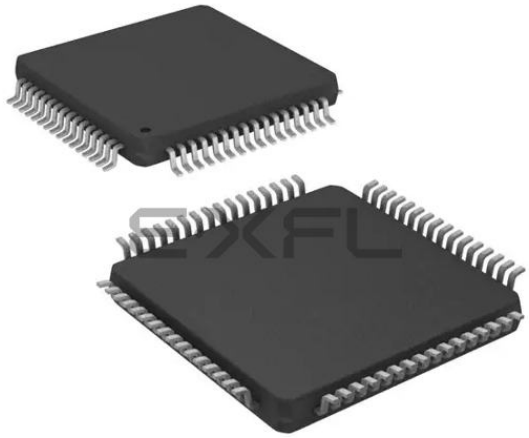


Welcome to E-XFL.COM

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	96KB (48K 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/pic18f66j65-i-pt

TABLE 1-5: PIC18F86J60/86J65/87J60 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
				PORTC is a bidirectional I/O port.
RC0/T1OSO/T13CKI	36			
RC0		I/O	ST	Digital I/O.
T1OSO		O	—	Timer1 oscillator output.
T13CKI		I	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/ECCP2/P2A	35			
RC1		I/O	ST	Digital I/O.
T1OSI		I	CMOS	Timer1 oscillator input.
ECCP2(1)		I/O	ST	Capture 2 input/Compare 2 output/PWM2 output.
P2A(1)		O	—	ECCP2 PWM Output A.
RC2/ECCP1/P1A	43			
RC2		I/O	ST	Digital I/O.
ECCP1		I/O	ST	Capture 1 input/Compare 1 output/PWM1 output.
P1A		O	—	ECCP1 PWM Output A.
RC3/SCK1/SCL1	44			
RC3		I/O	ST	Digital I/O.
SCK1		I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL1		I/O	ST	Synchronous serial clock input/output for I^2C™ mode.
RC4/SDI1/SDA1	45			
RC4		I/O	ST	Digital I/O.
SDI1		I	ST	SPI data in.
SDA1		I/O	ST	I^2C data I/O.
RC5/SDO1	46			
RC5		I/O	ST	Digital I/O.
SDO1		O	—	SPI data out.
RC6/TX1/CK1	37			
RC6		I/O	ST	Digital I/O.
TX1		O	—	EUSART1 asynchronous transmit.
CK1		I/O	ST	EUSART1 synchronous clock (see related RX1/DT1 pin).
RC7/RX1/DT1	38			
RC7		I/O	ST	Digital I/O.
RX1		I	ST	EUSART1 asynchronous receive.
DT1		I/O	ST	EUSART1 synchronous data (see related TX1/CK1 pin).

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})

Note 1: Default assignment for ECCP2/P2A when CCP2MX Configuration bit is set.

2: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

3: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared.

4: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

TABLE 1-6: PIC18F96J60/96J65/97J60 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
				PORTC is a bidirectional I/O port.
RC0/T1OSO/T13CKI	44			
RC0		I/O	ST	Digital I/O.
T1OSO		O	—	Timer1 oscillator output.
T13CKI		I	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/ECCP2/P2A	43			
RC1		I/O	ST	Digital I/O.
T1OSI		I	CMOS	Timer1 oscillator input.
ECCP2(2)		I/O	ST	Capture 2 input/Compare 2 output/PWM2 output.
P2A(2)		O	—	ECCP2 PWM Output A.
RC2/ECCP1/P1A	53			
RC2		I/O	ST	Digital I/O.
ECCP1		I/O	ST	Capture 1 input/Compare 1 output/PWM1 output.
P1A		O	—	ECCP1 PWM Output A.
RC3/SCK1/SCL1	54			
RC3		I/O	ST	Digital I/O.
SCK1		I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL1		I/O	ST	Synchronous serial clock input/output for I^2C™ mode.
RC4/SDI1/SDA1	55			
RC4		I/O	ST	Digital I/O.
SDI1		I	ST	SPI data in.
SDA1		I/O	ST	I^2C data I/O.
RC5/SDO1	56			
RC5		I/O	ST	Digital I/O.
SDO1		O	—	SPI data out.
RC6/TX1/CK1	45			
RC6		I/O	ST	Digital I/O.
TX1		O	—	EUSART1 asynchronous transmit.
CK1		I/O	ST	EUSART1 synchronous clock (see related RX1/DT1 pin).
RC7/RX1/DT1	46			
RC7		I/O	ST	Digital I/O.
RX1		I	ST	EUSART1 asynchronous receive.
DT1		I/O	ST	EUSART1 synchronous data (see related TX1/CK1 pin).

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})

Note 1: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared (Extended Microcontroller mode).

2: Default assignment for ECCP2/P2A for all devices in all operating modes (CCP2MX Configuration bit is set).

3: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

4: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared (Microcontroller mode).

5: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

6.4.3.2 FSR Registers and POSTINC, POSTDEC, PREINC and PLUSW

In addition to the INDF operand, each FSR register pair also has four additional indirect operands. Like INDF, these are "virtual" registers that cannot be indirectly read or written to. Accessing these registers actually accesses the associated FSR register pair, but also performs a specific action on its stored value. They are:

- POSTDEC: accesses the FSR value, then automatically decrements it by '1' afterwards
- POSTINC: accesses the FSR value, then automatically increments it by '1' afterwards
- PREINC: increments the FSR value by '1', then uses it in the operation
- PLUSW: adds the signed value of the W register (range of -128 to 127) to that of the FSR and uses the new value in the operation

In this context, accessing an INDF register uses the value in the FSR registers without changing them. Similarly, accessing a PLUSW register gives the FSR value offset by the value in the W register; neither value is actually changed in the operation. Accessing the other virtual registers changes the value of the FSR registers.

Operations on the FSRs with POSTDEC, POSTINC and PREINC affect the entire register pair; that is, roll-overs of the FSRnL register, from FFh to 00h, carry over to the FSRnH register. On the other hand, results of these operations do not change the value of any flags in the STATUS register (e.g., Z, N, OV, etc.).

The PLUSW register can be used to implement a form of Indexed Addressing in the data memory space. By manipulating the value in the W register, users can reach addresses that are fixed offsets from pointer addresses. In some applications, this can be used to implement some powerful program control structure, such as software stacks, inside of data memory.

6.4.3.3 Operations by FSRs on FSRs

Indirect Addressing operations that target other FSRs, or virtual registers, represent special cases. For example, using an FSR to point to one of the virtual registers will not result in successful operation. As a specific case, assume that the FSR0H:FSR0L pair contains FE7h, the address of INDF1. Attempts to read the value of the INDF1, using INDF0 as an operand, will return 00h. Attempts to write to INDF1, using INDF0 as the operand, will result in a NOP.

On the other hand, using the virtual registers to write to an FSR pair may not occur as planned. In these cases, the value will be written to the FSR pair but without any incrementing or decrementing. Thus, writing to INDF2 or POSTDEC2 will write the same value to the FSR2H:FSR2L pair.

Since the FSRs are physical registers mapped in the SFR space, they can be manipulated through all direct operations. Users should proceed cautiously when working on these registers, particularly if their code uses Indirect Addressing.

Similarly, operations by Indirect Addressing are generally permitted on all other SFRs. Users should exercise the appropriate caution that they do not inadvertently change settings that might affect the operation of the device.

6.5 Program Memory and the Extended Instruction Set

The operation of program memory is unaffected by the use of the extended instruction set.

Enabling the extended instruction set adds five additional two-word commands to the existing PIC18 instruction set: ADDFSR, CALLW, MOVSF, MOVSS and SUBFSR. These instructions are executed as described in **Section 6.2.4 "Two-Word Instructions"**.

6.6 Data Memory and the Extended Instruction Set

Enabling the PIC18 extended instruction set (XINST Configuration bit = 1) significantly changes certain aspects of data memory and its addressing. Specifically, the use of the Access Bank for many of the core PIC18 instructions is different. This is due to the introduction of a new addressing mode for the data memory space. This mode also alters the behavior of Indirect Addressing using FSR2 and its associated operands.

What does not change is just as important. The size of the data memory space is unchanged, as well as its linear addressing. The SFR map remains the same. Core PIC18 instructions can still operate in both Direct and Indirect Addressing mode; inherent and literal instructions do not change at all. Indirect Addressing with FSR0 and FSR1 also remains unchanged.

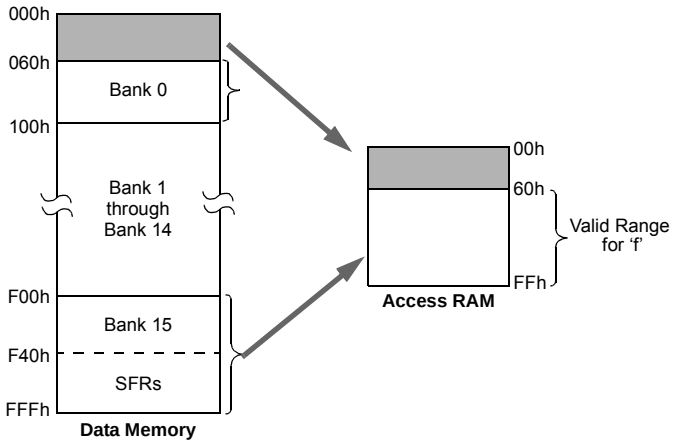
FIGURE 6-10: COMPARING ADDRESSING OPTIONS FOR BIT-ORIENTED AND BYTE-ORIENTED INSTRUCTIONS (EXTENDED INSTRUCTION SET ENABLED)

EXAMPLE INSTRUCTION: `ADDWF, f, d, a` (Opcode: `0010 01da ffff fff`

When a = 0 and f ≥ 60h:

The instruction executes in Direct Forced mode. 'f' is interpreted as a location in the Access RAM between 060h and FFFh. This is the same as locations F60h to FFFh (Bank 15) of data memory.

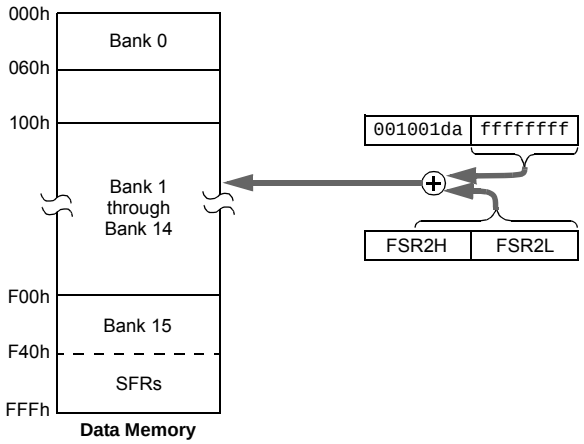
Locations below 060h are not available in this addressing mode.



When a = 0 and f ≤ 5Fh:

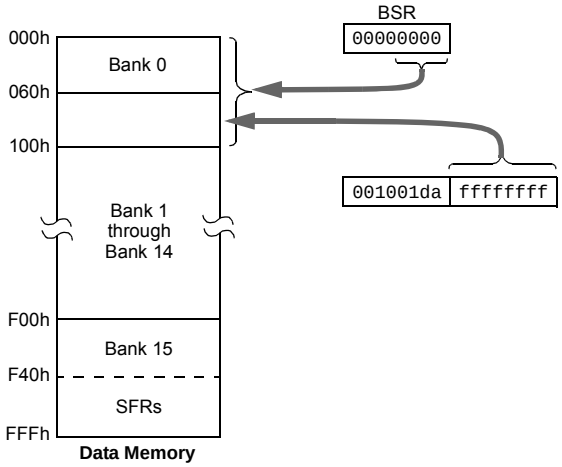
The instruction executes in Indexed Literal Offset mode. 'f' is interpreted as an offset to the address value in FSR2. The two are added together to obtain the address of the target register for the instruction. The address can be anywhere in the data memory space.

Note that in this mode, the correct syntax is now:
`ADDWF [k], d`
where 'k' is the same as 'f'.



When a = 1 (all values of f):

The instruction executes in Direct mode (also known as Direct Long mode). 'f' is interpreted as a location in one of the 16 banks of the data memory space. The bank is designated by the Bank Select Register (BSR). The address can be in any implemented bank in the data memory space.



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

17.4 PWM Mode

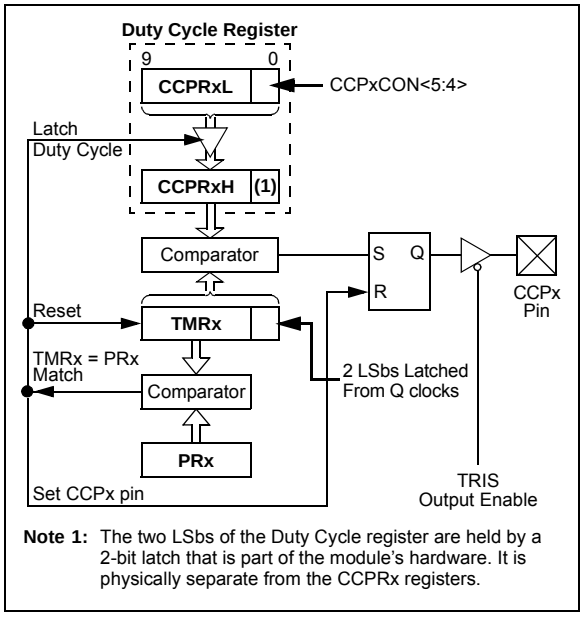
In Pulse-Width Modulation (PWM) mode, the CCPx pin produces up to a 10-bit resolution PWM output. Since the CCP4 and CCP5 pins are multiplexed with a PORTG data latch, the appropriate TRISG bit must be cleared to make the CCP4 or CCP5 pin an output.

Note:	Clearing the CCP4CON or CCP5CON register will force the RG3 or RG4 output latch (depending on device configuration) to the default low level. This is not the PORTG I/O data latch.

Figure 17-4 shows a simplified block diagram of the CCPx module in PWM mode.

For a step-by-step procedure on how to set up a CCPx module for PWM operation, see **Section 17.4.3 "Setup for PWM Operation"**.

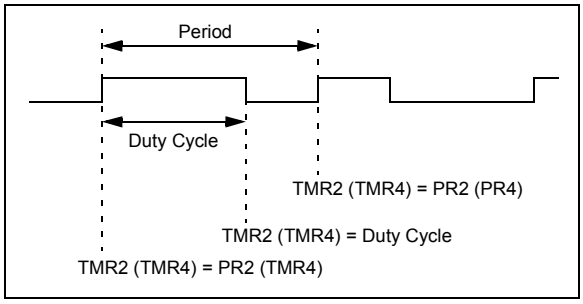
FIGURE 17-4: SIMPLIFIED PWM BLOCK DIAGRAM



Note 1: The two LSbs of the Duty Cycle register are held by a 2-bit latch that is part of the module's hardware. It is physically separate from the CCPRx registers.

A PWM output (Figure 17-5) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 17-5: PWM OUTPUT



17.4.1 PWM PERIOD

The PWM period is specified by writing to the PR2 (PR4) register. The PWM period can be calculated using Equation 17-1:

EQUATION 17-1:

$$\text{PWM Period} = [(\text{PR2}) + 1] \cdot 4 \cdot T_{OSC} \cdot (\text{TMR2 Prescale Value})$$

PWM frequency is defined as 1/[PWM period].

When TMR2 (TMR4) is equal to PR2 (PR4), the following three events occur on the next increment cycle:

- TMR2 (TMR4) is cleared
- The CCPx pin is set (exception: if PWM duty cycle = 0%, the CCPx pin will not be set)
- The PWM duty cycle is latched from CCPRxL into CCPRxH

Note:	The Timer2 and Timer4 postscalers (see **Section 14.0 "Timer2 Module"** and **Section 16.0 "Timer4 Module"**) are not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

17.4.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPRxL register and to the CCPxCON<5:4> bits. Up to 10-bit resolution is available. The CCPRxL contains the eight MSbs and the CCPxCON<5:4> contains the two LSbs. This 10-bit value is represented by CCPRxL:CCPxCON<5:4>. Equation 17-2 is used to calculate the PWM duty cycle in time.

EQUATION 17-2:

$$\text{PWM Duty Cycle} = (\text{CCPRxL:CCPxCON<5:4>}) \cdot T_{OSC} \cdot (\text{TMRx Prescale Value})$$

CCPRxL and CCPxCON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPRxH until after a match between PR2 (PR4) and TMR2 (TMR4) occurs (i.e., the period is complete). In PWM mode, CCPRxH is a read-only register.

REGISTER 19-9: PHCON1: PHY CONTROL REGISTER 1

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0	R/W-0
r	r	—	—	r	r	—	PDPXMD
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
r	—	—	—	—	—	—	—
bit 7							bit 0

Legend:	r = Reserved bit		
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 15-14 **Reserved:** Write as ‘0’

bit 13-12 **Unimplemented:** Read as ‘0’

bit 11-10 **Reserved:** Write as ‘0’

bit 9 **Unimplemented:** Read as ‘0’

bit 8 **PDPXMD:** PHY Duplex Mode bit
1 = PHY operates in Full-Duplex mode; application must also set FULDPX (MACON3<0>)
0 = PHY operates in Half-Duplex mode, application must also clear FULDP

bit 7 **Reserved:** Maintain as ‘0’

bit 6-0 **Unimplemented:** Read as ‘0’

REGISTER 19-10: PHSTAT1: PHYSICAL LAYER STATUS REGISTER 1

U-0	U-0	U-0	R-1	R-1	U-0	U-0	U-0
—	—	—	r	r	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/LL-0	R/LH-0	U-0
—	—	—	—	—	LLSTAT	r	—
bit 7							bit 0

Legend:	‘1’ = Bit is set	r = Reserved bit	
R = Read-only bit	‘0’ = Bit is cleared	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	R/L = Read-Only Latch bit	LL = Latches Low bit	LH = Latches High bit

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-11 **Reserved:** Read as ‘1’

bit 10-3 **Unimplemented:** Read as ‘0’

bit 2 **LLSTAT:** PHY Latching Link Status bit
1 = Link is up and has been up continously since PHSTAT1 was last read
0 = Link is down or was down for a period since PHSTAT1 was last read

bit 1 **Reserved:** Ignore on read

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 19-13: PHLCON: PHY MODULE LED CONTROL REGISTER

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-1	R/W-0	R/W-0
r	r	r	r	LACFG3	LACFG2	LACFG1	LACFG0
bit 15							bit 8

R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-1	R/W-x
LBCFG3	LBCFG2	LBCFG1	LBCFG0	LFRQ1	LFRQ0	STRCH	r
bit 7							bit 0

Legend:	r = Reserved bit		
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 15-14 **Reserved:** Write as '0'

bit 13-12 **Reserved:** Write as '1'

bit 11-8 **LACFG<3:0>:** LEDA Configuration bits
0000 = Reserved
0001 = Display transmit activity (stretchable)
0010 = Display receive activity (stretchable)
0011 = Display collision activity (stretchable)
0100 = Display link status
0101 = Display duplex status
0110 = Reserved
0111 = Display transmit and receive activity (stretchable)
1000 = On
1001 = Off
1010 = Blink fast
1011 = Blink slow
1100 = Display link status and receive activity (always stretched)
1101 = Display link status and transmit/receive activity (always stretched)
111x = Reserved

bit 7-4 **LBCFG<3:0>:** LEDB Configuration bits
0000 = Reserved
0001 = Display transmit activity (stretchable)
0010 = Display receive activity (stretchable)
0011 = Display collision activity (stretchable)
0100 = Display link status
0101 = Display duplex status
0110 = Reserved
0111 = Display transmit and receive activity (stretchable)
1000 = On
1001 = Off
1010 = Blink fast
1011 = Blink slow
1100 = Display link status and receive activity (always stretched)
1101 = Display link status and transmit/receive activity (always stretched)
111x = Reserved

bit 3-2 **LFRQ<1:0>:** LED Pulse Stretch Time Configuration bits (see Table 19-1)
11 = Reserved
10 = Stretch LED events by TLSTRCH
01 = Stretch LED events by TMSTRCH
00 = Stretch LED events by TNSTRCH

bit 1 **STRCH:** LED Pulse Stretching Enable bit
1 = Stretchable LED events will cause lengthened LED pulses based on LFRQ<1:0> configuration
0 = Stretchable LED events will only be displayed while they are occurring

bit 0 **Reserved:** Write as '0'

REGISTER 19-15: EIR: ETHERNET INTERRUPT REQUEST (FLAG) REGISTER

U-0	R-0	R/C-0	R-0	R/C-0	U-0	R/C-0	R/C-0
—	PKTIF	DMAIF	LINKIF	TXIF	—	TXERIF	RXERIF
bit 7							bit 0

Legend:			
R = Readable bit	C = Clearable 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 **PKTIF:** Receive Packet Pending Interrupt Flag bit
- 1 = Receive buffer contains one or more unprocessed packets; cleared only when EPKTCNT is decremented to 0 by setting PKTDEC (ECON2<6>)
- 0 = Receive buffer is empty

bit 5 **DMAIF:** DMA Interrupt Flag bit
- 1 = DMA copy or checksum calculation has completed
- 0 = No DMA interrupt is pending

bit 4 **LINKIF:** Link Change Interrupt Flag bit
- 1 = PHY reports that the link status has changed; read PHIR register to clear
- 0 = Link status has not changed

bit 3 **TXIF:** Transmit Interrupt Flag bit
- 1 = Transmit request has ended
- 0 = No transmit interrupt is pending

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

bit 1 **TXERIF:** Transmit Error Interrupt Flag bit
- 1 = A transmit error has occurred
- 0 = No transmit error has occurred

bit 0 **RXERIF:** Receive Error Interrupt Flag bit
- 1 = A packet was aborted because there is insufficient buffer space, or a buffer overrun has occurred
- 0 = No receive error interrupt is pending

20.4 I²C Mode

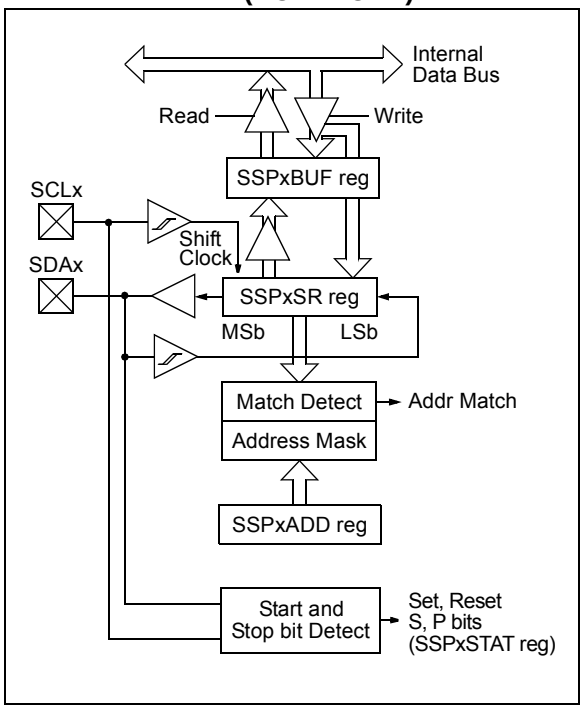
The MSSP module in I²C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer:

- Serial clock (SCLx) – RC3/SCK1/SCL1 (or RD6/SCK2/SCL2 for 100-pin devices)
- Serial data (SDAx) – RC4/SDI1/SDA1 (or RD5/SDI2/SDA2 for 100-pin devices)

The user must configure these pins as inputs by setting the TRISC<4:3> or TRISD<5:4> bits.

FIGURE 20-7: MSSP BLOCK DIAGRAM (I²C™ MODE)



20.4.1 REGISTERS

The MSSP module has six registers for I²C operation. These are:

- MSSPx Control Register 1 (SSPxCON1)
- MSSPx Control Register 2 (SSPxCON2)
- MSSPx Status Register (SSPxSTAT)
- MSSPx Receive Buffer/Transmit Register (SSPxBUF)
- MSSPx Shift Register (SSPxSR) – Not directly accessible
- MSSPx Address Register (SSPxADD)

SSPxCON1, SSPxCON2 and SSPxSTAT are the control and status registers in I²C mode operation. The SSPxCON1 and SSPxCON2 registers are readable and writable. The lower 6 bits of the SSPxSTAT are read-only. The upper two bits of the SSPxSTAT are read/write.

Many of the bits in SSPxCON2 assume different functions, depending on whether the module is operating in Master or Slave mode. SSPxCON2<5:1> also assume different names in Slave mode. The different aspects of SSPxCON2 are shown in Register 20-5 (for Master mode) and Register 20-6 (Slave mode).

SSPxSR is the shift register used for shifting data in or out. SSPxBUF is the buffer register to which data bytes are written to or read from.

The SSPxADD register holds the slave device address when the MSSP is configured in I²C Slave mode. When the MSSP is configured in Master mode, the lower seven bits of SSPxADD act as the Baud Rate Generator reload value.

In receive operations, SSPxSR and SSPxBUF together, create a double-buffered receiver. When SSPxSR receives a complete byte, it is transferred to SSPxBUF and the SSPxIF interrupt is set.

During transmission, the SSPxBUF is not double-buffered. A write to SSPxBUF will write to both SSPxBUF and SSPxSR.

20.4.17.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

a) A low level is sampled on SDAx when SCLx goes from low level to high level.
b) SCLx goes low before SDAx is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user deasserts SDAx and the pin is allowed to float high, the BRG is loaded with SSPxADD<6:0> and counts down to 0. The SCLx pin is then deasserted and when sampled high, the SDAx pin is sampled.

If SDAx is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', see Figure 20-31). If SDAx is sampled high, the BRG is reloaded and begins counting. If SDAx goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDAx at exactly the same time.

If SCLx goes from high-to-low before the BRG times out and SDAx has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition (see Figure 20-32).

If, at the end of the BRG time-out, both SCLx and SDAx are still high, the SDAx pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCLx pin, the SCLx pin is driven low and the Repeated Start condition is complete.

FIGURE 20-31: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

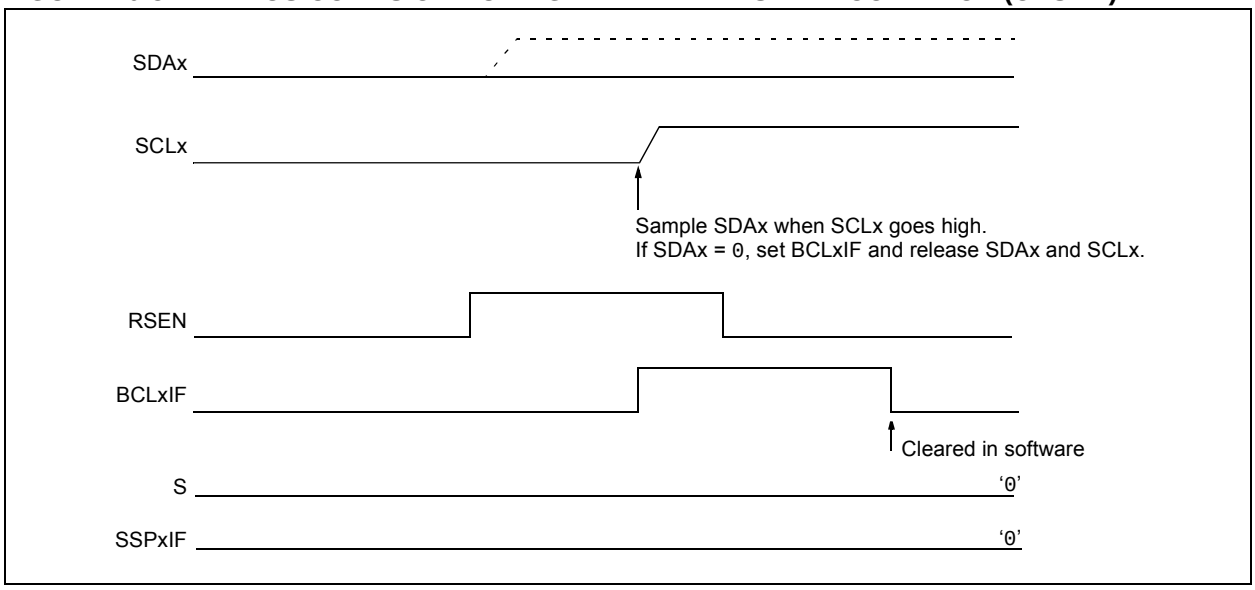
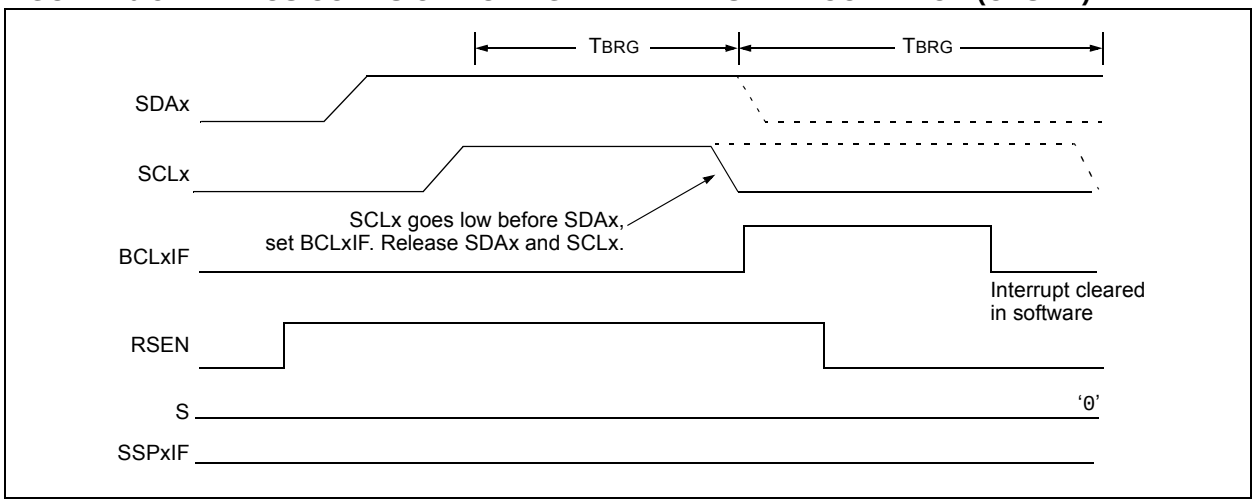


FIGURE 20-32: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



21.1 Baud Rate Generator (BRG)

The BRG is a dedicated, 8-bit or 16-bit generator that supports both the Asynchronous and Synchronous modes of the EUSARTx. By default, the BRG operates in 8-bit mode; setting the BRG16 bit (BAUDCONx<3>) selects 16-bit mode.

The SPBRGHx:SPBRGx register pair controls the period of a free-running timer. In Asynchronous mode, bits BRGH (TXSTAx<2>) and BRG16 (BAUDCONx<3>) also control the baud rate. In Synchronous mode, BRGH is ignored. Table 21-1 shows the formula for computation of the baud rate for different EUSARTx modes which only apply in Master mode (internally generated clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRGHx:SPBRGx registers can be calculated using the formulas in Table 21-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 21-1. Typical baud rates and error values for the various Asynchronous modes are shown in Table 21-2. It may be advantageous to use the high baud rate (BRGH = 1), or the 16-bit BRG to reduce the baud rate error, or achieve a slow baud rate for a fast oscillator frequency.

Writing a new value to the SPBRGHx:SPBRGx registers causes the BRG timer to be reset (or cleared). This ensures that the BRG does not wait for a timer overflow before outputting the new baud rate.

21.1.1 OPERATION IN POWER-MANAGED MODES

The device clock is used to generate the desired baud rate. When one of the power-managed modes is entered, the new clock source may be operating at a different frequency. This may require an adjustment to the value in the SPBRGx register pair.

21.1.2 SAMPLING

The data on the RXx pin (either RC7/RX1/DT1 or RG2/RX2/DT2) is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RXx pin.

TABLE 21-1: BAUD RATE FORMULAS

Configuration Bits			BRG/EUSARTx Mode	Baud Rate Formula
SYNC	BRG16	BRGH		
0	0	0	8-bit/Asynchronous	FOSC/[64 (n + 1)]
0	0	1	8-bit/Asynchronous	FOSC/[16 (n + 1)]
0	1	0	16-bit/Asynchronous	
0	1	1	16-bit/Asynchronous	FOSC/[4 (n + 1)]
1	0	x	8-bit/Synchronous	
1	1	x	16-bit/Synchronous	

Legend: x = Don't care, n = value of SPBRGHx:SPBRGx register pair

TABLE 21-3: BAUD RATES FOR ASYNCHRONOUS MODES

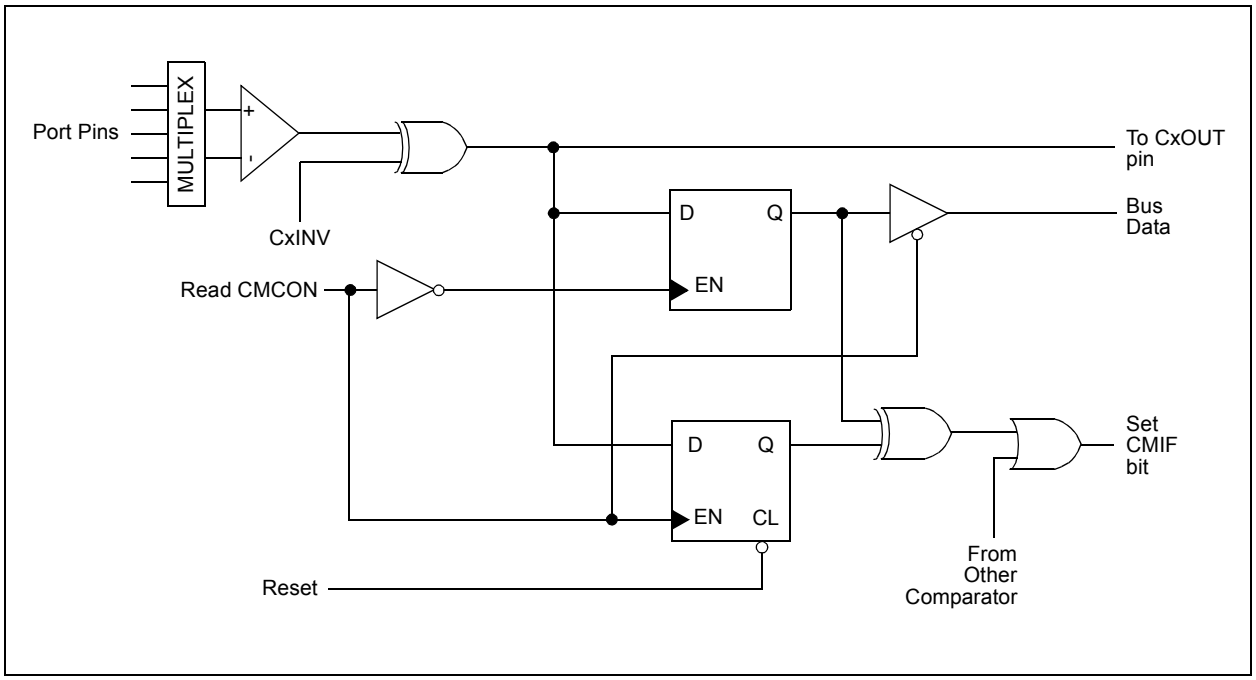
BAUD RATE (K)	SYNC = 0, BRG16 = 0, BRGH = 0											
	Fosc = 41.667 MHz			Fosc = 31.25 MHz			Fosc = 25.000 MHz			Fosc = 20.833 MHz		
	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	1.271	5.96	255
2.4	2.543	5.96	255	2.405	0.22	202	2.396	-0.15	162	2.393	-0.27	135
9.6	9.574	-0.27	67	9.574	-0.27	50	9.527	-0.76	40	9.574	-0.27	33
19.2	19.148	-0.27	33	19.531	1.73	24	19.531	1.73	19	19.147	-0.27	16
57.6	59.186	2.75	10	61.035	5.96	7	55.804	-3.12	6	54.253	-5.81	5
115.2	108.508	-5.81	5	122.070	5.96	3	130.208	13.03	2	108.505	-5.81	2

BAUD RATE (K)	SYNC = 0, BRG16 = 0, BRGH = 0								
	Fosc = 13.889 MHz			Fosc = 6.250 MHz			Fosc = 4.167 MHz		
	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)
0.3	—	—	—	—	—	—	0.300	0.01	216
1.2	1.198	-0.08	180	1.206	0.47	80	1.206	0.48	53
2.4	2.411	0.47	89	2.382	-0.76	40	2.411	0.48	26
9.6	9.435	-1.71	22	9.766	1.73	9	9.301	-3.11	6
19.2	19.279	2.75	10	19.531	1.73	4	21.703	13.04	2
57.6	54.254	-5.81	3	48.828	-15.23	1	65.109	13.04	0
115.2	108.508	-5.81	1	97.656	-15.23	0	65.109	-43.48	0

BAUD RATE (K)	SYNC = 0, BRG16 = 0, BRGH = 1											
	Fosc = 41.667 MHz			Fosc = 31.25 MHz			Fosc = 25.000 MHz			Fosc = 20.833 MHz		
	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	—	—	—	—	—	—
2.4	—	—	—	—	—	—	—	—	—	—	—	—
9.6	10.172	5.96	255	9.621	0.22	202	9.586	-0.15	162	9.573	-0.27	135
19.2	19.148	-0.27	135	19.148	-0.27	101	19.290	0.47	80	19.147	-0.27	67
57.6	57.871	0.47	44	57.445	-0.27	33	57.870	0.47	26	56.611	-1.72	22
115.2	113.226	-1.71	22	114.890	-0.27	16	111.607	-3.12	13	118.369	2.75	10

BAUD RATE (K)	SYNC = 0, BRG16 = 0, BRGH = 1								
	Fosc = 13.889 MHz			Fosc = 6.250 MHz			Fosc = 4.167 MHz		
	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)	Actual Rate (K)	% Error	SPBRG Value (decimal)
0.3	—	—	—	—	—	—	—	—	—
1.2	—	—	—	—	—	—	1.200	0.01	216
2.4	—	—	—	2.396	-0.15	162	2.389	-0.44	108
9.6	9.645	0.47	89	9.527	-0.76	40	9.645	0.48	26
19.2	19.290	0.47	44	19.531	1.73	19	18.603	-3.11	13
57.6	57.871	0.47	14	55.804	-3.12	6	52.088	-9.57	4
115.2	108.508	-5.81	7	130.208.	13.03	2	130.219	13.04	1

FIGURE 23-3: COMPARATOR OUTPUT BLOCK DIAGRAM



23.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that occurred. The CMIF bit (PIR2<6>) is the Comparator Interrupt Flag. The CMIF bit must be reset by clearing it. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

Both the CMIE bit (PIE2<6>) and the PEIE bit (INTCON<6>) must be set to enable the interrupt. In addition, the GIE bit (INTCON<7>) must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

Note:	If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR2 register) interrupt flag may not get set.

The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

a) Any read or write of CMCON will end the mismatch condition.

b) Clear flag bit, CMIF.

A mismatch condition will continue to set flag bit, CMIF. Reading CMCON will end the mismatch condition and allow flag bit, CMIF, to be cleared.

23.7 Comparator Operation During Sleep

When a comparator is active and the device is placed in Sleep mode, the comparator remains active and the interrupt is functional, if enabled. This interrupt will wake-up the device from Sleep mode, when enabled. Each operational comparator will consume additional current, as shown in the comparator specifications. To minimize power consumption while in Sleep mode, turn off the comparators (CM<2:0> = 111) before entering Sleep. If the device wakes up from Sleep, the contents of the CMCON register are not affected.

23.8 Effects of a Reset

A device Reset forces the CMCON register to its Reset state, causing the comparator modules to be turned off (CM<2:0> = 111). However, the input pins (RF3 through RF6) are configured as analog inputs by default on device Reset. The I/O configuration for these pins is determined by the setting of the PCFG<3:0> bits (ADCON1<3:0>). Therefore, device current is minimized when analog inputs are present at Reset time.

ANDWF **AND W with f**

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

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

Operation: (W) .AND. (f) → dest

Status Affected: N, Z

Encoding:

0001	01da	ffff	ffff

Description: The contents of W are ANDed with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: ANDWF REG, 0, 0

Before Instruction
W = 17h
REG = C2h
After Instruction
W = 02h
REG = C2h

BC **Branch if Carry**

Syntax: BC n

Operands: $-128 \le n \le 127$

Operation: if Carry bit is '1',
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0010	nnnn	nnnn

Description: If the Carry bit is '1', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE BC 5

Before Instruction
PC = address (HERE)
After Instruction
If Carry = 1;
PC = address (HERE + 12)
If Carry = 0;
PC = address (HERE + 2)

SLEEP **Enter Sleep Mode**

Syntax: 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. The Watchdog Timer and its postscaler 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: 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 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' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example 1: SUBFWB REG, 1, 0
Before Instruction
REG = 3
W = 2
C = 1
After Instruction
REG = FF
W = 2
C = 0
Z = 0
N = 1 ; 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

28.2 DC Characteristics: Power-Down and Supply Current PIC18F97J60 Family (Industrial) (Continued)

PIC18F97J60 Family (Industrial)		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	Supply Current (IDD)(2)						
	All devices	0.5	1.1	mA	-40°C	VDD = 2.0V, VDDCORE = 2.0V(4)	FOSC = 1 MHz (**PRI_IDLE** mode, EC oscillator)
		0.5	1.1	mA	+25°C		
		0.6	1.2	mA	+85°C		
	All devices	0.9	1.4	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	
		0.9	1.4	mA	+25°C		
		1.0	1.5	mA	+85°C		
	All devices	1.9	2.6	mA	-40°C	VDD = 3.3V(5)	
		1.8	2.6	mA	+25°C		
		1.9	2.6	mA	+85°C		
	All devices	5.9	9.5	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	FOSC = 25 MHz (**PRI_IDLE** mode, EC oscillator)
		5.6	9.5	mA	+25°C		
		5.9	9.5	mA	+85°C		
	All devices	7.5	13.2	mA	-40°C	VDD = 3.3V(5)	
		7.2	13.2	mA	+25°C		
		7.5	13.2	mA	+85°C		
	All devices	8.6	14.0	mA	-40°C	VDD = 2.5V, VDDCORE = 2.5V(4)	FOSC = 41.6667 MHz (**PRI_IDLE** mode, EC oscillator)
		8.0	14.0	mA	+25°C		
		8.6	14.0	mA	+85°C		
	All devices	9.8	16.0	mA	-40°C	VDD = 3.3V(5)	
		9.4	16.0	mA	+25°C		
		9.8	16.0	mA	+85°C		

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 high-impedance state and tied to VDD or VSS, and all features that add delta current disabled (such as WDT, Timer1 oscillator, 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.

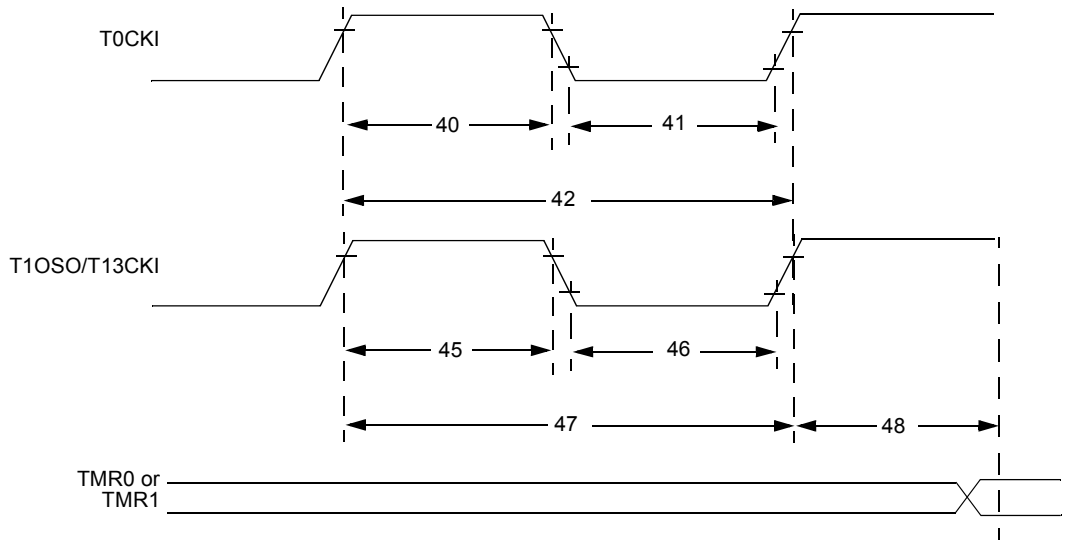
3: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

4: Voltage regulator disabled (ENVREG = 0, tied to VSS).

5: Voltage regulator enabled (ENVREG = 1, tied to VDD).

6: For ΔIETH, the specified current includes current sunk through TPOUT+ and TPOUT-. LEDA and LEDB are disabled for all testing.

FIGURE 28-9: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



Note: Refer to Figure 28-3 for load conditions.

TABLE 28-13: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Param No.	Symbol	Characteristic			Min	Max	Units	Conditions
40	TT0H	T0CKI High Pulse Width		No prescaler	0.5 TCY + 20	—	ns	
				With prescaler	10	—	ns	
41	TT0L	T0CKI Low Pulse Width		No prescaler	0.5 TCY + 20	—	ns	
				With prescaler	10	—	ns	
42	TT0P	T0CKI Period		No prescaler	TCY + 10	—	ns	
				With prescaler	Greater of: 20 ns or (TCY + 40)/N	—	ns	N = prescale value (1, 2, 4,..., 256)
45	TT1H	T13CKI High Time	Synchronous, no prescaler		0.5 TCY + 20	—	ns	
			Synchronous, with prescaler		10	—	ns	
			Asynchronous		30	—	ns	
46	TT1L	T13CKI Low Time	Synchronous, no prescaler		0.5 TCY + 5	—	ns	
			Synchronous, with prescaler		10	—	ns	
			Asynchronous		30	—	ns	
47	TT1P	T13CKI Input Period	Synchronous		Greater of: 20 ns or (TCY + 40)/N	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous		60	—	ns	
	FT1	T13CKI Oscillator Input Frequency Range			DC	50	kHz	
48	TCKE2TMRI	Delay from External T13CKI Clock Edge to Timer Increment			2 TOSC	7 TOSC	—	

TABLE 28-21: I²C™ BUS DATA REQUIREMENTS (SLAVE MODE)

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	4.0	—	μs	PIC18F97J60 family must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	PIC18F97J60 family must operate at a minimum of 10 MHz
			MSSP module	1.5 TCY	—		
101	TLOW	Clock Low Time	100 kHz mode	4.7	—	μs	PIC18F97J60 family must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	PIC18F97J60 family must operate at a minimum of 10 MHz
			MSSP module	1.5 TCY	—		
102	TR	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1 CB	300	ns	CB is specified to be from 10 to 400 pF
103	TF	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	
			400 kHz mode	20 + 0.1 CB	300	ns	CB is specified to be from 10 to 400 pF
90	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	μs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	μs	
91	THD:STA	Start Condition Hold Time	100 kHz mode	4.0	—	μs	After this period, the first clock pulse is generated
			400 kHz mode	0.6	—	μs	
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	**(Note 2)**
			400 kHz mode	100	—	ns	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
109	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns	**(Note 1)**
			400 kHz mode	—	—	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
D102	CB	Bus Capacitive Loading		—	400	pF	

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCLx to avoid unintended generation of Start or Stop conditions.

2: A Fast mode I²C™ bus device can be used in a Standard mode I²C bus system, but the requirement, TSU:DAT ≥ 250 ns, must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line,
TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCLx line is released.

APPENDIX A: REVISION HISTORY

Revision A (March 2006)

Original data sheet for the PIC18F97J60 family of devices.

Revision B (October 2006)

First revision. Includes preliminary electrical specifications; revised and updated material on the Ethernet module; updated material on Reset integration; and updates to the device memory map.

Revision C (June 2007)

Corrected Table 10.2: Input Voltage Levels; added content on Ethernet module's reading and writing to the buffer; added new, 100-lead PT 12x12x1 mm TQFP package to "Package Marking Information" and "Package Details" sections; updated other package details drawings; changed Product Identification System examples.

Revision D (January 2008)

Added one line to "Ethernet Features" description. Added land pattern schematics for each package.

Revision E (October 2009)

Updated to remove Preliminary status.

Revision F (April 2011)

Added Brown-out Reset (BOR) specs, added Ethernet RX Auto-Polarity circuit section, added EMI filter section, added **Section 2.0 "Guidelines for Getting Started with PIC18FJ Microcontrollers"**, changed the opcode encoding of the PUSHL instruction to 1110 1010 kkk kkkk and changed the 2 TOSC Maximum Device Frequency in Table 22-1 from 2.68 MHz to the correct value of 2.86 MHz. Updated comparator input offset voltage maximum to the correct value of 25 mV.