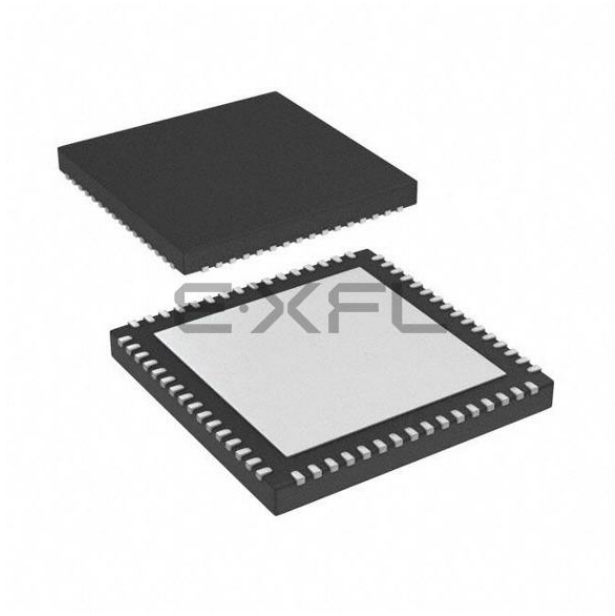


Welcome to E-XFL.COM

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	52
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 16x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f65j94-i-mr

REGISTER 3-6: REFOxCON: REFERENCE CLOCK OUTPUT CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	HC/R/W-0	HS/HC/R-0
ON	—	SIDL	OE	RSLP(1)	—	DIVSW_EN	ACTIVE
bit 7							bit 0

Legend:	HC = Hardware Clearable bit	HS = Hardware Settable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at all Resets	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **ON:** Reference Clock Output Enable bit
1 = Reference clock module is enabled
0 = Reference clock module is disabled

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

bit 5 **SIDL:** Peripheral Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 4 **OE:** Reference Clock Output Enable bit
1 = Reference clock is driven out on REFOx pin
0 = Reference clock is NOT driven out on REFOx pin

bit 3 **RSLP:** Reference Clock Output Run in Sleep bit(1)
1 = Reference Clock Output continues to run in Sleep
0 = Reference Clock Output is disabled in Sleep

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

bit 1 **DIVSW_EN:** Clock RODIV Switch Enabled Status bit
1 = Clock Divider Switching currently in progress
0 = Clock Divider Switching has completed

bit 0 **ACTIVE:** Reference Clock Output Request Status bit
1 = Reference clock request is active (user should not update the ROSEL and RODIV register fields)
0 = Reference clock request is not active (user may update the ROSEL and RODIV register fields)

Note 1: This bit has no effect when ROSEL<3:0> = 0000/0001, as the system clock and peripheral clock are always disabled in Sleep mode on PIC18 devices.

REGISTER 4-5: PMD0: PERIPHERAL MODULE DISABLE REGISTER 0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CCP10MD	CCP9MD	CCP8MD	CCP7MD	CCP6MD	CCP5MD	CCP4MD	ECCP3MD
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 **CCP10MD:** CCP10 Module Disable bit
1 = The CCP10 module is disabled. All CCP10 registers are held in Reset and are not writable.
0 = The CCP10 module is enable

bit 6 **CCP9MD:** CCP9 Module Disable bit
1 = The CCP9 module is disabled. All CCP9 registers are held in Reset and are not writable.
0 = The CCP9 module is enabled

bit 5 **CCP8MD:** CCP8 Module Disable bit
1 = The CCP8 module is disabled. All CCP8 registers are held in Reset and are not writable.
0 = The CCP8 module is enabled

bit 4 **CCP7MD:** CCP7 Module Disable bit
1 = The CCP7 module is disabled. All CCP7 registers are held in Reset and are not writable.
0 = The CCP7 module is enabled

bit 3 **CCP6MD:** CCP6 Module Disable bit
1 = The CCP6 module is disabled. All CCP6 registers are held in Reset and are not writable.
0 = The CCP6 module is enabled

bit 2 **CCP5MD:** CCP5 Module Disable bit
1 = The CCP5 module is disabled. All CCP5 registers are held in Reset and are not writable.
0 = The CCP5 module is enabled

bit 1 **CCP4MD:** CCP4 Module Disable bit
1 = The CCP4 module is disabled. All CCP4 registers are held in Reset and are not writable.
0 = The CCP4 module is enabled

bit 0 **ECCP3MD:** ECCP3 Module Disable bit
1 = The ECCP3 module is disabled. All ECCP3 registers are held in Reset and are not writable.
0 = The ECCP3 module is enabled

TABLE 5-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices			Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
SSP2CON3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
CM2CON	64-pin	80-pin	100-pin	0001 1111	0001 1111	uuuu uuuu
CM3CON	64-pin	80-pin	100-pin	0001 1111	0001 1111	uuuu uuuu
CCPTMRS0	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
CCPTMRS1	64-pin	80-pin	100-pin	00-0 -000	00-0 -000	uuuu uuuu
CCPTMRS2	64-pin	80-pin	100-pin	---0 -000	---0 -000	uuuu uuuu
RCSTA2	64-pin	80-pin	100-pin	0000 000x	0000 000x	uuuu uuuu
TXSTA2	64-pin	80-pin	100-pin	0000 0010	0000 0010	uuuu uuuu
BAUDCON2	64-pin	80-pin	100-pin	01x0 0000	01x0 0000	uuuu uuuu
SPBRGH1	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
RCSTA3	64-pin	80-pin	100-pin	0000 000x	0000 000x	uuuu uuuu
TXSTA3	64-pin	80-pin	100-pin	0000 0010	0000 0010	uuuu uuuu
BAUDCON3	64-pin	80-pin	100-pin	01x0 0000	01x0 0000	uuuu uuuu
SPBRGH3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
SPBRG3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
RCREG3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
TXREG3	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
DSCONL	64-pin	80-pin	100-pin	---- -000	---- -000	--- -uuu
DSCONH	64-pin	80-pin	100-pin	0-0- ---0	u-u- ---u	u-u- ---u
DSWAKEL	64-pin	80-pin	100-pin	0000 0001	uuuu uuuu	uuuu uuuu
DSWAKEH	64-pin	80-pin	100-pin	---- ---0	---- ---u	---- ---q
DSGPR0(6)	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
DSGPR1(6)	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
DSGPR2(6)	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
DSGPR3	64-pin	80-pin	100-pin	xxxx xxxx	uuuu uuuu	uuuu uuuu
SPBRGH2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
SPBRG2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
RCREG2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
TXREG2	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
PSTR2CON	64-pin	80-pin	100-pin	00-0 0001	00-0 0001	uu-u uuuu
PSTR3CON	64-pin	80-pin	100-pin	00-0 0001	00-0 0001	uu-u uuuu
SSP2STAT	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu
SSP2CON1	64-pin	80-pin	100-pin	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged; x = unknown; - = unimplemented bit, read as '0'; q = value depends on condition. Shaded cells indicate that 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).

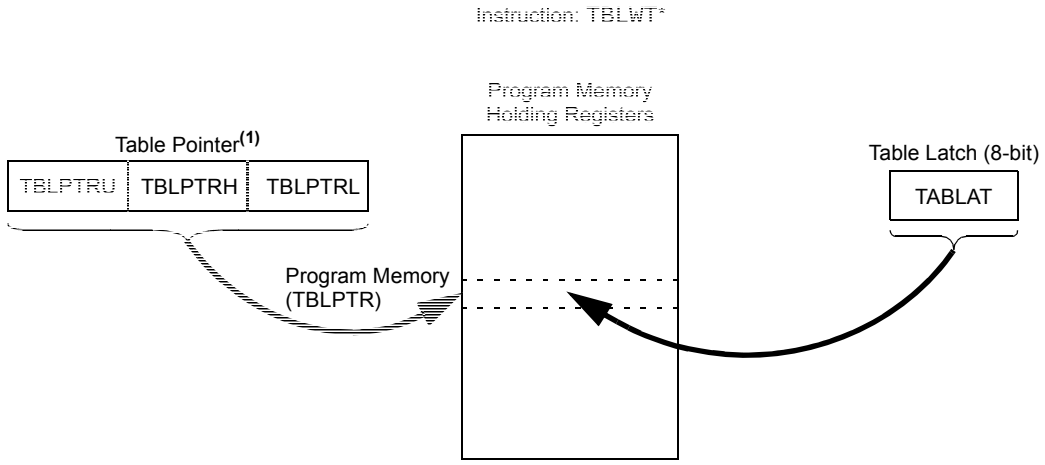
3: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

4: See Table 5-2 for Reset value for specific condition.

5: Bits 7,6 are unimplemented on 64 and 80-pin devices.

6: If the V_{BAT} is always powered, the DSGPx register values will remain unchanged after the first POR.

FIGURE 7-2: TABLE WRITE OPERATION



Note 1: The Table Pointer actually points to one of 64 holding registers; the address of which is determined by TBLPTRL<5:0>. The process for physically writing data to the program memory array is discussed in Section 7.5 "Writing to Flash Program Memory".

7.2 Control Registers

Several control registers are used in conjunction with the TBLRD and TBLWT instructions. These include:

- EECON1 register
- EECON2 register
- TABLAT register
- TBLPTR registers

7.2.1 EECON1 AND EECON2 REGISTERS

The EECON1 register (Register 7-1) is the control register for memory accesses. The EECON2 register is not a physical register; it is used exclusively in the memory write and erase sequences. Reading EECON2 will read all '0's.

The WWPROG bit, when set, will allow programming two bytes per word on the execution of the WR command. If this bit is cleared, the WR command will result in programming on a block of 64 bytes.

The FREE bit, when set, will allow a program memory erase operation. When FREE is set, the erase operation is initiated on the next WR command. When FREE is clear, only writes are enabled.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set in hardware when the WR bit is set, and cleared when the internal programming timer expires and the write operation is complete.

Note:	During normal operation, the WRERR is read as '1'. This can indicate that a write operation was prematurely terminated by a Reset or a write operation was attempted improperly.

8.1 External Memory Bus Control

The operation of the interface is controlled by the MEMCON register (Register 8-1). This register is available in all program memory operating modes, except Microcontroller mode. In this mode, the register is disabled and cannot be written to.

The EBDIS bit (MEMCON<7>) controls the operation of the bus and related port functions. Clearing EBDIS enables the interface and disables the I/O functions of the ports, as well as any other functions multiplexed to those pins. Setting the bit enables the I/O ports and other functions, but allows the interface to override everything else on the pins when an external memory operation is required. By default, the external bus is always enabled and disables all other I/O.

The operation of the EBDIS bit is also influenced by the program memory mode being used. This is discussed in more detail in **Section 8.5 "Program Memory Modes and the External Memory Bus"**.

The WAITx bits allow for the addition of Wait states to external memory operations. The use of these bits is discussed in **Section 8.3 "Wait States"**.

The WMx bits select the particular operating mode used when the bus is operating in 16-Bit Data Width mode. This is discussed in more detail in **Section 8.6 "16-Bit Data Width Modes"**. These bits have no effect when an 8-Bit Data Width mode is selected.

REGISTER 8-1: MEMCON: EXTERNAL MEMORY BUS CONTROL REGISTER(1)

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EBDIS	—	WAIT1	WAIT0	—	—	WM1	WM0
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 **EBDIS**: External Bus Disable bit
- 1 = External bus is enabled when microcontroller accesses external memory; otherwise, all external bus drivers are mapped as I/O ports
- 0 = External bus is always enabled, I/O ports are disabled

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

bit 5-4 **WAIT<1:0>:** Table Reads and Writes Bus Cycle Wait Count bits
- 11 = Table reads and writes will wait 0 TCY
- 10 = Table reads and writes will wait 1 TCY
- 01 = Table reads and writes will wait 2 TCY
- 00 = Table reads and writes will wait 3 TCY

bit 3-2 **Unimplemented**: Read as '0'

bit 1-0 **WM<1:0>:** TBLWT Operation with 16-Bit Data Bus Width Select bits
- 1x = Word Write mode: TABLAT word output, $\overline{WRH}$ is active when TABLAT is written
- 01 = Byte Select mode: TABLAT data is copied on both MSB and LSB, $\overline{WRH}$ and ($\overline{UB}$ or $\overline{LB}$) will activate
- 00 = Byte Write mode: TABLAT data is copied on both MSB and LSB, $\overline{WRH}$ or $\overline{WRL}$ will activate

Note 1: This register is unimplemented on 64-pin devices, read as '0'.

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 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			
				@ 64 MHz	@ 48 MHz	@ 10 MHz	@ 4 MHz
8 x 8 Unsigned	Without Hardware Multiply	13	69	4.3 μs	5.7 μs	27.6 μs	69 μs
	Hardware Multiply	1	1	62.5 ns	83.3 ns	400 ns	1 μs
8 x 8 Signed	Without Hardware Multiply	33	91	5.6 μs	7.5 μs	36.4 μs	91 μs
	Hardware Multiply	6	6	375 ns	500 ns	2.4 μs	6 μs
16 x 16 Unsigned	Without Hardware Multiply	21	242	15.1 μs	20.1 μs	96.8 μs	242 μs
	Hardware Multiply	28	28	1.7 μs	2.3 μs	11.2 μs	28 μs
16 x 16 Signed	Without Hardware Multiply	52	254	15.8 μs	21.2 μs	101.6 μs	254 μs
	Hardware Multiply	35	40	2.5 μs	3.3 μs	16.0 μs	40 μs

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

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	ACTORSIF	ACTLOCKIF	TMR8IF	—	TMR6IF	TMR5IF	TMR4IF
bit 7							bit 0

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

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

bit 6 **ACTORSIF:** Active Clock Tuning Out-of-Range Interrupt Flag bit
1 = Active clock tuning out-of-range occurred
0 = Active tuning out-of-range did not occur

bit 5 **ACTLOCKIF:** Active Clock Tuning Lock Interrupt Flag bit
1 = Active clock tuning lock/unlock occurred
0 = Active clock tuning lock/unlock did not occur

bit 4 **TMR8IF:** TMR8 to PR8 Match Interrupt Flag bit
1 = TMR8 to PR8 match occurred (must be cleared in software)
0 = No TMR8 to PR8 match occurred

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

bit 2 **TMR6IF:** TMR6 to PR6 Match Interrupt Flag bit
1 = TMR6 to PR6 match occurred (must be cleared in software)
0 = No TMR6 to PR6 match occurred

bit 1 **TMR5IF:** TMR5 Overflow Interrupt Flag bit
1 = TMR5 register overflowed (must be cleared in software)
0 = TMR5 register did not overflow

bit 0 **TMR4IF:** TMR4 to PR4 Match Interrupt Flag bit
1 = TMR4 to PR4 match occurred (must be cleared in software)
0 = No TMR4 to PR4 match occurred

REGISTER 10-12: PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TMR5GIE	LCDIE	RC2IE	TX2IE	CTMUIE	CCP2IE	CCP1IE	RTCCIE
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 **TMR5GIE:** TMR5 Gate Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 6 **LCDIE:** LCD Ready Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 5 **RC2IE:** EUSART2 Receive Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 4 **TX2IE:** EUSART2 Transmit Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 3 **CTMUIE:** CTMU Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2 **CCP2IE:** CCP2 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 1 **CCP1IE:** ECCP1 Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 0 **RTCCIE:** RTCC Interrupt Enable bit
1 = Enabled
0 = Disabled

11.12 PORTL, LATL and TRISL Registers

Note: PORTL is available only on 100-pin devices.

PORTL is an 8-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISL and LATL.

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

Each of the PORTL pins has a weak internal pull-up.

The pull-ups are provided to keep the inputs at a known state for the external memory interface while powering up. A single control bit can turn off all the pull-ups. This is performed by clearing bit, RLPU (PADCFG<0>).

The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on any device Reset.

EXAMPLE 11-11: INITIALIZING PORTL

```
BANKSEL PORTL   ; select correct bank
CLRF    PORTL   ; Initialize PORTL by
                ; clearing output latches
CLRF    LATL    ; Alternate method
                ; to clear output latches
MOVLW   0CFh    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISL   ; Set RL3:RL0 as inputs
                ; RL5:RL4 as output
                ; RL7:RL6 as inputs
```

TABLE 11-11: PORTL FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RL0/SEG48	RL0	0	O	DIG	LATL<0> data output.
		1	I	ST	PORTL<0> data input.
	SEG48	0	O	ANA	LCD Segment 48 output; disables all other pin functions.
RL1/SEG49	RL1	0	O	DIG	LATL<1> data output.
		1	I	ST	PORTL<1> data input.
	SEG49	0	O	ANA	LCD Segment 49 output; disables all other pin functions.
RL2/SEG50	RL2	0	O	DIG	LATL<2> data output.
		1	I	ST	PORTL<2> data input.
	SEG50	0	O	ANA	LCD Segment 50 output; disables all other pin functions.
RL3/SEG51	RL3	0	O	DIG	LATL<3> data output.
		1	I	ST	PORTL<3> data input.
	SEG51	0	O	ANA	LCD Segment 51 output; disables all other pin functions.
RL4/SEG52	RL4	0	O	DIG	LATL<4> data output.
		1	I	ST	PORTL<4> data input.
	SEG52	0	O	ANA	LCD Segment 52 output; disables all other pin functions.
RL5/SEG53	RL5	0	O	DIG	LATL<5> data output.
		1	I	ST	PORTL<5> data input.
	SEG53	0	O	ANA	LCD Segment 53 output; disables all other pin functions.
RL6/SEG54	RL6	0	O	DIG	LATL<6> data output.
		1	I	ST	PORTL<6> data input.
	SEG54	0	O	ANA	LCD Segment 54 output; disables all other pin functions.
RL7/SEG55	RL7	0	O	DIG	LATL<7> data output.
		1	I	ST	PORTL<7> data input.
	SEG55	0	O	ANA	LCD Segment 55 output; disables all other pin functions.

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

13.4 LCD Bias Types

The LCD module can be configured in one of three bias types:

- Static bias (two voltage levels: VSS and VDD)
- 1/2 bias (three voltage levels: VSS, 1/2 VDD and VDD)
- 1/3 bias (four voltage levels: VSS, 1/3 VDD, 2/3 VDD and VDD)

LCD bias voltages can be generated with internal resistor ladders, internal bias generator or external resistor ladder.

13.5 Internal Resistor Biasing

This mode does not use external resistors, but rather internal resistor ladders that are configured to generate the bias voltage.

The internal reference ladder actually consists of three separate ladders. Disabling the internal reference ladder disconnects all of the ladders, allowing external voltages to be supplied.

Depending on the total resistance of the resistor ladders, the biasing can be classified as low, medium or high power.

Table 13-3 shows the total resistance of each of the ladders. Table 13-3 shows the internal resister ladder connections. When the internal resistor ladder is selected, the bias voltage can either be from VDD or from VDDCORE, depending on the LCDIRS setting. It can also provide software contrast control (using LCDCST<2:0>)

.

TABLE 13-3: INTERNAL RESISTANCE LADDER POWER MODES

Power Mode	Nominal Resistance of Entire Ladder	IDD
Low	3 MΩ	1 µA
Medium	300 kΩ	10 µA
High	30 kΩ	100 µA

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

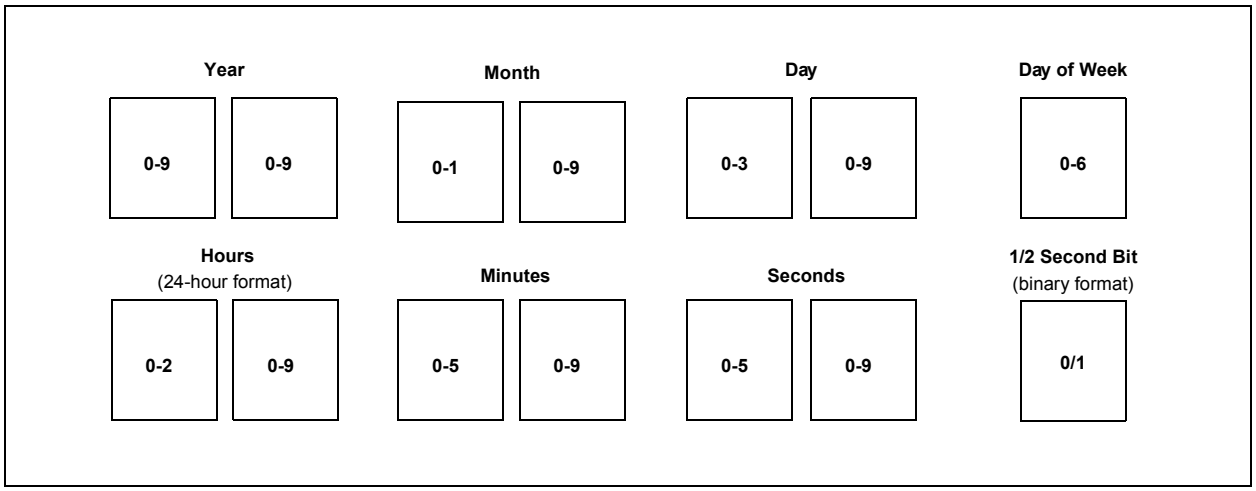
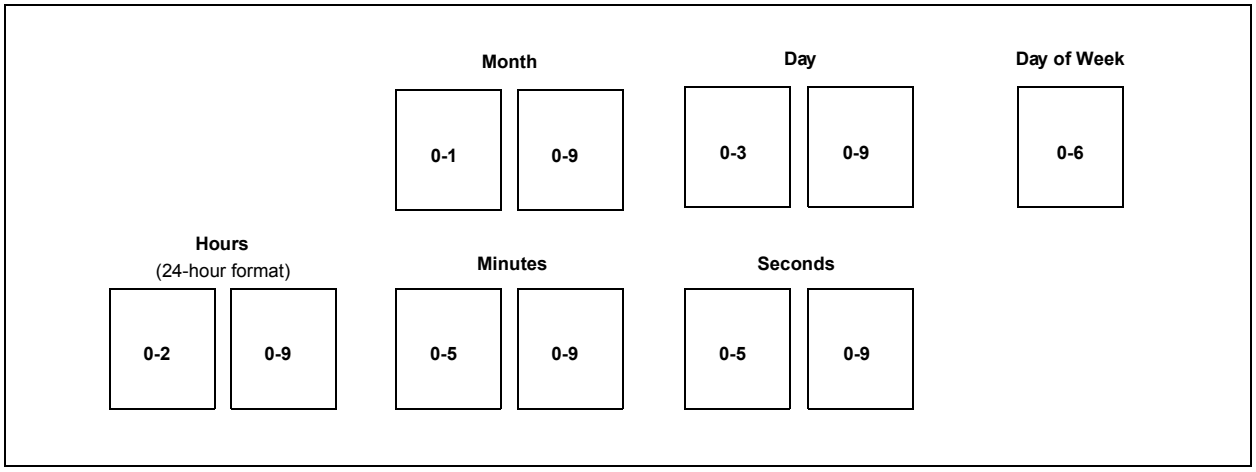


FIGURE 17-3: ALARM DIGIT FORMAT



20.3.7 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCKx. When the last bit is latched, the SSPxIF interrupt flag bit is set.

While in Slave mode, the external clock is supplied by the external clock source on the SCKx pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device can be configured to wake-up from Sleep.

20.3.8 SLAVE SELECT SYNCHRONIZATION

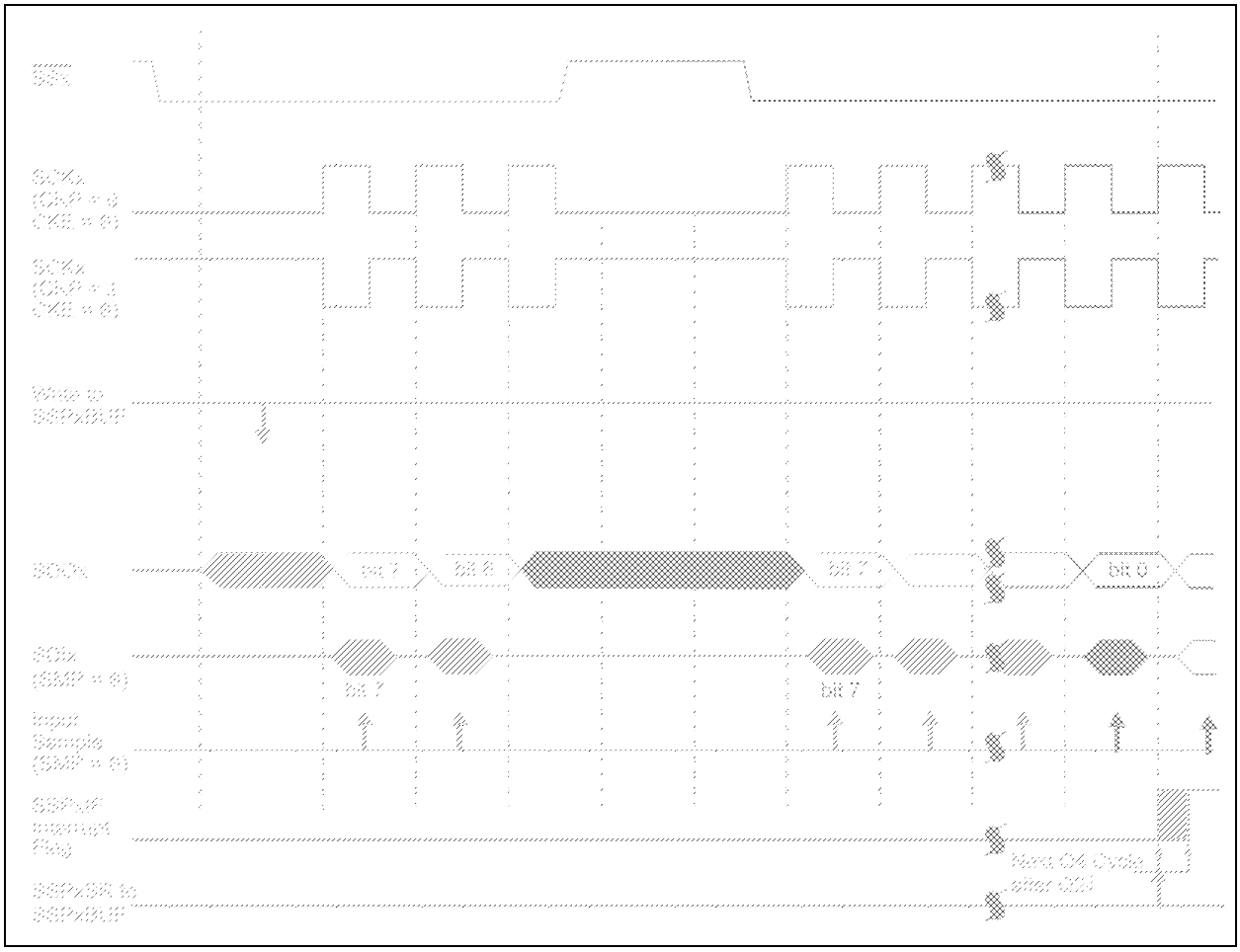
The $\overline{SSx}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with the $\overline{SSx}$ pin control enabled (SSPxCON1<3:0> = 04h). When the $\overline{SSx}$ pin is low, transmission and reception are enabled and the SDOx pin is driven. When the $\overline{SSx}$ pin goes high, the SDOx pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

Note:	When the SPI is in Slave mode with $\overline{SSx}$ pin control enabled (SSPxCON1<3:0> = 0100), the SPI module will reset if the $\overline{SSx}$ pin is set to VDD. If the SPI is used in Slave mode with CKE set, then the $\overline{SSx}$ pin control must be enabled.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SSx}$ pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDOx pin can be connected to the SDIx pin. When the SPI needs to operate as a receiver, the SDOx pin can be configured as an input. This disables transmissions from the SDOx. The SDIx can always be left as an input (SDIx function) since it cannot create a bus conflict.

FIGURE 20-4: SLAVE SYNCHRONIZATION WAVEFORM



20.5.9 MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPMx bits in SSPxCON1, and by setting the SSPEN bit. In Master mode, the SCLx and SDAx lines are manipulated by the MSSPx hardware if the TRIS bits are set.

The Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSPx module is disabled. Control of the I^2C bus may be taken when the P bit is set, or the bus is Idle, with both the S and P bits clear.

In Firmware Controlled Master mode, user code conducts all I^2C bus operations based on Start and Stop bit conditions.

Once Master mode is enabled, the user has six options.

1. Assert a Start condition on SDAx and SCLx.
2. Assert a Repeated Start condition on SDAx and SCLx.
3. Write to the SSPxBUF register initiating transmission of data/address.
4. Configure the I^2C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate a Stop condition on SDAx and SCLx.

Note: The MSSPx module, when configured in I^2C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPxBUF register to initiate transmission before the Start condition is complete. In this case, the SSPxBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPxBUF did not occur.

The following events will cause the MSSPx Interrupt Flag bit, SSPxIF, to be set (and MSSPx interrupt if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge transmitted
- Repeated Start

FIGURE 20-18: MSSPx BLOCK DIAGRAM (I^2C MASTER MODE)

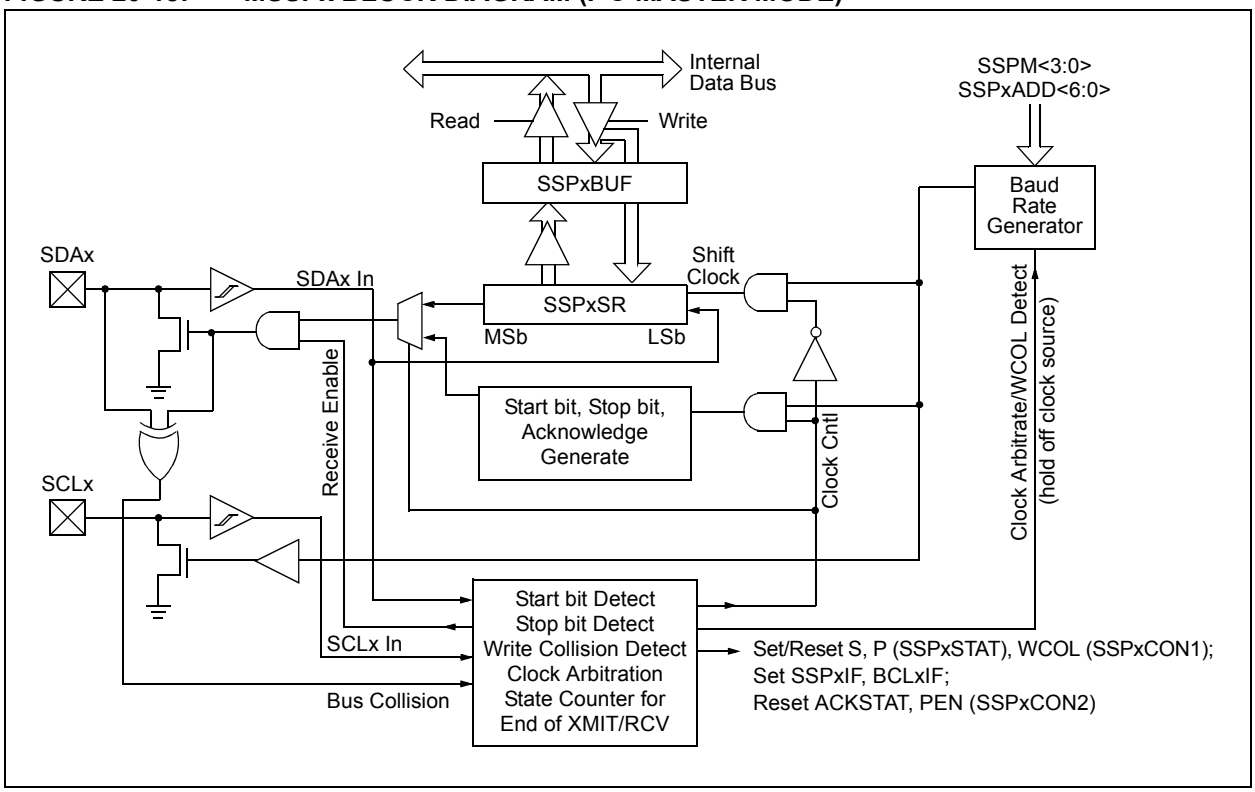
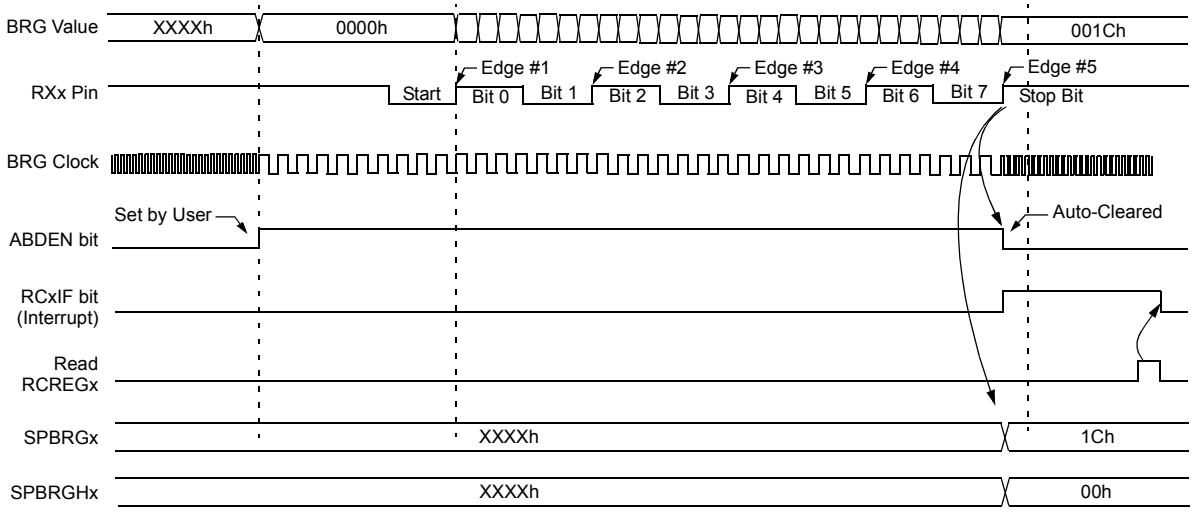
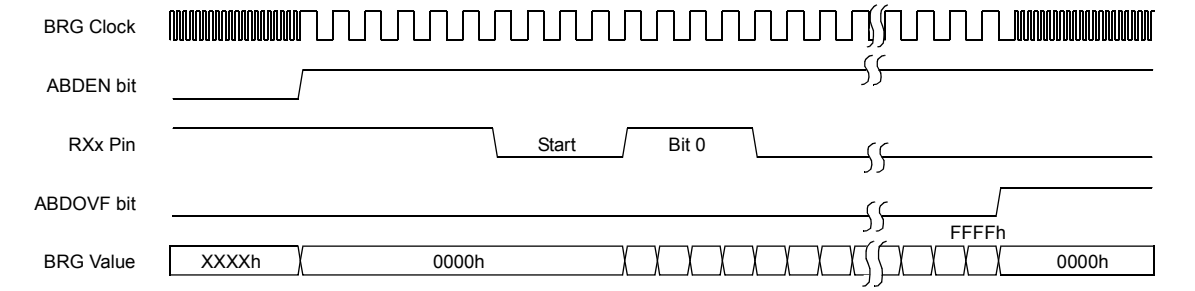


FIGURE 21-1: AUTOMATIC BAUD RATE CALCULATION



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

FIGURE 21-2: BRG OVERFLOW SEQUENCE



REGISTER 22-6: ADCON2H: A/D CONTROL REGISTER 2 HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
PVCFG1	PVCFG0	NVCFG0	OFFCAL	BUFREGEN	CSCNA	—	—
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 7-6 **PVCFG<1:0>:** Converter Positive Voltage Reference Configuration bits
1x =Unimplemented, do not use
01 =External V_{REF+}
00 =AV_{DD}

bit 5 **NVCFG0:** Converter Negative Voltage Reference Configuration bit
1 = External V_{REF-}
0 = AV_{SS}

bit 4 **OFFCAL:** Offset Calibration Mode Select bit
1 = Inverting and non-inverting inputs of channel Sample-and-Hold are connected to AV_{SS}
0 = Inverting and non-inverting inputs of channel Sample-and-Hold are connected to normal inputs

bit 3 **BUFREGEN:** A/D Buffer Register Enable bit
1 = Conversion result is loaded into the buffer location determined by the converted channel
0 = A/D result buffer is treated as a FIFO

bit 2 **CSCNA:** Scan Input Selections for CH0+ During Sample A bit
1 = Scans inputs
0 = Does not scan inputs

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

The LPENA bit (ADCON5H<6>) allows Threshold Detect to function with a low-power feature. By design, Threshold Detect can perform comparison operations when the device is in Sleep or Idle modes, waking the CPU when it generates an interrupt. Setting LPENA configures the device to return to low-power operation after the interrupt has been serviced.

The Compare Mode bits, CM<1:0> (ADCON5L<1:0>), select the type of comparison to be performed. Four types are available:

- The result of the current conversion is greater than a reference threshold
- The result of the current conversion is less than a reference threshold
- The result of the current conversion is between two predefined thresholds (“Inside Window”)
- The result of the current conversion is outside of the predefined thresholds (“Outside Window”)

The Write Mode bits, WM<1:0> (ADCON5L<3:2>), determine the disposition of the conversion. Three options are available:

- Discard the conversion after the comparison has been performed
- Store the conversion after the comparison has been performed
- Store the conversion without comparison (Legacy mode)

22.7.1.1 Buffer Operation And Comparisons

For Buffer Write modes that involve storing conversions (WM<1:0> = 0x), the BUFM and BUFREGEN bits control how the buffer functions (as a channel indexed, single FIFO or split FIFO buffer). However, when the compare and store option is selected (WM<1:0> = 01), using a FIFO mode may overwrite the buffers of other channels and cause unpredictable comparison results. For that reason, always use Channel Indexed Buffer mode (BUFREGEN = 1) when using the compare and store option.

22.7.1.2 Buffer Operation In Windowed Comparisons (Channel Mirroring)

The use of windowed comparisons changes the available options for the results buffer. To accommodate the storage of two threshold values, the buffer is automatically split into halves, similar to Split FIFO mode. Buffer addresses in each half are paired, with the lowest address in one buffer, matched to the buffer address in the upper half. (For example, in a 16-word buffer, ADCBUF0 is paired with ADCBUF9, ADCBUF1 is paired with ADCBUF10, and so on.) This pairing is referred to as “channel mirroring”.

Mirroring can obviously be applied only to the lower A/D channels; for most devices, this corresponds to the lower half of the external analog channels. This does not mean that those buffer locations cannot be used for other purposes. However, storing any other data in a particular buffer location, where channel mirroring is being used, may result in misleading comparison evaluations.

22.7.2 SETTING COMPARISON THRESHOLDS

The comparison thresholds for Threshold Detect are set by writing the desired values to an appropriate location in the A/D results buffer. This can only be done when the module is deactivated (ADCON1H<7> = 0).

The location of the threshold is determined by the comparison type. For simple greater than, and less than, comparisons, the value is written to the buffer location corresponding to the input channel to be monitored. For example, if AN0 is to be monitored for a voltage over a certain level, the ceiling threshold is stored in ADCBUF0.

The location of the thresholds for windowed comparisons are written to two addresses. The lower value is written to the address corresponding to the monitored channel. The upper value is stored in the corresponding mirrored address in the upper half of the buffer. To expand on the previous example, if the conversion on AN0 is to be a windowed comparison, the floor threshold is stored in ADCBUF0, while the ceiling threshold is stored in ADCBUF9.

22.7.3 COMPARE HIT REGISTERS

To determine if a particular event has occurred, the A/D module uses two registers to record match events. These registers are referred to as the Compare Hit registers and are designated, ADCHIT1H/L and ADCHIT0H/L. The registers map their individual bits sequentially to each of the (up to) 32 analog channels. If a particular channel in a device is not implemented, the corresponding Compare Hit bit (CHHn) is not implemented.

Each bit serves as an event semaphore for its corresponding channel. When the programmed event occurs on that channel, the bit becomes set and stays set until it is cleared by the application. It is the user's responsibility to clear the bits after the application has evaluated them.

Depending on the event, more than one Compare Hit bit may be set. The significance of a set bit must be interpreted by the application in the context of the Compare mode selected. Particular examples are covered in **Section 22.7.5 “Comparison Mode Examples”**.

22.7.4 THRESHOLD DETECT INTERRUPTS

The A/D module can generate an interrupt and set the ADIF flag based on Threshold Detect operation. This is based on completion of a Threshold Detect sequence and/or the occurrence of a valid comparison. When Threshold Detect is enabled (ASENA = 1), A/D module interrupt generation is governed by the ASINTMDx bits (ADCON5H<1:0>), superseding any configuration implemented by the SMPIx bits (ADCON2L<6:2>). For information on alternative interrupt settings, refer to **Section 22.6.1 "Number of Conversions Per Interrupt"**.

The Threshold Detect interrupt is configured by the ASINTMD<1:0> bits (ADCON5H<1:0>). Options include interrupt after a scan sequence, interrupt after a scan sequence with a valid match, interrupt after a valid match (without waiting for the sequence to end) or no interrupt.

22.7.5 COMPARISON MODE EXAMPLES

The following examples show the effect of valid comparisons on the results buffer and the

Compare Hit registers. In each figure, changes within the registers are indicated in bold.

For the sake of simplicity, the examples assume a device with only 16 analog inputs. Devices with a greater number of channels, and thus, larger results buffers and two Compare Hit registers, will function in a similar fashion.

Note:	When using any comparison mode, always use channel indexed buffer storage (BUFREGEN = 1). Otherwise, the threshold values for other channels may be overwritten, resulting in unpredictable comparisons.

22.7.5.1 Simple Comparisons (Greater And Less Than Results)

When the Compare Mode bits, CM<1:0> (ADCON5L<1:0>), are programmed as '0x', the converter compares the sampled value to see if it is greater than (CM<1:0> = 01), or less than (CM<1:0> = 00), the threshold value in the buffer location. If the condition is met, both of the following occur:

- The Compare Hit bit (CHHn) for the corresponding channel is set.
- If the Write Mode bits, WM<1:0> (ADCON5L<3:2>), are programmed to '01', the converted value is written to the buffer, replacing the threshold value. If WM<1:0> = 10, the converted value is discarded.

The changes to the result buffer and the Compare Hit register are shown in Figure 22-13. Note that they are the same for both types of simple comparison.

The PPBRST bit (UCON<6>) controls the Reset status when Double-Buffering mode (ping-pong buffering) is used. When the PPBRST bit is set, all Ping-Pong Buffer Pointers are set to the Even buffers. PPBRST has to be cleared by firmware. This bit is ignored in buffering modes not using ping-pong buffering.

The PKTDIS bit (UCON<4>) is a flag indicating that the SIE has disabled packet transmission and reception. This bit is set by the SIE when a SETUP token is received to allow setup processing. This bit cannot be set by the microcontroller, only cleared; clearing it allows the SIE to continue transmission and/or reception. Any pending events within the Buffer Descriptor Table (BDT) will still be available, indicated within the USTAT register's FIFO buffer.

The RESUME bit (UCON<2>) allows the peripheral to perform a remote wake-up by executing resume signaling. To generate a valid remote wake-up, firmware must set RESUME for 10 ms and then clear the bit. For more information on resume signaling, see Sections 7.1.7.5, 11.4.4 and 11.9 in the "*USB 2.0 Specification*".

The SUSPND bit (UCON<1>) places the module and supporting circuitry in a Low-Power mode. The input clock to the SIE is also disabled. This bit should be set by the software in response to an IDLEIF interrupt. It should be reset by the microcontroller firmware after an ACTVIF interrupt is observed. When this bit is active, the device remains attached to the bus but the transceiver outputs remain Idle. The voltage on the V_{USB3V3} pin may vary depending on the value of this bit. Setting this bit before a IDLEIF request will result in unpredictable bus behavior.

Note:	While in Suspend mode, a typical bus-powered USB device is limited to 2.5 mA of current. This is the complete current which may be drawn by the PIC MCU device and its supporting circuitry. Care should be taken to assure minimum current draw when the device enters Suspend mode.

27.2.2 USB CONFIGURATION REGISTER (UCFG)

Prior to communicating over USB, the module's associated internal and/or external hardware must be configured. Most of the configuration is performed with the UCFG register (Register 27-2).The UFCG register contains most of the bits that control the system-level behavior of the USB module. These include:

- Bus Speed (full speed versus low speed)
- On-Chip Pull-up Resistor Enable
- On-Chip Transceiver Enable
- Ping-Pong Buffer Usage

The UCFG register also contains two bits, which aid in module testing, debugging and USB certifications. These bits control output enable state monitoring and eye pattern generation.

Note:	The USB speed, transceiver and pull-up should only be configured during the module setup phase. It is not recommended to switch these settings while the module is enabled.

27.2.2.1 Internal Transceiver

The USB peripheral has a built-in, "*USB 2.0 Specification*", full-speed and low-speed capable transceiver, internally connected to the SIE. This feature is useful for low-cost, single chip applications. The UTRDIS bit (UCFG<3>) controls the transceiver; it is enabled by default (UTRDIS = 0). The FSEN bit (UCFG<2>) controls the transceiver speed; setting this bit enables full-speed operation.

The on-chip USB pull-up resistors are controlled by the UPUEN bit (UCFG<4>). They can only be selected when the on-chip transceiver is enabled.

The internal USB transceiver obtains power from the V_{USB3V3} pin. In order to meet USB signalling level specifications, V_{USB3V3} must be supplied with a voltage source between 3.0V and 3.6V. The best electrical signal quality is obtained when a 3.3V supply is used and locally bypassed with a high quality ceramic capacitor (ex: 0.1 μF). The capacitor should be placed as close as possible to the V_{USB3V3} and V_{SS} pins.

V_{USB3V3} should always be maintained $\geq V_{DD}$. If the USB module is not used, but RC4 or RC5 are used as general purpose inputs, V_{USB3V3} should still be connected to a power source (such as V_{DD}). The input thresholds for the RC4 and RC5 pins are dependent upon the V_{USB3V3} supply level.

The D+ and D- signal lines can be routed directly to their respective pins on the USB connector or cable (for hard-wired applications). No additional resistors, capacitors or magnetic components are required, as the D+ and D- drivers have controlled slew rate and output impedance, intended to match with the characteristic impedance of the USB cable.

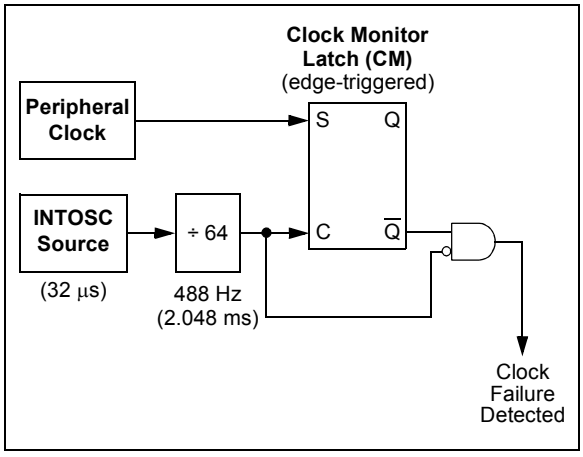
In order to achieve optimum USB signal quality, the D+ and D- traces between the microcontroller and USB connector (or cable) should be less than 19 cm long. Both traces should be equal in length and they should be routed parallel to each other. Ideally, these traces should be designed to have a characteristic impedance matching that of the USB cable.

28.4 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the microcontroller to continue operation in the event of an external oscillator failure by automatically switching the device clock to the internal oscillator block. The FSCM function is enabled by clearing the FSCMx Configuration bits.

When FSCM is enabled, the LF-INTOSC Oscillator runs at all times to monitor clocks to peripherals and provides a backup clock in the event of a clock failure. Clock monitoring (shown in Figure 28-3) is accomplished by creating a sample clock signal, which is the output from the LF-INTOSC, divided by 64. This allows ample time between FSCM sample clocks for a peripheral clock edge to occur. The peripheral device clock and the sample clock are presented as inputs to the Clock Monitor (CM) latch. The CM is set on the falling edge of the device clock source, but cleared on the rising edge of the sample clock.

FIGURE 28-3: FSCM BLOCK DIAGRAM



Clock failure is tested for on the falling edge of the sample clock. If a sample clock falling edge occurs while CM is still set, a clock failure has been detected (Figure 28-4). This causes the following:

- The FSCM generates an oscillator fail interrupt by setting bit, OSCFIF (PIR2<7>)
- The device clock source switches to the internal oscillator block (OSCCON is not updated to show the current clock source – this is the fail-safe condition)
- The WDT is reset

During switchover, the postscaler frequency from the internal oscillator block may not be sufficiently stable for timing-sensitive applications. In these cases, it may be desirable to select another clock configuration and enter an alternate power-managed mode. This can be done to attempt a partial recovery or execute a controlled shutdown. See **Section 28.3.1 "Special Considerations for Using Two-Speed Start-up"** for more details.

To use a higher clock speed on wake-up, the INTOSC or postscaler clock sources can be selected to provide a higher clock speed by setting bits, IRCF<2:0>, immediately after Reset. For wake-ups from Sleep, the INTOSC or postscaler clock sources can be selected by setting the IRCF<2:0> bits prior to entering Sleep mode.

The FSCM will detect only failures of the primary or secondary clock sources. If the internal oscillator block fails, no failure would be detected nor would any action be possible.

28.4.1 FSCM AND THE WATCHDOG TIMER

Both the FSCM and the WDT are clocked by the INTOSC Oscillator. Since the WDT operates with a separate divider and counter, disabling the WDT has no effect on the operation of the INTOSC Oscillator when the FSCM is enabled.

As already noted, the clock source is switched to the INTOSC clock when a clock failure is detected. Depending on the frequency selected by the IRCF<2:0> bits, this may mean a substantial change in the speed of code execution. If the WDT is enabled with a small prescale value, a decrease in clock speed allows a WDT time-out to occur and a subsequent device Reset. For this reason, Fail-Safe Clock events also reset the WDT and postscaler, allowing it to start timing from when execution speed was changed and decreasing the likelihood of an erroneous time-out.

28.4.2 EXITING FAIL-SAFE OPERATION

The Fail-Safe condition is terminated by either a device Reset or by entering a power-managed mode. On Reset, the controller starts the primary clock source, specified in Configuration Register 1H (with any required start-up delays that are required for the oscillator mode, such as the OST or PLL timer). The INTOSC multiplexer provides the device clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock source is then switched to the primary clock automatically after an OST. The Fail-Safe Clock Monitor then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INTOSC multiplexer. The OSCCON register will remain in its Reset state until a power-managed mode is entered.

29.2.5 SPECIAL CONSIDERATIONS WITH MICROCHIP MPLAB® IDE TOOLS

The latest versions of Microchip's software tools have been designed to fully support the extended instruction set for the PIC18F97J94 Family. This includes the MPLAB C18 C Compiler, MPASM assembly language and MPLAB Integrated Development Environment (IDE).

When selecting a target device for software development, MPLAB IDE will automatically set default Configuration bits for that device. The default setting for the XINST Configuration bit is '1', enabling the extended instruction set and Indexed Literal Offset Addressing. For proper execution of applications developed to take advantage of the extended instruction set, XINST must be set during programming.

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

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

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