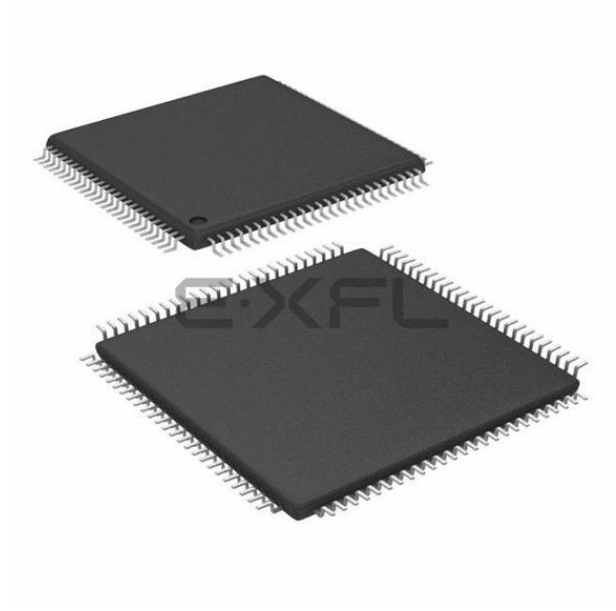


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, HLVD, LCD, POR, PWM, WDT
Number of I/O	86
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 24x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f96j94-i-pf

TABLE 3: 80-PIN ALLOCATION TABLE (PIC18F8XJ94)

I/O	80-Pin TQFP	ADC	Comparator	HLVD	CTMU	USB	LCD	MSSP	PSP	Interrupt	REFO	EMB	PPS-Lite(1)	Pull-up	Basic
RA0	30	AN0/ AN1-	—	—	—	—	SEG19	—	—	—	—	—	RP0	—	—
RA1	29	AN1	—	—	—	—	SEG18	—	—	—	—	—	RP1	—	—
RA2	28	AN2/ VREF-	—	—	—	—	SEG21	—	—	—	—	—	RP2	—	—
RA3	27	AN3/ VREF+	—	—	—	—	—	—	—	—	—	—	RP3	—	—
RA4	34	AN6	—	—	—	—	SEG14	—	—	—	—	—	RP4	—	—
RA5	33	AN4	C1INA/ C2INA/ C3INA	LVDIN	—	—	SEG15	—	—	—	—	—	RP5	—	—
RA6	50	—	—	—	—	—	—	—	—	—	—	—	RP6	—	OSC2/ CLKO
RA7	49	—	—	—	—	—	—	—	—	—	—	—	RP10	—	OSC1/ CLKI
RB0	58	—	—	—	CTED13	—	VLCAP1	—	—	INT0	—	—	RP8	—	—
RB1	57	—	—	—	—	—	VLCAP2	—	—	—	—	—	RP9	—	—
RB2	56	—	—	—	CTED1	—	SEG9	—	—	—	—	—	RP14	—	—
RB3	55	—	—	—	CTED2	—	SEG10	—	—	—	—	—	RP7	—	—
RB4	54	—	—	—	CTED3	—	SEG11	—	—	—	—	—	RP12	—	—
RB5	53	—	—	—	CTED4	—	SEG8	—	—	—	—	—	RP13	—	—
RB6	52	—	—	—	CTED5	—	—	—	—	—	—	—	—	—	PGC
RB7	47	—	—	—	CTED6	—	—	—	—	—	—	—	—	—	PGD
RC0	36	—	—	—	—	—	—	—	—	—	—	—	—	—	SOSCO/ SCKI/ PWRCLK
RC1	35	—	—	—	—	—	—	—	—	—	—	—	—	—	SOSCI
RC2	43	AN9	—	—	CTED7	—	SEG13	—	—	—	—	—	RP11	—	—
RC3	44	—	—	—	CTED8	—	SEG17	SCL1	—	—	—	—	RP15	—	—
RC4	45	—	—	—	CTED9	—	SEG16	SDA1	—	—	—	—	RP17	—	—
RC5	46	—	—	—	CTED10	—	SEG12	—	—	—	—	—	RP16		
RC6	37	—	—	—	CTED11	$\overline{UOE}$	SEG27	—	—	—	—	—	RP18	—	—
RC7	38	—	—	—	CTED12	—	SEG22	—	—	—	—	—	RP19	—	—
RD0	72	—	—	—	—	—	SEG0	—	PSP0	—	—	AD0	RP20	Y	—
RD1	69	—	—	—	—	—	SEG1	—	PSP1	—	—	AD1	RP21	Y	—
RD2	68	—	—	—	—	—	SEG2	—	PSP2	—	—	AD2	RP22	Y	—
RD3	67						SEG3		PSP3			AD3	RP23	Y	—
RD4	66	—	—	—	—	—	SEG4	—	PSP4	—	—	AD4	RP24	Y	—
RD5	65	—	—	—	—	—	SEG5	SDA2	PSP5	—	—	AD5	RP25	Y	—
RD6	64	—	—	—	—	—	SEG6	SCL2	PSP6	—	—	AD6	RP26	Y	—
RD7	63	—	—	—	—	—	SEG7	—	PSP7	—	REFO2	AD7	RP27	Y	—
RE0	4	—	—	—	—	—	LCDBIAS1	—	$\overline{RD}$	—	—	AD8	RP28	Y	—
RE1	3	—	—	—	—	—	LCDBIAS2	—	$\overline{WR}$	—	—	AD9	RP29	Y	—
RE2	78	—	—	—	—	—	LCDBIAS3	—	$\overline{CS}$	—	—	AD10	RP30	Y	—
RE3	77	—	—	—	—	—	COM0	—	—	—	REFO1	AD11	RP33	Y	—
RE4	76	—	—	—	—	—	COM1	—	—	—	—	AD12	RP32	Y	—
RE5	75	—	—	—	—	—	COM2	—	—	—	—	AD13	RP37	Y	—
RE6	74	—	—	—	—	—	COM3	—	—	—	—	AD14	RP34	Y	—
RE7	73	—	—	—	—	—	LCDBIAS0	—	—	—	—	AD15	RP31	Y	—
RF2	18	AN7	C2INB		CTMUI		SEG20	—	—	—	—	—	RP36	Y	—

ture and low VDD both have a limiting effect on the loop gain, such that if the circuit functions at these extremes, the designer can be more assured of proper operation at other temperatures and supply voltage combinations. The output sine wave should not be clipped in the highest gain environment (highest VDD and lowest temperature) and the sine output amplitude should be large enough in the lowest gain environment (lowest VDD and highest temperature) to cover the logic input requirements of the clock, as listed in the device data sheet. OSC1 may have specified VIL and VIH levels (refer to the specific product data sheet for more information).

A method for improving start-up is to use a value of C2 greater than C1. This causes a greater phase shift across the crystal at power-up, which speeds oscillator start-up. Besides loading the crystal for proper frequency response, these capacitors can have the effect of lowering loop gain if their value is increased. C2 can be selected to affect the overall gain of the circuit. A higher C2 can lower the gain if the crystal is being overdriven (also see discussion on Rs). Capacitance values that are too high can store and dump too much current through the crystal, so C1 and C2 should not become excessively large. Unfortunately, measuring the wattage through a crystal is difficult, but if you do not stray too far from the suggested values, you should not have to be concerned with this.

A series resistor, Rs, is added to the circuit if after all other external components are selected to satisfaction, and the crystal is still being overdriven. This can be determined by looking at the OSC2 pin, which is the driven pin, with an oscilloscope. Connecting the probe to the OSC1 pin will load the pin too much and negatively affect performance. Remember that a scope probe adds its own capacitance to the circuit, so this may have to be accounted for in your design (i.e., if the circuit worked best with a C2 of 22 pF and the scope probe was 10 pF, a 33 pF capacitor may actually be called for). The output signal should not be clipping or flattened. Overdriving the crystal can also lead to the circuit jumping to a higher harmonic level, or even, crystal damage.

The OSC2 signal should be a clean sine wave that easily spans the input minimum and maximum of the clock input pin. An easy way to set this is to again test the circuit at the minimum temperature and maximum VDD that the design will be expected to perform in; then, look at the output. This should be the maximum amplitude of the clock output. If there is clipping, or the sine wave is distorted near VDD and VSS, increasing load capacitors may cause too much current to flow through the crystal, or push the value too far from the manufacturer's load specification. To adjust the crystal current, add a trimmer potentiometer between the crystal inverter output pin and C2, and adjust it until the sine wave is clean. The crystal will experience the highest drive currents at the low temperature and high VDD extremes.

The trimmer potentiometer should be adjusted at these limits to prevent overdriving. A series resistor, Rs, of the closest standard value can now be inserted in place of the trimmer. If Rs is too high, perhaps more than 20 kΩ, the input will be too isolated from the output, making the clock more susceptible to noise. If you find a value this high is needed to prevent overdriving the crystal, try increasing C2 to compensate or changing the oscillator operating mode. Try to get a combination where Rs is around 10 kΩ or less, and load capacitance is not too far from the manufacturer's specification.

3.7 External Clock Input

In EC mode, the OSC1 pin is in a high-impedance state and can be driven by CMOS drivers. The OSC2 pin can be configured as either an I/O or the clock output (FOSC 4) by selecting the CLKOEN bit (CONFIG2L<5>). With CLKOEN set (Figure 3-5), the clock output is available for testing or synchronization purposes. With CLKOEN clear (Figure 3-6), the OSC2 pin becomes a general purpose I/O pin. The feedback device between OSC1 and OSC2 is turned off to save current.

FIGURE 3-5: EXTERNAL CLOCK INPUT OPERATION (CLKOEN = 1)

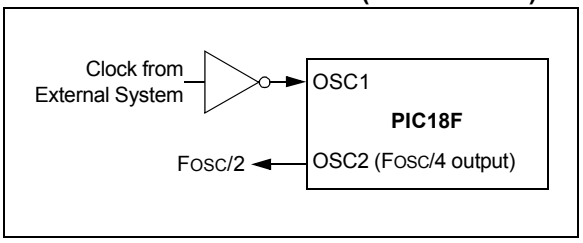


FIGURE 3-6: EXTERNAL CLOCK INPUT OPERATION (CLKOEN = 0)

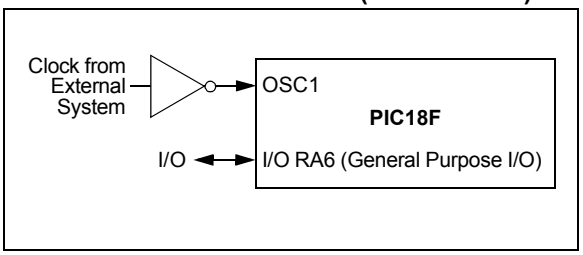


TABLE 6-2: REGISTER FILE SUMMARY (CONTINUED)

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F03h	SSP2BUF	MSSP2 Receive Buffer/Transmit Register							
F02h	SSP2ADD	MSSP2 Address Register in I^2C Slave Mode. MSSP1 Baud Rate Reload Register in I^2C Master Mode.							
F01h	ANCFG	—	—	—	—	—	VBG6EN	VBG2EN	VBGEN
F00h	DMACON2	DLYCYC3	DLYCYC2	DLYCYC1	DLYCYC0	INTLVL3	INTLVL2	INTLVL1	INTLVL0
EFFh	RCSTA4	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
EFEh	TXSTA4	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
EFDh	BAUDCON4	ABDOVF	RCIDL	RXDTP	TXCKP	BRG16	IREN	WUE	ABDEN
EFCh	SPBRGH4	EUSART4 Baud Rate Generator High Byte							
EFBh	SPBRG4	EUSART4 Baud Rate Generator							
EFAh	RCREG4	EUSART4 Receive Data FIFO							
EF9h	TXREG4	EUSART4 Transmit Data FIFO							
EF8h	CTMUCON1	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	TRIGEN
EF7h	CTMUCON2	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
EF6h	CTMUCON3	EDG2EN	EDG2POL	EDG2SEL3	EDG2SEL2	EDG2SEL1	EDG2SEL0	—	—
EF5h	CTMUCON4	EDG1EN	EDG1POL	EDG1SEL3	EDG1SEL2	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
EF4h	PMD0	CCP10MD	CCP9MD	CCP8MD	CCP7MD	CCP6MD	CCP5MD	CCP4MD	ECCP3MD
EF3h	PMD1	ECCP2MD	ECCP1MD	UART4MD	UART3MD	UART2MD	UART1MD	SSP2MD	SSP1MD
EF2h	PMD2	TMR8MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD
EF1h	PMD3	DSMMD	CTMUMD	ADCMD	RTCCMD	LCDMD	PSPMD	REFO1MD	REFO2MD
EF0h	PMD4	CMP1MD	CMP2MD	CMP3MD	USBMD	IOCMD	LVDMD	—	EMBMD
EEFh	MDCON	MDEN	MDOE	MDSLR	MDOPOL	MDO	—	—	MDBIT
EEEh	MDSRC	MDSODIS	—	—	—	MDSRC3	MDSRC2	MDSRC1	MDSRC0
EEDh	MDCARH	MDCHODIS	MDCHPOL	MDCHSYNC	—	MDCH3	MDCH2	MDCH1	MDCH0
EECh	MDCARL	MDCLODIS	MDCLPOL	MDCLSYNC	—	MDCL3	MDCL2	MDCL1	MDCL0
EEBh	ODCON1	ECCP2OD	ECCP1OD	USART4OD	USART3OD	USART2OD	USART1OD	SSP2OD	SSP1OD
EEAh	ODCON2	CCP10OD	CCP9OD	CCP8OD	CCP7OD	CCP6OD	CCP5OD	CCP4OD	ECCP3OD
EE9h	TRISK	TRISK7	TRISK6	TRISK5	TRISK4	TRISK3	TRISK2	TRISK1	TRISK0
EE8h	LATK	LATK7	LATK6	LATK5	LATK4	LATK3	LATK2	LATK1	LATK0
EE7h	PORTK	RK7	RK6	RK5	RK4	RK3	RK2	RK1	RK0
EE6h	TRISL	TRISL7	TRISL6	TRISL5	TRISL4	TRISL3	TRISL2	TRISL1	TRISL0
EE5h	LATL	LATL7	LATL6	LATL5	LATL4	LATL3	LATL2	LATL1	LATL0
EE4h	PORTL	RL7	RL6	RL5	RL4	RL3	RL2	RL1	RL0
EE3h	MEMCON	EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0
EE2h	REFO1CON	ON	—	SIDL	OE	RSLP	—	DIVSWEN	ACTIVE
EE1h	REFO1CON1	—	—	—	—	ROSEL3	ROSEL2	ROSEL1	ROSEL0
EE0h	REFO1CON2	RODIV7	RODIV6	RODIV5	RODIV4	RODIV3	RODIV2	RODIV1	RODIV0
EDFh	REFO1CON3	—	RODIV14	RODIV13	RODIV12	RODIV11	RODIV10	RODIV9	RODIV8
EDEh	REFO2CON	ON	—	SIDL	OE	RSLP	—	DIVSWEN	ACTIVE
EDDh	REFO2CON1	—	—	—	—	ROSEL3	ROSEL2	ROSEL1	ROSEL0
EDCh	REFO2CON2	RODIV7	RODIV6	RODIV5	RODIV4	RODIV3	RODIV2	RODIV1	RODIV0
EDBh	REFO2CON3	—	RODIV14	RODIV13	RODIV12	RODIV11	RODIV10	RODIV9	RODIV8
EDAh	LCDPS	WFT	BIASMD	LCDA	WA	LP3	LP2	LP1	LP0
ED9h	LCDCON	LCDEN	SLPEN	WERR	CS1	CS0	LMUX2	LMUX1	LMUX0
ED8h	LCDREG	CPEN	—	BIAS2	BIAS1	BIAS0	MODE13	CLKSEL1	CLKSEL0
ED7h	LCDREF	LCDIRE	—	LCDCST2	LCDCST1	LCDCST0	VLCD3PE	VLCD2PE	VLCD1PE
ED6h	LCDRL	LRLAP1	LRLAP0	LRLBP1	LRLBP0	—	LRLAT2	LRLAT1	LRLAT0
ED5h	LCDSE7	SE63	SE62	SE61	SE60	SE59	SE58	SE57	SE56
ED4h	LCDSE6	SE55	SE54	SE53	SE52	SE51	SE50	SE49	SE48
ED3h	LCDSE5	SE47	SE46	SE45	SE44	SE43	SE42	SE41	SE40
ED2h	LCDSE4	SE39	SE38	S37	SE36	SE35	SE34	SE33	SE32
ED1h	LCDSE3	SE31	SE30	SE29	SE28	SE27	SE26	SE25	SE24
ED0h	LCDSE2	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16

Legend: — = unimplemented, read as '0'.

8.0 EXTERNAL MEMORY BUS

Note: The External Memory Bus is not implemented on 64-pin devices.

The External Memory Bus (EMB) allows the device to access external memory devices (such as Flash, EPROM or SRAM) as program or data memory. It supports both 8 and 16-Bit Data Width modes, and three address widths of up to 20 bits.

The bus is implemented with 28 pins, multiplexed across four I/O ports. Three ports (PORTD, PORTE and PORTH) are multiplexed with the address/data bus for a total of 20 available lines, while PORTJ is multiplexed with the bus control signals.

A list of the pins and their functions is provided in Table 8-1.

TABLE 8-1: PIC18F97J94 FAMILY EXTERNAL BUS – I/O PORT FUNCTIONS

Name	Port	Bit	External Memory Bus Function
RD0/AD0	PORTD	0	Address Bit 0 or Data Bit 0
RD1/AD1	PORTD	1	Address Bit 1 or Data Bit 1
RD2/AD2	PORTD	2	Address Bit 2 or Data Bit 2
RD3/AD3	PORTD	3	Address Bit 3 or Data Bit 3
RD4/AD4	PORTD	4	Address Bit 4 or Data Bit 4
RD5/AD5	PORTD	5	Address Bit 5 or Data Bit 5
RD6/AD6	PORTD	6	Address Bit 6 or Data Bit 6
RD7/AD7	PORTD	7	Address Bit 7 or Data Bit 7
RE0/AD8	PORTE	0	Address Bit 8 or Data Bit 8
RE1/AD9	PORTE	1	Address Bit 9 or Data Bit 9
RE2/AD10	PORTE	2	Address Bit 10 or Data Bit 10
RE3/AD11	PORTE	3	Address Bit 11 or Data Bit 11
RE4/AD12	PORTE	4	Address Bit 12 or Data Bit 12
RE5/AD13	PORTE	5	Address Bit 13 or Data Bit 13
RE6/AD14	PORTE	6	Address Bit 14 or Data Bit 14
RE7/AD15	PORTE	7	Address Bit 15 or Data Bit 15
RH0/A16	PORTH	0	Address Bit 16
RH1/A17	PORTH	1	Address Bit 17
RH2/A18	PORTH	2	Address Bit 18
RH3/A19	PORTH	3	Address Bit 19
RJ0/ALE	PORTJ	0	Address Latch Enable (ALE) Control Pin
RJ1/$\overline{OE}$	PORTJ	1	Output Enable ($\overline{OE}$) Control Pin
RJ2/$\overline{WRL}$	PORTJ	2	Write Low ($\overline{WRL}$) Control Pin
RJ3/$\overline{WRH}$	PORTJ	3	Write High ($\overline{WRH}$) Control Pin
RJ4/BA0	PORTJ	4	Byte Address Bit 0 (BA0)
RJ5/$\overline{CE}$	PORTJ	5	Chip Enable ($\overline{CE}$) Control Pin
RJ6/$\overline{LB}$	PORTJ	6	Lower Byte Enable ($\overline{LB}$) Control Pin
RJ7/$\overline{UB}$	PORTJ	7	Upper Byte Enable ($\overline{UB}$) Control Pin

Note: For the sake of clarity, only I/O port and external bus assignments are shown here. One or more additional multiplexed features may be available on some pins.

Example 9-3 shows the sequence to do a 16 x 16 unsigned multiplication. Equation 9-1 shows the algorithm that is used. The 32-bit result is stored in four registers (RES3:RES0).

EQUATION 9-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

RES3:RES0 = ARG1H:ARG1L · ARG2H:ARG2L
= (ARG1H · ARG2H · 2^{16}) +
(ARG1H · ARG2L · 2^{8}) +
(ARG1L · ARG2H · 2^{8}) +
(ARG1L · ARG2L)

EXAMPLE 9-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
```

Example 9-4 shows the sequence to do a 16 x 16 signed multiply. Equation 9-2 shows the algorithm used. The 32-bit result is stored in four registers (RES3:RES0). To account for the sign bits of the arguments, the MSb for each argument pair is tested and the appropriate subtractions are done.

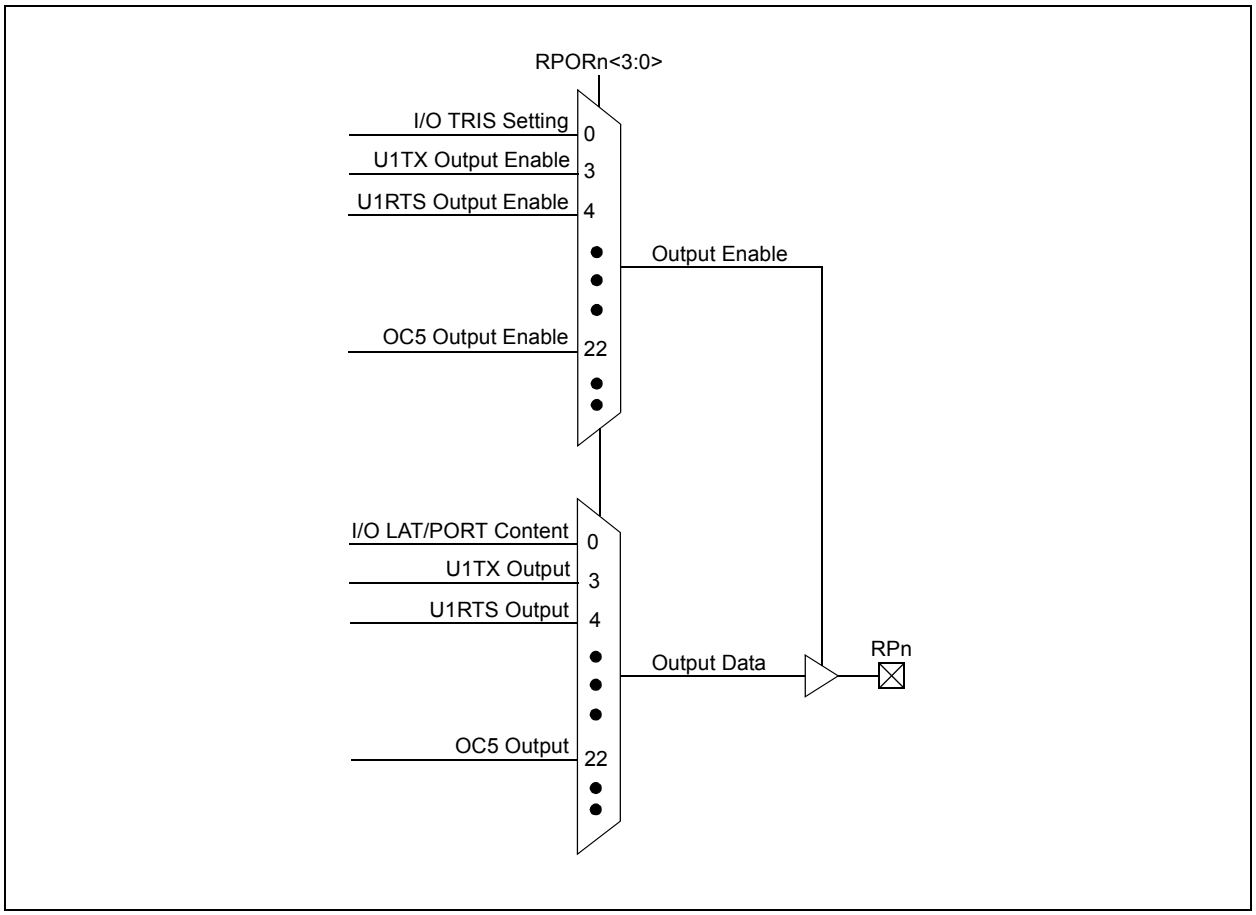
EQUATION 9-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

RES3:RES0= ARG1H:ARG1L · ARG2H:ARG2L
= (ARG1H · ARG2H · 2^{16}) +
(ARG1H · ARG2L · 2^{8}) +
(ARG1L · ARG2H · 2^{8}) +
(ARG1L · ARG2L) +
(-1 · ARG2H<7> · ARG1H:ARG1L · 2^{16}) +
(-1 · ARG1H<7> · ARG2H:ARG2L · 2^{16})

EXAMPLE 9-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
   MOVF    ARG1L, W
   MULWF   ARG2L        ; ARG1L * ARG2L ->
                        ; PRODH:PRODL
   MOVFF   PRODH, RES1 ;
   MOVFF   PRODL, RES0 ;
;
   MOVF    ARG1H, W
   MULWF   ARG2H        ; ARG1H * ARG2H ->
                        ; PRODH:PRODL
   MOVFF   PRODH, RES3 ;
   MOVFF   PRODL, RES2 ;
;
   MOVF    ARG1L, W
   MULWF   ARG2H        ; ARG1L * ARG2H ->
                        ; PRODH:PRODL
   MOVF    PRODL, W     ;
   ADDWF   RES1, F      ; Add cross
   MOVF    PRODH, W     ; products
   ADDWFC  RES2, F      ;
   CLRF    WREG         ;
   ADDWFC  RES3, F      ;
;
   MOVF    ARG1H, W     ;
   MULWF   ARG2L        ; ARG1H * ARG2L ->
                        ; PRODH:PRODL
   MOVF    PRODL, W     ;
   ADDWF   RES1, F      ; Add cross
   MOVF    PRODH, W     ; products
   ADDWFC  RES2, F      ;
   CLRF    WREG         ;
   ADDWFC  RES3, F      ;
;
   BTFSS   ARG2H, 7     ; ARG2H:ARG2L neg?
   BRA  SIGN_ARG1       ; no, check ARG1
   MOVF    ARG1L, W     ;
   SUBWF   RES2         ;
   MOVF    ARG1H, W     ;
   SUBWFB  RES3         ;
SIGN_ARG1
   BTFSS   ARG1H, 7     ; ARG1H:ARG1L neg?
   BRA     CONT_CODE    ; no, done
   MOVF    ARG2L, W     ;
   SUBWF   RES2         ;
   MOVF    ARG2H, W     ;
   SUBWFB  RES3
;
CONT_CODE
   :
```

FIGURE 11-8: MULTIPLEXING OF REMAPPABLE OUTPUT FOR RPn



REGISTER 11-5: RPORn_n: REMAPPED PERIPHERAL OUTPUT REGISTER n (FUNCTION MAPS TO PIN)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RPORn_3	RPORn_2	RPORn_1	RPORn_0	RPmR_3	RPmR_2	RPmR_1	RPmR_0
bit 7							bit 0

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

bit 7-4 **RPORn_<3:0>:** RPn peripheral output function mapping

bit 3-0 **RPmR<3:0>:** RPm peripheral output function mapping

Note 1: Register values can only be changed if IOLOCK = 0.

13.3 LCD Clock Source Selection

The LCD driver module has three possible clock sources:

- FRC/8192
- SOSC Clock/32
- LPRC/32

The first clock source is the 8 MHz Fast Internal RC (FRC) Oscillator divided by 8,192. This divider ratio is chosen to provide about 1 kHz output. The divider is not programmable. Instead, the LCD prescaler bits, LCDPS<3:0>, are used to set the LCD frame clock rate.

The second clock source is the SOSC Oscillator/32. This also outputs about 1 kHz when a 32.768 kHz crystal is used with the SOSC Oscillator. To use the SOSC Oscillator as a clock source, set the SOSCEN (T1CON<3>) bit.

The third clock source is a 31.25 kHz internal LPRC Oscillator/32 that provides approximately 1 kHz output.

The second and third clock sources may be used to continue running the LCD while the processor is in Sleep.

These clock sources are selected through the bits, CS<1:0> (LCDCON<4:3>).

13.3.1 LCD PRESCALER

A 16-bit counter is available as a prescaler for the LCD clock. The prescaler is not directly readable or writable. Its value is set by the LP<3:0> bits (LCDPS<3:0>) that determine the prescaler assignment and prescale ratio.

Selectable prescale values are from 1:1 through 1:16, in increments of one.

FIGURE 13-2: LCD CLOCK GENERATION

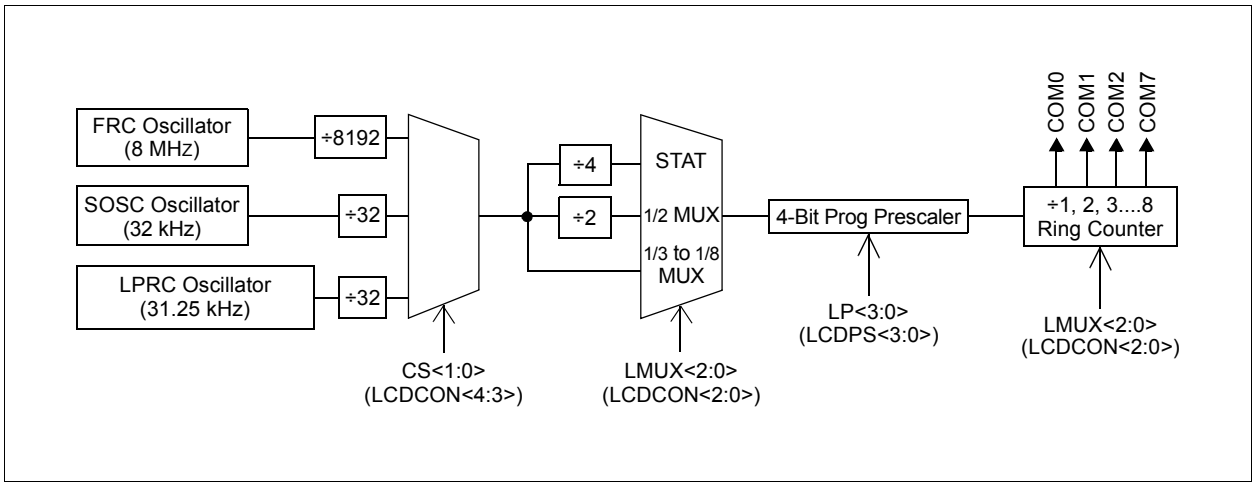
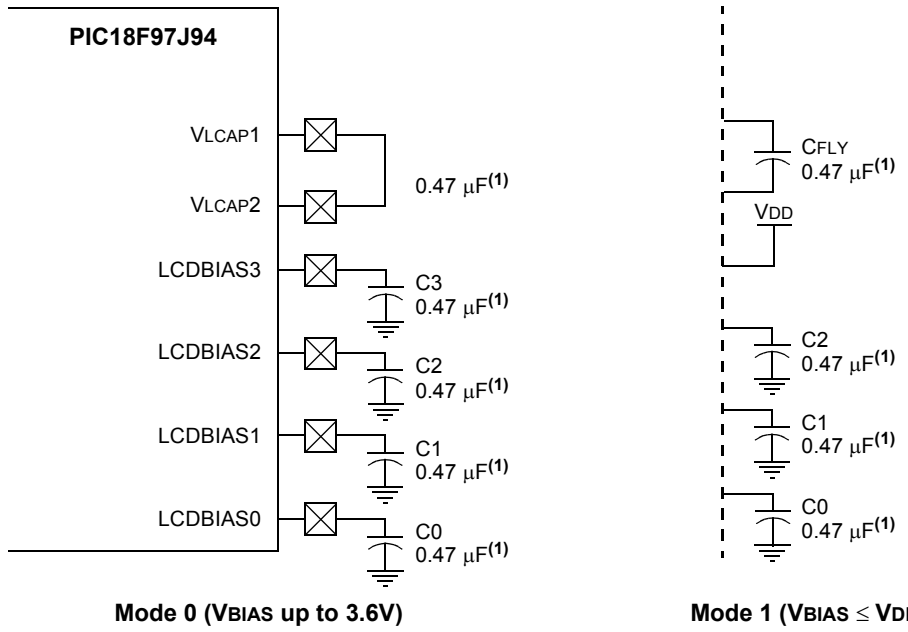


FIGURE 13-6: LCD REGULATOR CONNECTIONS FOR M0 AND M1 CONFIGURATIONS



Mode 0 (V_{BIAS} up to 3.6V) **Mode 1 ($V_{BIAS} \leq V_{DD}$)**

Note 1: These values are provided for design guidance only; they should be optimized for the application by the designer based on the actual LCD specifications.

15.6 Timer1/3/5 Interrupt

The TMRx register pair (TMRxH:TMRxL) increments from 0000h to FFFFh and overflows to 0000h. The Timerx interrupt, if enabled, is generated on overflow and is latched in the interrupt flag bit, TMRxIF. Table 15-3 gives each module's flag bit.

TABLE 15-3: TIMER1/3/5 INTERRUPT FLAG BITS

Timer Module	Flag Bit
1	PIR1<0>
3	PIR2<1>
5	PIR5<1>

This interrupt can be enabled or disabled by setting or clearing the TMRxIE bit, respectively. Table 15-4 gives each module's enable bit.

TABLE 15-4: TIMER1/3/5 INTERRUPT ENABLE BITS

Timer Module	Flag Bit
1	PIE1<0>
3	PIE2<1>
5	PIE5<1>

15.7 Resetting Timer1/3/5 Using the ECCP Special Event Trigger

If the ECCP modules are configured to use Timerx and to generate a Special Event Trigger in Compare mode (CCPxM<3:0> = 1011), this signal will reset Timerx. The trigger from ECCP2 will also start an A/D conversion if the A/D module is enabled (For more information, see **Section 18.3.4 "Special Event Trigger"**.)

The module must be configured as either a timer or synchronous counter to take advantage of this feature. When used this way, the CCPRxH:CCPRxL register pair effectively becomes a Period register for Timerx.

If Timerx is running in Asynchronous Counter mode, the Reset operation may not work.

In the event that a write to Timerx coincides with a Special Event Trigger from an ECCP module, the write will take precedence.

Note:	The Special Event Triggers from the ECCPx module will only clear the TMR3 register's content, but not set the TMR3IF interrupt flag bit (PIR1<0>).

Note:	The CCP and ECCP modules use Timers, 1 through 8, for some modes. The assignment of a particular timer to a CCP/ECCP module is determined by the Timer to CCP enable bits in the CCPTMRSx registers. For more details, see Register 18-2, Register 18-3 and Register 19-2

17.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

The key features of the Real-Time Clock and Calendar (RTCC) module are:

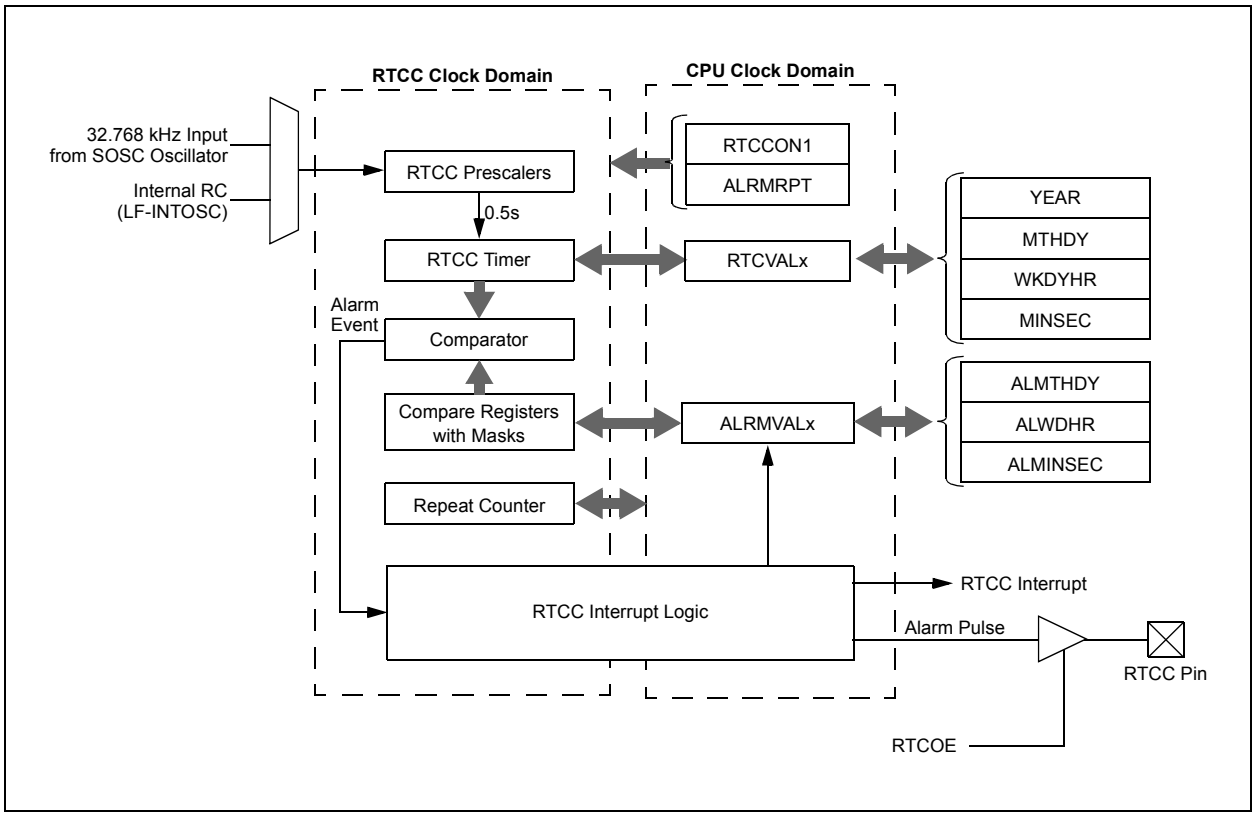
- Hardware Real-Time Clock and Calendar (RTCC)
- Provides hours, minutes and seconds using 24- hour format
- Visibility of one-half second period
- Provides calendar – weekday, date, month and year
- Alarm configurable for half a second, one second, 10 seconds, one minute, 10 minutes, one hour, one day, one week or one month
- Alarm repeat with decrementing counter
- Alarm with indefinite repeat – chime
- Year 2000 to 2099 leap year correction
- BCD format for smaller software overhead
- Optimized for long term battery operation
- Fractional second synchronization
- Multiple clock sources
  - SOSC
  - LPRC
  - 50 Hz
  - 60 Hz
- User calibration of the 32.768 kHz clock crystal frequency with periodic auto-adjust
- Calibration to within ±2.64 seconds error per month
- Calibrates up to 260 ppm of crystal error

The RTCC module is intended for applications where accurate time must be maintained for an extended period with minimum to no intervention from the CPU. The module is optimized for low-power usage in order to provide extended battery life, while keeping track of time.

The module is a 100-year clock and calendar with automatic leap year detection. The range of the clock is from 00:00:00 (midnight) on January 1, 2000 to 23:59:59 on December 31, 2099.

Hours are measured in 24-hour (military time) format. The clock provides a granularity of one second with half-second visibility to the user.

FIGURE 17-1: RTCC BLOCK DIAGRAM



17.1.4 RTCEN BIT WRITE

RTCWREN (RTCCON1<5>) must be set before a write to RTCEN can take place. Any write to the RTCEN bit, while RTCWREN = 0, will be ignored.

Like the RTCEN bit, the RTCVALH and RTCVALL registers can only be written to when RTCWREN = 1. A write to these registers, while RTCWREN = 0, will be ignored.

17.2 Operation

17.2.1 REGISTER INTERFACE

The register interface for the RTCC and alarm values is implemented using the Binary Coded Decimal (BCD) format. This simplifies the firmware when using the module, as each of the digits is contained within its own 4-bit value (see Figure 17-2 and Figure 17-3).

FIGURE 17-2: TIMER DIGIT FORMAT

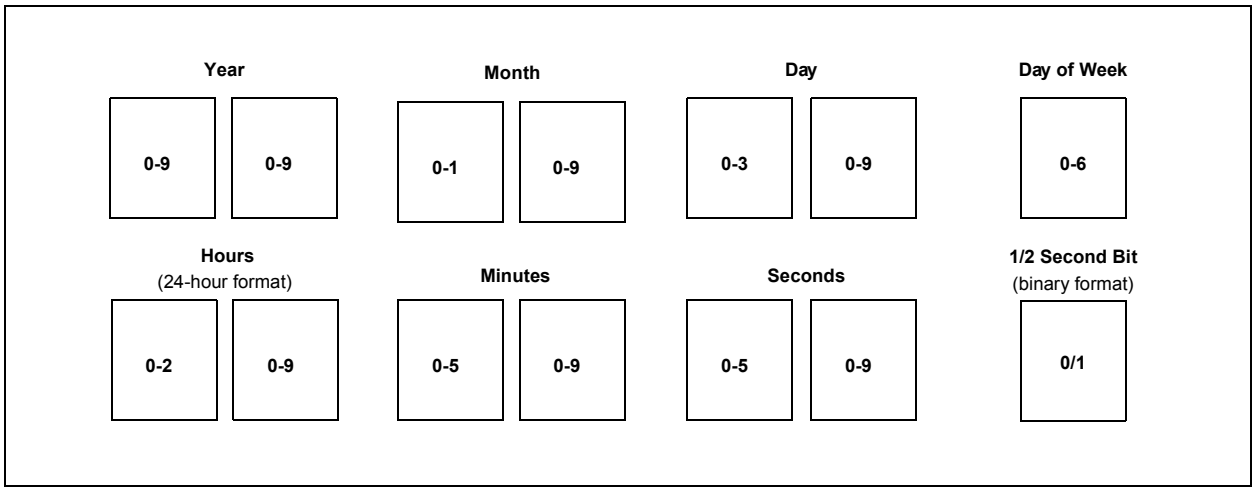
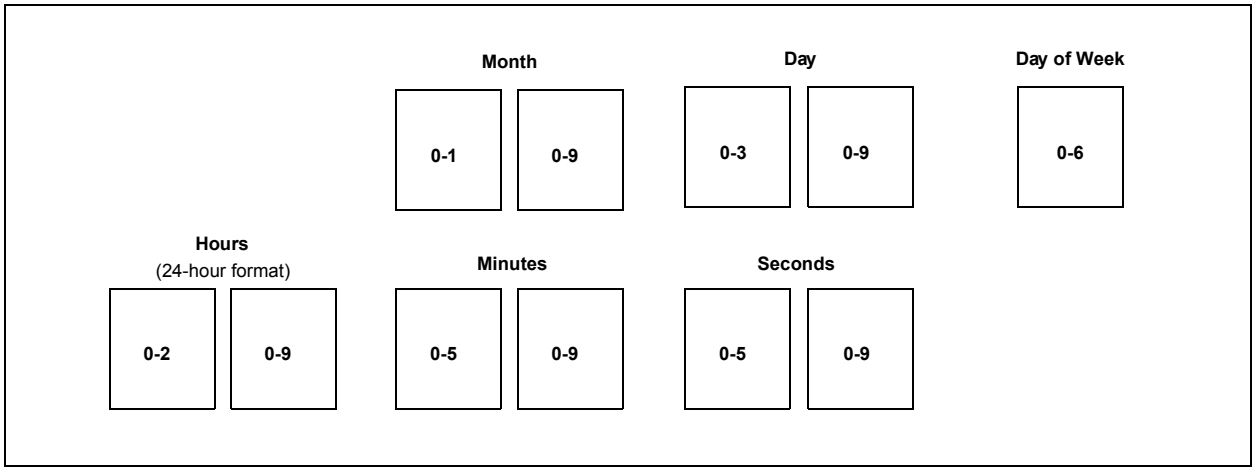


FIGURE 17-3: ALARM DIGIT FORMAT



20.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

20.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D Converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit™ (I^2C)
  - Full Master mode
  - Slave mode (with general address call)

The I^2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode with 5-bit and 7-bit address masking (with address masking for both 10-bit and 7-bit addressing)

All members of the PIC18FXXJ94 have two MSSP modules, designated as MSSP1 and MSSP2. Each module operates independently of the other.

Note:	Throughout this section, generic references to an MSSP module in any of its operating modes may be interpreted as being equally applicable to MSSP1 or MSSP2. Register names and module I/O signals use the generic designator 'x' to indicate the use of a numeral to distinguish a particular module when required. Control bit names are not individuated.

20.2 Control Registers

Each MSSP module has four associated control registers. These include a STATUS register (SSPxSTAT) and three control registers (SSPxCON1, SSPxCON2, and SSPxCON3). The use of these registers and their individual Configuration bits differ significantly depending on whether the MSSP module is operated in SPI or I^2C mode.

Additional details are provided under the individual sections. On all PIC18F97J94 family devices, the SPI DMA capability can only be used in conjunction with MSSP1. The SPI DMA feature is described in **Section 20.4 "SPI DMA Module"**.

Note:	In devices with more than one MSSP module, it is very important to pay close attention to SSPxCON register names. SSP1CON1 and SSP1CON2 control different operational aspects of the same module, while SSP1CON1 and SSP2CON1 control the same features for two different modules.

Note:	The SSPxBUF register cannot be used with read-modify-write instructions, such as BCF, COMF, etc. To avoid lost data in Master mode, a read of the SSPxBUF must be performed to clear the Buffer Full (BF) detect bit (SSPSTAT<0>) between each transmission.

20.3.4 ENABLING SPI I/O

To enable the serial port, the peripheral must first be mapped to I/O pins using the PPS-Lite feature. To enable the SPI peripheral, the MSSPx Enable bit, SSPEN (SSPxCON1<5>) must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPxCON registers and then set the SSPEN bit. This configures the SDIx, SDOx, SCKx and $\overline{SSx}$ pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDIx is automatically controlled by the SPI module
- SDOx must have the TRIS bit cleared for the corresponding RPn pin.
- SCKx (Master mode) must have the TRIS bit cleared for the corresponding RPn pin
- SCKx (Slave mode) must have the TRIS bit set for the corresponding RPn pin
- $\overline{SSx}$ must have the TRIS bit set for the corresponding RPn pin.

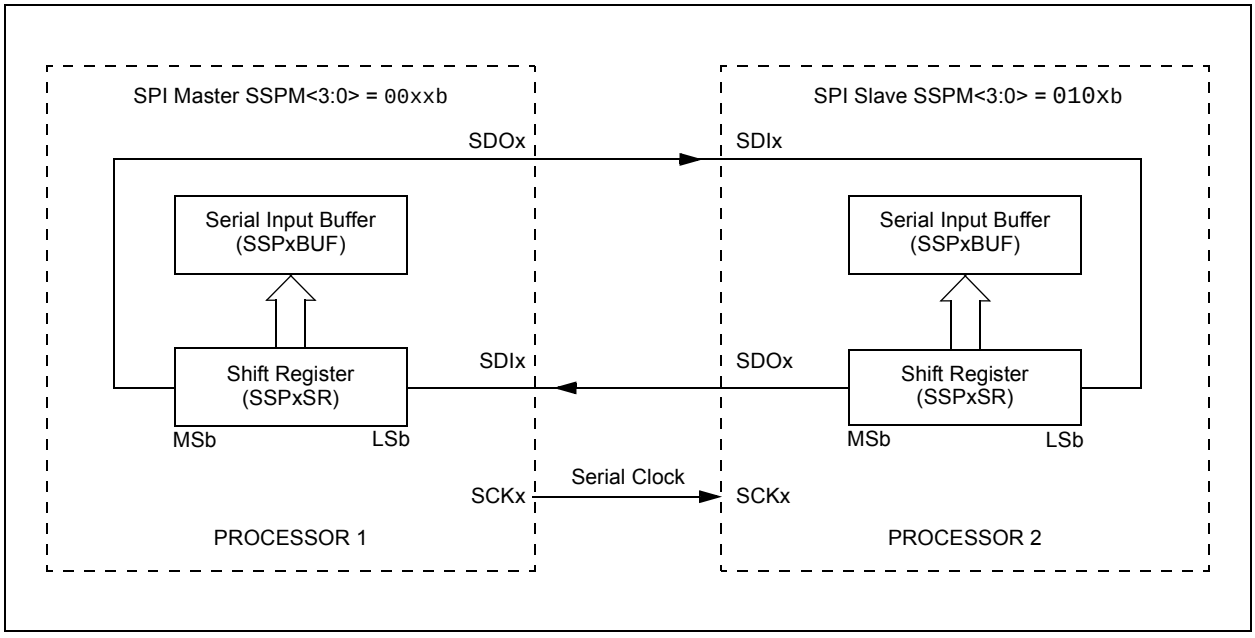
Any serial port function that is not desired may be overridden by programming the corresponding Data Direction (TRIS) register to the opposite value.

20.3.5 TYPICAL CONNECTION

Figure 20-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCKx signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

FIGURE 20-2: SPI MASTER/SLAVE CONNECTION



REGISTER 22-8: ADCON3H: A/D CONTROL REGISTER 3 HIGH

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	EXTSAM	PUMPEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
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 **ADRC:** A/D Conversion Clock Source bit
1 = RC Clock
0 = Clock derived from system clock

bit 6 **EXTSAM:** Extended Sampling Time bit
1 = A/D is still sampling after SAMP = 0
0 = A/D is finished sampling

bit 5 **PUMPEN:** Charge Pump Enable bit
1 = Charge pump for switches is enabled
0 = Charge pump for switches is disabled

bit 4-0 **SAMC<4:0>:** Auto-Sample Time Select bits
11111= 31 TAD
• • •
00001= 1 TAD
00000= 0 TAD

REGISTER 22-9: ADCON3L: A/D CONTROL REGISTER 3 LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
bit 7							bit 0

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

bit 7-0 **ADCS<7:0>:** A/D Conversion Clock Select bits ((ADCS<7:0> + 1) 2/Fosc) = TAD
11111111
• • •= Reserved
01000000
00111111= 64 · 2/Fosc = TAD
• • •
00000001= 2 · 2/Fosc = TAD
00000000= 2/Fosc = TAD

REGISTER 22-13: ADCHS0L: A/D SAMPLE SELECT REGISTER 0 LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA2	CH0NA1	CH0NA0	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0
bit 7							bit 0

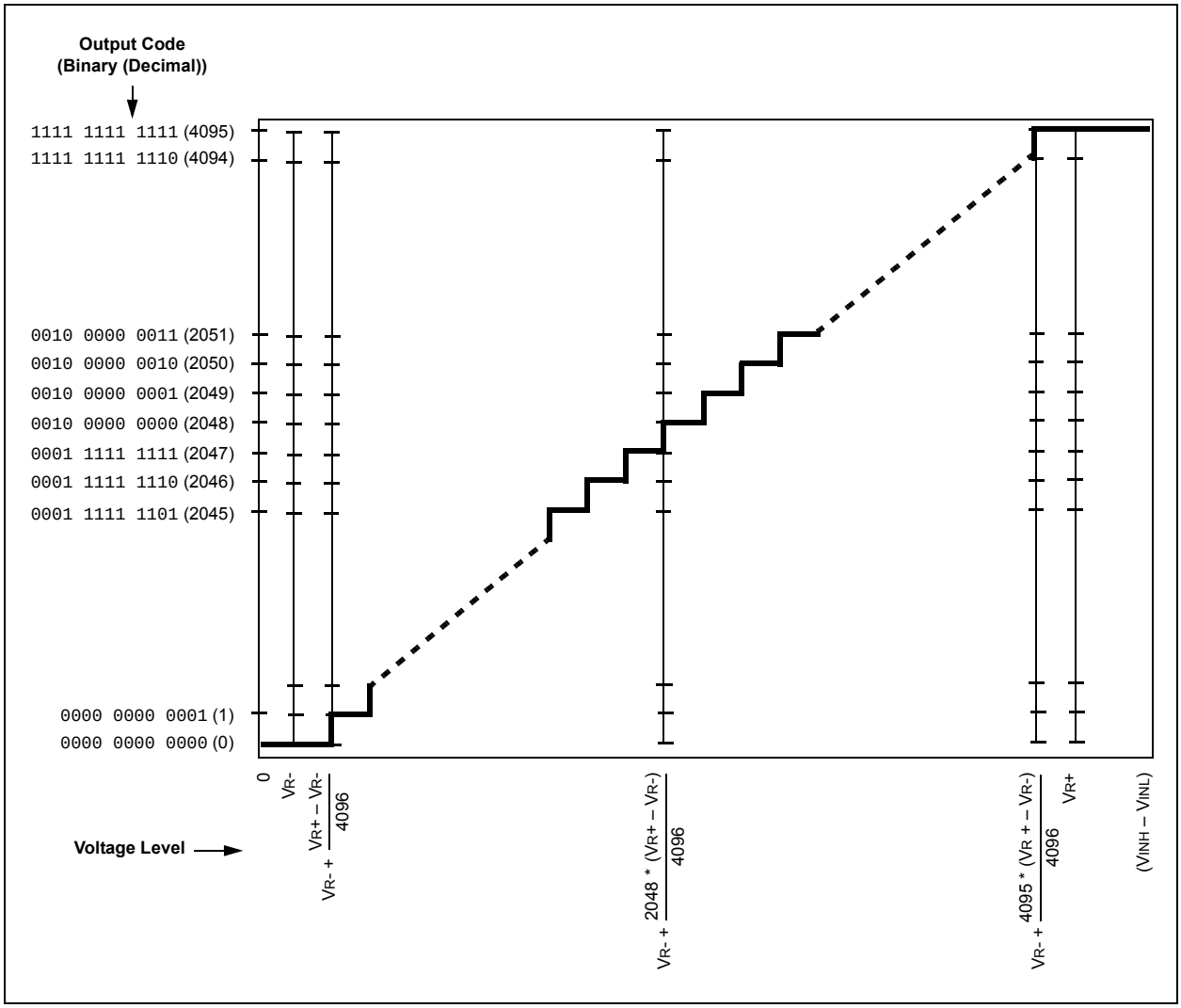
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-5 **CH0NA<2:0>:** Sample A Channel 0 Negative Input Select bits
1xx = Unimplemented
011 = Unimplemented
010 = AN1
001 = Unimplemented
000 = V_{REF-}/AV_{SS}

bit 4-0 **CH0SA<4:0>:** Sample A Channel 0 Positive Input Select bits
11111 = $V_{BAT}/2$ **(1)**
11110 = AV_{DD} **(1)**
11101 = AV_{SS} **(1)**
11100 = Band gap reference (V_{BG}) **(1)**
11011 = $V_{BG}/2$ **(1)**
11010 = $V_{BG}/6$ **(1)**
11001 = CTMU
11000 = CTMU temperature sensor input (does not require ADCTMUEN1H<0> to be set)
10111 = AN23 **(2)**
10110 = AN22 **(2)**
10101 = AN21 **(2)**
10100 = AN20 **(2)**
10011 = AN19
10010 = AN18
10001 = AN17
10000 = AN16
01111 = AN15
01110 = AN14
01101 = AN13
01100 = AN12
01011 = AN11
01010 = AN10
01001 = AN9
01000 = AN8
00111 = AN7
00110 = AN6
00101 = AN5
00100 = AN4
00011 = AN3
00010 = AN2
00001 = AN1
00000 = AN0

Note 1: These input channels do not have corresponding memory mapped result buffers.
2: These channels are implemented in 80-pin and 100-pin devices only.

FIGURE 22-21: 12-BIT A/D TRANSFER FUNCTION



For the 10-bit transfer function (when 10-bit resolution is available):

- The first code transition occurs when the input voltage is ((VR+) - (VR-))/1024 or 1.0 LSb.
- The '00 0000 0001' code is centered at VR- + (1.5 * (((VR+) - (VR-)) / 1024).
- The '10 0000 0000' code is centered at VREFL + (512.5 * (((VR+) - (VR-)) /1024).
- An input voltage less than VR- + (((VR-) - (VR-)) / 1024) converts as '00 0000 0000'.
- An input voltage greater than (VR-) + ((1023 (VR+)) - (VR-))/1024) converts as '11 1111 1111'.

REGISTER 23-2: CMSTAT: COMPARATOR STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	R-x	R-x	R-x
—	—	—	—	—	C3OUT	C2OUT	C1OUT
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 7-3 **Unimplemented:** Read as '0'

bit 2-0 **C3OUT:C1OUT:** Comparator x Status bits

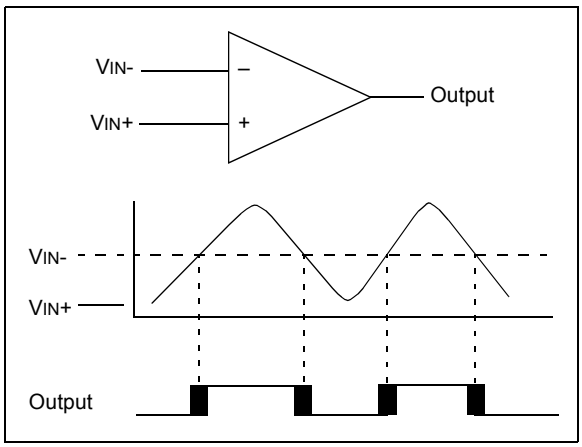
If CPOL (CMxCON<5>)= 0 (non-inverted polarity):
1 = Comparator x's VIN+ > VIN-
0 = Comparator x's VIN+ < VIN-

CPOL = 1 (inverted polarity):
1 = Comparator x's VIN+ < VIN-
0 = Comparator x's VIN+ > VIN-

23.2 Comparator Operation

A single comparator is shown in Figure 23-2, along with the relationship between the analog input levels and the digital output. When the analog input at VIN+ is less than the analog input, VIN-, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input, VIN-, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 23-2 represent the uncertainty due to input offsets and response time.

FIGURE 23-2: SINGLE COMPARATOR



23.3 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output has a valid level. The response time of the comparator differs from the settling time of the voltage reference. Therefore, both of these times must be considered when determining the total response to a comparator input change. Otherwise, the maximum delay of the comparators should be used (see **Section 30.0 "Electrical Specifications"**).

23.4 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 23-3. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up condition may occur.

A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

REGISTER 24-2: CVRCONL: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
CVREN	CVROE	CVRPSS1	CVRPSS0	—	—	—	CVRNSS
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 **CVREN:** Comparator Voltage Reference Enable bit
1 = CVREF circuit is powered on
0 = CVREF circuit is powered down

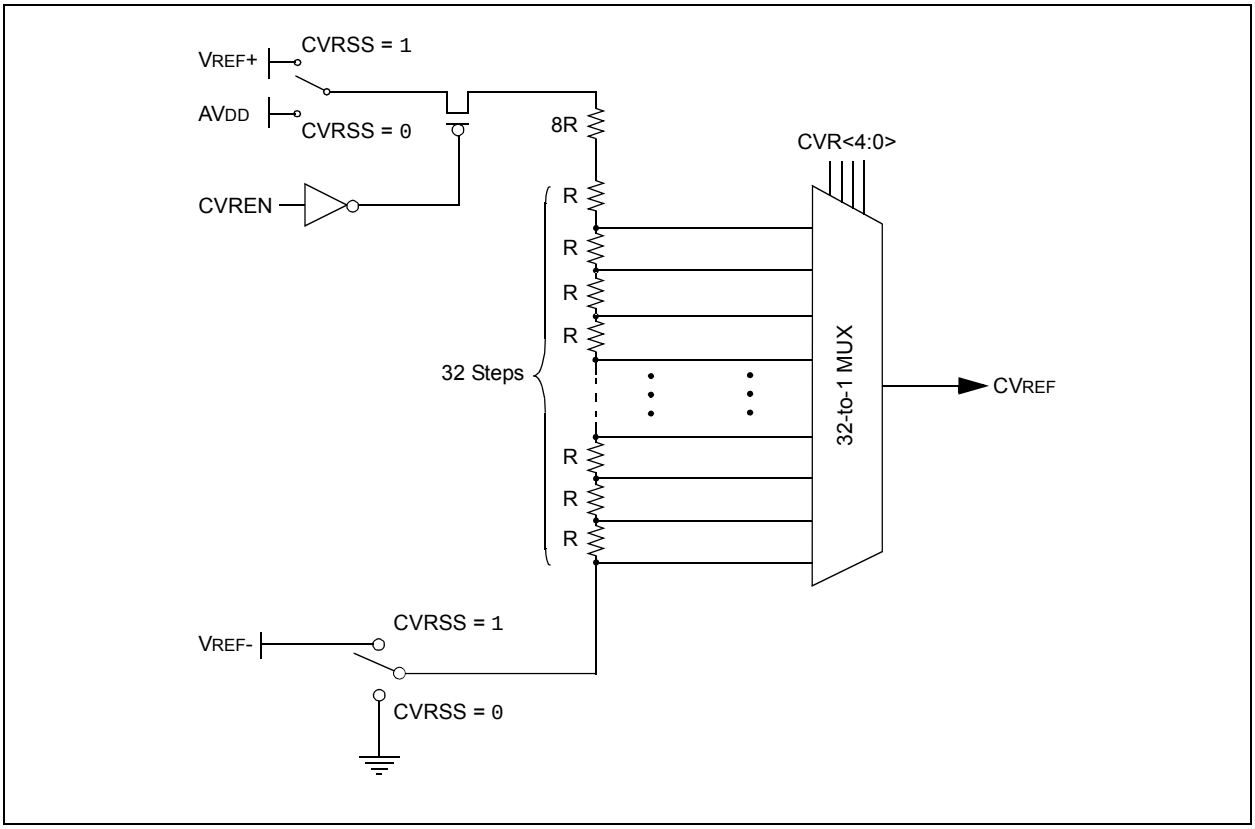
bit 6 **CVROE:** Comparator VREF Output Enable bit
1 = CVREF voltage level is output on CVREF pin
0 = CVREF voltage level is disconnected from CVREF pin

bit 5-4 **CVRPSS<1:0>:** Comparator VREF Positive Source (VPOSSRC) Selection bits
11 = Reserved, do not use. Positive source is floating
10 = VBG (Band gap)
01 = VREF+
00 = AVDD

bit 3-1 **Unimplemented:** Read as '0'

bit 0 **CVRNSS:** Comparator VREF Negative Source (VNEGSRC) Selection bit
01 = VREF-
00 = AVSS

FIGURE 24-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM



The USB Specification limits the power taken from the bus. Each device is ensured 100 mA at approximately 5V (one unit load). Additional power may be requested, up to a maximum of 500 mA.

Note that power above one unit load is a request and the host or hub is not obligated to provide the extra current. Thus, a device capable of consuming more than one unit load must be able to maintain a low-power configuration of a one unit load or less, if necessary.

The USB Specification also defines a Suspend mode. In this situation, current must be limited to 500 μA, averaged over one second. A device must enter a suspend state after 3 ms of inactivity (i.e., no SOF tokens for 3 ms). A device entering Suspend mode must drop current consumption within 10 ms after suspend. Likewise, when signaling a wake-up, the device must signal a wake-up within 10 ms of drawing current above the suspend limit.

27.9.5 ENUMERATION

When the device is initially attached to the bus, the host enters an enumeration process in an attempt to identify the device. Essentially, the host interrogates the device, gathering information, such as power consumption, data rates and sizes, protocol and other descriptive information; descriptors contain this information. A typical enumeration process would be as follows:

1. USB Reset – Reset the device. Thus, the device is not configured and does not have an address (Address 0).
2. Get Device Descriptor – The host requests a small portion of the device descriptor.
3. USB Reset – Reset the device again.
4. Set Address – The host assigns an address to the device.
5. Get Device Descriptor – The host retrieves the device descriptor, gathering info, such as manufacturer, type of device, maximum control packet size.
6. Get configuration descriptors.
7. Get any other descriptors.
8. Set a configuration.

The exact enumeration process depends on the host.

27.9.6 DESCRIPTORS

There are eight different standard descriptor types, of which, five are most important for this device.

27.9.6.1 Device Descriptor

The device descriptor provides general information, such as manufacturer, product number, serial number, the class of the device and the number of configurations. There is only one device descriptor.

27.9.6.2 Configuration Descriptor

The configuration descriptor provides information on the power requirements of the device and how many different interfaces are supported when in this configuration. There may be more than one configuration for a device (i.e., low-power and high-power configurations).

27.9.6.3 Interface Descriptor

The interface descriptor details the number of endpoints used in this interface, as well as the class of the interface. There may be more than one interface for a configuration.

27.9.6.4 Endpoint Descriptor

The endpoint descriptor identifies the transfer type (**Section 27.9.3 “Transfers”**) and direction, and some other specifics for the endpoint. There may be many endpoints in a device and endpoints may be shared in different configurations.

27.9.6.5 String Descriptor

Many of the previous descriptors reference one or more string descriptors. String descriptors provide human readable information about the layer (**Section 27.9.1 “Layered Framework”**) they describe. Often these strings show up in the host to help the user identify the device. String descriptors are generally optional to save memory and are encoded in a unicode format.

27.9.7 BUS SPEED

Each USB device must indicate its bus presence and speed to the host. This is accomplished through a 1.5 kΩ resistor, which is connected to the bus at the time of the attachment event.

Depending on the speed of the device, the resistor either pulls up the D+ or D- line to 3.3V. For a low-speed device, the pull-up resistor is connected to the D- line. For a full-speed device, the pull-up resistor is connected to the D+ line.

27.9.8 CLASS SPECIFICATIONS AND DRIVERS

USB specifications include class specifications, which operating system vendors optionally support. Examples of classes include: Audio, Mass Storage, Communications and Human Interface (HID). In most cases, a driver is required at the host side to ‘talk’ to the USB device. In custom applications, a driver may need to be developed. Fortunately, drivers are available for most common host systems for the most common classes of devices. Thus, these drivers can be reused.

REGISTER 28-9: CONFIG5H: CONFIGURATION REGISTER 5 HIGH (BYTE ADDRESS 300009h)

U-1	U-1	U-1	U-1	R/WO-1	R/WO-1	R/WO-1	R/WO-1
—	—	—	—	MSSPMSK1	MSSPMSK2	LS48MHZ	IOL1WAY
bit 7							bit 0

Legend:	P = Programmable bit	WO = Write-Once bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7-4 **Unimplemented:** Program the corresponding Flash Configuration bit to '1'

bit 3 **MSSPMSK1:** MSSP1 7-Bit Address Masking Mode Enable bit
1 = 7-Bit Address Masking mode enable
0 = 5-Bit Address Masking mode enable

bit 2 **MSSPMSK2:** MSSP2 7-Bit Address Masking Mode Enable bit
1 = 7-Bit Address Masking mode enable
0 = 5-Bit Address Masking mode enable

bit 1 **LS48MHZ:** Low-Speed USB Clock Selection bit
1 = 48 MHz system clock is expected; divide-by-8 generates low-speed USB clock
0 = 24 MHz system clock is expected; divide-by-4 generates low-speed USB clock

bit 0 **IOL1WAY:** IOLOCK Bit One-Way Set Enable bit
1 = The IOLOCK bit can only be set once (provided an unlocking sequence is executed); this prevents any possible future RP register changes
0 = The IOLOCK bit can be set and cleared as needed (provided an unlocking sequence is executed)