then a measurement of the capacitance of the circuit. In the case of time measurement, the current, as well as the capacitance of the circuit, is fixed. In this case, the voltage read by the A/D is then representative of the amount of time elapsed from the time the current source starts and stops charging the circuit.

If the CTMU is being used as a time delay, both capacitance and current source are fixed, as well as the voltage supplied to the comparator circuit. The delay of a signal is determined by the amount of time it takes the voltage to charge to the comparator threshold voltage.

19.1.1 THEORY OF OPERATION

The operation of the CTMU is based on the equation for charge:

$$I = C \cdot \frac{dV}{dT}$$

More simply, the amount of charge measured in coulombs in a circuit is defined as current in amperes (I) multiplied by the amount of time in seconds that the current flows (t). Charge is also defined as the capacitance in farads (C) multiplied by the voltage of the circuit (V). It follows that:

$$I \cdot t = C \cdot V.$$

The CTMU module provides a constant, known current source. The A/D Converter is used to measure (V) in the equation, leaving two unknowns: capacitance (C) and time (t). The above equation can be used to calculate capacitance or time, by either the relationship using the known fixed capacitance of the circuit:

$$t = (C \cdot V)/I$$

or by:

$$C = (I \cdot t)/V$$

using a fixed time that the current source is applied to the circuit.

19.1.2 CURRENT SOURCE

At the heart of the CTMU is a precision current source, designed to provide a constant reference for measurements. The level of current is user-selectable across three ranges, with the ability to trim the output. The current range is selected by the IRNG<1:0> bits (CTMUICON<1:0>), with a value of '00' representing the lowest range.

Current trim is provided by the ITRIM<5:0> bits (CTMUICON<7:2>). Note that half of the range adjusts the current source positively and the other half reduces the current source. A value of '000000' is the neutral position (no change). A value of '100000' is the maximum negative adjustment, and '011111' is the maximum positive adjustment.

19.1.3 EDGE SELECTION AND CONTROL

CTMU measurements are controlled by edge events occurring on the module's two input channels. Each channel, referred to as Edge 1 and Edge 2, can be configured to receive input pulses from one of the edge input pins (CTED1 and CTED2) or ECCPx Special Event Triggers. The input channels are level-sensitive, responding to the instantaneous level on the channel rather than a transition between levels. The inputs are selected using the EDG1SEL and EDG2SEL bit pairs (CTMUCONL<3:2 and 6:5>).

In addition to source, each channel can be configured for event polarity using the EDGE2POL and EDGE1POL bits (CTMUCONL<7,4>). The input channels can also be filtered for an edge event sequence (Edge 1 occurring before Edge 2) by setting the EDGSEQEN bit (CTMUCONH<2>).

19.1.4 EDGE STATUS

The CTMUCONL register also contains two Status bits: EDG2STAT and EDG1STAT (CTMUCONL<1:0>). Their primary function is to show if an edge response has occurred on the corresponding channel. The CTMU automatically sets a particular bit when an edge response is detected on its channel. The level-sensitive nature of the input channels also means that the Status bits become set immediately if the channel's configuration is changed and is the same as the channel's current state.

19.6 Creating a Delay with the CTMU Module

A unique feature on board the CTMU module is its ability to generate system clock independent output pulses based on an external capacitor value. This is accomplished using the internal comparator voltage reference module, Comparator 2 input pin and an external capacitor. The pulse is output onto the CTPLS pin. To enable this mode, set the TGEN bit.

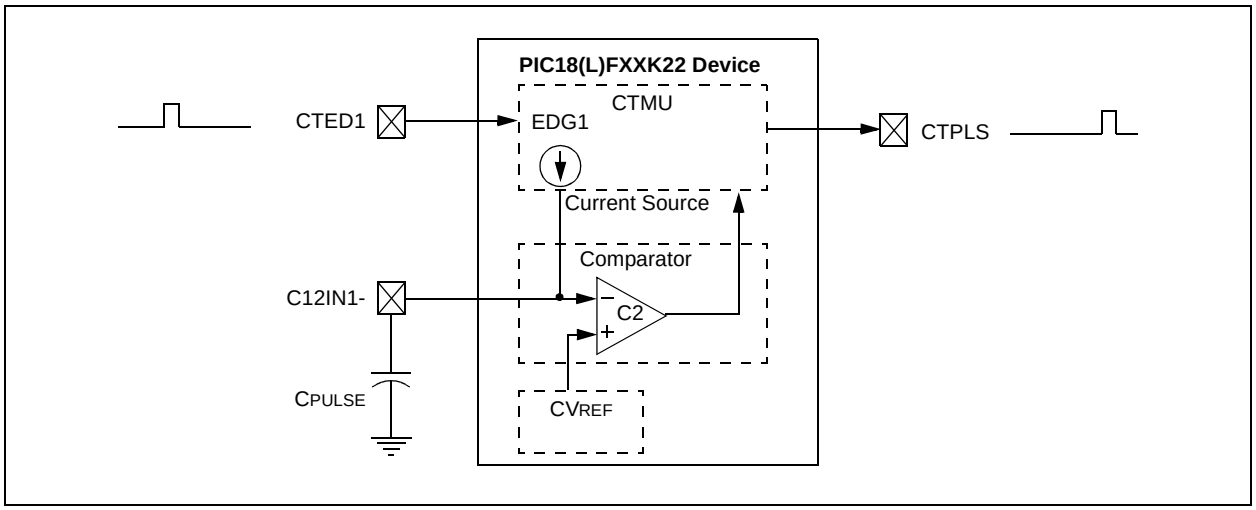
See Figure 19-4 for an example circuit. C_{PULSE} is chosen by the user to determine the output pulse width on CTPLS. The pulse width is calculated by $T = (C_{PULSE}/I) * V$, where I is known from the current source measurement step (**Section 19.3.1 "Current Source Calibration"**) and V is the internal reference voltage (CV_{REF}).

An example use of this feature is for interfacing with variable capacitive-based sensors, such as a humidity sensor. As the humidity varies, the pulse width output on CTPLS will vary. The CTPLS output pin can be connected to an input capture pin and the varying pulse width is measured to determine the humidity in the application.

Follow these steps to use this feature:

1. Initialize Comparator 2.
2. Initialize the comparator voltage reference.
3. Initialize the CTMU and enable time delay generation by setting the TGEN bit.
4. Set EDG1STAT.
5. When C_{PULSE} charges to the value of the voltage reference trip point, an output pulse is generated on CTPLS.

FIGURE 19-4: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR PULSE DELAY GENERATION



19.7 Operation During Sleep/Idle Modes

19.7.1 SLEEP MODE AND DEEP SLEEP MODES

When the device enters any Sleep mode, the CTMU module current source is always disabled. If the CTMU is performing an operation that depends on the current source when Sleep mode is invoked, the operation may not terminate correctly. Capacitance and time measurements may return erroneous values.

19.7.2 IDLE MODE

The behavior of the CTMU in Idle mode is determined by the CTMUSIDL bit (CTMUCONH<5>). If CTMUSIDL is cleared, the module will continue to operate in Idle mode. If CTMUSIDL is set, the module's current source is disabled when the device enters Idle mode. If the module is performing an operation when Idle mode is invoked, in this case, the results will be similar to those with Sleep mode.

19.8 CTMU Peripheral Module Disable (PMD)

When this peripheral is not used, the Peripheral Module Disable bit can be set to disconnect all clock sources to the module, reducing power consumption to an absolute minimum. See **Section 3.6 "Selective Peripheral Module Control"**.

REGISTER 20-2: SRCON1: SR LATCH CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SRSPE	SRSCKE	SRSC2E	SRSC1E	SRRPE	SRRCKE	SRRC2E	SRRC1E
bit 7							bit 0

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

bit 7 **SRSPE:** SR Latch Peripheral Set Enable bit
1 = SRI pin status sets SR latch
0 = SRI pin status has no effect on SR latch

bit 6 **SRSCKE:** SR Latch Set Clock Enable bit
1 = Set input of SR latch is pulsed with DIVSRCLK
0 = Set input of SR latch is not pulsed with DIVSRCLK

bit 5 **SRSC2E:** SR Latch C2 Set Enable bit
1 = C2 Comparator output sets SR latch
0 = C2 Comparator output has no effect on SR latch

bit 4 **SRSC1E:** SR Latch C1 Set Enable bit
1 = C1 Comparator output sets SR latch
0 = C1 Comparator output has no effect on SR latch

bit 3 **SRRPE:** SR Latch Peripheral Reset Enable bit
1 = SRI pin resets SR latch
0 = SRI pin has no effect on SR latch

bit 2 **SRRCKE:** SR Latch Reset Clock Enable bit
1 = Reset input of SR latch is pulsed with DIVSRCLK
0 = Reset input of SR latch is not pulsed with DIVSRCLK

bit 1 **SRRC2E:** SR Latch C2 Reset Enable bit
1 = C2 Comparator output resets SR latch
0 = C2 Comparator output has no effect on SR latch

bit 0 **SRRC1E:** SR Latch C1 Reset Enable bit
1 = C1 Comparator output resets SR latch
0 = C1 Comparator output has no effect on SR latch

TABLE 20-2: REGISTERS ASSOCIATED WITH THE SR LATCH

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
SRCON0	SRLEN	SRCLK<2:0>			SRQEN	SRNQEN	SRPS	SRPR	329
SRCON1	SRSPE	SRSCKE	SRSC2E	SRSC1E	SRRPE	SRRCKE	SRRC2E	SRRC1E	330
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	151
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	151
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	152

Legend: Shaded bits are not used with this module.

CLRF **Clear f**

Syntax: CLRF f {,a}

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

Operation: 000h → f
1 → Z

Status Affected: Z

Encoding:

0110	101a	ffff	ffff

Description: Clears the contents of the specified register.
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 ≤ 95 (5Fh). See **Section 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: CLRF FLAG_REG, 1

Before Instruction
FLAG_REG = 5Ah
After Instruction
FLAG_REG = 00h

CLRWDT **Clear Watchdog Timer**

Syntax: CLRWDT

Operands: None

Operation: 000h → WDT,
000h → WDT postscaler,
1 → $\overline{TO}$,
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0000	0100

Description: CLRWDT instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits, $\overline{TO}$ and $\overline{PD}$, are set.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	No operation

Example: CLRWDT

Before Instruction
WDT Counter = ?
After Instruction
WDT Counter = 00h
WDT Postscaler = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

SUBLW **Subtract W from literal**

Syntax: SUBLW k

Operands: $0 \le k \le 255$

Operation: $k - (W) \rightarrow W$

Status Affected: N, OV, C, DC, Z

Encoding:

0000	1000	kkkk	kkkk

Description: W is subtracted from the 8-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example 1:
```
SUBLW   02h
Before Instruction
    W   =   01h
    C   =   ?
After Instruction
    W   =   01h
    C   =   1   ; result is positive
    Z   =   0
    N   =   0
```

Example 2:
```
SUBLW   02h
Before Instruction
    W   =   02h
    C   =   ?
After Instruction
    W   =   00h
    C   =   1   ; result is zero
    Z   =   1
    N   =   0
```

Example 3:
```
SUBLW   02h
Before Instruction
    W   =   03h
    C   =   ?
After Instruction
    W   =   FFh ; (2's complement)
    C   =   0   ; result is negative
    Z   =   0
    N   =   1
```

SUBWF **Subtract W from f**

Syntax: SUBWF f {,d {,a}}

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

Operation: $(f) - (W) \rightarrow dest$

Status Affected: N, OV, C, DC, Z

Encoding:

0101	11da	ffff	ffff

Description: Subtract W from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).
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 25.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example 1:
```
SUBWF   REG, 1, 0
Before Instruction
    REG   =   3
    W     =   2
    C     =   ?
After Instruction
    REG   =   1
    W     =   2
    C     =   1   ; result is positive
    Z     =   0
    N     =   0
```

Example 2:
```
SUBWF   REG, 0, 0
Before Instruction
    REG   =   2
    W     =   2
    C     =   ?
After Instruction
    REG   =   2
    W     =   0
    C     =   1   ; result is zero
    Z     =   1
    N     =   0
```

Example 3:
```
SUBWF   REG, 1, 0
Before Instruction
    REG   =   1
    W     =   2
    C     =   ?
After Instruction
    REG   =   FFh ;(2's complement)
    W     =   2
    C     =   0   ; result is negative
    Z     =   0
    N     =   1
```

26.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

26.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

26.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

26.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

26.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

26.9 PICkit 3 In-Circuit Debugger/ Programmer

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

26.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

FIGURE 28-70: PIC18F2X/4XK22 TYPICAL IDD: PRI_IDLE EC with PLL

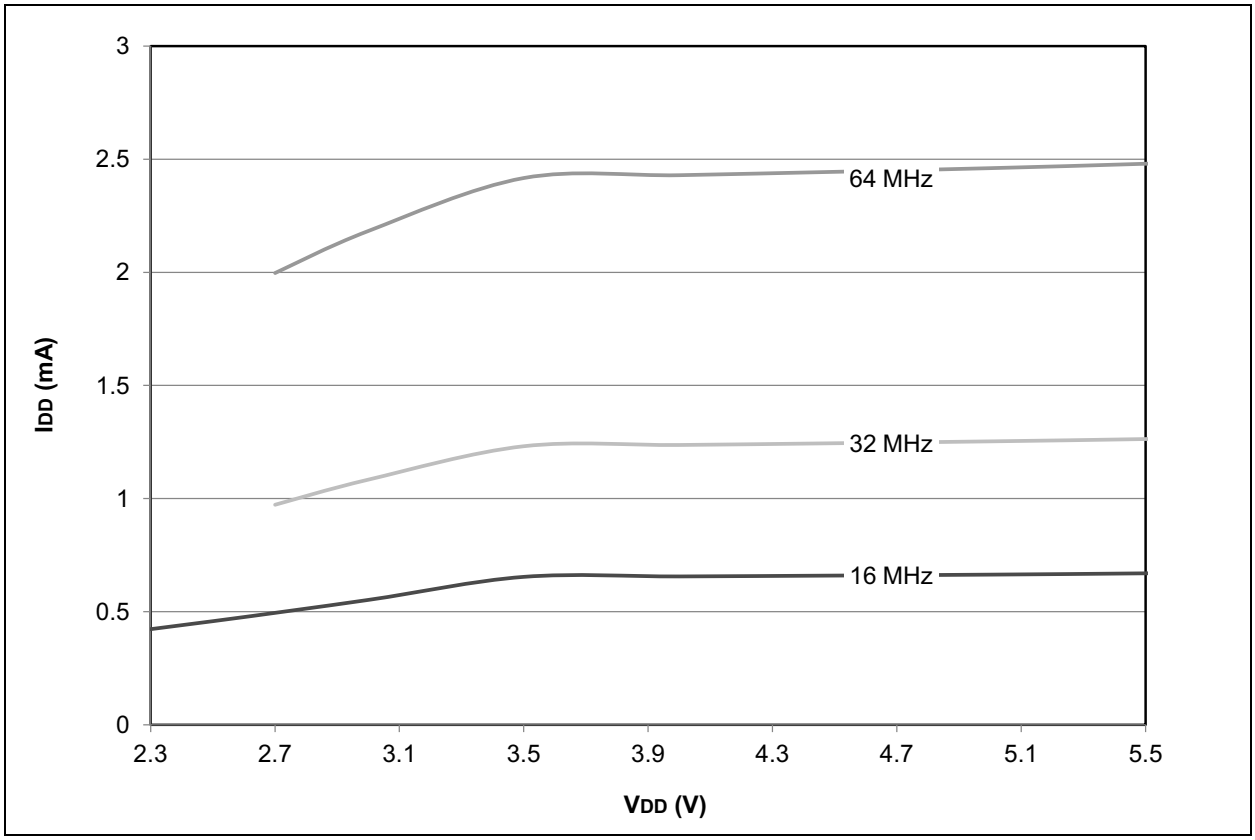
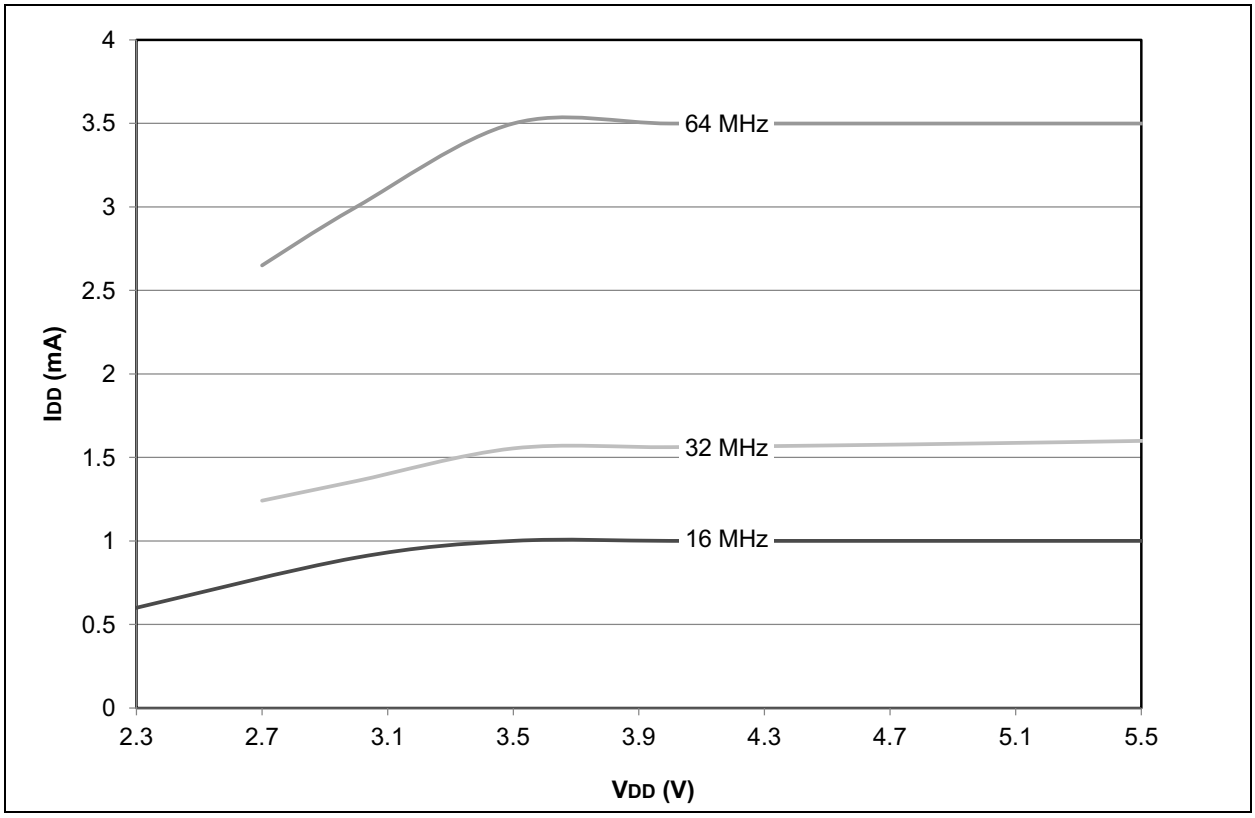


FIGURE 28-71: PIC18F2X/4XK22 MAXIMUM IDD: PRI_IDLE EC with PLL



THE MICROCHIP WEBSITE

Microchip provides online support via our website at www.microchip.com. This website is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the website contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip website at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the website at: http://microchip.com/support