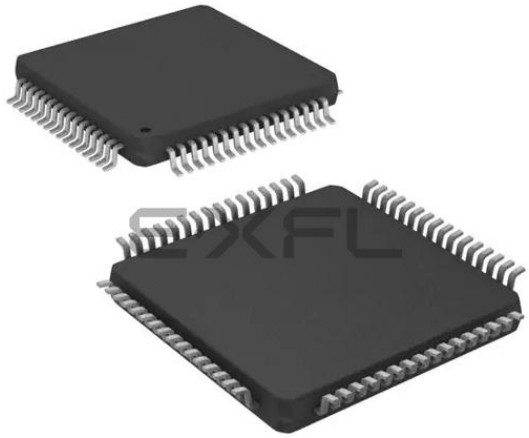


Welcome to E-XFL.COM

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	41.667MHz
Connectivity	Ethernet, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	39
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3808 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f66j60-i-pt

TABLE 1-4: PIC18F66J60/66J65/67J60 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
$\overline{MCLR}$	7	I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
OSC1/CLKI	39			Oscillator crystal or external clock input.
OSC1		I	ST	Oscillator crystal input or external clock source input. ST buffer when configured in internal RC mode; CMOS otherwise.
CLKI		I	CMOS	External clock source input. Always associated with pin function, OSC1. (See related OSC2/CLKO pin.)
OSC2/CLKO	40			Oscillator crystal or clock output.
OSC2		O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO		O	—	In Internal RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
				PORTA is a bidirectional I/O port.
RA0/LEDA/AN0	24			
RA0		I/O	TTL	Digital I/O.
LEDA		O	—	Ethernet LEDA indicator output.
AN0		I	Analog	Analog Input 0.
RA1/LEDB/AN1	23			
RA1		I/O	TTL	Digital I/O.
LEDB		O	—	Ethernet LEDB indicator output.
AN1		I	Analog	Analog Input 1.
RA2/AN2/V_{REF-}	22			
RA2		I/O	TTL	Digital I/O.
AN2		I	Analog	Analog Input 2.
V_{REF-}		I	Analog	A/D reference voltage (low) input.
RA3/AN3/V_{REF+}	21			
RA3		I/O	TTL	Digital I/O.
AN3		I	Analog	Analog Input 3.
V_{REF+}		I	Analog	A/D reference voltage (high) input.
RA4/T0CKI	28			
RA4		I/O	ST	Digital I/O.
T0CKI		I	ST	Timer0 external clock input.
RA5/AN4	27			
RA5		I/O	TTL	Digital I/O.
AN4		I	Analog	Analog Input 4.

Legend: TTL = TTL compatible input
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
CMOS = CMOS compatible input or output
Analog = Analog input
O = Output
OD = Open-Drain (no P diode to V_{DD})

TABLE 1-4: PIC18F66J60/66J65/67J60 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
				PORTD is a bidirectional I/O port.
RD0/P1B	60			
RD0		I/O	ST	Digital I/O.
P1B		O	—	ECCP1 PWM Output B.
RD1/ECCP3/P3A	59			
RD1		I/O	ST	Digital I/O.
ECCP3		I/O	ST	Capture 3 input/Compare 3 output/PWM3 output.
P3A		O	—	ECCP3 PWM Output A.
RD2/CCP4/P3D	58			
RD2		I/O	ST	Digital I/O.
CCP4		I/O	ST	Capture 4 input/Compare 4 output/PWM4 output.
P3D		O	—	CCP4 PWM Output D.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD})

TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices			Power-on Reset, Brown-out Reset	MCLR Reset, WDT Reset, RESET Instruction, Stack Resets, CM Reset	Wake-up via WDT or Interrupt
CCP4CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--00 0000	--00 0000	--uu uuuu
CCPR5H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR5L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP5CON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--00 0000	--00 0000	--uu uuuu
SPBRG2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
RCREG2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TXREG2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TXSTA2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0010	0000 0010	uuuu uuuu
RCSTA2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 000x	0000 000x	uuuu uuuu
ECCP3AS	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ECCP3DEL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ECCP2AS	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ECCP2DEL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
SSP2BUF	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP2ADD	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
SSP2STAT	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
SSP2CON1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
SSP2CON2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EDATA	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
EIR	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-000 0-00	-000 0-00	-uuu u-uu
ECON2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	100- ----	100- ----	uuu- ----
ESTAT	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-0-0 -000	-0-0 -000	-u-u -uuu
EIE	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	-000 0-00	-000 0-00	-uuu u-uu
EDMACSH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EDMACSL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EDMADSTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
EDMADSTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EDMANDH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
EDMANDL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
EDMASTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
EDMASTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ERXWRPTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
ERXWRPTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
ERXRDPTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0101	---0 0101	---u uuuu
ERXRDPTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1010	1111 1010	uuuu uuuu
ERXNDH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---1 1111	---1 1111	---u uuuu
ERXNDL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu
ERXSTH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0101	---0 0101	---u uuuu
ERXSTL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1010	1111 1010	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.
Shaded cells indicate conditions do not apply for the designated device.

Note 1: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

4: See Table 5-1 for Reset value for specific condition.

8.6.4 16-BIT MODE TIMING

The presentation of control signals on the external memory bus is different for the various operating modes. Typical signal timing diagrams are shown in Figure 8-4 and Figure 8-5.

FIGURE 8-4: EXTERNAL MEMORY BUS TIMING FOR TBLRD (EXTENDED MICROCONTROLLER MODE)

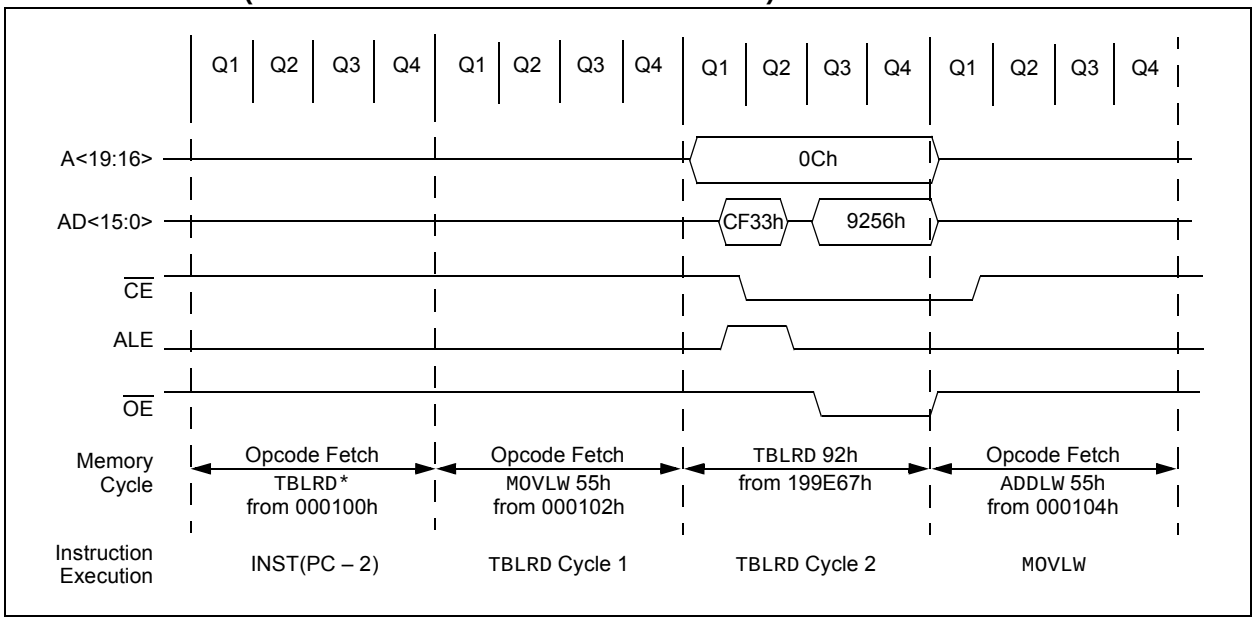
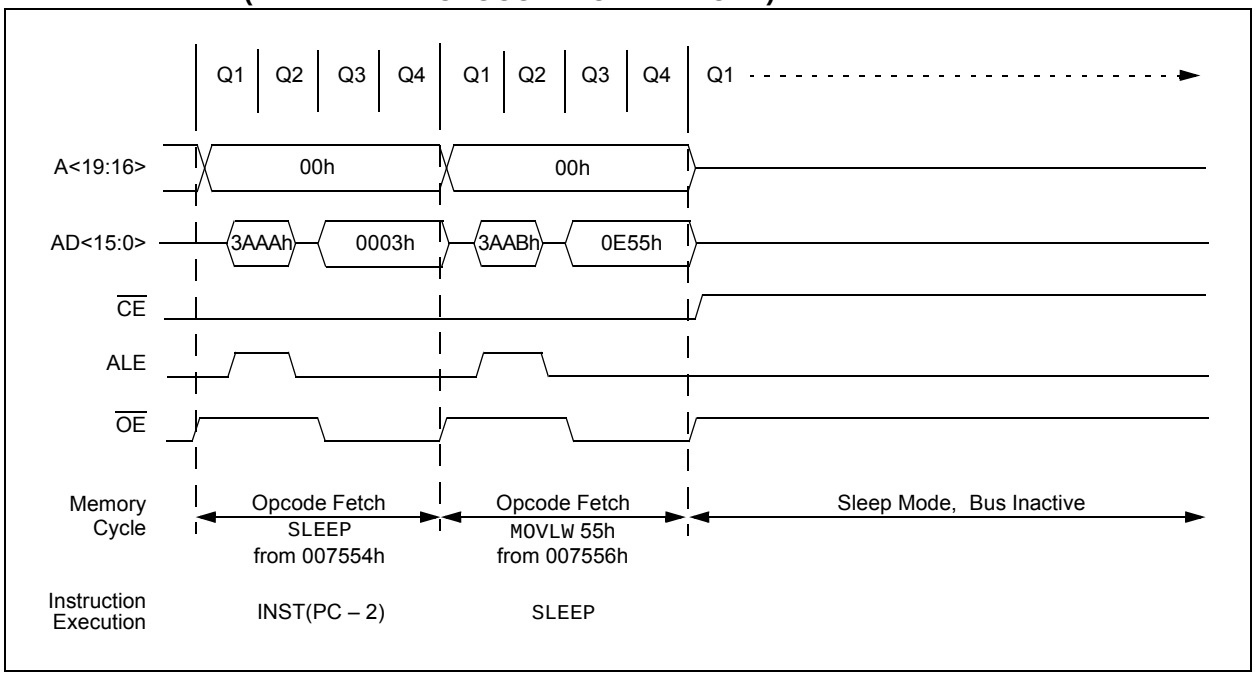


FIGURE 8-5: EXTERNAL MEMORY BUS TIMING FOR SLEEP (EXTENDED MICROCONTROLLER MODE)



9.0 8 x 8 HARDWARE MULTIPLIER

9.1 Introduction

All PIC18 devices include an 8 x 8 hardware multiplier as part of the ALU. The multiplier performs an unsigned operation and yields a 16-bit result that is stored in the product register pair, PRODH:PRODL. The multiplier's operation does not affect any flags in the STATUS register.

Making multiplication a hardware operation allows it to be completed in a single instruction cycle. This has the advantages of higher computational throughput and reduced code size for multiplication algorithms and allows the PIC18 devices to be used in many applications previously reserved for digital signal processors. A comparison of various hardware and software multiply operations, along with the savings in memory and execution time, is shown in Table 9-1.

9.2 Operation

Example 9-1 shows the instruction sequence for an 8 x 8 unsigned multiplication. Only one instruction is required when one of the arguments is already loaded in the WREG register.

Example 9-2 shows the sequence to do an 8 x 8 signed multiplication. To account for the sign bits of the arguments, each argument's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 9-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

```
MOVF    ARG1, W     ;
MULWF   ARG2        ; ARG1 * ARG2 ->
                    ; PRODH:PRODL
```

EXAMPLE 9-2: 8 x 8 SIGNED MULTIPLY ROUTINE

```
MOVF    ARG1, W
MULWF   ARG2        ; ARG1 * ARG2 ->
                    ; PRODH:PRODL
BTFSC   ARG2, SB    ; Test Sign Bit
SUBWF   PRODH, F    ; PRODH = PRODH
                    ;         - ARG1
MOVF    ARG2, W
BTFSC   ARG1, SB    ; Test Sign Bit
SUBWF   PRODH, F    ; PRODH = PRODH
                    ;         - ARG2
```

TABLE 9-1: PERFORMANCE COMPARISON FOR VARIOUS MULTIPLY OPERATIONS

Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time		
				@ 40 MHz	@ 10 MHz	@ 4 MHz
8 x 8 unsigned	Without hardware multiply	13	69	6.9 μs	27.6 μs	69 μs
	Hardware multiply	1	1	100 ns	400 ns	1 μs
8 x 8 signed	Without hardware multiply	33	91	9.1 μs	36.4 μs	91 μs
	Hardware multiply	6	6	600 ns	2.4 μs	6 μs
16 x 16 unsigned	Without hardware multiply	21	242	24.2 μs	96.8 μs	242 μs
	Hardware multiply	28	28	2.8 μs	11.2 μs	28 μs
16 x 16 signed	Without hardware multiply	52	254	25.4 μs	102.6 μs	254 μs
	Hardware multiply	35	40	4.0 μs	16.0 μs	40 μs

TABLE 11-11: PORTE FUNCTIONS (CONTINUED)

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RE5/AD13/ P1C	RE5	0	O	DIG	LATE<5> data output.
		1	I	ST	PORTE<5> data input; weak pull-up when REPU bit is set.
	AD13(1)	x	O	DIG	External memory interface, Address/Data Bit 13 output.(2)
		x	I	TTL	External memory interface, Data Bit 13 input.(2)
	P1C(3)	0	O	DIG	ECCP1 Enhanced PWM output, Channel C; takes priority over port and PSP data. May be configured for tri-state during Enhanced PWM shutdown events.
RE6/AD14/ P1B(4)	RE6	0	O	DIG	LATE<6> data output.
		1	I	ST	PORTE<6> data input; weak pull-up when REPU bit is set.
	AD14(1)	x	O	DIG	External memory interface, Address/Data Bit 14 output.(2)
		x	I	TTL	External memory interface, Data Bit 14 input.(2)
	P1B(3)	0	O	DIG	ECCP1 Enhanced PWM output, Channel B; takes priority over port and PSP data. May be configured for tri-state during Enhanced PWM shutdown events.
RE7/AD15/ ECCP2/P2A(4)	RE7	0	O	DIG	LATE<7> data output.
		1	I	ST	PORTE<7> data input; weak pull-up when REPU bit is set.
	AD15(1)	x	O	DIG	External memory interface, Address/Data Bit 15 output.(2)
		x	I	TTL	External memory interface, Data Bit 15 input.(2)
	ECCP2(5)	0	O	DIG	ECCP2 compare output and PWM output; takes priority over port data.
		1	I	ST	ECCP2 capture input.
	P2A(5)	0	O	DIG	ECCP2 Enhanced PWM output, Channel A; takes priority over port and PSP data. May be configured for tri-state during Enhanced PWM shutdown events.

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

Note 1: EMB functions are implemented on 100-pin devices only.

2: External memory interface I/O takes priority over all other digital and PSP I/O.

3: Default assignments for P1B/P1C and P3B/P3C when ECCPMX Configuration bit is set (80-pin and 100-pin devices).

4: Unimplemented on 64-pin devices.

5: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared (80-pin and 100-pin devices in Microcontroller mode).

6: Unimplemented on 64-pin and 80-pin devices.

TABLE 11-12: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTE	RE7(1)	RE6(1)	RE5	RE4	RE3	RE2	RE1	RE0	72
LATE	LATE7(1)	LATE6(1)	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	72
TRISE	TRISE7(1)	TRISE6(1)	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	71
LATA	RDPU	REPU	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	72

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

Note 1: Unimplemented on 64-pin devices; read as '0'.

15.0 TIMER3 MODULE

The Timer3 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR3H and TMR3L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt on overflow
- Module Reset on CCPx/ECCPx Special Event Trigger

A simplified block diagram of the Timer3 module is shown in Figure 15-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 15-2.

The Timer3 module is controlled through the T3CON register (Register 15-1). It also selects the clock source options for the CCPx and ECCPx modules; see **Section 17.1.1 "CCPx/ECCPx Modules and Timer Resources"** for more information.

REGISTER 15-1: T3CON: TIMER3 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
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	$\overline{\text{T3SYNC}}$	TMR3CS	TMR3ON
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 **RD16:** 16-Bit Read/Write Mode Enable bit
1 = Enables register read/write of Timer3 in one 16-bit operation
0 = Enables register read/write of Timer3 in two 8-bit operations

bit 6,3 **T3CCP<2:1>:** Timer3 and Timer1 to CCPx/ECCPx Enable bits
11 = Timer3 and Timer4 are the clock sources for all CCPx/ECCPx modules
10 = Timer3 and Timer4 are the clock sources for ECCP3, CCP4 and CCP5;
Timer1 and Timer2 are the clock sources for ECCP1 and ECCP2
01 = Timer3 and Timer4 are the clock sources for ECCP2, ECCP3, CCP4 and CCP5;
Timer1 and Timer2 are the clock sources for ECCP1
00 = Timer1 and Timer2 are the clock sources for all CCPx/ECCPx modules

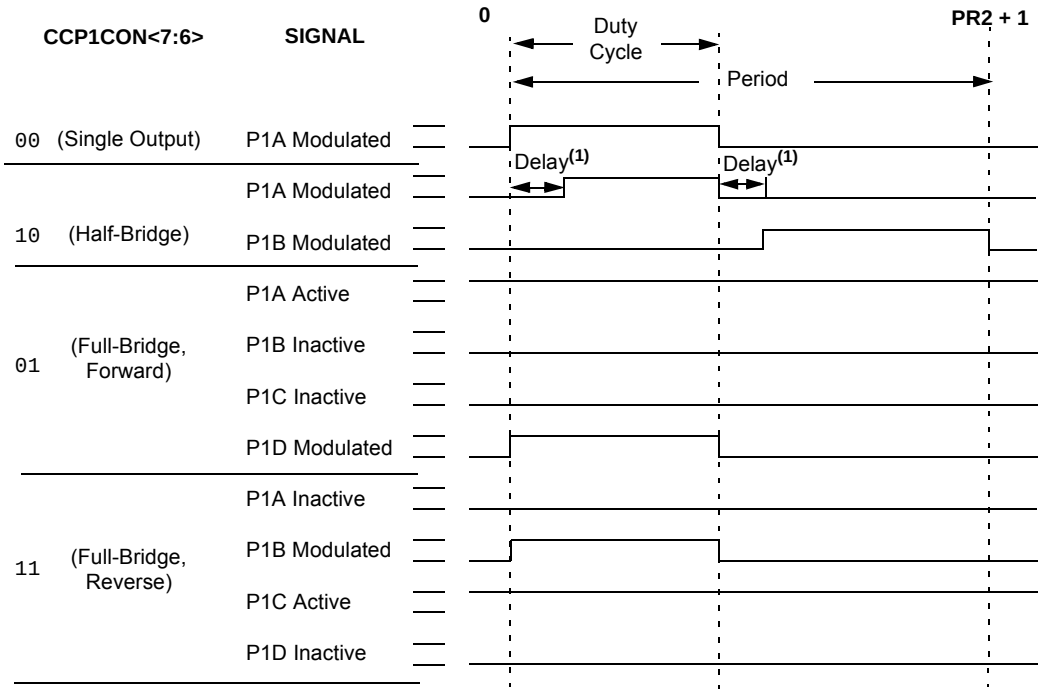
bit 5-4 **T3CKPS<1:0>:** Timer3 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 2 **$\overline{\text{T3SYNC}}$:** Timer3 External Clock Input Synchronization Select bit
(not usable if the device clock comes from Timer1/Timer3)
When TMR3CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR3CS = 0:
This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.

bit 1 **TMR3CS:** Timer3 Clock Source Select bit
1 = External clock input from Timer1 oscillator or T13CKI (on the rising edge after the first falling edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR3ON:** Timer3 On bit
1 = Enables Timer3
0 = Stops Timer3

FIGURE 18-2: PWM OUTPUT RELATIONSHIPS (ACTIVE-HIGH STATE)



Relationships:

- Period = 4 * TOSC * (PR2 + 1) * (TMR2 Prescale Value)
- Duty Cycle = TOSC * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 Prescale Value)
- Delay = 4 * TOSC * (ECCP1DEL<6:0>)

Note 1: Dead-band delay is programmed using the ECCP1DEL register (**Section 18.4.6 "Programmable Dead-Band Delay"**).

18.4.9 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the ECCP1 module for PWM operation:

1. Configure the PWM pins, P1A and P1B (and P1C and P1D, if used), as inputs by setting the corresponding TRIS bits.
2. Set the PWM period by loading the PR2 (PR4) register.
3. Configure the ECCP1 module for the desired PWM mode and configuration by loading the CCP1CON register with the appropriate values:
   - Select one of the available output configurations and direction with the P1M<1:0> bits.
   - Select the polarities of the PWM output signals with the CCP1M<3:0> bits.
4. Set the PWM duty cycle by loading the CCPR1L register and the CCP1CON<5:4> bits.
5. For auto-shutdown:
   - Disable auto-shutdown; ECCP1ASE = 0
   - Configure auto-shutdown source
   - Wait for Run condition
6. For Half-Bridge Output mode, set the dead-band delay by loading ECCP1DEL<6:0> with the appropriate value.
7. If auto-shutdown operation is required, load the ECCP1AS register:
   - Select the auto-shutdown sources using the ECCP1AS<2:0> bits.
   - Select the shutdown states of the PWM output pins using PSS1AC<1:0> and PSS1BD<1:0> bits.
   - Set the ECCP1ASE bit (ECCP1AS<7>).
8. If auto-restart operation is required, set the P1RSEN bit (ECCP1DEL<7>).
9. Configure and start TMR2 (TMR4):
   - Clear the TMRx interrupt flag bit by clearing the TMRxIF bit (PIR1<1> for Timer2 or PIR3<3> for Timer4).
   - Set the TMRx prescale value by loading the TxCKPS bits (T2CON<1:0> for Timer2 or T4CON<1:0> for Timer4).
   - Enable Timer2 (or Timer4) by setting the TMRxON bit (T2CON<2> for Timer2 or T4CON<2> for Timer4).
10. Enable PWM outputs after a new PWM cycle has started:
    - Wait until TMR2 (TMR4) overflows (TMRxIF bit is set).
    - Enable the ECCP1/P1A, P1B, P1C and/or P1D pin outputs by clearing the respective TRIS bits.
    - Clear the ECCP1ASE bit (ECCP1AS<7>).

18.4.10 EFFECTS OF A RESET

Both Power-on Reset and subsequent Resets will force all ports to Input mode and the CCPx/ECCPx registers to their Reset states.

This forces the Enhanced CCPx modules to reset to a state compatible with the standard CCPx modules.

19.2.1.3 Transmit Buffer

Any space within the 8-Kbyte memory which is not programmed as part of the receive FIFO buffer is considered to be the transmit buffer. The responsibility of managing where packets are located in the transmit buffer belongs to the application. Whenever the application decides to transmit a packet, the ETXST and ETXND Pointers are programmed with addresses specifying where, within the transmit buffer, the particular packet to transmit is located. The hardware does not check that the start and end addresses do not overlap with the receive buffer. To prevent buffer corruption, the firmware must not transmit a packet while the ETXST and ETXND Pointers are overlapping the receive buffer, or while the ETXND Pointers are too close to the receive buffer. See **Section 19.5.2 "Transmitting Packets"** for more information.

19.2.1.4 Buffer Arbiter and Access Arbitration

The Ethernet buffer is clocked at one-half of the microcontroller clock rate. Varying amounts of memory access bandwidth are available depending on the clock speed. The total bandwidth available, in bytes per second, is equal to twice the instruction rate (2 * FCY or FOSC/2). For example, at a system clock speed of 41.667 MHz, the total available memory bandwidth that is available is 20.834 Mbyte/s. At an Ethernet signaling rate of 10 Mbit/s, the Ethernet RX engine requires 1.25 Mbyte/s of buffer memory bandwidth to operate without causing an overrun. If Full-Duplex mode is used, an additional 1.25 Mbyte/s is required to allow for simultaneous RX and TX activity.

Because of the finite available memory bandwidth, a three-channel arbiter is used to allocate bandwidth between the RX engine, the TX and DMA engines, and the microcontroller's CPU (i.e., the application accessing EDATA). The arbiter gives the EDATA register accesses first priority, while all remaining bandwidth is shared between the RX and TX/DMA blocks.

With arbitration, bandwidth limitations require that some care be taken in balancing the needs of the module's hardware with that of the application. Accessing the EDATA register too often may result in the RX or TX blocks causing a buffer overrun or underrun, respectively. If such a memory access failure occurs, the BUFER bit (ESTAT<6>), and either the TXERIF or RXERIF interrupt flag, becomes set, and a TX or RX interrupt occurs (if enabled). In either case, the current packet will be lost or aborted.

To eliminate the risk of lost packets, run the microcontroller core at higher speeds. Following the arbitration restrictions, shown in Table 19-2, will prevent memory access failures from occurring. Also, avoid using segments of application code which perform back-to-back accesses of the EDATA register. Instead, insert one or more instructions (including NOP instructions) between each read or write to EDATA.

19.2.1.5 DMA Access to the Buffer

The integrated DMA controller must read from the buffer when calculating a checksum, and it must read and write to the buffer when copying memory. The DMA follows the same wrapping rules as previously described for the receive buffer. While it sequentially reads, it will be subject to a wrapping condition at the end of the receive buffer. All writes it does will not be subject to any wrapping conditions. See **Section 19.9 "Direct Memory Access Controller"** for more information.

TABLE 19-2: BUFFER ARBITRATION RESTRICTIONS VS. CLOCK SPEED

FOSC (MHz)	FCY (MHz)	Available Bandwidth (Mbyte/s)			Application Restrictions to Prevent Underrun/Overrun
		Total	After RX	After TX	
41.667	10.42	20.83	19.58	18.33	Access EDATA no more than once every 2 TCY
31.250	7.81	15.63	14.38	13.13	Access EDATA no more than once every 2 TCY
25.000	6.25	12.50	11.25	10.00	Access EDATA no more than once every 2 TCY
20.833	5.21	10.42	9.17	7.92	Access EDATA no more than once every 2 TCY
13.889	3.47	6.94	5.69	4.44	Access EDATA no more than once every 2 TCY
12.500	3.13	6.25	5.00	3.75	Access EDATA no more than once every 2 TCY
8.333	2.08	4.17	2.92	1.67	Access EDATA no more than once every 3 TCY
6.250	1.56	3.13	1.88	0.63	Access EDATA no more than once every 5 TCY
4.167	1.04	2.08	0.83	< 0	Do not use DMA, do not use full duplex, access EDATA no more than once every 3 TCY
2.778	0.69	1.39	0.14	< 0	Do not use DMA, do not use full duplex, access EDATA no more than once every 10 TCY

19.3.1.1 Receive Error Interrupt (RXERIF)

The receive error interrupt is used to indicate that a packet being received was aborted due to an error condition. Three errors are possible:

1. No buffer space is available to store the incoming packet (buffer overflow);
2. Receiving another packet would cause the EPKTCNT counter to overflow, because it already contains the value, 255; or
3. The Ethernet RX hardware was not allocated enough memory bandwidth to write the incoming data to the buffer.

When a packet is being received and the receive error occurs, the packet being received will be aborted (permanently lost) and the RXERIF bit will be set to '1'. Once set, RXERIF can only be cleared by firmware or by a Reset condition. If the receive error interrupt and Ethernet interrupt are enabled (both RXERIE and ETHIE are set), an Ethernet interrupt is generated. If the receive error interrupt is not enabled (either RXERIE or ETHIE is cleared), the application may poll RXERIF and take appropriate action.

Normally, upon the first two receive error conditions (buffer overflow or potential EPKTCNT overflow), the application would process any packets pending from the receive buffer, and then make additional room for future packets by advancing the ERXRDPT registers (low byte first) and decrementing the EPKTCNT register. See **Section 19.5.3.3 "Freeing Receive Buffer Space"** for more information on processing packets. Once processed, the application should clear the RXERIF bit.

The third condition (insufficient RX memory bandwidth) can be identified by checking if the BUFER bit (ESTAT<6>) has been set. Memory access errors that set BUFER are generally transient in nature, and do not require run-time resolution. Adjustments to the application and its allocation of buffer memory bandwidth may be necessary if BUFER errors are frequent or persistent.

19.3.1.2 Transmit Error Interrupt (TXERIF)

The transmit error interrupt is used to indicate that a transmit abort has occurred. An abort can occur because of any of the following conditions:

1. More than 15 collisions occurred while attempting to transmit a given packet.
2. A late collision (collision after 64 bytes of a packet had been transmitted) has occurred.
3. The transmission was unable to gain an opportunity to transmit the packet because the medium was constantly occupied for too long. The deferral limit was reached and the DEFER bit (MACON4<6>) was clear.
4. An attempt to transmit a packet larger than the maximum frame length, defined by the MAMXFL registers, was made without setting the HFRMEN bit (MACON3<2>) or per-packet POVERRIDE and PHUGEEN bits.
5. The Ethernet buffer did not have enough memory bandwidth to maintain the required 10 Mbit/s transfer rate (buffer underrun).

Upon any of these conditions, the TXERIF flag is set to '1'. Once set, it can only be cleared by firmware or by a Reset condition. If the transmit error interrupt is enabled (TXERIE and ETHIE are both set), an Ethernet interrupt is generated. If the transmit error interrupt is not enabled (either TXERIE or ETHIE is cleared), the application may poll TXERIF and take appropriate action. Once the interrupt is processed, the flag bit should be cleared.

After a transmit abort, the TXRTS bit (ECON1<3>) will be cleared, the TXABRT bit (ESTAT<1>) becomes set and the transmit status vector will be written at the ETXND registers + 1. The MAC will not automatically attempt to retransmit the packet. The application may wish to read the transmit status vector and BUFER bit to determine the cause of the abort. After determining the problem and solution, the application should clear the BUFER (if set) and TXABRT bits so that future aborts can be detected accurately.

In Full-Duplex mode, Conditions 4 and 5 are the only ones that should cause this interrupt. Condition 5 can be further distinguished as it also sets the BUFER bit. Collisions and other problems related to sharing the network are not possible on full-duplex networks. The conditions, which cause the transmit error interrupt, meet the requirements of the transmit interrupt. As a result, when this interrupt occurs, TXIF will also be simultaneously set.

19.3.1.3 Transmit Interrupt (TXIF)

The transmit interrupt is used to indicate that the requested packet transmission has ended (the TXRTS bit has transitioned from '1' to '0'). Upon transmission completion, abort, or transmission cancellation by the application, the TXIF flag will be set to '1'. If the application did not clear the TXRTS bit, and the TXABRT bit is not set, the packet was successfully transmitted. Once TXIF is set, it can only be cleared in software or by a Reset condition. If the transmit interrupt is enabled (TXIE and ETHIE are both set), an interrupt is generated. If the transmit interrupt is not enabled (either TXIE or ETHIE is cleared), the application may poll the TXIF bit and take appropriate action.

19.3.1.4 Link Change Interrupt (LINKIF)

The LINKIF indicates that the link status has changed. The actual current link status can be obtained from the LLSTAT (PHSTAT1<2>) or LSTAT (PHSTAT2<10>) bits (see Register 19-10 and Register 19-12). Unlike other interrupt sources, the link status change interrupt is created in the integrated PHY module; additional steps must be taken to enable it.

By Reset default, LINKIF is never set for any reason. To receive it, both the PLNKIE and PGEIE bits must be set. When the interrupt is enabled, the LINKIF bit will shadow the contents of the PGIF bit. The PHY only supports one interrupt, so the PGIF bit will always be the same as the PLNKIF bit (when both PHY enable bits are set).

Once LINKIF is set, it can only be cleared in software or by a Reset. If the link change interrupt is enabled (LINKIE, PLNKIE, PGEIE and ETHIE are all set), an interrupt is generated. If the link change interrupt is not enabled (LINKIE, PLNKIE, PGEIE or ETHIE are cleared), the user application may poll the PLNKIF flag and take appropriate action.

The LINKIF bit is read-only. Because reading PHY registers requires a non-negligible period of time, the application may instead set PLNKIE and PGEIE, then poll the LINKIF flag bit. Performing an MII read on the PHIR register will clear the LINKIF, PGIF and PLNKIF bits automatically, and allow for future link status change interrupts. See **Section 19.2.5 “PHY Registers”** for information on accessing the PHY registers.

19.3.1.5 DMA Interrupt (DMAIF)

The DMA interrupt indicates that the DMA module has completed its memory copy or checksum calculation (the DMAST bit has transitioned from ‘1’ to ‘0’). Additionally, this interrupt will be caused if the application cancels a DMA operation by manually clearing the DMAST bit. Once set, DMAIF can only be cleared by the firmware or by a Reset condition. If the DMA interrupt is enabled, an Ethernet interrupt is generated. If the DMA interrupt is not enabled, the user application may poll the DMAIF flag status and take appropriate action. Once processed, the flag bit should be cleared.

19.3.1.6 Receive Packet Pending Interrupt (PKTIF)

The receive packet pending interrupt is used to indicate the presence of one or more data packets in the receive buffer and to provide a notification means for the arrival of new packets. When the receive buffer has at least one packet in it, the PKTIF flag bit is set. In other words, this interrupt flag will be set any time the Ethernet Packet Count register (EPKTCNT) is non-zero.

When the receive packet pending interrupt is enabled (both PKTIE and ETHIE are set), an Ethernet interrupt is generated whenever a new packet is successfully received and written into the receive buffer. If the receive packet pending interrupt is not enabled (either PKTIE or ETHIE is cleared), the user application may poll the PKTIF bit and take appropriate action.

The PKTIF bit can only be cleared indirectly in software, by decrementing the EPKTCNT register to ‘0’, or by a Reset condition. See **Section 19.5.3 “Receiving Packets”** for more information about clearing the EPKTCNT register. When the last data packet in the receive buffer is processed, EPKTCNT becomes zero and the PKTIF bit is automatically cleared.

19.3.2 ETHERNET INTERRUPTS AND WAKE-ON-LAN

The Ethernet interrupt structure implements a version of Wake-on-LAN, also called Remote Wake-up, using a Magic Packet data packet. This allows the application to conserve power in Idle mode, and then return to full-power operation only when a specific wake-up packet is received.

For Remote Wake-up to work, the Ethernet module must remain enabled at all times. It is also necessary to configure the receive filters to select for Magic Packets. For more information on filter configuration, see **Section 19.8 “Receive Filters”**.

To configure the microcontroller for Remote Wake-up:

1. With the Ethernet module enabled and in normal operating configuration, enable the CRC post-filter and Magic Packets filter (ERXFCON<5,3> = 1).
2. Finish processing any pending packets in the Ethernet buffer.
3. Enable Ethernet interrupts at the microcontroller level (PIE2<5> = 1) and the receive packet pending interrupt at the module level (EIE<6> = 1).
4. Place the microcontroller in PRI_IDLE mode (with the primary clock source selected and OSCCON<7> = 1, execute the SLEEP instruction).

In this configuration, the receipt of a Magic Packet data packet will cause a receive packet pending interrupt. This, in turn, will cause the microcontroller to wake-up from the interrupt.

20.4.8 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Enable bit, SEN (SSPxCON2<0>). If the SDAx and SCLx pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<6:0> and starts its count. If SCLx and SDAx are both sampled high when the Baud Rate Generator times out (TBRG), the SDAx pin is driven low. The action of the SDAx being driven low while SCLx is high is the Start condition and causes the S bit (SSPxSTAT<3>) to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<6:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit (SSPxCON2<0>) will be automatically cleared by hardware. The Baud Rate Generator is suspended, leaving the SDAx line held low and the Start condition is complete.

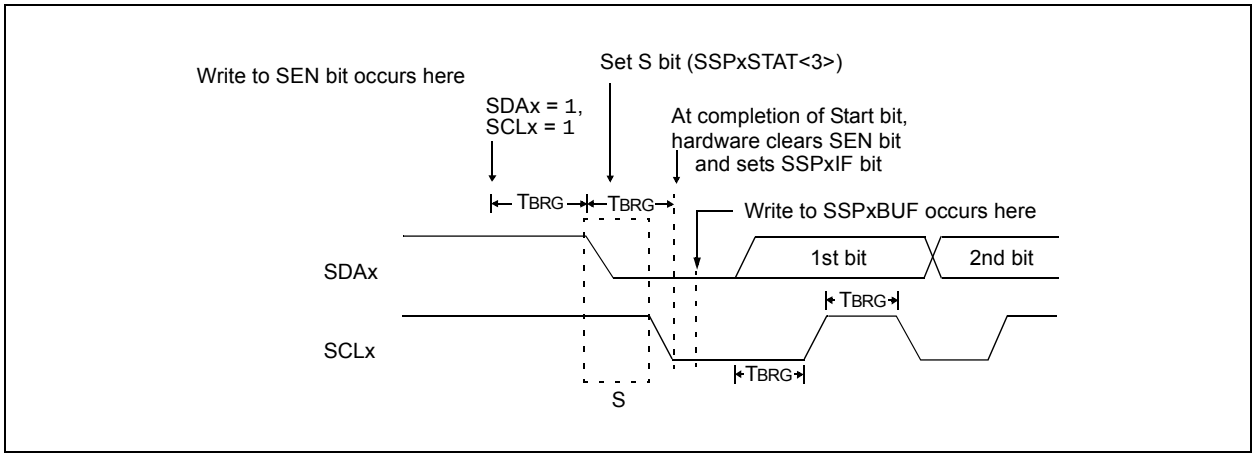
Note:	If, at the beginning of the Start condition, the SDAx and SCLx pins are already sampled low, or if during the Start condition, the SCLx line is sampled low before the SDAx line is driven low, a bus collision occurs. The Bus Collision Interrupt Flag, BCLxIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

20.4.8.1 WCOL Status Flag

If the user writes the SSPxBUF when a Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

Note:	Because queueing of events is not allowed, writing to the lower 5 bits of SSPxCON2 is disabled until the Start condition is complete.

FIGURE 20-21: FIRST START BIT TIMING



21.2.4 AUTO-WAKE-UP ON SYNC BREAK CHARACTER

During Sleep mode, all clocks to the EUSARTx are suspended. Because of this, the Baud Rate Generator is inactive and a proper byte reception cannot be performed. The auto-wake-up feature allows the controller to wake-up due to activity on the RXx/DTx line while the EUSARTx is operating in Asynchronous mode.

The auto-wake-up feature is enabled by setting the WUE bit (BAUDCONx<1>). Once set, the typical receive sequence on RXx/DTx is disabled and the EUSARTx remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RXx/DTx line. (This coincides with the start of a Sync Break or a Wake-up Signal character for the LIN/J2602 protocol.)

Following a wake-up event, the module generates an RCxIF interrupt. The interrupt is generated synchronously to the Q clocks in normal operating modes (Figure 21-8) and asynchronously if the device is in Sleep mode (Figure 21-9). The interrupt condition is cleared by reading the RCREGx register.

The WUE bit is automatically cleared once a low-to-high transition is observed on the RXx line following the wake-up event. At this point, the EUSARTx module is in Idle mode and returns to normal operation. This signals to the user that the Sync Break event is over.

21.2.4.1 Special Considerations Using Auto-Wake-up

Since auto-wake-up functions by sensing rising edge transitions on RXx/DTx, information with any state changes before the Stop bit may signal a false End-of-Character (EOC) and cause data or framing errors. To work properly, therefore, the initial character in the transmission must be all '0's. This can be 00h (8 bytes) for standard RS-232 devices or 000h (12 bits) for LIN/J2602 bus.

Oscillator start-up time must also be considered, especially in applications using oscillators with longer start-up intervals (i.e., HS or HSPLL mode). The Sync Break (or Wake-up Signal) character must be of sufficient length and be followed by a sufficient interval to allow enough time for the selected oscillator to start and provide proper initialization of the EUSARTx.

21.2.4.2 Special Considerations Using the WUE Bit

The timing of WUE and RCxIF events may cause some confusion when it comes to determining the validity of received data. As noted, setting the WUE bit places the EUSARTx in an Idle mode. The wake-up event causes a receive interrupt by setting the RCxIF bit. The WUE bit is cleared after this when a rising edge is seen on RXx/DTx. The interrupt condition is then cleared by reading the RCREGx register. Ordinarily, the data in RCREGx will be dummy data and should be discarded.

The fact that the WUE bit has been cleared (or is still set), and the RCxIF flag is set, should not be used as an indicator of the integrity of the data in RCREGx. Users should consider implementing a parallel method in firmware to verify received data integrity.

To assure that no actual data is lost, check the RCIDL bit to verify that a receive operation is not in process. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

22.1 A/D Acquisition Requirements

For the A/D Converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 22-2. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor, CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD). The source impedance affects the offset voltage at the analog input (due to pin leakage current). **The maximum recommended impedance for analog sources is 2.5 kΩ.** After the analog input channel is selected (changed), the channel must be sampled for at least the minimum acquisition time before starting a conversion.

Note:	When the conversion is started, the holding capacitor is disconnected from the input pin.

To calculate the minimum acquisition time, Equation 22-1 may be used. This equation assumes that 1/2 LSb error is used (1024 steps for the A/D). The 1/2 LSb error is the maximum error allowed for the A/D to meet its specified resolution.

Equation 22-3 shows the calculation of the minimum required acquisition time, TACQ. This calculation is based on the following application system assumptions:

CHOLD	=	25 pF
Rs	=	2.5 kΩ
Conversion Error	≤	1/2 LSb
VDD	=	3V → Rss = 2 kΩ
Temperature	=	85°C (system max.)

EQUATION 22-1: ACQUISITION TIME

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Holding Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$

EQUATION 22-2: A/D MINIMUM CHARGING TIME

$$V_{HOLD} = (V_{REF} - (V_{REF}/2048)) \bullet (1 - e^{(-T_C/C_{HOLD}(R_{IC} + R_{SS} + R_S))})$$
or
$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S)\ln(1/2048)$$

EQUATION 22-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

$$T_{ACQ} = T_{AMP} + T_C + T_{COFF}$$
$$T_{AMP} = 0.2\ \mu s$$
$$T_{COFF} = (\text{Temp} - 25°C)(0.02\ \mu s/°C)$$
$$(85°C - 25°C)(0.02\ \mu s/°C)$$
$$1.2\ \mu s$$

Temperature coefficient is only required for temperatures > 25°C. Below 25°C, T_{COFF} = 0 ms.

$$T_C = -(C_{HOLD})(R_{IC} + R_{SS} + R_S)\ln(1/2048)\ \mu s$$
$$-(25\ pF)(1\ k\Omega + 2\ k\Omega + 2.5\ k\Omega)\ln(0.0004883)\ \mu s$$
$$1.05\ \mu s$$
$$T_{ACQ} = 0.2\ \mu s + 1\ \mu s + 1.2\ \mu s$$
$$2.4\ \mu s$$

22.7 A/D Converter Calibration

The A/D Converter in the PIC18F97J60 family of devices includes a self-calibration feature which compensates for any offset generated within the module. The calibration process is automated and is initiated by setting the ADCAL bit (ADCON0<7>). The next time the GO/$\overline{\text{DONE}}$ bit is set, the module will perform a "dummy" conversion (that is, with reading none of the input channels) and store the resulting value internally to compensate for offset. Thus, subsequent offsets will be compensated.

The calibration process assumes that the device is in a relatively steady-state operating condition. If A/D calibration is used, it should be performed after each device Reset, or if there are other major changes in operating conditions.

22.8 Operation in Power-Managed Modes

The selection of the automatic acquisition time and A/D conversion clock is determined in part by the clock source and frequency while in a power-managed mode.

If the A/D is expected to operate while the device is in a power-managed mode, the ACQT<2:0> and ADCS<2:0> bits in ADCON2 should be updated in accordance with the power-managed mode clock that will be used. After the power-managed mode is entered (either of the power-managed Run modes), an A/D acquisition or conversion may be started. Once an acquisition or conversion is started, the device should continue to be clocked by the same power-managed mode clock source until the conversion has been completed. If desired, the device may be placed into the corresponding power-managed Idle mode during the conversion.

If the power-managed mode clock frequency is less than 1 MHz, the A/D RC clock source should be selected.

Operation in Sleep mode requires the A/D RC clock to be selected. If bits, ACQT<2:0>, are set to '000' and a conversion is started, the conversion will be delayed one instruction cycle to allow execution of the SLEEP instruction and entry to Sleep mode. The IDLEN and SCS bits in the OSCCON register must have already been cleared prior to starting the conversion.

TABLE 22-2: SUMMARY OF A/D REGISTERS

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	69
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	71
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	71
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	CCP1IP	TMR2IP	TMR1IP	71
PIR2	OSCFIF	CMIF	ETHIF	r	BCL1IF	—	TMR3IF	CCP2IF	71
PIE2	OSCFIE	CMIE	ETHIE	r	BCL1IE	—	TMR3IE	CCP2IE	71
IPR2	OSCFIP	CMIP	ETHIP	r	BCL1IP	—	TMR3IP	CCP2IP	71
ADRESH	A/D Result Register High Byte								70
ADRESL	A/D Result Register Low Byte								70
ADCON0	ADCAL	—	CHS3	CHS3	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	70
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	70
ADCON2	ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0	70
CCP2CON	P2M1	P2M0	DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0	70
PORTA	RJPU	—	RA5	RA4	RA3	RA2	RA1	RA0	72
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	71
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0(1)	72
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0(1)	71
PORTH(2)	RH7	RH6	RH5	RH4	RH3	RH2	RH1	RH0	72
TRISH(2)	TRISH7	TRISH6	TRISH5	TRISH4	TRISH3	TRISH2	TRISH1	TRISH0	71

Legend: — = unimplemented, read as '0', r = reserved. Shaded cells are not used for A/D conversion.

Note 1: Implemented in 100-pin devices only.

2: This register is not implemented in 64-pin devices.

NOTES:

REGISTER 25-5: CONFIG3L: CONFIGURATION REGISTER 3 LOW (BYTE ADDRESS 300004h)

R/WO-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1	U-0	U-0	U-0
WAIT(1)	BW(1)	EMB1(1)	EMB0(1)	EASHFT(1)	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	WO = Write-Once bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed	'1' = Bit is set	'0' = Bit is cleared

bit 7 **WAIT:** External Bus Wait Enable bit(1)
- 1 = Wait states for operations on external memory bus is disabled
- 0 = Wait states for operations on external memory bus is enabled and selected by MEMCON<5:4>

bit 6 **BW:** Data Bus Width Select bit(1)
- 1 = 16-Bit Data Width mode
- 0 = 8-Bit Data Width mode

bit 5-4 **EMB<1:0>:** External Memory Bus Configuration bits(1)
- 11 = Microcontroller mode, external bus disabled
- 10 = Extended Microcontroller mode,12-Bit Addressing mode
- 01 = Extended Microcontroller mode,16-Bit Addressing mode
- 00 = Extended Microcontroller mode, 20-Bit Addressing mode

bit 3 **EASHFT:** External Address Bus Shift Enable bit(1)
- 1 = Address shifting is enabled; address on external bus is offset to start at 000000h
- 0 = Address shifting is disabled; address on external bus reflects the PC value

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

Note 1: Implemented on 100-pin devices only.

REGISTER 25-7: DEVID1: DEVICE ID REGISTER 1 FOR PIC18F97J60 FAMILY DEVICES

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7							bit 0

Legend:		
R = Read-only bit	P = Programmable bit	U = Unimplemented bit, read as ‘0’
-n = Value when device is unprogrammed		u = Unchanged from programmed state

bit 7-5 **DEV<2:0>:** Device ID bits
See Register 25-8 for a complete listing.

bit 4-0 **REV<4:0>:** Revision ID bits
These bits are used to indicate the device revision.

REGISTER 25-8: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F97J60 FAMILY DEVICES

R	R	R	R	R	R	R	R
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7							bit 0

Legend:		
R = Read-only bit	P = Programmable bit	U = Unimplemented bit, read as ‘0’
-n = Value when device is unprogrammed		u = Unchanged from programmed state

bit 7-0 **DEV<10:3>:** Device ID bits:

DEV<10:3> (DEVID2<7:0>)	DEV<2:0> (DEVID1<7:5>)	Device
0001 1000	000	PIC18F66J60
0001 1111	000	PIC18F66J65
0001 1111	001	PIC18F67J60
0001 1000	001	PIC18F86J60
0001 1111	010	PIC18F86J65
0001 1111	011	PIC18F87J60
0001 1000	010	PIC18F96J60
0001 1111	100	PIC18F96J65
0001 1111	101	PIC18F97J60

RETFIE **Return from Interrupt**

Syntax: RETFIE {s}

Operands: s ∈ [0,1]

Operation: (TOS) → PC,
1 → GIE/GIEH or PEIE/GIEL;
if s = 1,
(WS) → W,
(STATUSS) → STATUS,
(BSRS) → BSR,
PCLATU, PCLATH are unchanged

Status Affected: GIE/GIEH, PEIE/GIEL.

Encoding:

0000	0000	0001	000s

Description: Return from interrupt. Stack is popped and Top-of-Stack (TOS) is loaded into the PC. Interrupts are enabled by setting either the high or low-priority Global Interrupt Enable bit. If 's' = 1, the contents of the shadow registers WS, STATUSS and BSRS are loaded into their corresponding registers W, STATUS and BSR. If 's' = 0, no update of these registers occurs (default).

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	POP PC from stack Set GIEH or GIEL
No operation	No operation	No operation	No operation

Example: RETFIE 1

After Interrupt
PC = TOS
W = WS
BSR = BSRS
STATUS = STATUSS
GIE/GIEH, PEIE/GIEL = 1

RETLW **Return Literal to W**

Syntax: RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → W,
(TOS) → PC,
PCLATU, PCLATH are unchanged

Status Affected: None

Encoding:

0000	1100	kkkk	kkkk

Description: W is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). The high address latch (PCLATH) remains unchanged.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	POP PC from stack, write to W
No operation	No operation	No operation	No operation

Example:

```
    CALL TABLE ; W contains table
               ; offset value
               ; W now has
               ; table value
    :
TABLE
    ADDWF PCL  ; W = offset
    RETLW k0   ; Begin table
    RETLW k1   ;
    :
    :
    RETLW kn   ; End of table
```

Before Instruction
W = 07h
After Instruction
W = value of kn