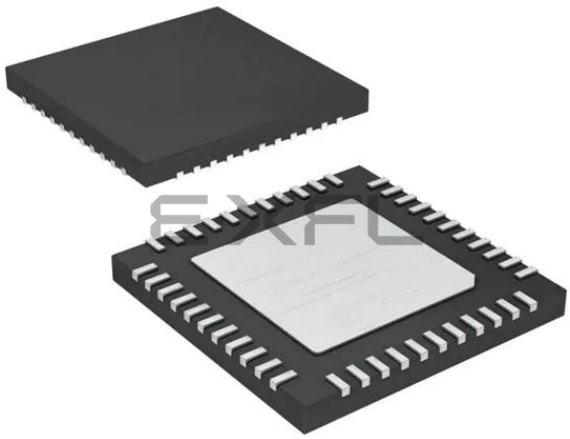


Welcome to E-XFL.COM

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	48MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 30x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/pic18f43k22-e-ml

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

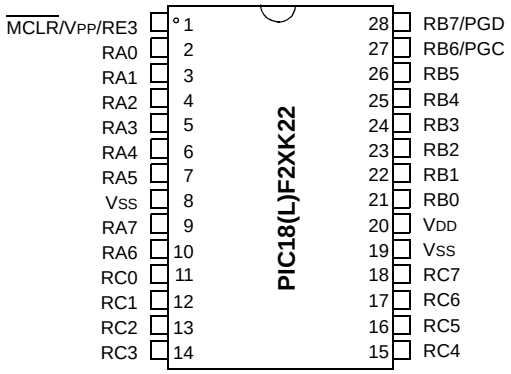
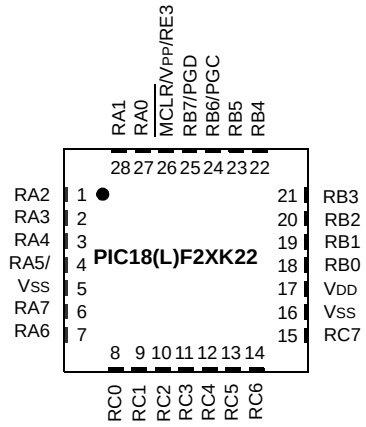


FIGURE 2: 28-PIN QFN, UQFN(1) DIAGRAM



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

TABLE 1-2: PIC18(L)F2XK22 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Number		Pin Name	Pin Type	Buffer Type	Description
PDIP, SOIC	QFN, UQFN				
25	22	RB4/IOC0/P1D/T5G/AN11			
		RB4	I/O	TTL	Digital I/O.
		IOC0	I	TTL	Interrupt-on-change pin.
		P1D	O	CMOS	Enhanced CCP1 PWM output.
		T5G	I	ST	Timer5 external clock gate input.
		AN11	I	Analog	Analog input 11.
26	23	RB5/IOC1/P2B/P3A/CCP3/T3CKI/T1G/AN13			
		RB5	I/O	TTL	Digital I/O.
		IOC1	I	TTL	Interrupt-on-change pin.
		P2B(1)	O	CMOS	Enhanced CCP2 PWM output.
		P3A(1)	O	CMOS	Enhanced CCP3 PWM output.
		CCP3(1)	I/O	ST	Capture 3 input/Compare 3 output/PWM 3 output.
		T3CKI(2)	I	ST	Timer3 clock input.
		T1G	I	ST	Timer1 external clock gate input.
		AN13	I	Analog	Analog input 13.
27	24	RB6/IOC2/TX2/CK2/PGC			
		RB6	I/O	TTL	Digital I/O.
		IOC2	I	TTL	Interrupt-on-change pin.
		TX2	O	—	EUSART asynchronous transmit.
		CK2	I/O	ST	EUSART synchronous clock (see related RXx/DTx).
		PGC	I/O	ST	In-Circuit Debugger and ICSP™ programming clock pin.
28	25	RB7/IOC3/RX2/DT2/PGD			
		RB7	I/O	TTL	Digital I/O.
		IOC3	I	TTL	Interrupt-on-change pin.
		RX2	I	ST	EUSART asynchronous receive.
		DT2	I/O	ST	EUSART synchronous data (see related TXx/CKx).
		PGD	I/O	ST	In-Circuit Debugger and ICSP™ programming data pin.
11	8	RC0/P2B/T3CKI/T3G/T1CKI/SOSCO			
		RC0	I/O	ST	Digital I/O.
		P2B(2)	O	CMOS	Enhanced CCP1 PWM output.
		T3CKI(1)	I	ST	Timer3 clock input.
		T3G	I	ST	Timer3 external clock gate input.
		T1CKI	I	ST	Timer1 clock input.
		SOSCO	O	—	Secondary oscillator output.
12	9	RC1/P2A/CCP2/SOSCI			
		RC1	I/O	ST	Digital I/O.
		P2A	O	CMOS	Enhanced CCP2 PWM output.
		CCP2(1)	I/O	ST	Capture 2 input/Compare 2 output/PWM 2 output.
		SOSCI	I	Analog	Secondary oscillator input.

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 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.

2.3 Register Definitions: Oscillator Control

REGISTER 2-1: OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0	R/W-0	R/W-1	R/W-1	R-q	R-0	R/W-0	R/W-0
IDLEN	IRCF<2:0>			OSTS(1)	HFIOFS	SCS<1:0>	
bit 7							bit 0

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

bit 7 **IDLEN:** Idle Enable bit
1 = Device enters Idle mode on SLEEP instruction
0 = Device enters Sleep mode on SLEEP instruction

bit 6-4 **IRCF<2:0>:** Internal RC Oscillator Frequency Select bits(2)
111 = HFINTOSC – (16 MHz)
110 = HFINTOSC/2 – (8 MHz)
101 = HFINTOSC/4 – (4 MHz)
100 = HFINTOSC/8 – (2 MHz)
011 = HFINTOSC/16 – (1 MHz)(3)

If INTSRC = 0 and MFIOSEL = 0:
010 = HFINTOSC/32 – (500 kHz)
001 = HFINTOSC/64 – (250 kHz)
000 = LFINTOSC – (31.25 kHz)

If INTSRC = 1 and MFIOSEL = 0:
010 = HFINTOSC/32 – (500 kHz)
001 = HFINTOSC/64 – (250 kHz)
000 = HFINTOSC/512 – (31.25 kHz)

If INTSRC = 0 and MFIOSEL = 1:
010 = MFINTOSC – (500 kHz)
001 = MFINTOSC/2 – (250 kHz)
000 = LFINTOSC – (31.25 kHz)

If INTSRC = 1 and MFIOSEL = 1:
010 = MFINTOSC – (500 kHz)
001 = MFINTOSC/2 – (250 kHz)
000 = MFINTOSC/16 – (31.25 kHz)

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit
1 = Device is running from the clock defined by FOSC<3:0> of the CONFIG1H register
0 = Device is running from the internal oscillator (HFINTOSC, MFINTOSC or LFINTOSC)

bit 2 **HFIOFS:** HFINTOSC Frequency Stable bit
1 = HFINTOSC frequency is stable
0 = HFINTOSC frequency is not stable

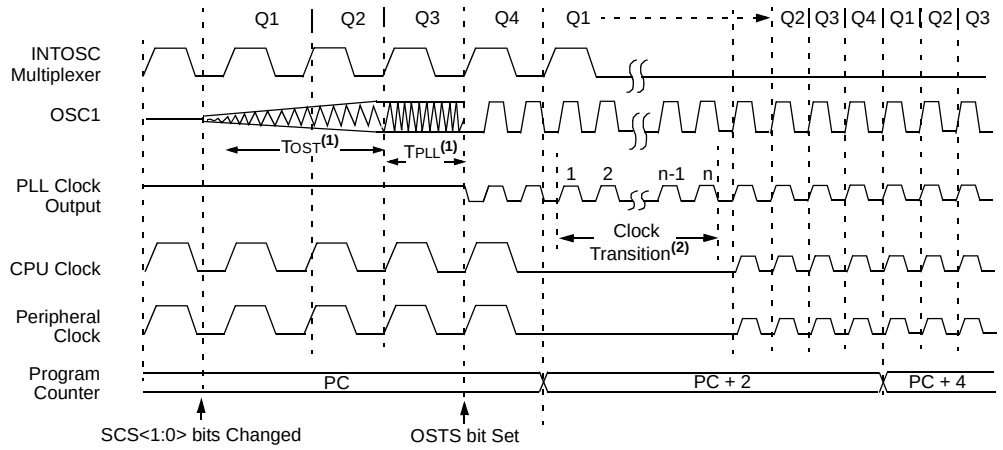
bit 1-0 **SCS<1:0>:** System Clock Select bit
1x = Internal oscillator block
01 = Secondary (SOSC) oscillator
00 = Primary clock (determined by FOSC<3:0> in CONFIG1H).

Note 1: Reset state depends on state of the IESO Configuration bit.
2: INTOSC source may be determined by the INTSRC bit in OSCTUNE and the MFIOSEL bit in OSCCON2.
3: Default output frequency of HFINTOSC on Reset.

TABLE 3-2: INTERNAL OSCILLATOR FREQUENCY STABILITY BITS

IRCF<2:0>	INTSRC	MFIOSEL	Selected Oscillator	Selected Oscillator Stable when:
000	0	x	LFINTOSC	LFIOFS = 1
000	1	0	HFINTOSC	HFIOFS = 1
000	1	1	MFINTOSC	MFIOFS = 1
010 or 001	x	0	HFINTOSC	HFIOFS = 1
010 or 001	x	1	MFINTOSC	MFIOFS = 1
011 – 111	x	x	HFINTOSC	HFIOFS = 1

FIGURE 3-3: TRANSITION TIMING FROM RC_RUN MODE TO PRI_RUN MODE



Note 1: TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.

2: Clock transition typically occurs within 2-4 TOSC.

6.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range.

A read from program memory is executed one byte at a time. A write to program memory is executed on blocks of 64 bytes at a time. Program memory is erased in blocks of 64 bytes at a time. A bulk erase operation cannot be issued from user code.

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

6.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

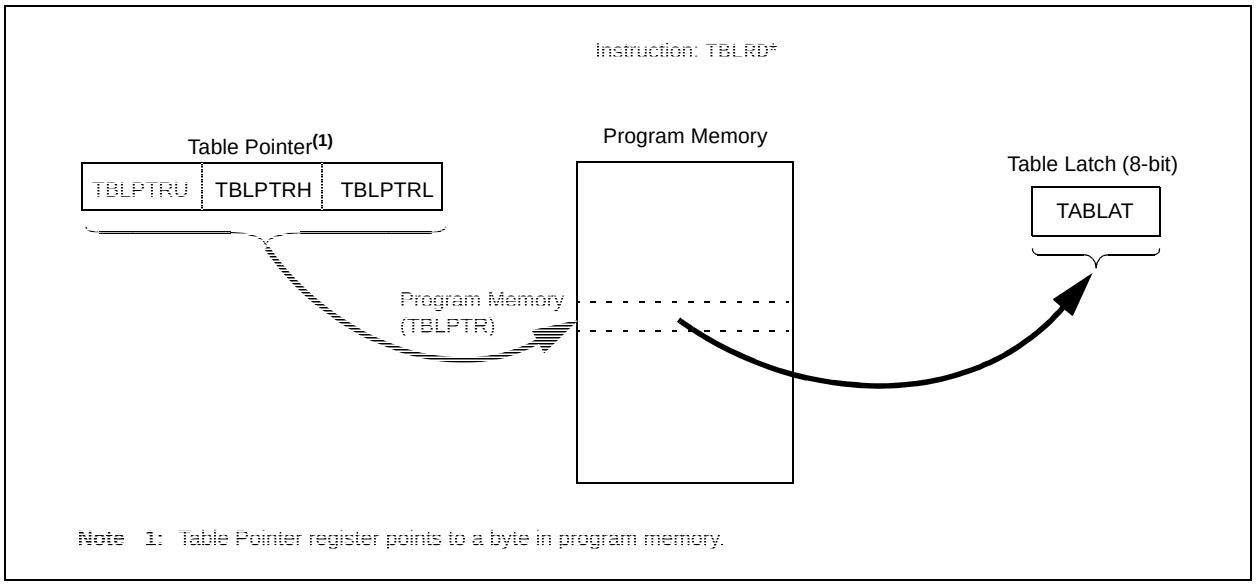
The program memory space is 16 bits wide, while the data RAM space is 8 bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

The table read operation retrieves one byte of data directly from program memory and places it into the TABLAT register. Figure 6-1 shows the operation of a table read.

The table write operation stores one byte of data from the TABLAT register into a write block holding register. The procedure to write the contents of the holding registers into program memory is detailed in **Section 6.6 "Writing to Flash Program Memory"**. Figure 6-2 shows the operation of a table write with program memory and data RAM.

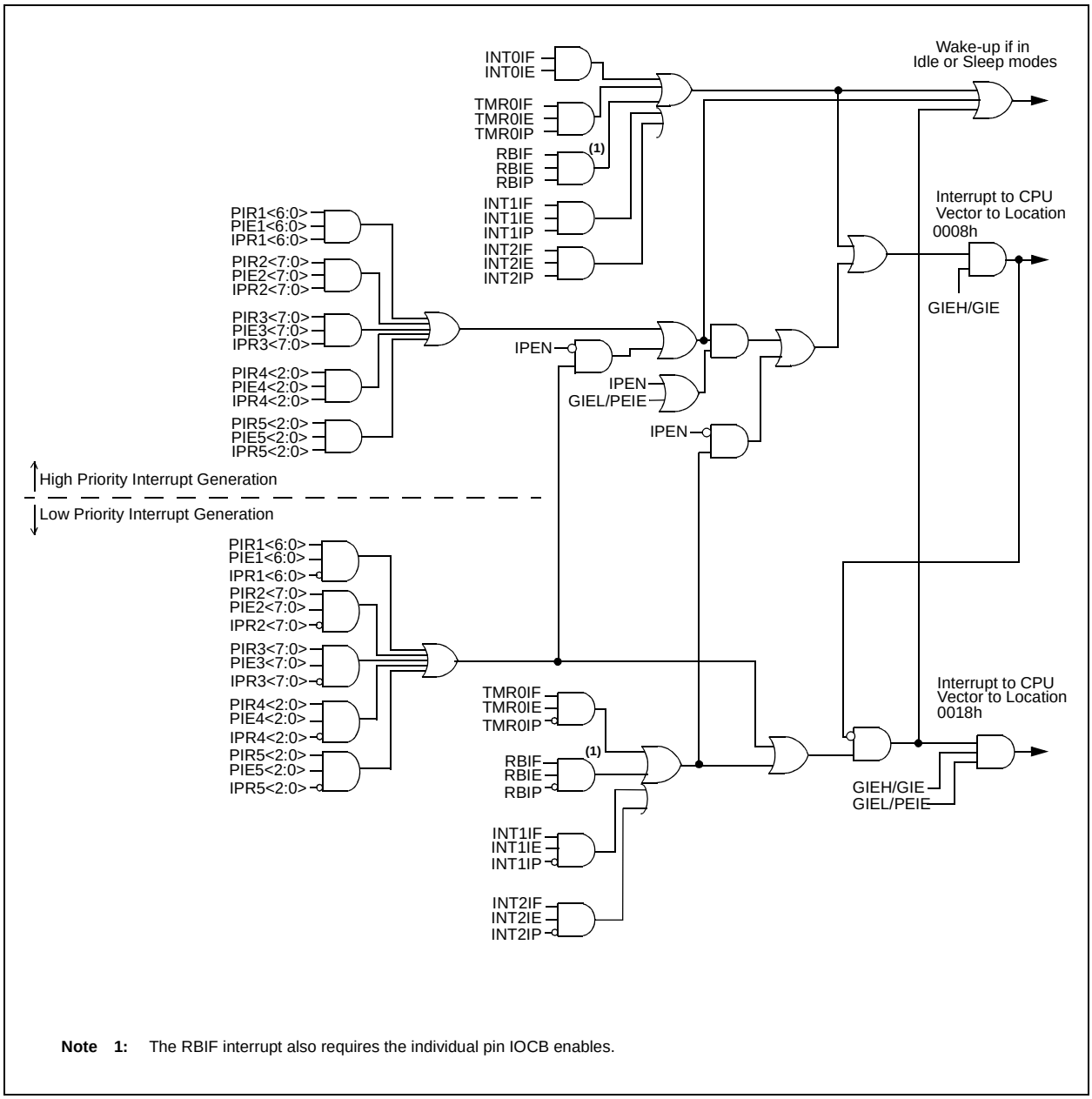
Table operations work with byte entities. Tables containing data, rather than program instructions, are not required to be word aligned. Therefore, a table can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word aligned.

FIGURE 6-1: TABLE READ OPERATION



Note:	Do not use the MOVFF instruction to modify any of the interrupt control registers while **any** interrupt is enabled. Doing so may cause erratic microcontroller behavior.

FIGURE 9-1: PIC18 INTERRUPT LOGIC



Note 1: The RBIF interrupt also requires the individual pin IOCB enables.

10.1.1 PORTA OUTPUT PRIORITY

Each PORTA pin is multiplexed with other functions. The pins, their combined functions and their output priorities are briefly described here. For additional information, refer to the appropriate section in this data sheet.

When multiple outputs are enabled, the actual pin control goes to the peripheral with the higher priority. Table 10-4 lists the PORTA pin functions from the highest to the lowest priority.

Analog input functions, such as ADC and comparator, are not shown in the priority lists.

These inputs are active when the I/O pin is set for Analog mode using the ANSELx registers. Digital output functions may control the pin when it is in Analog mode with the priority shown below.

TABLE 10-4: PORT PIN FUNCTION PRIORITY

Port bit	Port Function Priority by Port Pin				
	PORTA	**PORTB**	**PORTC**	**PORTD(2)**	**PORTE(2)**
0	RA0	CCP4(1)	SOSCO	SCL2	CCP3(8)
		RB0	P2B(6)	SCK2	P3A(8)
			RC0	RD0	RE0
1	RA1	SCL2(1)	SOSCI	SDA2	P3B
		SCK2(1)	CCP2(3)	CCP4	RE1
		P1C(1)	P2A(3)	RD1	
		RB1	RC1		
2	RA2	SDA2(1)	CCP1	P2B	CCP5
		P1B(1)	P1A	RD2(4)	RE2
		RB2	CTPLS		
			RC2		
3	RA3	SDO2(1)	SCL1	P2C	$\overline{\text{MCLR}}$
		CCP2(6)	SCK1	RD3	V_{PP}
		P2A(6)	RC3		RE3
		RB3			
4	SRQ	P1D(1)	SDA1	SDO2	
	C1OUT	RB4	RC4	P2D	
	CCP5(1)			RD4	
	RA4				

Note 1: PIC18(L)F2XK22 devices.
2: PIC18(L)F4XK22 devices.
3: Function default pin.
4: Function default pin (28-pin devices).
5: Function default pin (40/44-pin devices).
6: Function alternate pin.
7: Function alternate pin (28-pin devices).
8: Function alternate pin (40/44-pin devices)

TABLE 10-4: PORT PIN FUNCTION PRIORITY (CONTINUED)

Port bit	Port Function Priority by Port Pin				
	PORTA	**PORTB**	**PORTC**	**PORTD(2)**	**PORTE(2)**
5	SRNQ	CCP3(3)	SDO1	P1B	
	C2OUT	P3A(3)	RC5	RD5	
	RA5	P2B(1)(4)			
		RB5			
6	OSC2	PGC	TX1/CK1	TX2/CK2	
	CLKO	TX2/CK2(1)	CCP3(1)(7)	P1C	
	RA6	RB6	P3A(1)(7)	RD6	
		ICDCK	RC6		
7	RA7				
	OSC1	PGD	RX1/DT1	RX2/DT2	
	RA7	RX2/DT2(1)	P3B(1)	P1D	
		RB7	RC7	RD7	
		ICDDT			

Note 1: PIC18(L)F2XK22 devices.
2: PIC18(L)F4XK22 devices.
3: Function default pin.
4: Function default pin (28-pin devices).
5: Function default pin (40/44-pin devices).
6: Function alternate pin.
7: Function alternate pin (28-pin devices).
8: Function alternate pin (40/44-pin devices)

TABLE 10-6: REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	150
ECCP2AS	CCP2ASE	CCP2AS<2:0>			PSS2AC<1:0>		PSS2BD<1:0>		202
CCP2CON	P2M<1:0>		DC2B<1:0>		CCP2M<3:0>				198
ECCP3AS	CCP3ASE	CCP3AS<2:0>			PSS3AC<1:0>		PSS3BD<1:0>		202
CCP3CON	P3M<1:0>		DC3B<1:0>		CCP3M<3:0>				198
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	109
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	110
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	111
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—	153
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	152
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	148
SLRCON	—	—	—	SLRE(1)	SLRD(1)	SLRC	SLRB	SLRA	153
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/$\overline{\text{DONE}}$	T1GVAL	T1GSS<1:0>		167
T3CON	TMR3CS<1:0>		T3CKPS<1:0>		T3SOSCEN	$\overline{\text{T3SYNC}}$	T3RD16	TMR3ON	166
T5GCON	TMR5GE	T5GPOL	T5GTM	T5GSPM	T5GGO/$\overline{\text{DONE}}$	T5GVAL	T5GSS<1:0>		167
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	151
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	152

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for PORTB.

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

TABLE 10-7: CONFIGURATION REGISTERS ASSOCIATED WITH PORTB

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
CONFIG4L	$\overline{\text{DEBUG}}$	XINST	—	—	—	LVP(1)	—	STRVEN	349

Legend: — = unimplemented locations, read as '0'. Shaded bits are not used for PORTB.

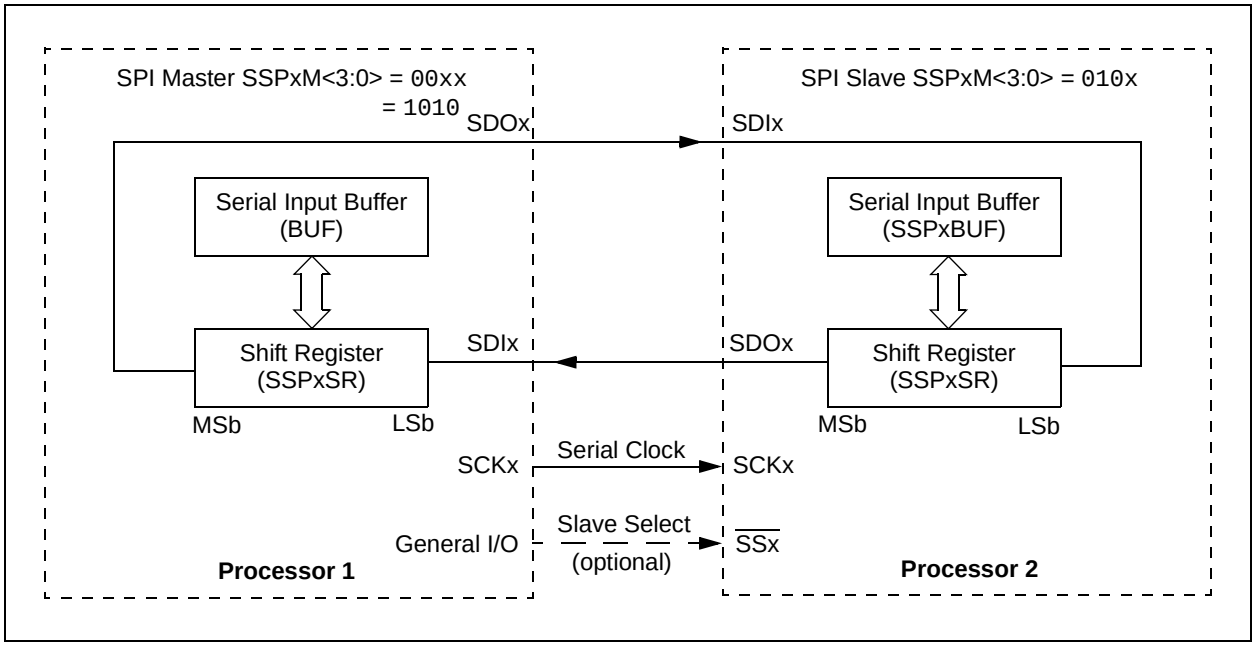
Note 1: Can only be changed when in high voltage programming mode.

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

The MSSPx consists of a transmit/receive shift register (SSPxSR) and a buffer register (SSPxBUF). The SSPxSR shifts the data in and out of the device, MSb first. The SSPxBUF holds the data that was written to the SSPxSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPxBUF register. Then, the Buffer Full Detect bit, BF of the SSPxSTAT register, and the interrupt flag bit, SSPxIF, are set. This double-buffering of the received data (SSPxBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPxBUF register during transmission/ reception of data will be ignored and the write collision detect bit, WCOL of the SSPxCON1 register, will be set. User software must clear the WCOL bit to allow the following write(s) to the SSPxBUF register to complete successfully.

When the application software is expecting to receive valid data, the SSPxBUF should be read before the next byte of data to transfer is written to the SSPxBUF. The Buffer Full bit, BF of the SSPxSTAT register, indicates when SSPxBUF has been loaded with the received data (transmission is complete). When the SSPxBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSPx interrupt is used to determine when the transmission/reception has completed. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur.

FIGURE 15-5: SPI MASTER/SLAVE CONNECTION



15.6.13 MULTI -MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDAx pin, arbitration takes place when the master outputs a '1' on SDAx, by letting SDAx float high and another master asserts a '0'. When the SCLx pin floats high, data should be stable. If the expected data on SDAx is a '1' and the data sampled on the SDAx pin is '0', then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLxIF, and reset the I^2C port to its Idle state (Figure 15-32).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDAx and SCLx lines are deasserted and the SSPxBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

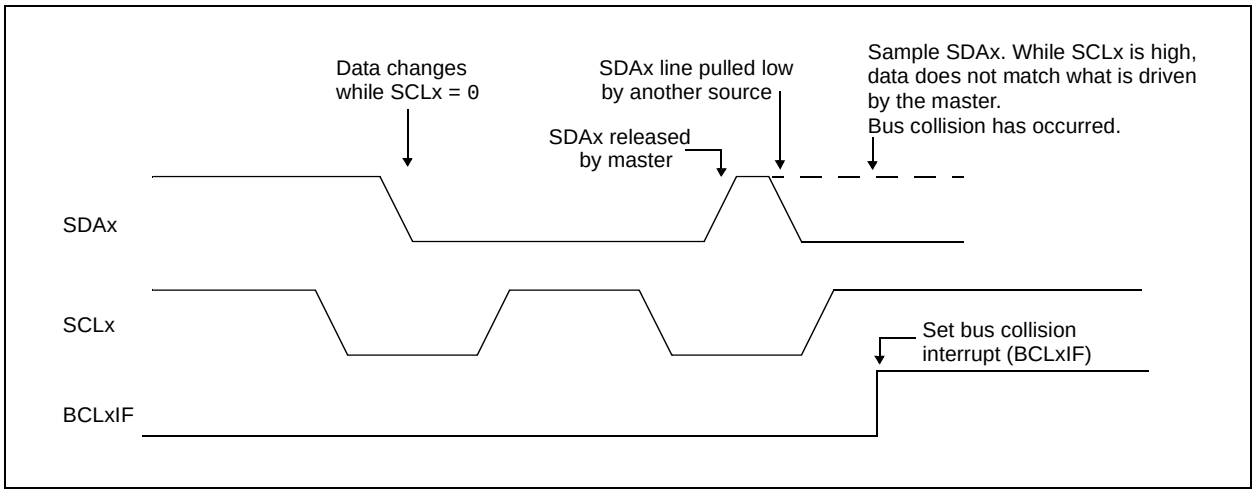
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDAx and SCLx lines are deasserted and the respective control bits in the SSPxCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDAx and SCLx pins. If a Stop condition occurs, the SSPxIF bit will be set.

A write to the SSPxBUF will start the transmission of data at the first data bit, regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I^2C bus can be taken when the P bit is set in the SSPxSTAT register, or the bus is Idle and the S and P bits are cleared.

FIGURE 15-32: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



REGISTER 16-3: BAUDCONx: BAUD RATE CONTROL REGISTER

R/W-0	R-1	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	DTRXP	CKTXP	BRG16	—	WUE	ABDEN
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 **ABDOVF:** Auto-Baud Detect Overflow bit
Asynchronous mode:
1 = Auto-baud timer overflowed
0 = Auto-baud timer did not overflow
Synchronous mode:
Don't care

bit 6 **RCIDL**: Receive Idle Flag bit
Asynchronous mode:
1 = Receiver is Idle
0 = Start bit has been detected and the receiver is active
Synchronous mode:
Don't care

bit 5 **DTRXP**: Data/Receive Polarity Select bit
Asynchronous mode:
1 = Receive data (RXx) is inverted (active-low)
0 = Receive data (RXx) is not inverted (active-high)
Synchronous mode:
1 = Data (DTx) is inverted (active-low)
0 = Data (DTx) is not inverted (active-high)

bit 4 **CKTXP**: Clock/Transmit Polarity Select bit
Asynchronous mode:
1 = Idle state for transmit (TXx) is low
0 = Idle state for transmit (TXx) is high
Synchronous mode:
1 = Data changes on the falling edge of the clock and is sampled on the rising edge of the clock
0 = Data changes on the rising edge of the clock and is sampled on the falling edge of the clock

bit 3 **BRG16:** 16-bit Baud Rate Generator bit
1 = 16-bit Baud Rate Generator is used (SPBRGHx:SPBRGx)
0 = 8-bit Baud Rate Generator is used (SPBRGx)

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

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = Receiver is waiting for a falling edge. No character will be received but RCxIF will be set on the falling edge. WUE will automatically clear on the rising edge.
0 = Receiver is operating normally
Synchronous mode:
Don't care

bit 0 **ABDEN**: Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Auto-Baud Detect mode is enabled (clears when auto-baud is complete)
0 = Auto-Baud Detect mode is disabled
Synchronous mode:
Don't care

16.4.1 AUTO-BAUD DETECT

The EUSART module supports automatic detection and calibration of the baud rate.

In the Auto-Baud Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RXx signal, the RXx signal is timing the BRG. The Baud Rate Generator is used to time the period of a received 55h (ASCII “U”) which is the Sync character for the LIN bus. The unique feature of this character is that it has five rising edges including the Stop bit edge.

Setting the ABDEN bit of the BAUDCONx register starts the auto-baud calibration sequence (**Section 16.4.2 “Auto-baud Overflow”**). While the ABD sequence takes place, the EUSART state machine is held in Idle. On the first rising edge of the receive line, after the Start bit, the SPBRGx begins counting up using the BRG counter clock as shown in Table 16-6. The fifth rising edge will occur on the RXx/DTx pin at the end of the eighth bit period. At that time, an accumulated value totaling the proper BRG period is left in the SPBRGHx:SPBRGx register pair, the ABDEN bit is automatically cleared, and the RCxIF interrupt flag is set. A read operation on the RCREGx needs to be performed to clear the RCxIF interrupt. RCREGx content should be discarded. When calibrating for modes that do not use the SPBRGHx register the user can verify that the SPBRGx register did not overflow by checking for 00h in the SPBRGHx register.

The BRG auto-baud clock is determined by the BRG16 and BRGH bits as shown in Table 16-6. During ABD, both the SPBRGHx and SPBRGx registers are used as a 16-bit counter, independent of the BRG16 bit setting. While calibrating the baud rate period, the SPBRGHx and SPBRGx registers are clocked at 1/8th the BRG base clock rate. The resulting byte measurement is the average bit time when clocked at full speed.

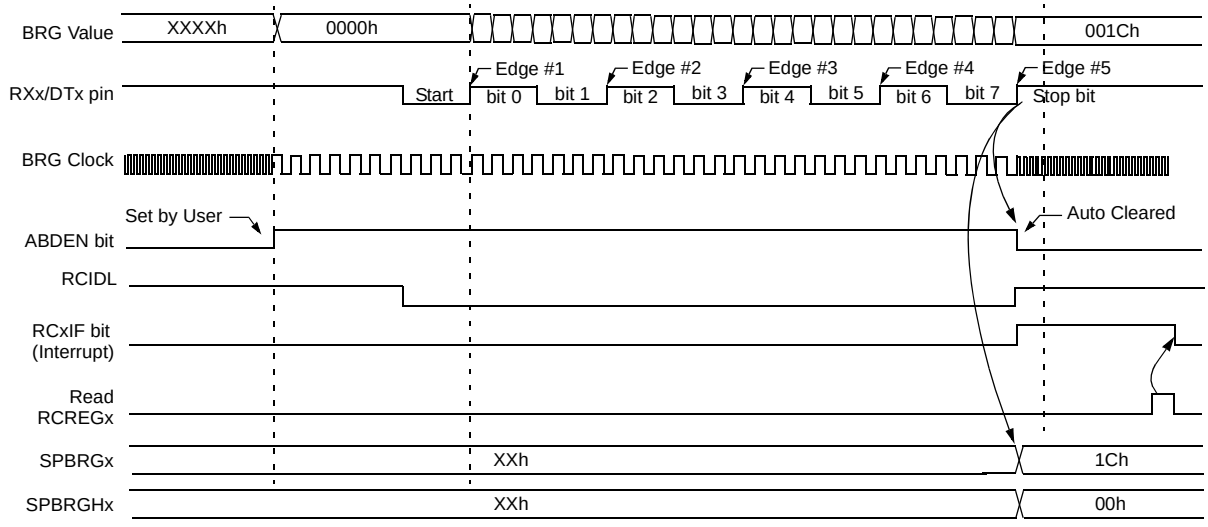
Note 1:	If the WUE bit is set with the ABDEN bit, auto-baud detection will occur on the byte following the Break character (see **Section 16.4.3 “Auto-Wake-up on Break”**).
2:	It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible.
3:	During the auto-baud process, the auto-baud counter starts counting at one. Upon completion of the auto-baud sequence, to achieve maximum accuracy, subtract one from the SPBRGHx:SPBRGx register pair.

TABLE 16-6: BRG COUNTER CLOCK RATES

BRG16	BRGH	BRG Base Clock	BRG ABD Clock
0	0	FOSC/64	FOSC/512
0	1	FOSC/16	FOSC/128
1	0	FOSC/16	FOSC/128
1	1	FOSC/4	FOSC/32

Note: During the ABD sequence, SPBRGx and SPBRGHx registers are both used as a 16-bit counter, independent of BRG16 setting.

FIGURE 16-6: AUTOMATIC BAUD RATE CALIBRATION



Note 1: The ABD sequence requires the **EUSART** module to be configured in Asynchronous mode.

16.5.1.6 Synchronous Master Reception

Data is received at the RXx/DTx pin. The RXx/DTx pin output driver must be disabled by setting the corresponding TRIS bits when the EUSART is configured for synchronous master receive operation.

In Synchronous mode, reception is enabled by setting either the Single Receive Enable bit (SREN of the RCSTAx register) or the Continuous Receive Enable bit (CREN of the RCSTAx register).

When SREN is set and CREN is clear, only as many clock cycles are generated as there are data bits in a single character. The SREN bit is automatically cleared at the completion of one character. When CREN is set, clocks are continuously generated until CREN is cleared. If CREN is cleared in the middle of a character the CK clock stops immediately and the partial character is discarded. If SREN and CREN are both set, then SREN is cleared at the completion of the first character and CREN takes precedence.

To initiate reception, set either SREN or CREN. Data is sampled at the RXx/DTx pin on the trailing edge of the TXx/CKx clock pin and is shifted into the Receive Shift Register (RSR). When a complete character is received into the RSR, the RCxIF bit is set and the character is automatically transferred to the two character receive FIFO. The Least Significant eight bits of the top character in the receive FIFO are available in RCREGx. The RCxIF bit remains set as long as there are un-read characters in the receive FIFO.

16.5.1.7 Slave Clock

Synchronous data transfers use a separate clock line, which is synchronous with the data. A device configured as a slave receives the clock on the TXx/CKx line. The TXx/CKx pin output driver must be disabled by setting the associated TRIS bit when the device is configured for synchronous slave transmit or receive operation. Serial data bits change on the leading edge to ensure they are valid at the trailing edge of each clock. One data bit is transferred for each clock cycle. Only as many clock cycles should be received as there are data bits.

16.5.1.8 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before RCREGx is read to access the FIFO. When this happens the OERR bit of the RCSTAx register is set. Previous data in the FIFO will not be overwritten. The two characters in the FIFO buffer can be read, however, no additional characters will be received until the error is cleared. The OERR bit can only be cleared by clearing the overrun condition. If the overrun error occurred when the SREN bit is set and CREN is clear then the error is cleared by reading RCREGx.

If the overrun occurred when the CREN bit is set then the error condition is cleared by either clearing the CREN bit of the RCSTAx register or by clearing the SPEN bit which resets the EUSART.

16.5.1.9 Receiving 9-bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCSTAx register is set the EUSART will shift 9-bits into the RSR for each character received. The RX9D bit of the RCSTAx register is the ninth, and Most Significant, data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCREGx.

16.5.1.10 Synchronous Master Reception Setup:

1. Initialize the SPBRGHx, SPBRGx register pair for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Set the RXx/DTx and TXx/CKx TRIS controls to ‘1’.
3. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC. Disable RXx/DTx and TXx/CKx output drivers by setting the corresponding TRIS bits.
4. Ensure bits CREN and SREN are clear.
5. If using interrupts, set the GIE/GIEH and PEIE/GIEL bits of the INTCON register and set RCxIE.
6. If 9-bit reception is desired, set bit RX9.
7. Start reception by setting the SREN bit or for continuous reception, set the CREN bit.
8. Interrupt flag bit RCxIF will be set when reception of a character is complete. An interrupt will be generated if the enable bit RCxIE was set.
9. Read the RCSTAx register to get the ninth bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCREGx register.
11. If an overrun error occurs, clear the error by either clearing the CREN bit of the RCSTAx register or by clearing the SPEN bit which resets the EUSART.

17.2 ADC Operation

17.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Setting the GO/DONE bit of the ADCON0 register to a '1' will, depending on the ACQT bits of the ADCON2 register, either immediately start the Analog-to-Digital conversion or start an acquisition delay followed by the Analog-to-Digital conversion.

Figure 17-3 shows the operation of the A/D converter after the GO bit has been set and the ACQT<2:0> bits are cleared. A conversion is started after the following instruction to allow entry into SLEEP mode before the conversion begins.

Figure 17-4 shows the operation of the A/D converter after the GO bit has been set and the ACQT<2:0> bits are set to '010' which selects a 4 TAD acquisition time before the conversion starts.

Note:	The GO/DONE bit should not be set in the same instruction that turns on the ADC. Refer to **Section 17.2.10 "A/D Conversion Procedure"**.

FIGURE 17-3: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 000, TACQ = 0)

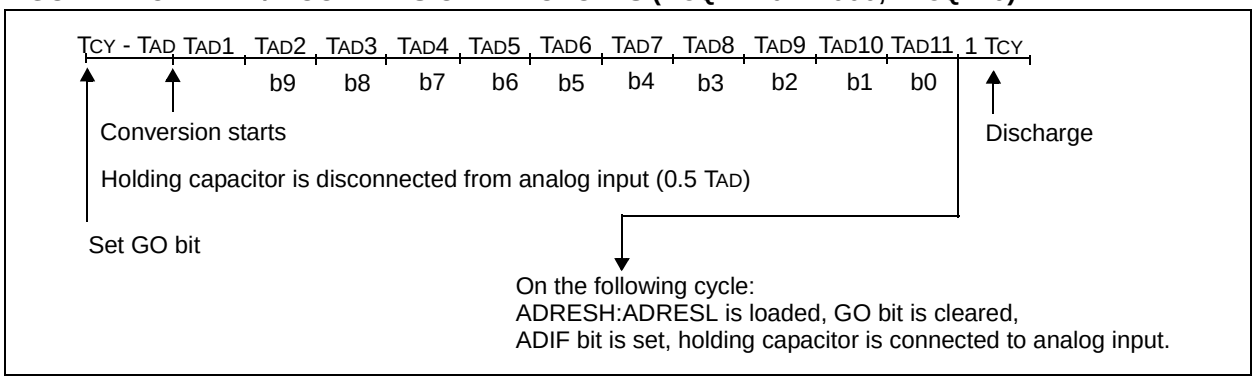
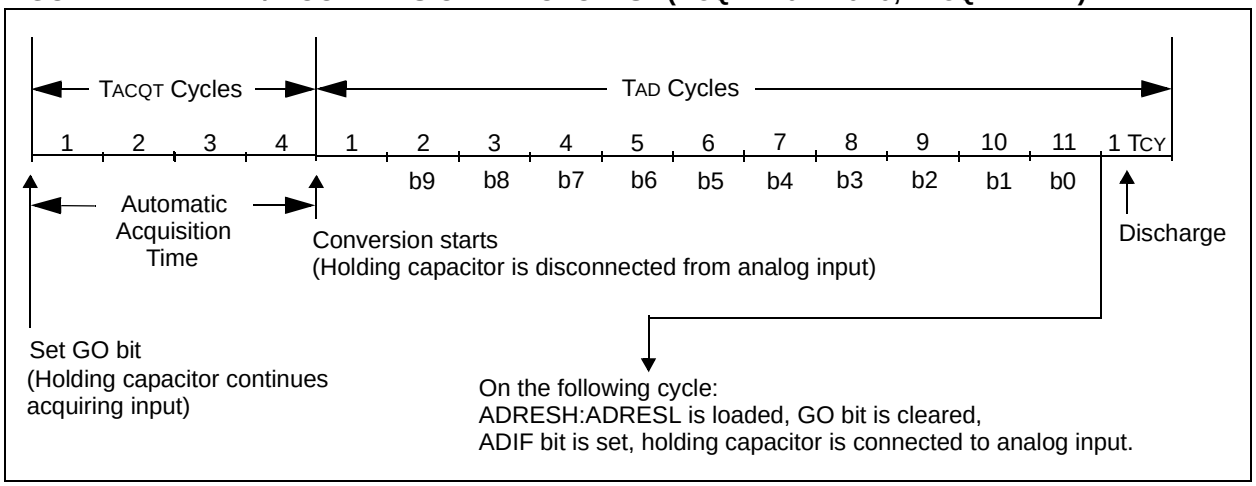


FIGURE 17-4: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



18.4 Comparator Interrupt Operation

The comparator interrupt flag will be set whenever there is a change in the output value of the comparator. Changes are recognized by means of a mismatch circuit which consists of two latches and an exclusive-or gate (see Figure 18-2). The first latch is updated with the comparator output value, when the CMxCON0 register is read or written. The value is latched on the third cycle of the system clock, also known as Q3. This first latch retains the comparator value until another read or write of the CMxCON0 register occurs or a Reset takes place. The second latch is updated with the comparator output value on every first cycle of the system clock, also known as Q1. When the output value of the comparator changes, the second latch is updated and the output values of both latches no longer match one another, resulting in a mismatch condition. The latch outputs are fed directly into the inputs of an exclusive-or gate. This mismatch condition is detected by the exclusive-or gate and sent to the interrupt circuitry. The mismatch condition will persist until the first latch value is updated by performing a read of the CMxCON0 register or the comparator output returns to the previous state.

Note 1: A write operation to the CMxCON0 register will also clear the mismatch condition because all writes include a read operation at the beginning of the write cycle.

2: Comparator interrupts will operate correctly regardless of the state of CxOE.

When the mismatch condition occurs, the comparator interrupt flag is set. The interrupt flag is triggered by the edge of the changing value coming from the exclusive-or gate. This means that the interrupt flag can be reset once it is triggered without the additional step of reading or writing the CMxCON0 register to clear the mismatch latches. When the mismatch registers are cleared, an interrupt will occur upon the comparator's return to the previous state, otherwise no interrupt will be generated.

Software will need to maintain information about the status of the comparator output, as read from the CMxCON0 register, or CM2CON1 register, to determine the actual change that has occurred. See Figures 18-3 and 18-4.

The CxIF bit of the PIR2 register is the comparator interrupt flag. This bit must be reset by software by clearing it to '0'. Since it is also possible to write a '1' to this register, an interrupt can be generated.

In mid-range Compatibility mode the CxIE bit of the PIE2 register and the PEIE/GIEL and GIE/GIEH bits of the INTCON register must all be set to enable comparator interrupts. If any of these bits are cleared, the interrupt is not enabled, although the CxIF bit of the PIR2 register will still be set if an interrupt condition occurs.

18.4.1 PRESETTING THE MISMATCH LATCHES

The comparator mismatch latches can be preset to the desired state before the comparators are enabled. When the comparator is off the CxPOL bit controls the CxOUT level. Set the CxPOL bit to the desired CxOUT non-interrupt level while the CxON bit is cleared. Then, configure the desired CxPOL level in the same instruction that the CxON bit is set. Since all register writes are performed as a read-modify-write, the mismatch latches will be cleared during the instruction read phase and the actual configuration of the CxON and CxPOL bits will be occur in the final write phase.

FIGURE 18-3: COMPARATOR INTERRUPT TIMING W/O CMxCON0 READ

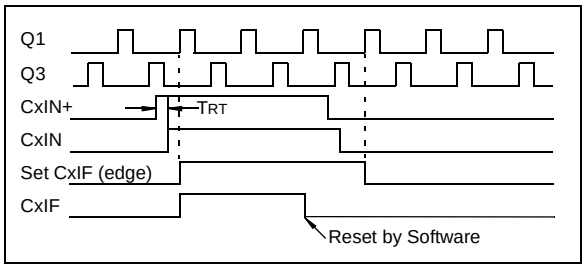
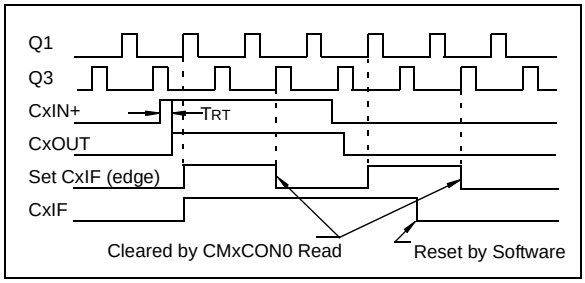


FIGURE 18-4: COMPARATOR INTERRUPT TIMING WITH CMxCON0 READ



Note 1: If a change in the CMxCON0 register (CxOUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CxIF interrupt flag of the PIR2 register may not get set.

2: When either comparator is first enabled, bias circuitry in the comparator module may cause an invalid output from the comparator until the bias circuitry is stable. Allow about 1 μs for bias settling then clear the mismatch condition and interrupt flags before enabling comparator interrupts.

REGISTER 19-3: CTMUICON: CTMU CURRENT 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
ITRIM<5:0>						IRNG<1:0>	
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 **ITRIM<5:0>:** Current Source Trim bits
011111 = Maximum positive change from nominal current
011110
.
.
.
000001 = Minimum positive change from nominal current
000000 = Nominal current output specified by IRNG<1:0>
111111 = Minimum negative change from nominal current
.
.
.
100010
100001 = Maximum negative change from nominal current

bit 1-0 **IRNG<1:0>:** Current Source Range Select bits (see Table 27-4)
11 = 100 × Base current
10 = 10 × Base current
01 = Base current level
00 = Current source disabled

TABLE 19-1: REGISTERS ASSOCIATED WITH CTMU MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
CTMUCONH	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	323
CTMUCONL	EDG2POL	EDG2SEL<1:0>		EDG1POL	EDG1SEL<1:0>		EDG2STAT	EDG1STAT	324
CTMUICON	ITRIM<5:0>						IRNG<1:0>		325
IPR3	SSP2IP	BCL2IP	RC2IP	TX2IP	CTMUIP	TMR5GIP	TMR3GIP	TMR1GIP	123
PIE3	SSP2IE	BCL2IE	RC2IE	TX2IE	CTMUIE	TMR5GIE	TMR3GIE	TMR1GIE	119
PIR3	SSP2IF	BCL2IF	RC2IF	TX2IF	CTMUIF	TMR5GIF	TMR3GIF	TMR1GIF	114
PMD2	—	—	—	—	CTMUMD	CMP2MD	CMP1MD	ADCMD	54

Legend: — = unimplemented, read as '0'. Shaded bits are not used during CTMU operation.

FIGURE 22-1: DIGITAL-TO-ANALOG CONVERTER BLOCK DIAGRAM

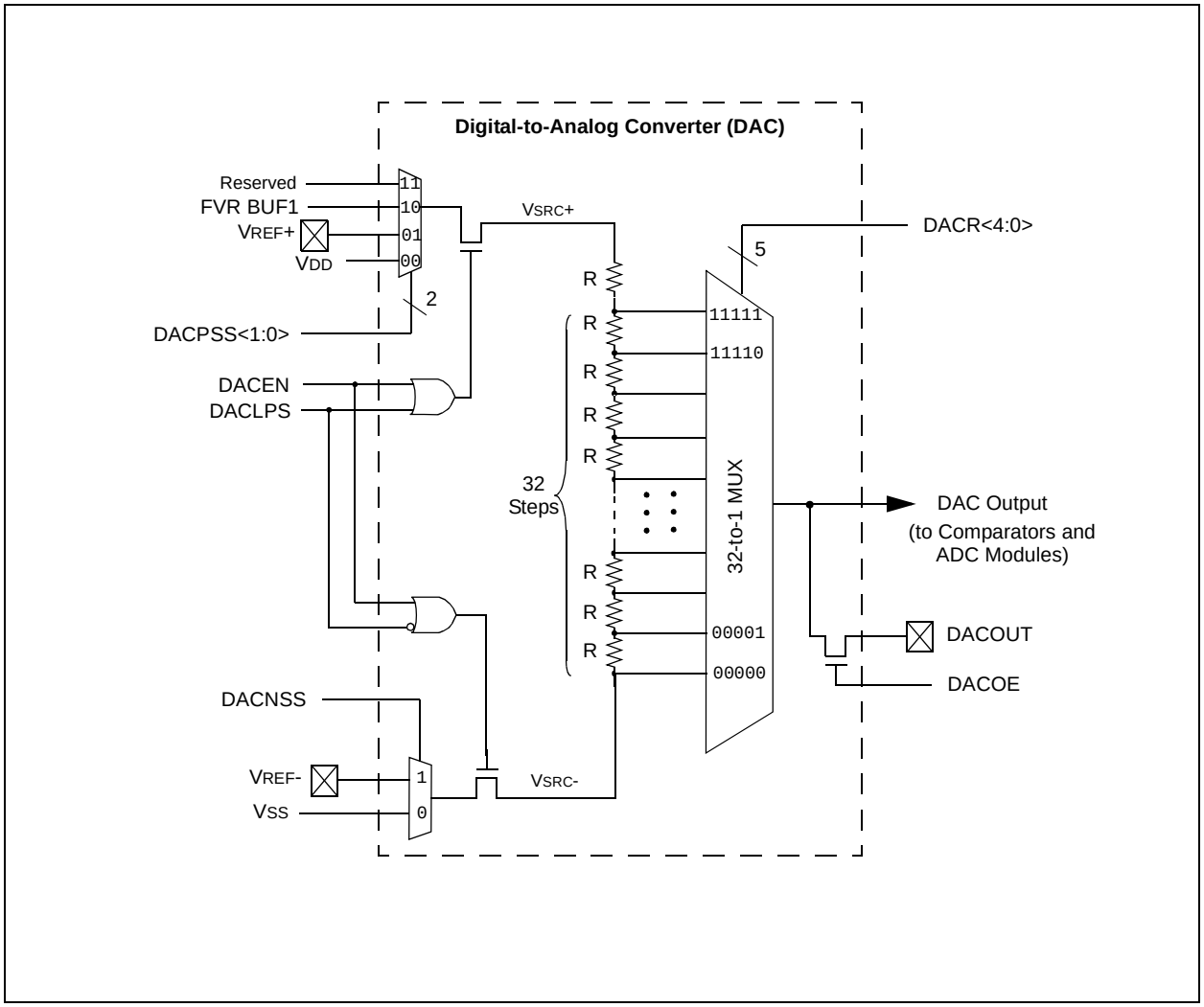
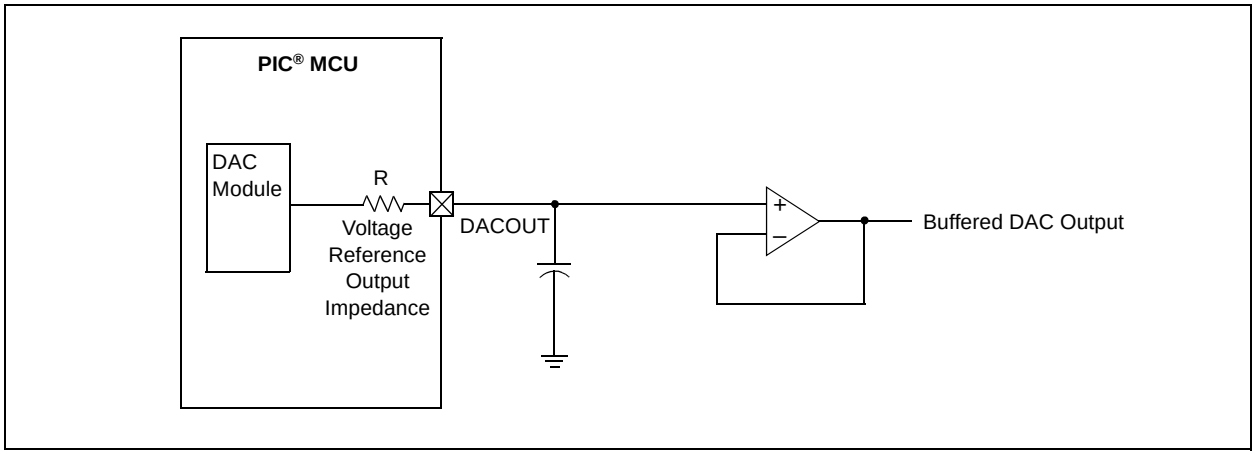


FIGURE 22-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE



REGISTER 22-2: VREFCON2: VOLTAGE REFERENCE CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DACR<4:0>				
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

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

bit 4-0 **DACR<4:0>:** DAC Voltage Output Select bits

$V_{OUT} = ((V_{SRC}+) - (V_{SRC}-))*(DACR<4:0>/(2^5)) + V_{SRC}-$

TABLE 22-1: REGISTERS ASSOCIATED WITH DAC MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
VREFCON0	FVREN	FVRST	FVRS<1:0>		—	—	—	—	332
VREFCON1	DACEN	DACLPS	DACOE	—	DACPSS<1:0>		—	DACNSS	335
VREFCON2	—	—	—	DACR<4:0>					336

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

24.5 Program Verification and Code Protection

The overall structure of the code protection on the PIC18 Flash devices differs significantly from other PIC microcontroller devices.

The user program memory is divided into three or five blocks, depending on the device. One of these is a Boot Block of 0.5K or 2K bytes, depending on the device. The remainder of the memory is divided into individual blocks on binary boundaries.

Each of the blocks has three code protection bits associated with them. They are:

- Code-Protect bit (CPn)
- Write-Protect bit (WRTn)
- External Block Table Read bit (EBTRn)

Figure 24-2 shows the program memory organization for 8, 16 and 32-Kbyte devices and the specific code protection bit associated with each block. The actual locations of the bits are summarized in Table 24-5.

FIGURE 24-2: CODE-PROTECTED PROGRAM MEMORY FOR PIC18(L)F2X/4XK22

MEMORY SIZE/DEVICE				Block Code Protection Controlled By:
8 Kbytes (PIC18(L)FX3K22)	16 Kbytes (PIC18(L)FX4K22)	32 Kbytes (PIC18(L)FX5K22)	64 Kbytes (PIC18(L)FX6K22)	
Boot Block (000h-1FFh)	Boot Block (000h-7FFh)	Boot Block (000h-7FFh)	Boot Block (000h-7FFh)	CPB, WRTB, EBTRB
Block 0 (200h-FFFh)	Block 0 (800h-1FFFh)	Block 0 (800h-1FFFh)	Block 0 (800h-3FFFh)	CP0, WRT0, EBTR0
Block 1 (1000h-1FFFh)	Block 1 (2000h-3FFFh)	Block 1 (2000h-3FFFh)	Block 1 (4000h-7FFFh)	CP1, WRT1, EBTR1
Unimplemented Read '0's (2000h-1FFFFFh)	Unimplemented Read '0's (4000h-1FFFFFh)	Block 2 (4000h-5FFFh)	Block 2 (8000h-BFFFh)	CP2, WRT2, EBTR2
		Block 3 (6000h-7FFFh)	Block 3 (C000h-FFFFh)	CP3, WRT3, EBTR3
		Unimplemented Read '0's (8000h-1FFFFFh)	Unimplemented Read '0's (10000h-1FFFFFh)	(Unimplemented Memory Space)

TABLE 24-5: CONFIGURATION REGISTERS ASSOCIATED WITH CODE PROTECTION

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
300008h	CONFIG5L	—	—	—	—	CP3(1)	CP2(1)	CP1	CP0
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—
30000Ah	CONFIG6L	—	—	—	—	WRT3(1)	WRT2(1)	WRT1	WRT0
30000Bh	CONFIG6H	WRTD	WRTB	WRTC(2)	—	—	—	—	—
30000Ch	CONFIG7L	—	—	—	—	EBTR3(1)	EBTR2(1)	EBTR1	EBTR0
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—

Legend: Shaded bits are unimplemented.

Note 1: Available on PIC18(L)FX5K22 and PIC18(L)FX6K22 devices only.

2: In user mode, this bit is read-only and cannot be self-programmed.