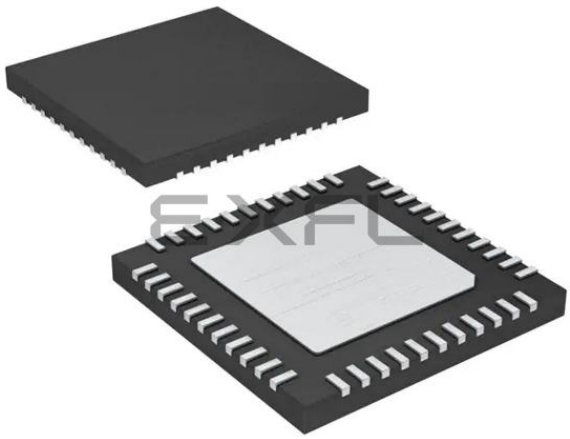


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	ECANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x12b
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/pic18f45k80-e-ml

REGISTER 3-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)

bit 2 **HFIOFS:** HF-INTOSC Frequency Stable bit
1 = HF-INTOSC oscillator frequency is stable
0 = HF-INTOSC oscillator frequency is not stable

bit 1-0 **SCS<1:0>:** System Clock Select bits(4)
1x = Internal oscillator block (LF-INTOSC, MF-INTOSC or HF-INTOSC)
01 = SOSC oscillator
00 = Default primary oscillator (OSC1/OSC2 or HF-INTOSC with or without PLL; defined by the FOSC<3:0> Configuration bits, CONFIG1H<3:0>)

Note 1: The Reset state depends on the state of the IESO Configuration bit (CONFIG1H<7>).
2: Modifying these bits will cause an immediate clock frequency switch if the internal oscillator is providing the device clocks.
3: The source is selected by the INTSRC bit (OSCTUNE<7>).
4: Modifying these bits will cause an immediate clock source switch.
5: INTSRC = OSCTUNE<7> and MFIOSEL = OSCCON2<0>.
6: This is the lowest power option for an internal source.

REGISTER 3-2: OSCCON2: OSCILLATOR CONTROL REGISTER 2

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

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

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

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

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

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

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

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

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

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

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

REGISTER 3-3: OSCTUNE: OSCILLATOR TUNING 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
INTSRC	PLLEN	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
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 **INTSRC:** Internal Oscillator Low-Frequency Source Select bit
1 = 31.25 kHz device clock is derived from 16 MHz INTOSC source (divide-by-512 enabled, HF-INTOSC)
0 = 31 kHz device clock is derived from INTOSC 31 kHz oscillator (LF-INTOSC)

bit 6 **PLLEN:** Frequency Multiplier PLL Enable bit
1 = PLL is enabled
0 = PLL is disabled

bit 5-0 **TUN<5:0>:** Fast RC Oscillator (INTOSC) Frequency Tuning bits
011111 = Maximum frequency
•
•
000001
000000 = Center frequency; fast RC oscillator is running at the calibrated frequency
111111
•
•
100000 = Minimum frequency

REGISTER 10-3: INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	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 **INT3IE:** INT3 External Interrupt Enable bit
1 = Enables the INT3 external interrupt
0 = Disables the INT3 external interrupt

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 **INT3IF:** INT3 External Interrupt Flag bit
1 = The INT3 external interrupt occurred (must be cleared in software)
0 = The INT3 external interrupt did not occur

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 Interrupt Enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

REGISTER 10-8: PIR5: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 5

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IRXIF	WAKIF	ERRIF	TXB2IF	TXB1IF	TXB0IF	RXB1IF	RXB0IF/ FIFOFIF
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 **IRXIF:** Invalid Message Received Interrupt Flag bits
1 = An invalid message occurred on the CAN bus
0 = No invalid message occurred on the CAN bus

bit 6 **WAKIF:** Bus Wake-up Activity Interrupt Flag bit
1 = Activity on the CAN bus has occurred
0 = No activity on the CAN bus

bit 5 **ERRIF:** Error Interrupt Flag bit (Multiple sources in COMSTAT register)
1 = An error has occurred in the CAN module (multiple sources)
0 = No CAN module errors have occurred

bit 4 **TXB2IF:** Transmit Buffer 2 Interrupt Flag bit
1 = Transmit Buffer 2 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 2 has not completed transmission of a message

bit 3 **TXB1IF:** Transmit Buffer 1 Interrupt Flag bit
1 = Transmit Buffer 1 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 1 has not completed transmission of a message

bit 2 **TXB0IF:** Transmit Buffer 0 Interrupt Flag bit
1 = Transmit Buffer 0 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 0 has not completed transmission of a message

bit 1 **RXB1IF:** Receive Buffer 1 Interrupt Flag bit
Mode 0:
1 = CAN Receive Buffer 1 has received a new message
0 = CAN Receive Buffer 1 has not received a new message
Modes 1 and 2:
1 = A CAN Receive Buffer/FIFO has received a new message
0 = A CAN Receive Buffer/FIFO has not received a new message

bit 0 Bit operation is dependent on the selected mode:
Mode 0:
RXB0IF: Receive Buffer 0 Interrupt Flag bit
1 = CAN Receive Buffer 0 has received a new message
0 = CAN Receive Buffer 0 has not received a new message
Mode 1:
Unimplemented: Read as '0'
Mode 2:
FIFOFIF: FIFO Full Interrupt Flag bit
1 = FIFO has reached full status as defined by the FIFO_HF bit
0 = FIFO has not reached full status as defined by the FIFO_HF bit

17.2 Timer4 Interrupt

The Timer4 module has an eight-bit Period register, PR4, that is both readable and writable. Timer4 increment from 00h until it matches PR4 and then resets to 00h on the next increment cycle. The PR4 register is initialized to FFh upon Reset.

17.3 Output of TMR4

The outputs of TMR4 (before the postscaler) are used only as a PWM time base for the ECCP modules. They are not used as baud rate clocks for the MSSP module as is the Timer2 output.

FIGURE 17-1: TIMER4 BLOCK DIAGRAM

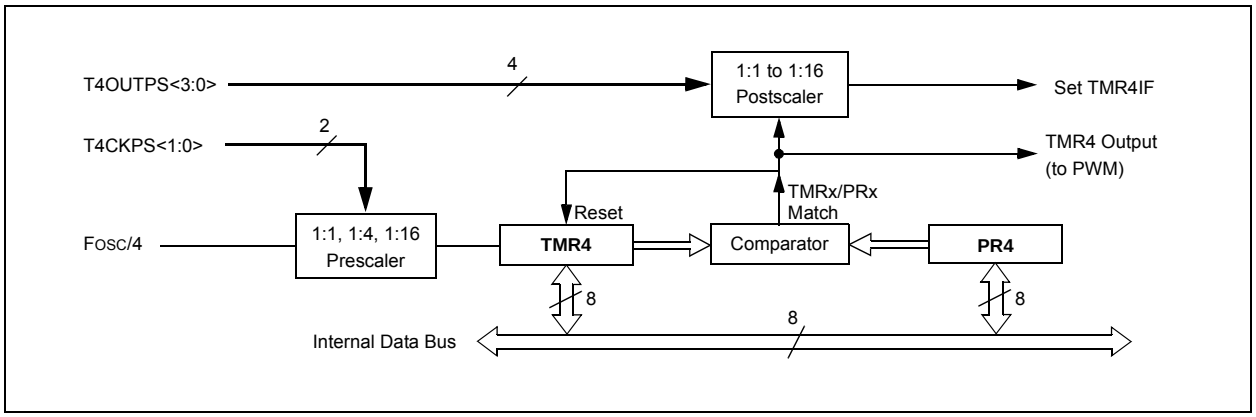


TABLE 17-1: REGISTERS ASSOCIATED WITH TIMER4 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
IPR4	TMR4IP	EEIP	CMP2IP	CMP1IP	—	CCP5IP	CCP4IP	CCP3IP
PIR4	TMR4IF	EEIF	CMP2IF	CMP1IF	—	CCP5IF	CCP4IF	CCP3IF
PIE4	TMR4IE	EEIE	CMP2IE	CMP1IE	—	CCP5IE	CCP4IE	CCP3IE
TMR4	Timer4 Register							
T4CON	—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0
PR4	Timer4 Period Register							
PMD1	PSPMD	CTMUMD	ADCMD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD

Legend: — = unimplemented, read as '0'. Shaded cells are not used by the Timer4 module.

REGISTER 20-2: CCPTMRS: CCP TIMER SELECT REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	C5TSEL	C4TSEL	C3TSEL	C2TSEL	C1TSEL
bit 7							bit 0

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

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

bit 4 **C5TSEL:** CCP5 Timer Selection bit
0 = CCP5 is based off of TMR1/TMR2
1 = CCP5 is based off of TMR3/TMR4

bit 3 **C4TSEL:** CCP4 Timer Selection bit
0 = CCP4 is based off of TMR1/TMR2
1 = CCP4 is based off of TMR3/TMR4

bit 2 **C3TSEL:** CCP3 Timer Selection bit
0 = CCP3 is based off of TMR1/TMR2
1 = CCP3 is based off of TMR3/TMR4

bit 1 **C2TSEL:** CCP2 Timer Selection bit
0 = CCP2 is based off of TMR1/TMR2
1 = CCP2 is based off of TMR3/TMR4

bit 0 **C1TSEL:** CCP1 Timer Selection bit
0 = ECCP1 is based off of TMR1/TMR2
1 = ECCP1 is based off of TMR3/TMR4

REGISTER 20-4: ECCP1DEL: ENHANCED PWM 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
P1RSEN	P1DC6	P1DC5	P1DC4	P1DC3	P1DC2	P1DC1	P1DC0
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 **P1RSEN:** PWM Restart Enable bit

1 = Upon auto-shutdown, the ECCP1ASE bit clears automatically once the shutdown event goes away; the PWM restarts automatically

0 = Upon auto-shutdown, ECCP1ASE must be cleared by software to restart the PWM

bit 6-0 **P1DC<6:0>:** PWM Delay Count bits

P1DCn = Number of FOSC/4 (4 * TOSC) cycles between the scheduled time when a PWM signal **should** transition active and the **actual** time it does transition active.

20.4.7 PULSE STEERING MODE

In Single Output mode, pulse steering allows any of the PWM pins to be the modulated signal. Additionally, the same PWM signal can simultaneously be available on multiple pins.

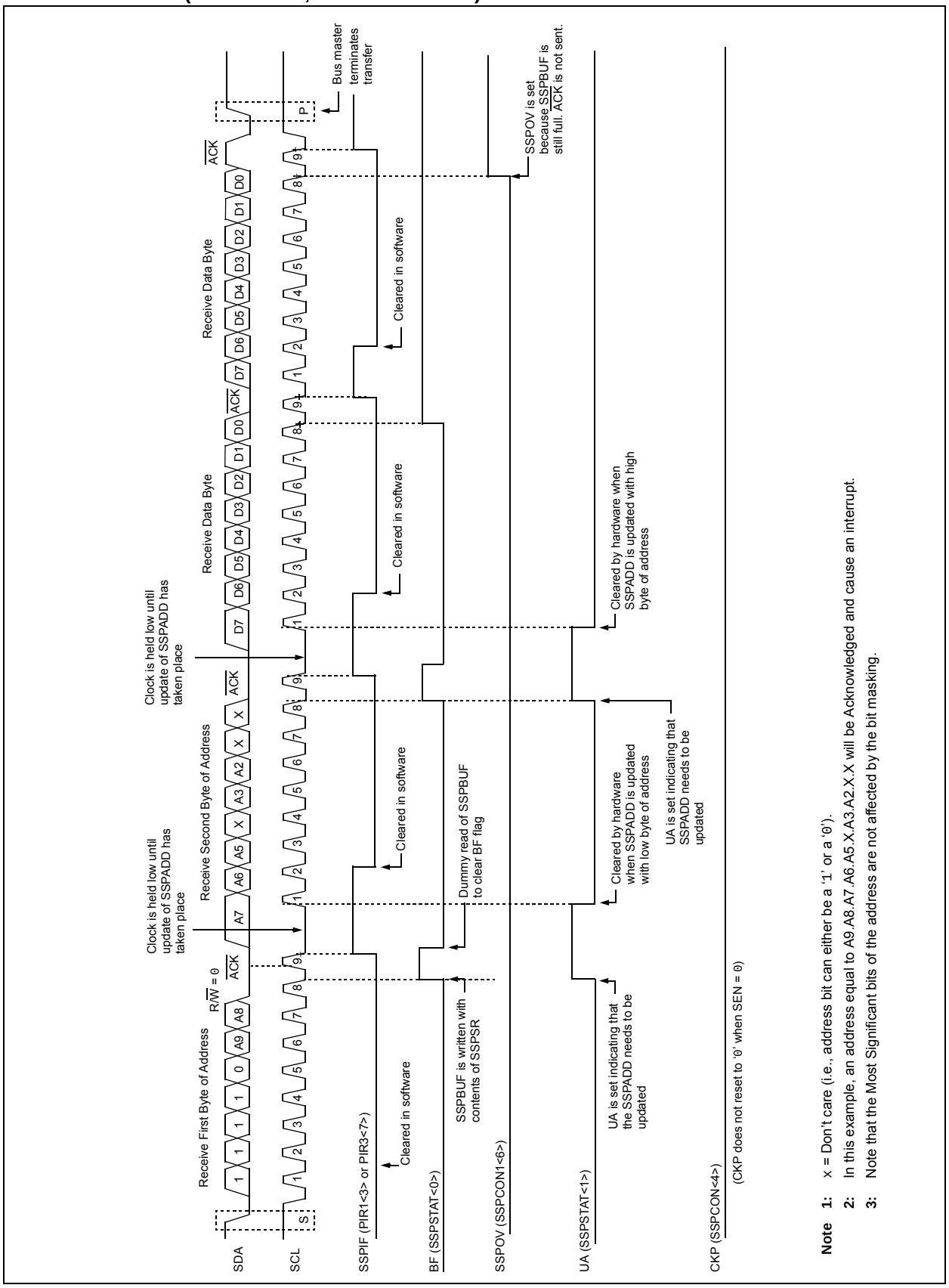
Once the Single Output mode is selected (CCP1M<3:2> = 11 and P1M<1:0> = 00 of the CCP1CON register), the user firmware can bring out the same PWM signal to one, two, three or four output pins by setting the appropriate STR<D:A> bits (PSTR1CON<3:0>), as provided in Table 20-2.

Note:	The associated TRIS bits must be set to output ('0') to enable the pin output driver in order to see the PWM signal on the pin.

While the PWM Steering mode is active, the CCP1M<1:0> bits (CCP1CON<1:0>) select the PWM output polarity for the P1<D:A> pins.

The PWM auto-shutdown operation also applies to the PWM Steering mode, as described in **Section 20.4.4 "Enhanced PWM Auto-shutdown mode"**. An auto-shutdown event will only affect pins that have PWM outputs enabled.

FIGURE 21-11: I²C™ SLAVE MODE TIMING WITH SEN = 0 AND ADMSK<5:1> = 01001 (RECEPTION, 10-BIT ADDRESS)



Note 1: x = Don't care (i.e., address bit can either be a '1' or a '0').

2: In this example, an address equal to A9.A8.A7.A6.A5.X.A3.A2.X.X will be Acknowledged and cause an interrupt.

3: Note that the Most Significant bits of the address are not affected by the bit masking.

REGISTER 22-3: BAUDCONx: BAUD RATE CONTROL REGISTER

R/W-0	R-1	R/W-x	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	RXDTP	TXCKP	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 Acquisition Rollover Status bit
1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode (must be cleared in software)
0 = No BRG rollover has occurred

bit 6 **RCIDL:** Receive Operation Idle Status bit
1 = Receive operation is Idle
0 = Receive operation is active

bit 5 **RXDTP:** Received Data Polarity Select bit (Asynchronous mode only)
Asynchronous mode:
1 = Receive data (RXx) is inverted
0 = Receive data (RXx) is not inverted

bit 4 **TXCKP:** Clock and Data Polarity Select bit
Asynchronous mode:
1 = Idle state for transmit (TXx) is a low level
0 = Idle state for transmit (TXx) is a high level
Synchronous mode:
1 = Idle state for clock (CKx) is a high level
0 = Idle state for clock (CKx) is a low level

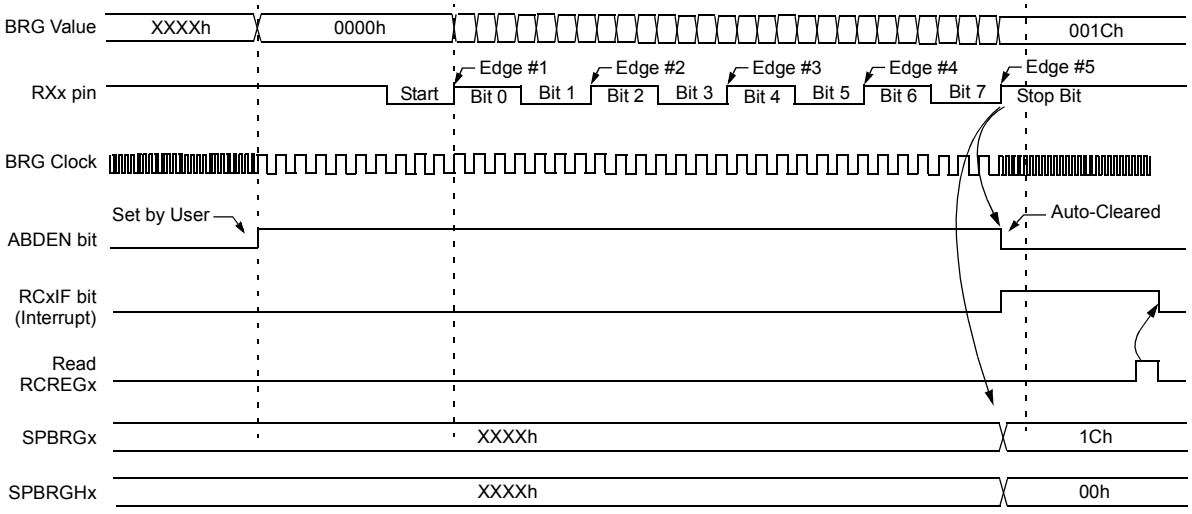
bit 3 **BRG16:** 16-Bit Baud Rate Register Enable bit
1 = 16-bit Baud Rate Generator – SPBRGHx and SPBRGx
0 = 8-bit Baud Rate Generator – SPBRGx only (Compatible mode), SPBRGHx value is ignored

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

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

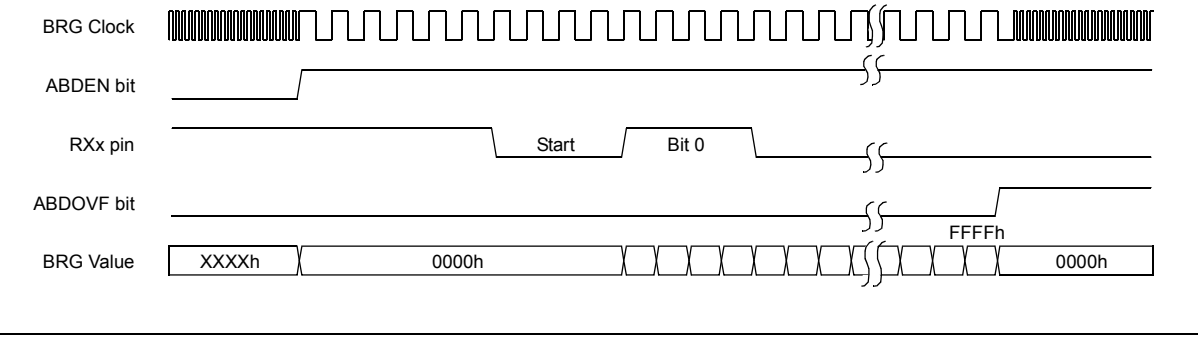
bit 0 **ABDEN:** Auto-Baud Detect Enable bit
Asynchronous mode:
1 = Enables baud rate measurement on the next character: requires reception of a Sync field (55h); cleared in hardware upon completion
0 = Baud rate measurement is disabled or completed
Synchronous mode:
Unused in this mode.

FIGURE 22-1: AUTOMATIC BAUD RATE CALCULATION



Note: The ABD sequence requires the EUSARTx module to be configured in Asynchronous mode and WUE = 0.

FIGURE 22-2: BRG OVERFLOW SEQUENCE



22.3.2 EUSARTx ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 22-6. The data is received on the RXx pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC. This mode would typically be used in RS-232 systems.

To set up an Asynchronous Reception:

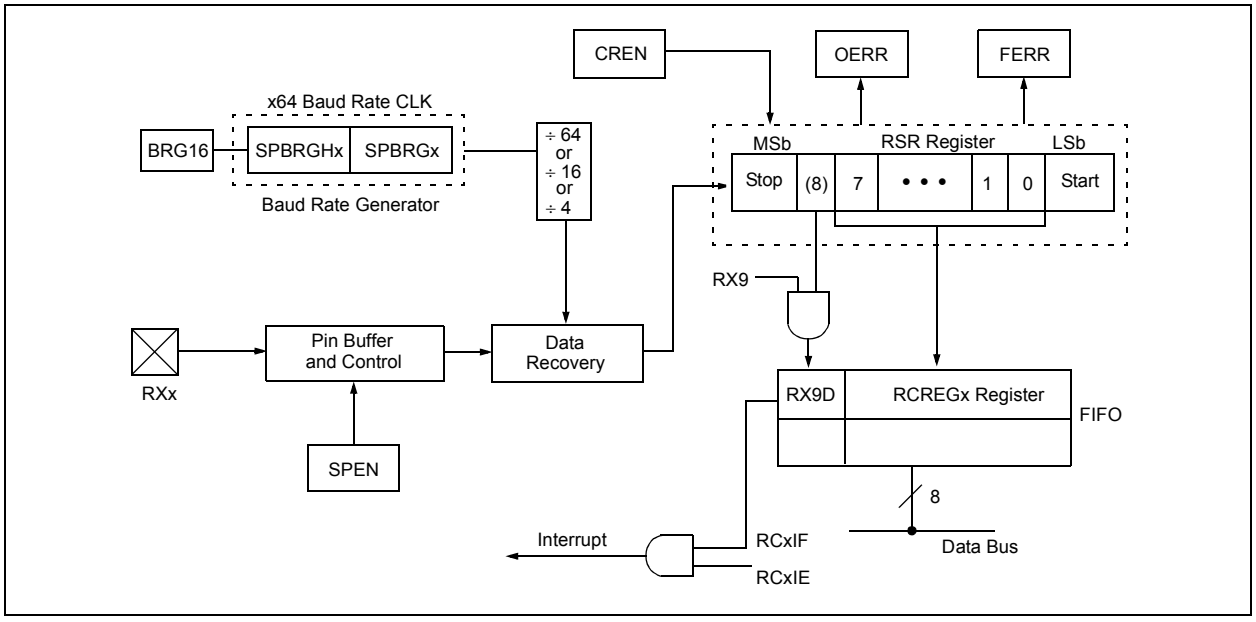
1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If interrupts are desired, set enable bit, RCxIE.
4. If 9-bit reception is desired, set bit, RX9.
5. Enable the reception by setting bit, CREN.
6. Flag bit, RCxIF, will be set when reception is complete and an interrupt will be generated if enable bit, RCxIE, was set.
7. Read the RCSTAx register to get the 9th bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREGx register.
9. If any error occurred, clear the error by clearing enable bit, CREN.
10. If using interrupts, ensure that the GIE and PEIE bits (INTCON<7:6>) are set.

22.3.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RCxIP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RCxIF bit will be set when reception is complete. The interrupt will be Acknowledged if the RCxIE and GIE bits are set.
8. Read the RCSTAx register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREGx to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

FIGURE 22-6: EUSARTx RECEIVE BLOCK DIAGRAM



22.4 EUSARTx Synchronous Master Mode

The Synchronous Master mode is entered by setting the CSRC bit (TXSTAx<7>). In this mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit, SYNC (TXSTAx<4>). In addition, enable bit, SPEN (RCSTAx<7>), is set in order to configure the TXx and RXx pins to CKx (clock) and DTx (data) lines, respectively.

The Master mode indicates that the processor transmits the master clock on the CKx line. Clock polarity is selected with the TXCKP bit (BAUDCONx<4>). Setting TXCKP sets the Idle state on CKx as high, while clearing the bit sets the Idle state as low. This option is provided to support Microwire devices with this module.

22.4.1 EUSARTx SYNCHRONOUS MASTER TRANSMISSION

The EUSARTx transmitter block diagram is shown in Figure 22-3. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The shift register obtains its data from the Read/Write Transmit Buffer register, TXREGx. The TXREGx register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREGx (if available).

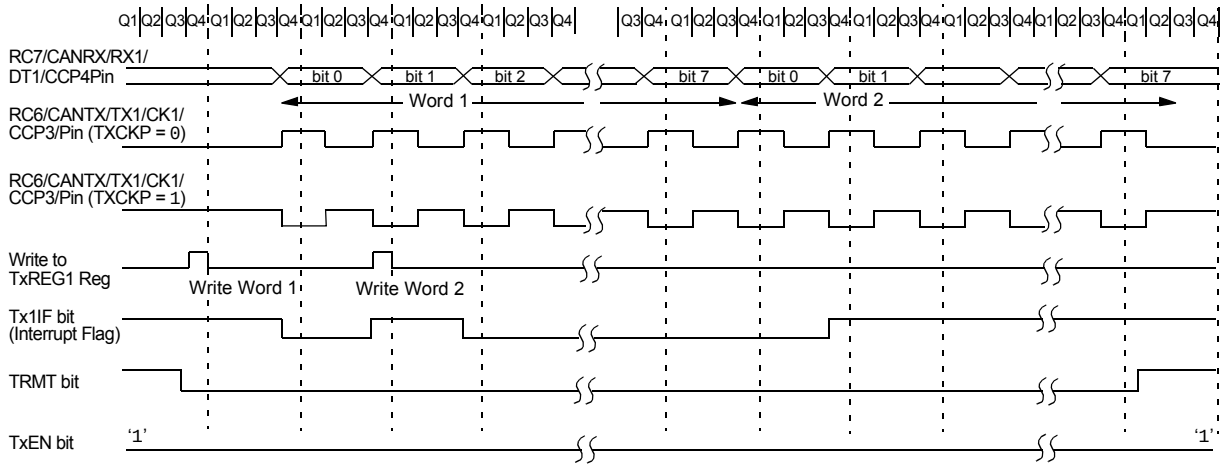
Once the TXREGx register transfers the data to the TSR register (occurs in one TCY), the TXREGx is empty and the TXxIF flag bit is set. The interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXxIE. TXxIF is set regardless of the state of enable bit, TXxIE; it cannot be cleared in software. It will reset only when new data is loaded into the TXREGx register.

While flag bit, TXxIF, indicates the status of the TXREGx register, another bit, TRMT (TXSTAx<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR is empty. No interrupt logic is tied to this bit, so the user must poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

To set up a Synchronous Master Transmission:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits, SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit, TXxIE.
4. If 9-bit transmission is desired, set bit, TX9.
5. Enable the transmission by setting bit, TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit, TX9D.
7. Start transmission by loading data to the TXREGx register.
8. If using interrupts, ensure that the GIE and PEIE bits (INTCON<7:6>) are set.

FIGURE 22-11: SYNCHRONOUS TRANSMISSION



Note: Sync Master mode, SPBRGx = 0; continuous transmission of two 8-bit words. This example is equally applicable to EUSART2 (RB6/PGC/TX2/CK2/KBI2 and RB7/PGD/T3G/RX2/DT2/KBI3).

23.2 A/D Registers

23.2.1 A/D CONTROL REGISTERS

REGISTER 23-1: ADCON0: A/D CONTROL REGISTER 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CHS4	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
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 6-2 **CHS<4:0>:** Analog Channel Select bits

00000 = Channel 00 (AN0)	10000 = (Reserved)(2)
00001 = Channel 01 (AN1)	10001 = (Reserved)(2)
00010 = Channel 02 (AN2)	10010 = (Reserved)(2)
00011 = Channel 03 (AN3)	10011 = (Reserved)(2)
00100 = Channel 04 (AN4)	10100 = (Reserved)(2)
00101 = Channel 05 (AN5)(1,2)	10101 = (Reserved)(2)
00110 = Channel 06 (AN6)(1,2)	10110 = (Reserved)(2)
00111 = Channel 07 (AN7)(1,2)	10111 = (Reserved)(2)
01000 = Channel 08 (AN8)	11000 = (Reserved)(2)
01001 = Channel 09 (AN9)	11001 = (Reserved)(2)
01010 = Channel 10 (AN10)	11010 = (Reserved)(2)
01011 = (Reserved)(2)	11011 = (Reserved)(2)
01100 = (Reserved)(2))	11100 = (MUX disconnect)(3)
01101 = (Reserved)(2))	11101 = Channel 29 (temperature diode)
01110 = (Reserved)(2))	11110 = Channel 30 (V_{DDCORE})
01111 = (Reserved)(2)	11111 = Channel 31 (1.024V band gap)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit

1 = A/D cycle is in progress. Setting this bit starts an A/D conversion cycle. The bit is cleared automatically by hardware when the A/D conversion is completed.

0 = A/D conversion has completed or is not in progress

bit 0 **ADON:** A/D On bit

1 = A/D Converter is operating

0 = A/D conversion module is shut off and consuming no operating current

Note 1: These channels are not implemented on 28-pin devices.

2: Performing a conversion on unimplemented channels will return random values.

3: Channel 28 turns off analog MUX switches to allow for minimum capacitive loading of the A/D input, for finer resolution CTMU time measurements.

26.2 HLVD Setup

To set up the HLVD module:

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

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

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

26.3 Current Consumption

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

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

26.4 HLVD Start-up Time

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

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

EXAMPLE 27-2: WIN AND ICODE BITS USAGE IN INTERRUPT SERVICE ROUTINE TO ACCESS TX/RX BUFFERS (CONTINUED)

```
ErrorInterrupt
    BCF     PIR3, ERRIF                 ; Clear the interrupt flag
    …                                   ; Handle error.
    RETFIE
TXB2Interrupt
    BCF     PIR3, TXB2IF                ; Clear the interrupt flag
    GOTO    AccessBuffer
TXB1Interrupt
    BCF     PIR3, TXB1IF                ; Clear the interrupt flag
    GOTO    AccessBuffer
TXB0Interrupt
    BCF     PIR3, TXB0IF                ; Clear the interrupt flag
    GOTO    AccessBuffer
RXB1Interrupt
    BCF     PIR3, RXB1IF                ; Clear the interrupt flag
    GOTO    Accessbuffer
RXB0Interrupt
    BCF     PIR3, RXB0IF                ; Clear the interrupt flag
    GOTO    AccessBuffer
AccessBuffer                            ; This is either TX or RX interrupt
    ; Copy CANSTAT.ICODE bits to CANCON.WIN bits
    MOVF    TempCANCON, W               ; Clear CANCON.WIN bits before copying
                                        ; new ones.
    ANDLW   B’11110001’                 ; Use previously saved CANCON value to
                                        ; make sure same value.
    MOVWF   TempCANCON                  ; Copy masked value back to TempCANCON
    MOVF    TempCANSTAT, W              ; Retrieve ICODE bits
    ANDLW   B’00001110’                 ; Use previously saved CANSTAT value
                                        ; to make sure same value.
    IORWF   TempCANCON                  ; Copy ICODE bits to WIN bits.
    MOVFF   TempCANCON, CANCON          ; Copy the result to actual CANCON
    ; Access current buffer…
    ; User code
    ; Restore CANCON.WIN bits
    MOVF    CANCON, W                   ; Preserve current non WIN bits
    ANDLW   B’11110001’
    IORWF   TempCANCON                  ; Restore original WIN bits
    ; Do not need to restore CANSTAT - it is read-only register.
    ; Return from interrupt or check for another module interrupt source
```

REGISTER 27-19: RXBnDLC: RECEIVE BUFFER 'n' DATA LENGTH CODE REGISTERS [0 ≤ n ≤ 1]

U-0	R-x	R-x	R-x	R-x	R-x	R-x	R-x
—	RXRTR	RB1	R0	DLC3	DLC2	DLC1	DLC0
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 **RXRTR:** Receiver Remote Transmission Request bit
1 = Remote transfer request
0 = No remote transfer request

bit 5 **RB1:** Reserved bit 1
Reserved by CAN Spec and read as '0'.

bit 4 **RB0:** Reserved bit 0
Reserved by CAN Spec and read as '0'.

bit 3-0 **DLC<3:0>:** Data Length Code bits
1111 = Invalid
1110 = Invalid
1101 = Invalid
1100 = Invalid
1011 = Invalid
1010 = Invalid
1001 = Invalid
1000 = Data length = 8 bytes
0111 = Data length = 7 bytes
0110 = Data length = 6 bytes
0101 = Data length = 5 bytes
0100 = Data length = 4 bytes
0011 = Data length = 3 bytes
0010 = Data length = 2 bytes
0001 = Data length = 1 byte
0000 = Data length = 0 bytes

REGISTER 27-20: RXBnDm: RECEIVE BUFFER 'n' DATA FIELD BYTE 'm' REGISTERS [0 ≤ n ≤ 1, 0 ≤ m ≤ 7]

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
RXBnDm7	RXBnDm6	RXBnDm5	RXBnDm4	RXBnDm3	RXBnDm2	RXBnDm1	RXBnDm0
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 **RXBnDm<7:0>:** Receive Buffer n Data Field Byte m bits (where 0 ≤ n < 1 and 0 < m < 7)
Each receive buffer has an array of registers. For example, Receive Buffer 0 has 8 registers: RXB0D0 to RXB0D7.

REGISTER 27-28: BnEIDH: TX/RX BUFFER 'n' EXTENDED IDENTIFIER REGISTERS, HIGH BYTE IN RECEIVE MODE [0 ≤ n ≤ 5, TXnEN (BSEL0<n>) = 0](1)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 7							bit 0

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

bit 7-0 **EID<15:8>:** Extended Identifier bits

Note 1: These registers are available in Mode 1 and 2 only.

REGISTER 27-29: BnEIDH: TX/RX BUFFER 'n' EXTENDED IDENTIFIER REGISTERS, HIGH BYTE IN TRANSMIT MODE [0 ≤ n ≤ 5, TXnEN (BSEL0<n>) = 1](1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 7							bit 0

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

bit 7-0 **EID<15:8>:** Extended Identifier bits

Note 1: These registers are available in Mode 1 and 2 only.

REGISTER 27-30: BnEIDL: TX/RX BUFFER 'n' EXTENDED IDENTIFIER REGISTERS, LOW BYTE IN RECEIVE MODE [0 ≤ n ≤ 5, TXnEN (BSEL<n>) = 0](1)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7							bit 0

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

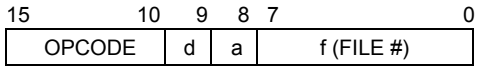
bit 7-0 **EID<7:0>:** Extended Identifier bits

Note 1: These registers are available in Mode 1 and 2 only.

FIGURE 29-1: GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations

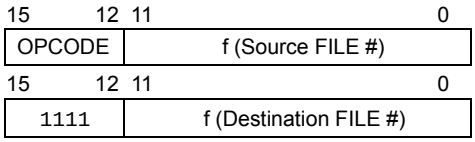
Example Instruction



ADDWF MYREG, W, B

d = 0 for result destination to be WREG register
d = 1 for result destination to be file register (f)
a = 0 to force Access Bank
a = 1 for BSR to select bank
f = 8-bit file register address

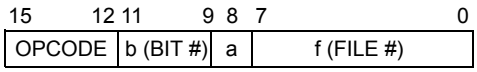
Byte to Byte move operations (2-word)



MOVFF MYREG1, MYREG2

f = 12-bit file register address

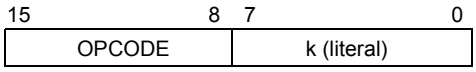
Bit-oriented file register operations



BSF MYREG, bit, B

b = 3-bit position of bit in file register (f)
a = 0 to force Access Bank
a = 1 for BSR to select bank
f = 8-bit file register address

Literal operations

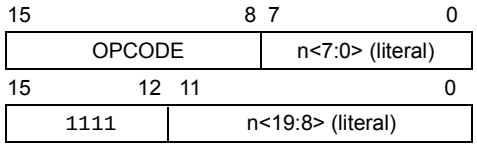


MOVLW 7Fh

k = 8-bit immediate value

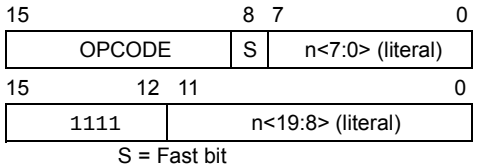
Control operations

CALL, GOTO and Branch operations



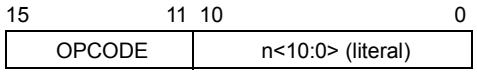
GOTO Label

n = 20-bit immediate value

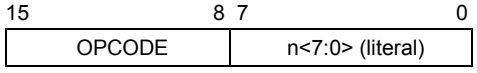


CALL MYFUNC

S = Fast bit



BRA MYFUNC



BC MYFUNC

29.2.5 SPECIAL CONSIDERATIONS WITH MICROCHIP MPLAB® IDE TOOLS

The latest versions of Microchip's software tools have been designed to fully support the extended instruction set for the PIC18F66K80 family. This includes the MPLAB C18 C Compiler, MPASM assembly language and MPLAB Integrated Development Environment (IDE).

When selecting a target device for software development, MPLAB IDE will automatically set default Configuration bits for that device. The default setting for the XINST Configuration bit is '0', disabling the extended instruction set and Indexed Literal Offset Addressing. For proper execution of applications developed to take advantage of the extended instruction set, XINST must be set during programming.

To develop software for the extended instruction set, the user must enable support for the instructions and the Indexed Addressing mode in their language tool(s). Depending on the environment being used, this may be done in several ways:

- A menu option or dialog box within the environment that allows the user to configure the language tool and its settings for the project
- A command line option
- A directive in the source code

These options vary between different compilers, assemblers and development environments. Users are encouraged to review the documentation accompanying their development systems for the appropriate information.

31.2 DC Characteristics: Power-Down and Supply Current PIC18F66K80 Family (Industrial/Extended) (Continued)

PIC18F66K80 Family (Industrial/Extended)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	Module Differential Currents (ΔIWDT, ΔIBOR, ΔIHLVD, ΔIADC)						
D022 (ΔIWDT)	**Watchdog Timer**						
	PIC18**LF**XXK80	0.4	2	µA	-40°C to +125°C	VDD = 1.8V Regulator Disabled	
	PIC18**LF**XXK80	0.6	3	µA	-40°C to +125°C	VDD = 3.3V Regulator Disabled	
	PIC18**F**XXK80	0.6	3	µA	-40°C to +125°C	VDD = 3.3V Regulator Enabled	
	PIC18**F**XXK80	0.8	4	µA	-40°C to +125°C	VDD = 5.5V Regulator Enabled	
D022A (ΔIBOR)	**Brown-out Reset**						
	PIC18**LF**XXK80	4.6	20	µA	-40°C to +125°C	VDD = 1.8V Regulator Disabled	High-Power BOR
	PIC18**F**XXK80	4.6	20	µA	-40°C to +125°C	VDD = 3.3V Regulator Enabled	
	PIC18**F**XXK80	4.6	20	µA	-40°C to +125°C	VDD = 5.5V Regulator Enabled	
D022B ΔIHLVD	**High/Low-Voltage Detect**						
	PIC18**LF**XXK80	3.8	10	µA	-40°C to +125°C	VDD = 1.8V Regulator Disabled	
	PIC18L**F**XXK80	4.5	12	µA	-40°C to +125°C	VDD = 3.3V Regulator Disabled	
	PIC18**F**XXK80	3.8	12	µA	-40°C to +125°C	VDD = 3.3V Regulator Enabled	
	PIC18**F**XXK80	4.9	13	µA	-40°C to +125°C	VDD = 5.5V Regulator Enabled	
D026 ΔIADC	**A/D Converter**						
	PIC18**LF**XXK80	0.4	1.5		-40°C to +125°C	VDD = 1.8V Regulator Disabled	A/D on, not converting
	PIC18L**F**XXK80	0.5	2	µA	-40°C to +125°C	VDD = 3.3V Regulator Disabled	
	PIC18**F**XXK80	0.5	3	µA	-40°C to +125°C	VDD = 3.3V Regulator Enabled	
	PIC18**F**XXK80	1	3	µA	-40°C to +125°C	VDD = 5.5V Regulator Enabled	

Legend: Shading of rows is to assist in readability of the table.

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in a high-impedance state and tied to VDD or VSS, and all features that add delta current are disabled (such as WDT, SOSC oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: For LF devices, $\overline{RETEN}$ (CONFIG1L<0>) = 1.

5: For F devices, SRETEN (WDTCON<4>) = 1 and $\overline{RETEN}$ (CONFIG1L<0>) = 0.