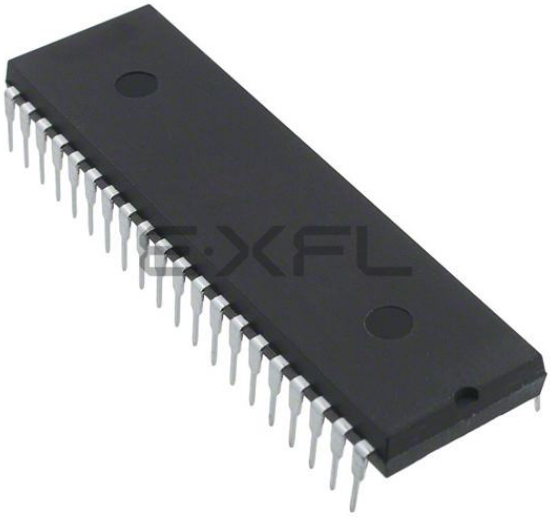


Welcome to E-XFL.COM

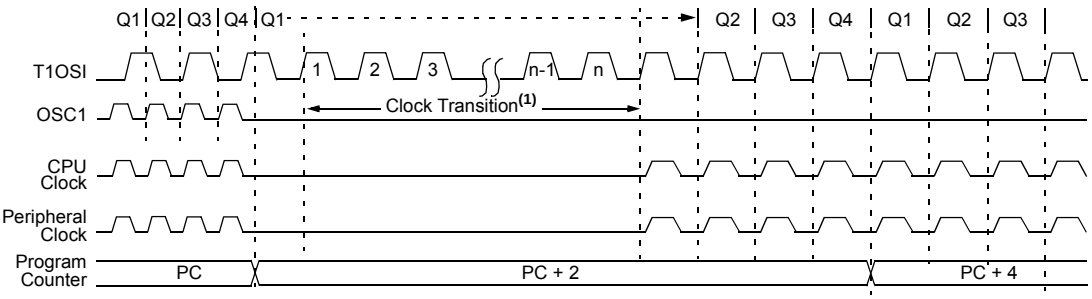
What is "Embedded - Microcontrollers"?

"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"

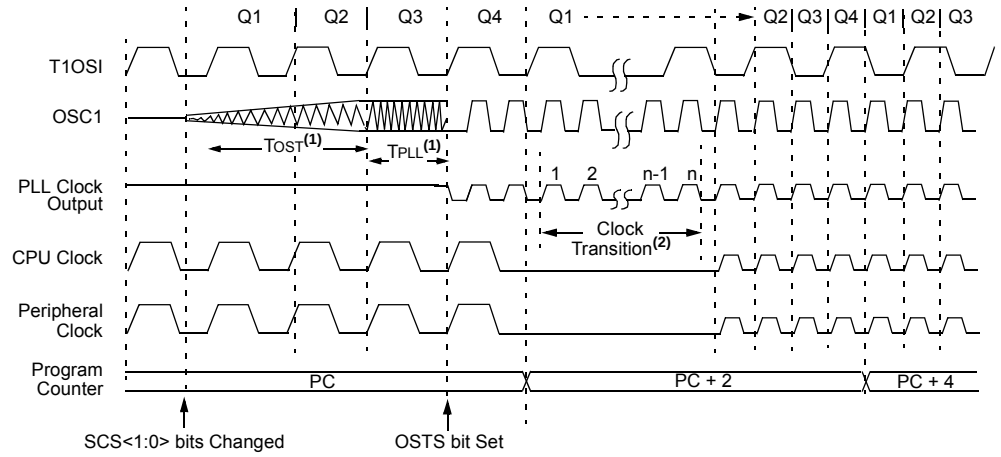
Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, Power Control PWM, QEI, POR, PWM, WDT
Number of I/O	36
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 9x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf4431-i-p

FIGURE 4-1: TRANSITION TIMING FOR ENTRY TO SEC_RUN MODE



Note 1: Clock transition typically occurs within 2-4 TOSC.

FIGURE 4-2: TRANSITION TIMING FROM SEC_RUN MODE TO PRI_RUN MODE (HSPLL)



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.

4.2.3 RC_RUN MODE

In RC_RUN mode, the CPU and peripherals are clocked from the internal oscillator block using the INTOSC multiplexer. In this mode, the primary clock is shut down. When using the INTRC source, this mode provides the best power conservation of all the Run modes, while still executing code. It works well for user applications which are not highly timing-sensitive or do not require high-speed clocks at all times.

If the primary clock source is the internal oscillator block (either INTRC or INTOSC), there are no distinguishable differences between PRI_RUN and RC_RUN modes during execution. However, a clock switch delay will occur during entry to and exit from RC_RUN mode. Therefore, if the primary clock source is the internal oscillator block, the use of RC_RUN mode is not recommended.

This mode is entered by setting the SCS1 bit to '1'. Although it is ignored, it is recommended that the SCS0 bit also be cleared; this is to maintain software compatibility with future devices. When the clock source is switched to the INTOSC multiplexer (see Figure 4-3), the primary oscillator is shut down and the OSTS bit is cleared. The IRCF bits may be modified at any time to immediately change the clock speed.

Note:	Caution should be used when modifying a single IRCF bit. If VDD is less than 3V, it is possible to select a higher clock speed than is supported by the low VDD. Improper device operation may result if the VDD/FOSC specifications are violated.

NOTES:

TABLE 7-1: REGISTERS ASSOCIATED WITH DATA EEPROM MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	54
EEADR	EEPROM Address Register								56
EEDATA	EEPROM Data Register								56
EECON2	EEPROM Control Register 2 (not a physical register)								56
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	56
IPR2	OSCFIP	—	—	EEIP	—	LVDIP	—	CCP2IP	57
PIR2	OSCFIF	—	—	EEIF	—	LVDIF	—	CCP2IF	57
PIE2	OSCFIE	—	—	EEIE	—	LVDIE	—	CCP2IE	57

Legend: — = unimplemented, read as '0'. Shaded cells are not used during Flash/EEPROM access.

EXAMPLE 8-3: WRITING TO FLASH PROGRAM MEMORY (CONTINUED)

```
PROGRAM_MEMORY
            BCF     INTCON, GIE           ; disable interrupts
            MOVLW   55h                   ; required sequence
            MOVWF   EECON2                ; write 55h
            MOVLW   0AAh
            MOVWF   EECON2                ; write 0AAh
            BSF     EECON1, WR            ; start program (CPU stall)
            NOP
            BSF     INTCON, GIE           ; re-enable interrupts
            DECFSZ  COUNTER_HI            ; loop until done
            GOTO    PROGRAM_LOOP
            BCF     EECON1, WREN          ; disable write to memory
```

8.5.2 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

8.5.3 UNEXPECTED TERMINATION OF WRITE OPERATION

If a write is terminated by an unplanned event, such as loss of power or an unexpected Reset, the memory location just programmed should be verified and reprogrammed if needed. The WRERR bit is set when a write operation is interrupted by a $\overline{\text{MCLR}}$ Reset, or a WDT Time-out Reset during normal operation. In these situations, users can check the WRERR bit and rewrite the location.

8.6 Flash Program Operation During Code Protection

See **Section 23.5 "Program Verification and Code Protection"** for details on code protection of Flash program memory.

TABLE 8-2: REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
TBLPTRU	—	—	bit 21(1)	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					54
TBPLTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								54
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								54
TABLAT	Program Memory Table Latch								54
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	54
EECON2	EEPROM Control Register 2 (not a physical register)								56
EECON1	EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD	56
IPR2	OSCFIP	—	—	EEIP	—	LVDIP	—	CCP2IP	57
PIR2	OSCFIF	—	—	EEIF	—	LVDIF	—	CCP2IF	57
PIE2	OSCFIE	—	—	EEIE	—	LVDIE	—	CCP2IE	57

Legend: — = unimplemented, read as '0'. Shaded cells are not used during Flash/EEPROM access.

Note 1: Bit 21 of the PC is only available in Test mode and Serial Programming modes.

REGISTER 10-3: INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF
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 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

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

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

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

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared in software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared in software)
0 = The INT1 external interrupt did not occur

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

TABLE 11-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	57
LATB	LATB Data Output Register								57
TRISB	PORTB Data Direction Register								57
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	54
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	54
INTCON3	INT2IP	INT1IP	—	INT2IE	INT1IE	—	INT2IF	INT1IF	54

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTB.

11.4 PORTD, TRISD and LATD Registers

Note: PORTD is only available on PIC18F4331/4431 devices.

PORTD is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISD. Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., put the contents of the output latch on the selected pin).

The Data Latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register read and write the latched output value for PORTD.

All pins on PORTD are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

Note: On a Power-on Reset, these pins are configured as digital inputs.

PORTD includes PWM<7:6> complementary fourth channel PWM outputs. PWM4 is the complementary output of PWM5 (the third channel), which is multiplexed with the RB5 pin. This output can be used as the alternate output using the PWM4MX Configuration bit in CONFIG3H when the Single-Supply Programming pin (PGM) is used on RB5.

RD1, RD2 and RD3 can be used as the alternate output for SDO, SDI/SDA and SCK/SCL using the SSPMX Configuration bit in CONFIG3H.

RD4 an be used as the alternate output for $\overline{\text{FLTA}}$ using the FLTAMX Configuration bit in CONFIG3H.

EXAMPLE 11-4: INITIALIZING PORTD

```
CLRF    PORTD   ; Initialize PORTD by
                ; clearing output
                ; data latches
CLRF    LATD    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0xCF    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISD   ; Set RD<3:0> as inputs
                ; RD<5:4> as outputs
                ; RD<7:6> as inputs
```

The three input capture channels are controlled through the Input Capture Control registers, CAP1CON, CAP2CON and CAP3CON. Each channel is configured independently with its dedicated register. The implementation of the registers is identical except for the Special Event Trigger (see **Section 17.1.8 "Special Event Trigger (CAP1 Only)"**). The typical Capture Control register is shown in Register 17-1.

Note: Throughout this section, references to registers and bit names that may be associated with a specific capture channel will be referred to generically by the use of the term 'x' in place of the channel number. For example, 'CAPxREN' may refer to the Capture Reset Enable bit in CAP1CON, CAP2CON or CAP3CON.

REGISTER 17-1: CAPxCON: INPUT CAPTURE x CONTROL REGISTER

U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CAPxREN	—	—	CAPxM3	CAPxM2	CAPxM1	CAPxM0
bit 7							bit 0

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

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

bit 6 **CAPxREN:** Time Base Reset Enable bit
- 1 = Enabled
- 0 = Disable selected time base Reset on capture

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **CAPxM<3:0>:** Input Capture x (ICx) Mode Select bits
- 1111 = Special Event Trigger mode; the trigger occurs on every rising edge on CAP1 input(1)
- 1110 = Special Event Trigger mode; the trigger occurs on every falling edge on CAP1 input(1)
- 1101 = Unused
- 1100 = Unused
- 1011 = Unused
- 1010 = Unused
- 1001 = Unused
- 1000 = Capture on every CAPx input state change
- 0111 = Pulse-Width Measurement mode, every rising to falling edge
- 0110 = Pulse-Width Measurement mode, every falling to rising edge
- 0101 = Frequency Measurement mode, every rising edge
- 0100 = Capture mode, every 16th rising edge
- 0011 = Capture mode, every 4th rising edge
- 0010 = Capture mode, every rising edge
- 0001 = Capture mode, every falling edge
- 0000 = Input Capture x (ICx) off

Note 1: Special Event Trigger is only available on CAP1. For CAP2 and CAP3, this configuration is unused.

17.2 Quadrature Encoder Interface

The Quadrature Encoder Interface (QEI) decodes speed and motion sensor information. It can be used in any application that uses a quadrature encoder for feedback. The interface implements these features:

- Three QEI inputs: two phase signals (QEA and QEB) and one index signal (INDX)
- Direction of movement detection with a direction change interrupt (IC3DRIF)
- 16-bit up/down position counter
- Standard and High-Precision Position Tracking modes
- Two Position Update modes (x2 and x4)
- Velocity measurement with a programmable postscaler for high-speed velocity measurement
- Position counter interrupt (IC2QEIF in the PIR3 register)
- Velocity control interrupt (IC1IF in the PIR3 register)

The QEI submodule has three main components: the QEI control logic block, the position counter and velocity postscaler.

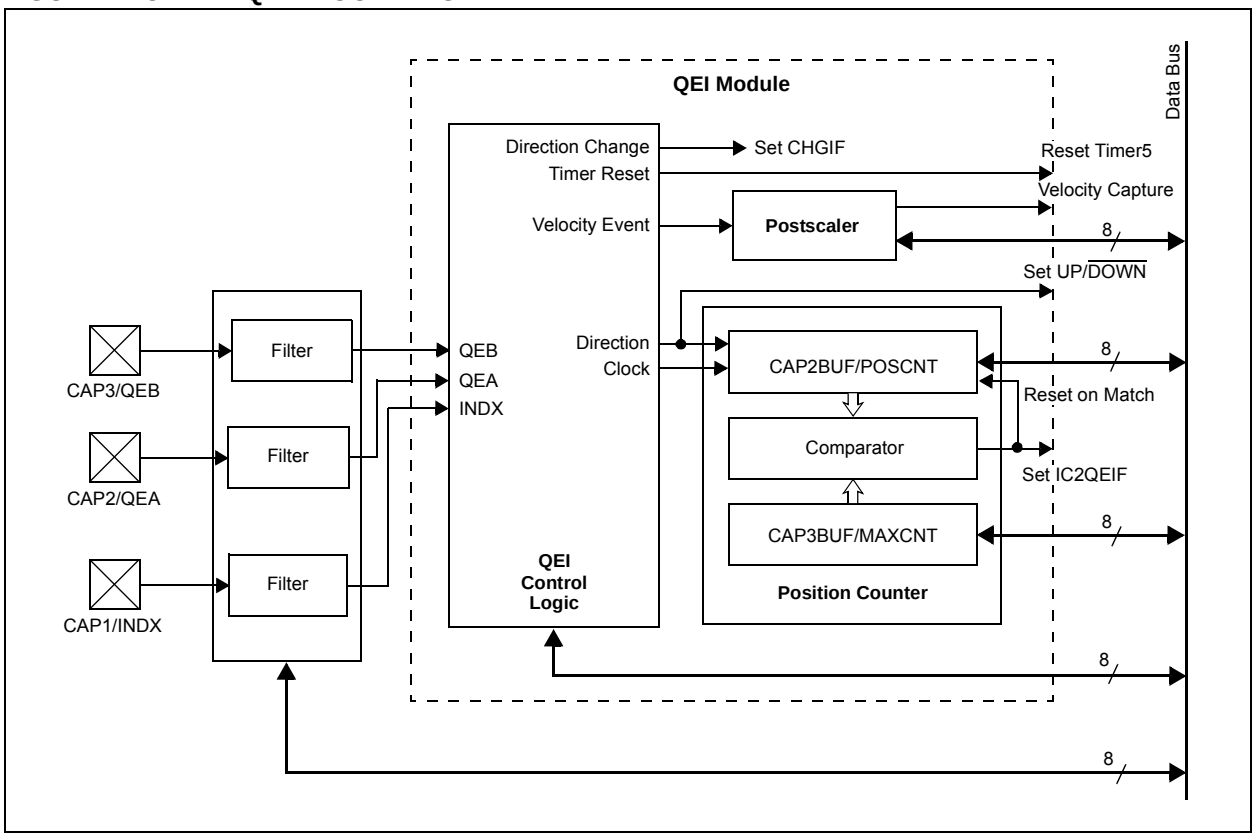
The QEI control logic detects the leading edge on the QEA or QEB phase input pins and generates the count pulse, which is sent to the position counter logic. It also samples the index input signal (INDX) and generates the direction of the rotation signal (up/down) and the velocity event signals.

The position counter acts as an integrator for tracking distance traveled. The QEA and QEB input edges serve as the stimulus to create the input clock which advances the Position Counter register (POSCNT). The register is incremented on either the QEA input edge, or the QEA and QEB input edges, depending on the operating mode. It is reset either by a rollover on match to the Period register, MAXCNT, or on the external index pulse input signal (INDX). An interrupt is generated on a Reset of POSCNT if the position counter interrupt is enabled.

The velocity postscaler down samples the velocity pulses used to increment the velocity counter by a specified ratio. It essentially divides down the number of velocity pulses to one output per so many inputs, preserving the pulse width in the process.

A simplified block-diagram of the QEI module is shown in Figure 17-8.

FIGURE 17-8: QEI BLOCK DIAGRAM



17.2.2 QEI MODES

Position measurement resolution depends on how often the Position Counter register, POSCNT, is incremented. There are two QEI Update modes to measure the rotor's position: QEI x2 and QEI x4.

TABLE 17-4: QEI MODES

QEIM<2:0>	Mode/ Reset	Description
000	—	QEI disabled.(1)
001	x2 update/ index pulse	Two clocks per QEA pulse. INDX resets POSCNT.
010	x2 update/ period match	Two clocks per QEA pulse. POSCNT is reset by the period match (MAXCNT).
011	—	Unused.
100	—	Unused.
101	x4 update/ index pulse	Four clocks per QEA and QEB pulse pair. INDX resets POSCNT.
110	x4 update/ period match	Four clocks per QEA and QEB pulse pair. POSCNT is reset by the period match (MAXCNT).
111	—	Unused.

Note 1: QEI module is disabled. The position counter and the velocity measurement functions are fully disabled in this mode.

17.2.2.1 QEI x2 Update Mode

QEI x2 Update mode is selected by setting the QEI Mode Select bits (QEIM<2:0>) to '001' or '010'. In this mode, the QEI logic detects every edge on the QEA input only. Every rising and falling edge on the QEA signal clocks the position counter.

The position counter can be reset by either an input on the INDX pin (QEIM<2:0> = 001), or by a period match, even when the POSCNT register pair equals MAXCNT (QEIM<2:0> = 010).

17.2.2.2 QEI x4 Update Mode

QEI x4 Update mode provides for a finer resolution of the rotor position, since the counter increments or decrements more frequently for each QEA/QEB input pulse pair than in QEI x2 mode. This mode is selected by setting the QEI mode select bits to '101' or '110'. In QEI x4, the phase measurement is made on the rising and the falling edges of both QEA and QEB inputs. The position counter is clocked on every QEA and QEB edge.

Like QEI x2 mode, the position counter can be reset by an input on the pin (QEIM<2:0> = 101), or by the period match event (QEIM<2:0> = 010).

17.2.3 QEI OPERATION

The Position Counter register pair (POSCNTH: POSCNTL) acts as an integrator, whose value is proportional to the position of the sensor rotor that corresponds to the number of active edges detected. POSCNT can either increment or decrement, depending on a number of selectable factors which are decoded by the QEI logic block. These include the Count mode selected, the phase relationship of QEA to QEB ("lead/lag"), the direction of rotation and if a Reset event occurs. The logic is detailed in the sections that follow.

17.2.3.1 Edge and Phase Detect

In the first step, the active edges of QEA and QEB are detected, and the phase relationship between them is determined. The position counter is changed based on the selected QEI mode.

In QEI x2 Update mode, the position counter increments or decrements on every QEA edge based on the phase relationship of the QEA and QEB signals.

In QEI x4 Update mode, the position counter increments or decrements on every QEA and QEB edge based on the phase relationship of the QEA and QEB signals. For example, if QEA leads QEB, the position counter is incremented by '1'. If QEB lags QEA, the position counter is decremented by '1'.

17.2.3.2 Direction of Count

The QEI control logic generates a signal that sets the UP/$\overline{\text{DOWN}}$ bit (QEICON<5>); this, in turn, determines the direction of the count. When QEA leads QEB, UP/$\overline{\text{DOWN}}$ is set (= 1) and the position counter increments on every active edge. When QEA lags QEB, UP/$\overline{\text{DOWN}}$ is cleared and the position counter decrements on every active edge.

TABLE 17-5: DIRECTION OF ROTATION

Current Signal Detected	Previous Signal Detected				Pos. Cntrl.(1)
	Rising		Falling		
	QEA	QEB	QEA	QEB	
QEA Rising				x	INC
		x			DEC
QEA Falling				x	DEC
		x			INC
QEB Rising	x				INC
			x		DEC
QEB Falling			x		INC
	x				DEC

Note 1: When UP/$\overline{\text{DOWN}}$ = 1, the position counter is incremented. When UP/$\overline{\text{DOWN}}$ = 0, the position counter is decremented.

REGISTER 18-8: FLTCONFIG: FAULT CONFIGURATION 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
BRFEN	FLTBS(1)	FLTBMOD(1)	FLTBEN(1)	FLTCON(2)	FLTAS	FLTAMOD	FLTAEN
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 **BRFEN:** Breakpoint Fault Enable bit
- 1 = Enable Fault condition on a breakpoint (i.e., only when PWMPIN = 1)
- 0 = Disable Fault condition

bit 6 **FLTBS:** Fault B Status bit(1)
- 1 = $\overline{\text{FLTB}}$ is asserted:
  if FLTBMOD = 0, cleared by the user;
  if FLTBMOD = 1, cleared automatically at beginning of the new period when $\overline{\text{FLTB}}$ is deasserted
- 0 = No Fault

bit 5 **FLTBMOD:** Fault B Mode bit(1)
- 1 = Cycle-by-Cycle mode: Pins are inactive for the remainder of the current PWM period or until $\overline{\text{FLTB}}$ is deasserted; FLTBS is cleared automatically when $\overline{\text{FLTB}}$ is inactive (no Fault present)
- 0 = Inactive mode: Pins are deactivated (catastrophic failure) until $\overline{\text{FLTB}}$ is deasserted and FLTBS is cleared by the user only

bit 4 **FLTBEN:** Fault B Enable bit(1)
- 1 = Enable Fault B
- 0 = Disable Fault B

bit 3 **FLTCON:** Fault Configuration bit(2)
- 1 = $\overline{\text{FLTA}}$, $\overline{\text{FLTB}}$ or both deactivates all PWM outputs
- 0 = $\overline{\text{FLTA}}$ or $\overline{\text{FLTB}}$ deactivates PWM<5:0>

bit 2 **FLTAS:** Fault A Status bit
- 1 = $\overline{\text{FLTA}}$ is asserted:
  if FLTAMOD = 0, cleared by the user;
  if FLTAMOD = 1, cleared automatically at beginning of the new period when $\overline{\text{FLTA}}$ is deasserted
- 0 = No Fault

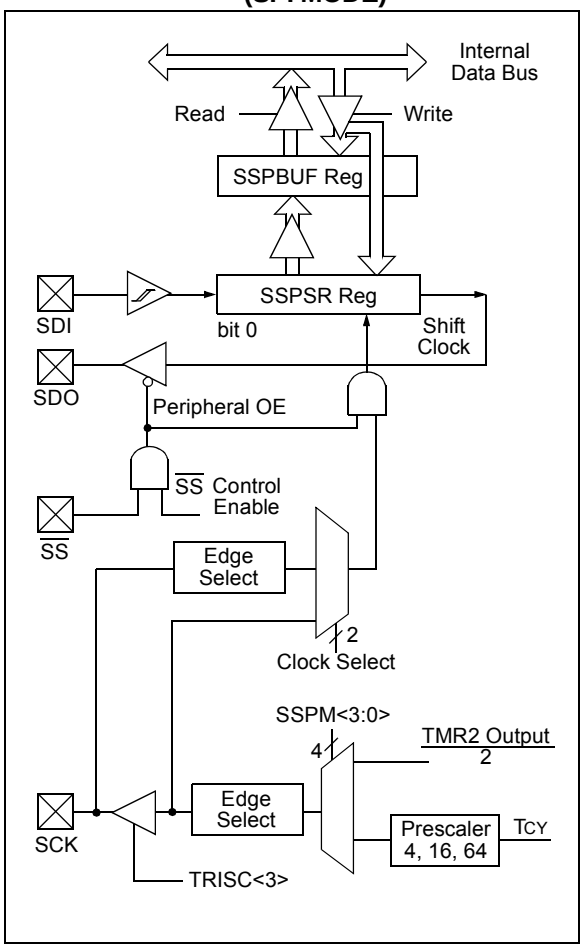
bit 1 **FLTAMOD:** Fault A Mode bit
- 1 = Cycle-by-Cycle mode: Pins are inactive for the remainder of the current PWM period or until $\overline{\text{FLTA}}$ is deasserted; FLTAS is cleared automatically
- 0 = Inactive mode: Pins are deactivated (catastrophic failure) until $\overline{\text{FLTA}}$ is deasserted and FLTAS is cleared by the user only

bit 0 **FLTAEN:** Fault A Enable bit
- 1 = Enable Fault A
- 0 = Disable Fault A

Note 1: Unimplemented in PIC18F2331/2431 devices; maintain these bits clear.

2: PWM<7:6> are implemented only on PIC18F4331/4431 devices. On PIC18F2331/2431 devices, setting or clearing FLTCON has no effect.

FIGURE 19-1: SSP BLOCK DIAGRAM (SPI MODE)



To enable the serial port, SSP Enable bit, SSPEN (SSPCON<5>), must be set. To reset or reconfigure SPI mode, clear bit SSPEN, reinitialize the SSPCON register and then set bit SSPEN. This configures the SDI, SDO, SCK and $\overline{SS}$ pins as serial port pins. For the pins to behave as the serial port function, they must have their data direction bits (in the TRISC register) appropriately programmed. That is:

- Serial Data Out (SDO) – RC7/RX/DT/SDO or RD1/SDO
- SDI must have TRISC<4> or TRISD<2> set
- SDO must have TRISC<7> or TRISD<1> cleared
- SCK (Master mode) must have TRISC<5> or TRISD<3> cleared
- SCK (Slave mode) must have TRISC<5> or TRISD<3> set
- $\overline{SS}$ must have TRISA<6> set

Note 1: When the SPI is in Slave mode, with the $\overline{SS}$ pin control enabled, (SSPCON<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to VDD.

2: If the SPI is used in Slave mode with CKE = 1, then the $\overline{SS}$ pin control must be enabled.

3: When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the state of the $\overline{SS}$ pin can affect the state read back from the TRISC<6> bit. The peripheral OE signal from the SSP module into PORTC controls the state that is read back from the TRISC<6> bit (see **Section 11.3 "PORTC, TRISC and LATC Registers"** for information on PORTC). If Read-Modify-Write instructions, such as BSF, are performed on the TRISC register while the $\overline{SS}$ pin is high, this will cause the TRISC<6> bit to be set, thus disabling the SDO output.

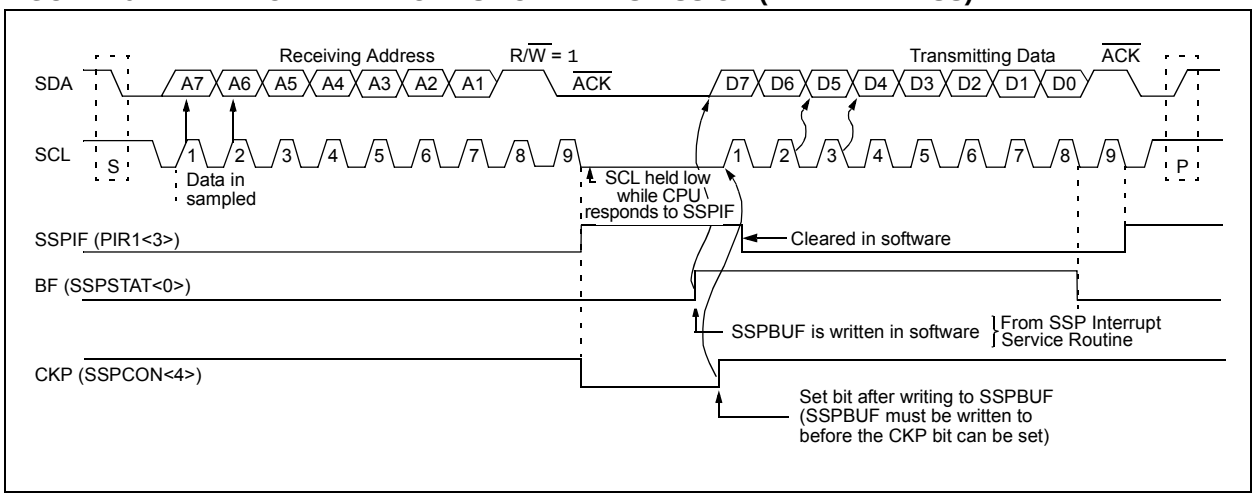
19.3.1.3 Transmission

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit and pin, SCK/SCL, is held low. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then, pin, SCK/SCL, should be enabled by setting bit, CKP (SSPCON<4>). The master must monitor the SCL pin prior to asserting another clock pulse. The slave devices may be holding off the master by stretching the clock. The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 19-7).

An SSP interrupt is generated for each data transfer byte. Flag bit, SSPIF, must be cleared in software and the SSPSTAT register is used to determine the status of the byte. Flag bit, SSPIF, is set on the falling edge of the ninth clock pulse.

As a slave-transmitter, the $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not $\overline{ACK}$), then the data transfer is complete. When the $\overline{ACK}$ is latched by the slave, the slave logic is reset and the slave then monitors for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin, SCK/SCL, should be enabled by setting bit CKP.

FIGURE 19-7: I²C™ WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)



REGISTER 20-3: BAUDCON: BAUD RATE CONTROL REGISTER

U-0	R-1	U-0	R/W-1	R/W-0	U-0	R/W-0	R/W-0
—	RCIDL	—	SCKP	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 **Unimplemented:** Read as ‘0’

bit 6 **RCIDL:** Receive Operation Idle Status bit
1 = Receiver is Idle
0 = Receive in progress

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **SCKP:** Synchronous Clock Polarity Select bit
Asynchronous mode:
Unused in this mode.
Synchronous mode:
1 = Idle state for clock (CK) is a high level
0 = Idle state for clock (CK) is a low level

bit 3 **BRG16:** 16-Bit Baud Rate Register Enable bit
1 = 16-bit Baud Rate Generator – SPBRGH and SPBRG
0 = 8-bit Baud Rate Generator – SPBRG only (Compatible mode), SPBRGH value ignored

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **WUE:** Wake-up Enable bit
Asynchronous mode:
1 = EUSART will continue to sample the RX pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge
0 = RX pin not monitored or rising edge detected
Synchronous mode:
Unused in this mode.

bit 0 **ABDEN:** Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Enable baud rate measurement on the next character – requires reception of a Sync field (55h); cleared in hardware upon completion.
0 = Baud rate measurement disabled or completed
Synchronous mode:
Unused in this mode.

REGISTER 23-4: CONFIG3L: CONFIGURATION REGISTER 3 LOW (BYTE ADDRESS 300004h)

U-0	U	R/P-1	R/P-1	R/P-1	R/P-1	U	U
—	—	$\overline{\text{T1OSCMX}}$	HPOL(1)	LPOL(1)	PWMPIN(3)	—	—
bit 7							bit 0

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		U = Unchanged from programmed state

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **T1OSCMX:** Timer1 Oscillator Mode bit
1 = Low-power Timer1 operation when microcontroller is in Sleep mode
0 = Standard (legacy) Timer1 oscillator operation

bit 4 **HPOL:** High Side Transistors Polarity bit (i.e., Odd PWM Output Polarity Control bit)(1)
1 = PWM1, 3, 5 and 7 are active-high (default)(2)
0 = PWM1, 3, 5 and 7 are active-low(2)

bit 3 **LPOL:** Low Side Transistors Polarity bit (i.e., Even PWM Output Polarity Control bit)(1)
1 = PWM0, 2, 4 and 6 are active-high (default)(2)
0 = PWM0, 2, 4 and 6 are active-low(2)

bit 2 **PWMPIN:** PWM Output Pins Reset State Control bit(3)
1 = PWM outputs are disabled upon Reset (default)
0 = PWM outputs drive active states upon Reset

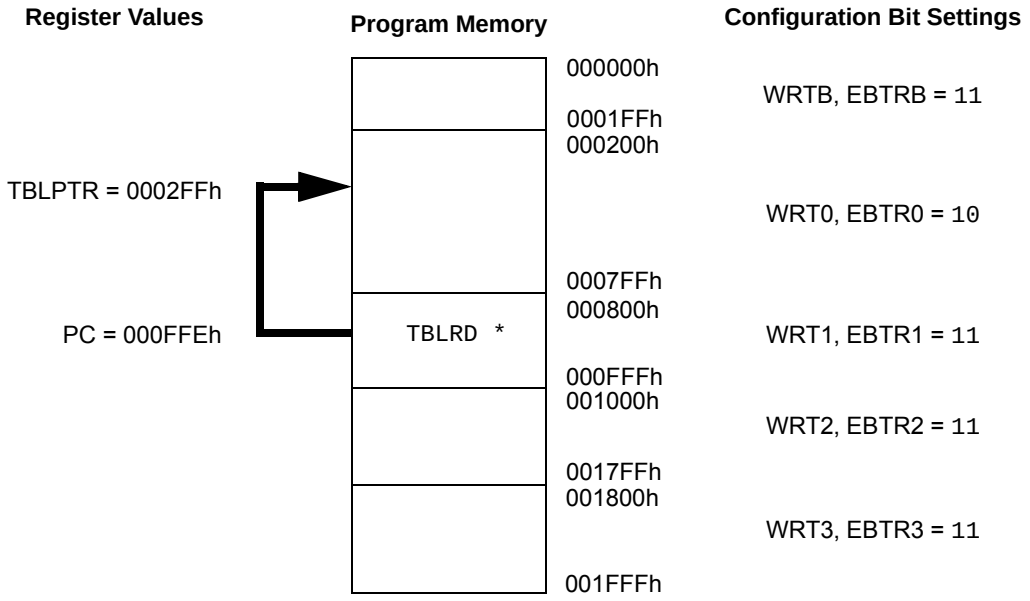
bit 1-0 **Unimplemented:** Read as '0'

Note 1: Polarity control bits, HPOL and LPOL, define PWM signal output active and inactive states; PWM states generated by the Fault inputs or PWM manual override.

2: PWM6 and PWM7 output channels are only available on PIC18F4331/4431 devices.

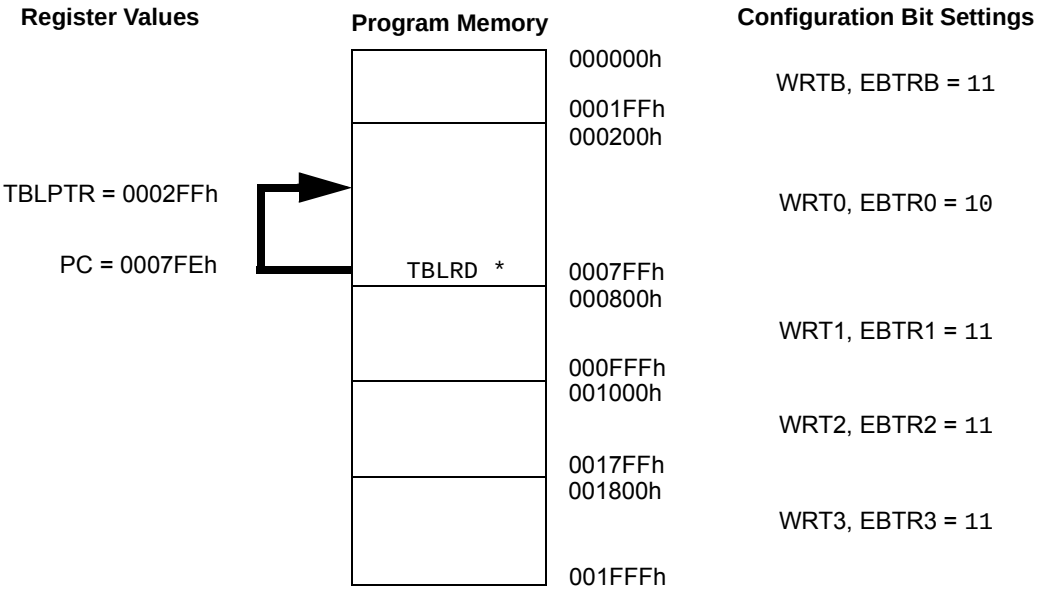
3: When PWMPIN = 0, PWMEN<2:0> = 101 if the device has eight PWM output pins (40 and 44-pin devices) and PWMEN<2:0> = 100 if the device has six PWM output pins (28-pin devices). PWM output polarity is defined by HPOL and LPOL.

FIGURE 23-7: EXTERNAL BLOCK TABLE READ (EBTRn) DISALLOWED



Results: All table reads from external blocks to Blockn are disabled whenever EBTRn = 0.
The TABLAT register returns a value of '0'.

FIGURE 23-8: EXTERNAL BLOCK TABLE READ (EBTRn) ALLOWED



Results: Table reads permitted within Blockn, even when EBTRBn = 0.
The TABLAT register returns the value of the data at the location TBLPTR.

TABLE 24-2: PIC18FXXXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word MSb	LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS							
ADDWF	f, d, a	Add WREG and f	1	0010 01da	ffff ffff	C, DC, Z, OV, N	1, 2
ADDWFC	f, d, a	Add WREG and Carry bit to f	1	0010 00da	ffff ffff	C, DC, Z, OV, N	1, 2
ANDWF	f, d, a	AND WREG with f	1	0001 01da	ffff ffff	Z, N	1,2
CLRF	f, a	Clear f	1	0110 101a	ffff ffff	Z	2
COMF	f, d, a	Complement f	1	0001 11da	ffff ffff	Z, N	1, 2
CPFSEQ	f, a	Compare f with WREG, Skip =	1 (2 or 3)	0110 001a	ffff ffff	None	4
CPFSGT	f, a	Compare f with WREG, Skip >	1 (2 or 3)	0110 010a	ffff ffff	None	4
CPFSLT	f, a	Compare f with WREG, Skip <	1 (2 or 3)	0110 000a	ffff ffff	None	1, 2
DECF	f, d, a	Decrement f	1	0000 01da	ffff ffff	C, DC, Z, OV, N	1, 2, 3, 4
DECFSZ	f, d, a	Decrement f, Skip if 0	1 (2 or 3)	0010 11da	ffff ffff	None	1, 2, 3, 4
DCFSNZ	f, d, a	Decrement f, Skip if Not 0	1 (2 or 3)	0100 11da	ffff ffff	None	1, 2
INCF	f, d, a	Increment f	1	0010 10da	ffff ffff	C, DC, Z, OV, N	1, 2, 3, 4
INCFSZ	f, d, a	Increment f, Skip if 0	1 (2 or 3)	0011 11da	ffff ffff	None	4
INFSNZ	f, d, a	Increment f, Skip if Not 0	1 (2 or 3)	0100 10da	ffff ffff	None	1, 2
IORWF	f, d, a	Inclusive OR WREG with f	1	0001 00da	ffff ffff	Z, N	1, 2
MOVF	f, d, a	Move f	1	0101 00da	ffff ffff	Z, N	1
MOVFF	f_s, f_d	Move f_s (source) to 1st word f_d (destination) 2nd word	2	1100 ffff 1111 ffff	ffff ffff ffff ffff	None	
MOVWF	f, a	Move WREG to f	1	0110 111a	ffff ffff	None	
MULWF	f, a	Multiply WREG with f	1	0000 001a	ffff ffff	None	
NEGF	f, a	Negate f	1	0110 110a	ffff ffff	C, DC, Z, OV, N	1, 2
RLCF	f, d, a	Rotate Left f through Carry	1	0011 01da	ffff ffff	C, Z, N	
RLNCF	f, d, a	Rotate Left f (No Carry)	1	0100 01da	ffff ffff	Z, N	1, 2
RRCF	f, d, a	Rotate Right f through Carry	1	0011 00da	ffff ffff	C, Z, N	
RRNCF	f, d, a	Rotate Right f (No Carry)	1	0100 00da	ffff ffff	Z, N	
SETF	f, a	Set f	1	0110 100a	ffff ffff	None	
SUBFWB	f, d, a	Subtract f from WREG with Borrow	1	0101 01da	ffff ffff	C, DC, Z, OV, N	1, 2
SUBWF	f, d, a	Subtract WREG from f	1	0101 11da	ffff ffff	C, DC, Z, OV, N	
SUBWFB	f, d, a	Subtract WREG from f with Borrow	1	0101 10da	ffff ffff	C, DC, Z, OV, N	1, 2
SWAPF	f, d, a	Swap Nibbles in f	1	0011 10da	ffff ffff	None	4
TSTFSZ	f, a	Test f, Skip if 0	1 (2 or 3)	0110 011a	ffff ffff	None	1, 2
XORWF	f, d, a	Exclusive OR WREG with f	1	0001 10da	ffff ffff	Z, N	
BIT-ORIENTED FILE REGISTER OPERATIONS							
BCF	f, b, a	Bit Clear f	1	1001 bbba	ffff ffff	None	1, 2
BSF	f, b, a	Bit Set f	1	1000 bbba	ffff ffff	None	1, 2
BTFSC	f, b, a	Bit Test f, Skip if Clear	1 (2 or 3)	1011 bbba	ffff ffff	None	3, 4
BTFSS	f, b, a	Bit Test f, Skip if Set	1 (2 or 3)	1010 bbba	ffff ffff	None	3, 4
BTG	f, b, a	Bit Toggle f	1	0111 bbba	ffff ffff	None	1, 2

Note 1: When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as an input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned.

3: If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

4: Some instructions are 2-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

5: If the table write starts the write cycle to internal memory, the write will continue until terminated.

SLEEP **Enter Sleep Mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT postscaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

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

Encoding:

0000	0000	0000	0011

Description: The Power-Down status bit ($\overline{PD}$) is cleared. The Time-out status bit ($\overline{TO}$) is set. Watchdog Timer and its post-scaler are cleared.
The processor is put into Sleep mode with the oscillator stopped.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	Go to Sleep

Example: SLEEP

Before Instruction
$\overline{TO}$ = ?
$\overline{PD}$ = ?
After Instruction
$\overline{TO}$ = 1 †
$\overline{PD}$ = 0

† If WDT causes wake-up, this bit is cleared.

SUBFWB **Subtract f from W with Borrow**

Syntax: [*label*] SUBFWB f [,d [,a]]

Operands: 0 ≤ f ≤ 255
d ∈ [0,1]
a ∈ [0,1]

Operation: (W) – (f) – ($\overline{C}$) → dest

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

Encoding:

0101	01da	ffff	ffff

Description: Subtract register, 'f', and the Carry flag (borrow) from W (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored in register, 'f'. If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example 1: SUBFWB REG

Before Instruction
REG = 0x03
W = 0x02
C = 0x01
After Instruction
REG = 0xFF
W = 0x02
C = 0x00
Z = 0x00
N = 0x01 ; result is negative

Example 2: SUBFWB REG, 0, 0

Before Instruction
REG = 2
W = 5
C = 1
After Instruction
REG = 2
W = 3
C = 1
Z = 0
N = 0 ; result is positive

Example 3: SUBFWB REG, 1, 0

Before Instruction
REG = 1
W = 2
C = 0
After Instruction
REG = 0
W = 2
C = 1
Z = 1 ; result is zero
N = 0

NOTES:

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	X	/XX	XXX
Device	Temperature Range	Package	Pattern

Device	PIC18F2331/2431/4331/4431(1), PIC18F2331/2431/4331/4431T(1,2); VDD range 4.2V to 5.5V PIC18LF2331/2431/4331/4431(1), PIC18LF2331/2431/4331/44310T(1,2); VDD range 2.0V to 5.5V
Temperature Range	I = -40°C to +85°C (Industrial) E = -40°C to +125°C (Extended)
Package	PT = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny Plastic DIP P = PDIP ML = QFN
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)

Examples:

a) PIC18LF4431-I/P 301 = Industrial temp., PDIP package, Extended VDD limits, QTP pattern #301.

b) PIC18LF2331-I/SO = Industrial temp., SOIC package, Extended VDD limits.

c) PIC18F4331-I/P = Industrial temp., PDIP package, normal VDD limits.

Note 1: F = Standard Voltage Range
LF = Wide Voltage Range

2: T = in Tape and Reel – SOIC and TQFP Packages only.