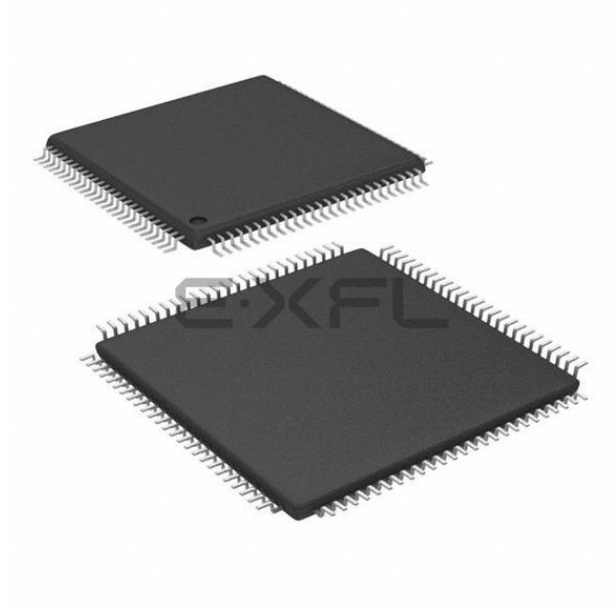


Welcome to E-XFL.COM

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	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 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/pic18f95j94-i-pf

REGISTER 3-1: OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0	R-x	R-x	R-x	U-0	R/W-x	R/W-x	R/W-x
IDLEN	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
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 **IDLEN:** Idle Enable bit
1 = SLEEP instruction invokes Idle mode
0 = SLEEP instruction invokes Sleep mode

bit 6-4 **COSC<2:0>:** Current Oscillator Selection bits (read-only)
000 = Fast RC Oscillator (FRC)
001 = Fast RC Oscillator (FRC), divided by N, with PLL module
010 = Primary Oscillator (MS, HS, EC)
011 = Primary Oscillator (MS, HS, EC) with PLL module
100 = Secondary Oscillator (SOSC)
101 = Low-Power RC Oscillator (LPRC)
110 = Fast RC Oscillator (FRC) divided by 16 (500 kHz)
111 = Fast RC Oscillator (FRC) divided by N

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **NOSC<2:0>:** New Oscillator Selection bits
000 = Fast RC Oscillator (FRC)
001 = Fast RC Oscillator (FRC), divided by N, with PLL module
010 = Primary Oscillator (MS, HS, EC)
011 = Primary Oscillator (MS, HS, EC) with PLL module
100 = Secondary Oscillator (SOSC)
101 = Low-Power RC Oscillator (LPRC)
110 = Fast RC Oscillator (FRC) divided by 16 (500 kHz)
111 = Fast RC Oscillator (FRC) divided by N

REGISTER 5-2: RCON2: RESET CONTROL REGISTER 2

R/W-0, HS	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
EXTR(1)	—	SWDTEN(2)	—	—	—	—	—
bit 7							bit 0

Legend:	HS = Hardware Settable 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 7 **EXTR:** External Reset ($\overline{MCLR}$) Pin bit(1)
1 = A Master Clear (pin) Reset has occurred
0 = A Master Clear (pin) Reset has not occurred

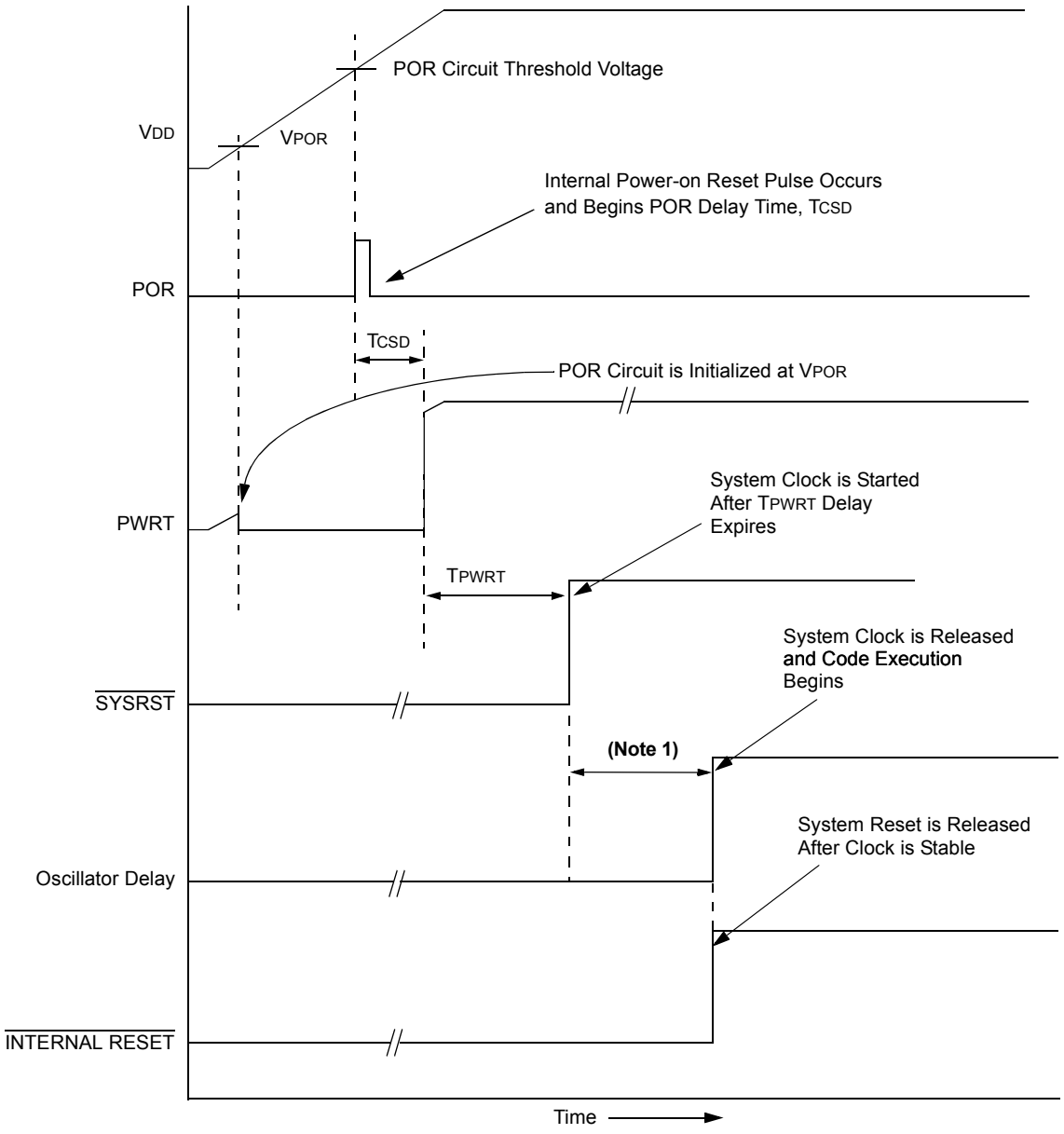
bit 6 **Unimplemented**: Read as '0'

bit 5 **SWDTEN:** Software Controlled Watchdog Timer Enable bit(2)
1 = Watchdog Timer is on
0 = Watchdog Timer is off

bit 4-0 **Unimplemented**: Read as '0'

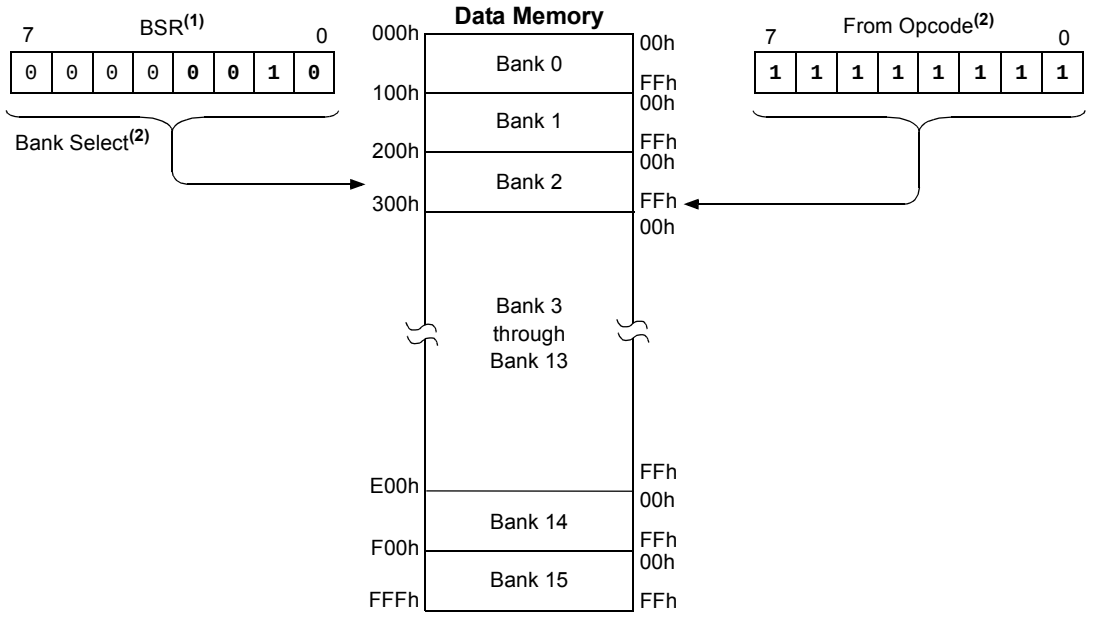
Note 1: This bit is set in hardware; it can be cleared in software.
2: This bit has no effect unless the Configuration bits, WDTEN<1:0> = 10.

FIGURE 5-2: POR MODULE TIMING SEQUENCE FOR RISING VDD



Note 1: Timer and interval are determined by the initial start-up oscillator configuration; TOSC is for external oscillator modes, TFRC is for the FRC Oscillator or TLPRC for the internal 31 kHz RC Oscillator.

FIGURE 6-7: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)



Note 1: The Access RAM bit of the instruction can be used to force an override of the selected bank (BSR<3:0>) to the registers of the Access Bank.

2: The MOVFF instruction embeds the entire 12-bit address in the instruction.

6.3.2 ACCESS BANK

While the use of the BSR, with an embedded 8-bit address, allows users to address the entire range of data memory, it also means that the user must ensure that the correct bank is selected. If not, data may be read from, or written to, the wrong location. This can be disastrous if a GPR is the intended target of an operation, but an SFR is written to instead. Verifying and/or changing the BSR for each read or write to data memory can become very inefficient.

To streamline access for the most commonly used data memory locations, the data memory is configured with an Access Bank, which allows users to access a mapped block of memory without specifying a BSR. The Access Bank consists of the first 96 bytes of memory (00h-5Fh) in Bank 0 and the last 160 bytes of memory (60h-FFh) in Bank 15. The lower half is known as the “Access RAM” and is composed of GPRs. The upper half is where the device’s SFRs are mapped. These two areas are mapped contiguously in the Access Bank and can be addressed in a linear fashion by an 8-bit address (Figure 6-6).

The Access Bank is used by core PIC18 instructions that include the Access RAM bit (the ‘a’ parameter in the instruction). When ‘a’ is equal to ‘1’, the instruction uses the BSR and the 8-bit address included in the opcode for the data memory address. When ‘a’ is ‘0’, however, the instruction is forced to use the Access Bank address map. In that case, the current value of the BSR is ignored entirely.

Using this “forced” addressing allows the instruction to operate on a data address in a single cycle without updating the BSR first. For 8-bit addresses of 60h and above, this means that users can evaluate and operate on SFRs more efficiently. The Access RAM below 60h is a good place for data values that the user might need to access rapidly, such as immediate computational results or common program variables.

Access RAM also allows for faster and more code efficient context saving and switching of variables.

The mapping of the Access Bank is slightly different when the extended instruction set is enabled (XINST Configuration bit = 1). This is discussed in more detail in **Section 6.6.3 “Mapping the Access Bank in Indexed Literal Offset Mode”**.

6.3.3 GENERAL PURPOSE REGISTER FILE

PIC18 devices may have banked memory in the GPR area. This is data RAM which is available for use by all instructions. GPRs start at the bottom of Bank 0 (address 000h) and grow upwards towards the bottom of the SFR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

10.7 IPR Registers

The IPR registers contain the individual priority bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are six Peripheral Interrupt Priority registers (IPR1 through IPR6). Using the priority bits requires that the Interrupt Priority Enable (IPEN) bit (RCON<7>) be set.

REGISTER 10-16: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PSPIP	ADIP	RC1IP	TX1IP	SSP1IP	TMR1GIP	TMR2IP	TMR1IP
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 **PSPIP:** Parallel Slave Port Read/Write Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **ADIP:** A/D Converter Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **RC1IP:** EUSART1 Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **TX1IP:** EUSART1 Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **SSP1IP:** Master Synchronous Serial Port 1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **TMR1GIP:** Timer1 Gate Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **TMR2IP:** TMR2 to PR2 Match Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR1IP:** TMR1 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 10-19: IPR4: PERIPHERAL INTERRUPT PRIORITY REGISTER 4

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
CCP10IP	CCP9IP	CCP8IP	CCP7IP	CCP6IP	CCP5IP	CCP4IP	ECCP3IP
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 **CCP10IP:** CCP10 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **CCP9IP:** CCP9 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **CCP8IP:** CCP8 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **CCP7IP:** CCP7 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **CCP6IP:** CCP6 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **CCP5IP:** CCP5 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **CCP4IP:** CCP4 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **ECCP3IP:** ECCP3 Interrupt Priority bits
1 = High priority
0 = Low priority

FIGURE 12-1: SIMPLIFIED BLOCK DIAGRAM OF THE DATA SIGNAL MODULATOR

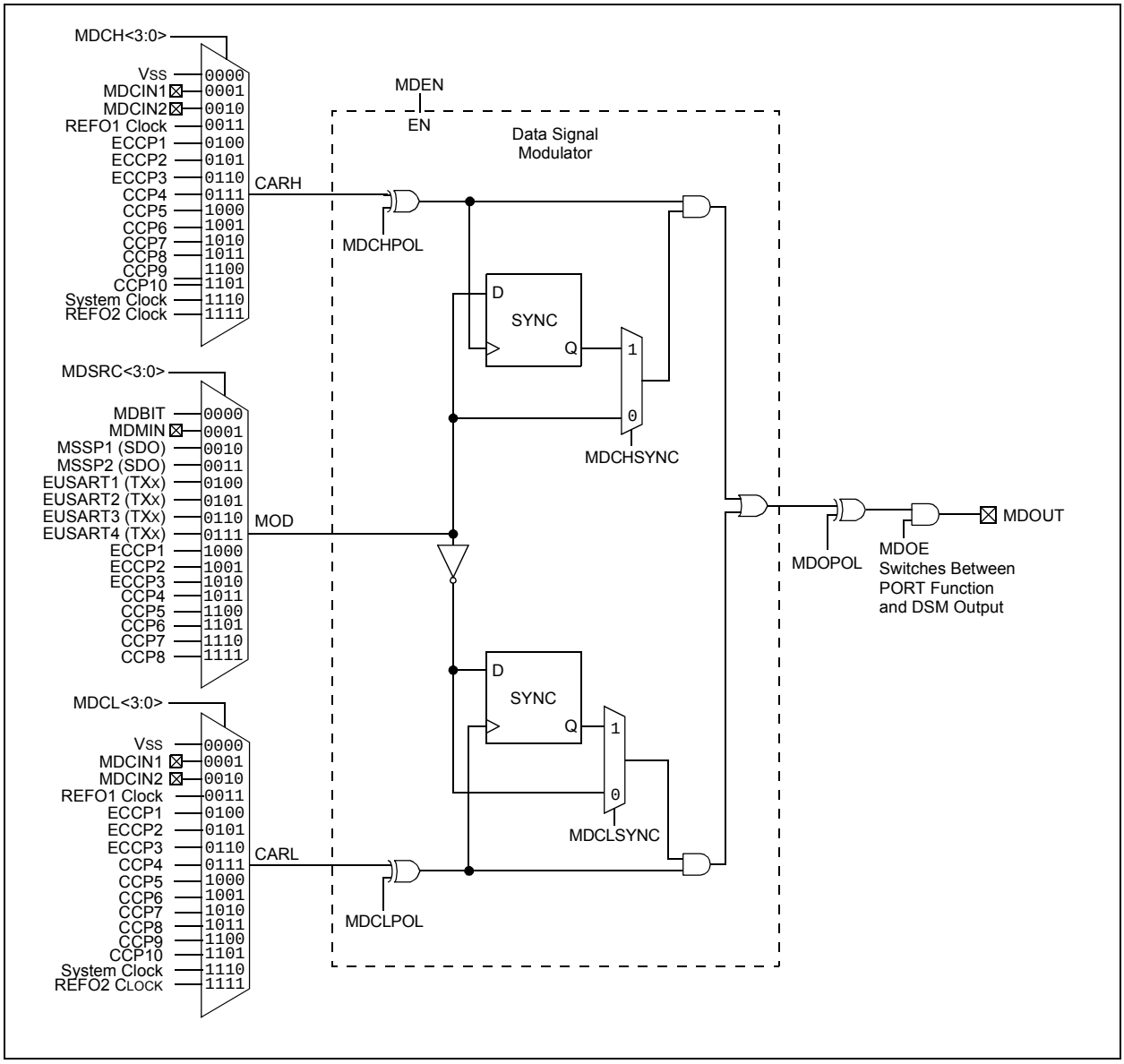
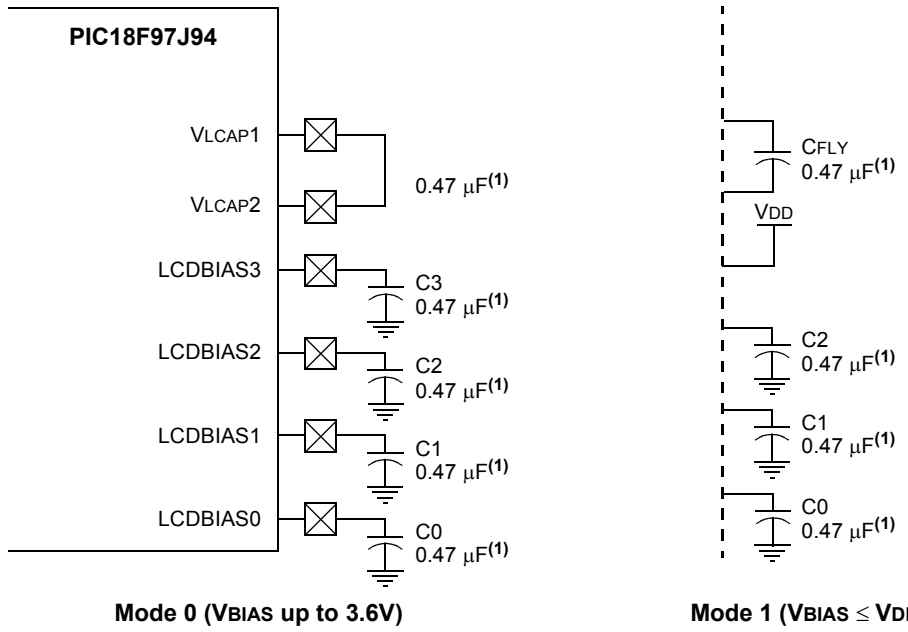


FIGURE 13-6: LCD REGULATOR CONNECTIONS FOR M0 AND M1 CONFIGURATIONS



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.

13.13 LCD Interrupts

The LCD timing generation provides an interrupt that defines the LCD frame timing. This interrupt can be used to coordinate the writing of the pixel data with the start of a new frame, which produces a visually crisp transition of the image.

This interrupt can also be used to synchronize external events to the LCD. For example, the interface to an external segment driver can be synchronized for segment data updates to the LCD frame.

A new frame is defined as beginning at the leading edge of the COM0 common signal. The interrupt will be set immediately after the LCD controller completes accessing all pixel data required for a frame. This will occur at a fixed interval before the frame boundary (TFINT), as shown in Figure 13-22.

The LCD controller will begin to access data for the next frame within the interval from the interrupt to when the controller begins accessing data after the interrupt (TFWR). New data must be written within TFWR, as this is when the LCD controller will begin to access the data for the next frame.

When the LCD driver is running with Type-B waveforms, and the LMUX<2:0> bits are not equal to '000', there are some additional issues.

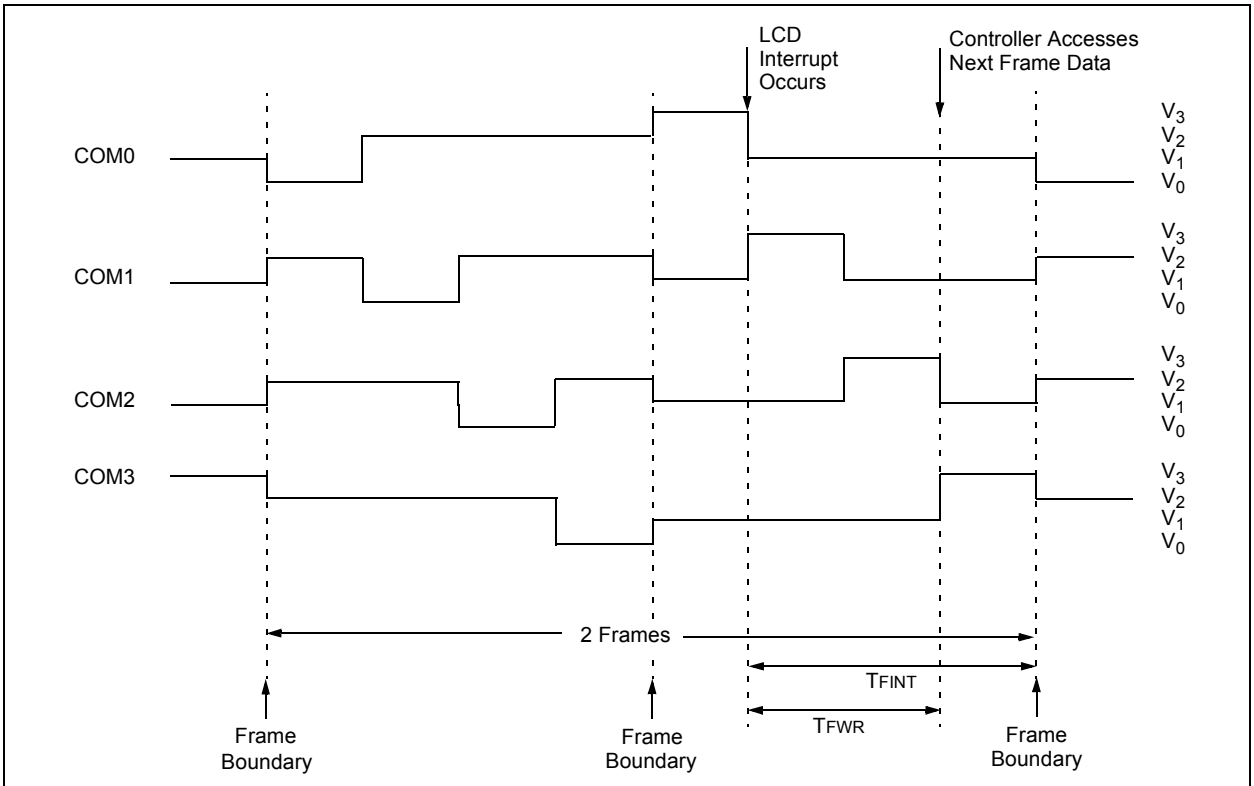
Since the DC voltage on the pixel takes two frames to maintain 0V, the pixel data must not change between subsequent frames. If the pixel data were allowed to change, the waveform for the odd frames would not necessarily be the complement of the waveform generated in the even frames and a DC component would be introduced into the panel.

Because of this, using Type-B waveforms requires synchronizing the LCD pixel updates to occur within a subframe after the frame interrupt.

To correctly sequence writing in Type-B, the interrupt only occurs on complete phase intervals. If the user attempts to write when the write is disabled, the WERR bit (LCDCON<5>) is set.

Note:	The interrupt is not generated when the Type-A waveform is selected and when the Type-B with no multiplex (static) is selected.

FIGURE 13-22: EXAMPLE WAVEFORMS AND INTERRUPT TIMING IN QUARTER DUTY CYCLE DRIVE



TFWR = TFRAME/2 * (LMUX<2:0> + 1) + TCY/2

TFINT = (TFWR/2 – (2 TCY + 40 ns)) → Minimum = 1.5(TFRAME/4) – (2 TCY + 40 ns)
(TFWR/2 – (1 TCY + 40 ns)) → Maximum = 1.5(TFRAME/4) – (1 TCY + 40 ns)

18.2 Capture Mode

In Capture mode, the CCPRxH:CCPRxL register pair captures the 16-bit value of the TMR1 or TMR3 registers when an event occurs on the corresponding ECCPx pin. An event is defined as one of the following:

- Every falling edge
- Every rising edge
- Every fourth rising edge
- Every 16th rising edge

The event is selected by the mode select bits, CCPxM<3:0> (CCPxCON<3:0>). When a capture is made, the interrupt request flag bit, CCPxIF, is set (see Table 18-2). The flag must be cleared by software. If another capture occurs before the value in the CCPRxH/L register is read, the old captured value is overwritten by the new captured value.

TABLE 18-2: ECCP1/2/3 INTERRUPT FLAG BITS

ECCP Module	Flag Bit
1	PIR3<1>
2	PIR3<2>
3	PIR4<0>

18.2.1 ECCP PIN CONFIGURATION

In Capture mode, the appropriate ECCPx pin should be configured as an input by setting the corresponding TRIS direction bit.

Note:	If the ECCPx pin is configured as an output, a write to the port can cause a capture condition.

18.2.2 TIMER1/2/3/4/5/6/8 MODE SELECTION

The timers that are to be used with the capture feature (Timer1/2/3/4/5/6 or 8) must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation will not work. The timer to be used with each ECCP module is selected in the CCPTMRS0 register (Register 18-2).

18.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE interrupt enable bit clear to avoid false interrupts. The interrupt flag bit, CCPxIF, should also be cleared following any such change in operating mode.

18.2.4 ECCP PRESCALER

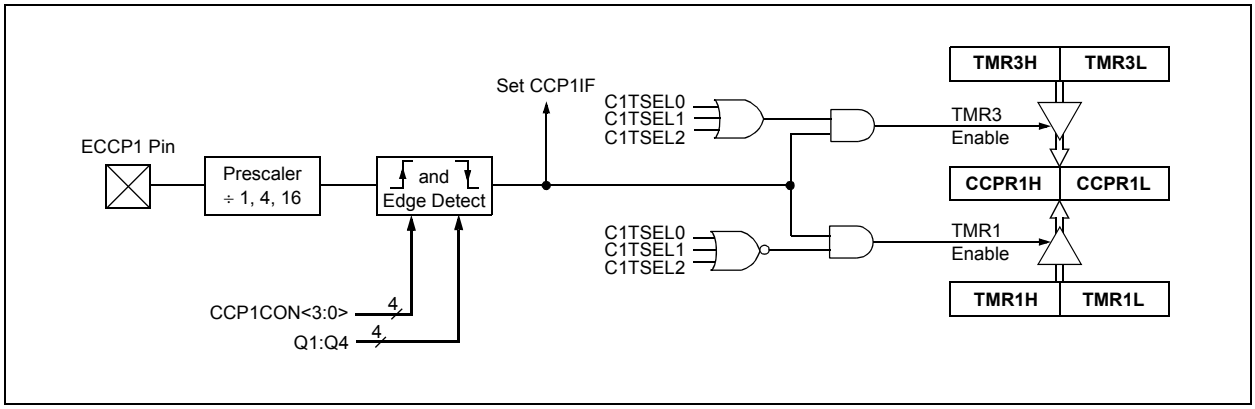
There are four prescaler settings in Capture mode; they are specified as part of the operating mode selected by the mode select bits (CCPxM<3:0>). Whenever the ECCP module is turned off, or Capture mode is disabled, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 18-1 provides the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

EXAMPLE 18-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF   CCP1CON     ; Turn ECCP module off
MOVLW  NEW_CAPT_PS ; Load WREG with the
                   ; new prescaler mode
                   ; value and ECCP ON
MOVWF  CCP1CON     ; Load CCP1CON with
                   ; this value
```

FIGURE 18-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



18.4.8 OPERATION IN POWER-MANAGED MODES

In Sleep mode, all clock sources are disabled. Timer2/4/6/8 will not increment and the state of the module will not change. If the ECCPx pin is driving a value, it will continue to drive that value. When the device wakes up, it will continue from this state. If Two-Speed Start-ups are enabled, the initial start-up frequency from HF-INTOSC and the postscaler may not be stable immediately.

In PRI_IDLE mode, the primary clock will continue to clock the ECCPx module without change.

18.4.8.1 Operation with Fail-Safe Clock Monitor (FSCM)

If the Fail-Safe Clock Monitor (FSCM) is enabled, a clock failure will force the device into the power-managed RC_RUN mode and the OSCFIF bit of the PIR2 register will be set. The ECCPx will then be clocked from the internal oscillator clock source, which may have a different clock frequency than the primary clock.

18.4.9 EFFECTS OF A RESET

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

This forces the ECCP module to reset to a state compatible with previous, non-enhanced CCP modules used on other PIC18 and PIC16 devices.

REGISTER 20-7: SSPxCON1: MSSPx CONTROL REGISTER 1 (I^2C MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN(1)	CKP	SSPM3(2)	SSPM2(2)	SSPM1(2)	SSPM0(2)
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 **WCOL:** Write Collision Detect bit

In Master Transmit mode:

1 = A write to the SSPxBUF register was attempted while the I^2C conditions were not valid for a transmission to be started (must be cleared in software)

0 = No collision

In Slave Transmit mode:

1 = The SSPxBUF register is written while it is still transmitting the previous word (must be cleared in software)

0 = No collision

In Receive mode (Master or Slave modes):

This is a "don't care" bit.

bit 6 **SSPOV:** Receive Overflow Indicator bit

In Receive mode:

1 = A byte is received while the SSPxBUF register is still holding the previous byte (must be cleared in software)

0 = No overflow

In Transmit mode:

This is a "don't care" bit in Transmit mode.

bit 5 **SSPEN:** Master Synchronous Serial Port Enable bit(1)

1 = Enables the serial port and configures the SDAx and SCLx pins as the serial port pins

0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** SCKx Release Control bit

In Slave mode:

1 = Releases clock

0 = Holds clock low (clock stretch), used to ensure data setup time

In Master mode:

Unused in this mode.

bit 3-0 **SSPM<3:0>:** Master Synchronous Serial Port Mode Select bits(2)

1111 = I^2C Slave mode: 10-bit address with Start and Stop bit interrupts enabled

1110 = I^2C Slave mode: 7-bit address with Start and Stop bit interrupts enabled

1011 = I^2C Firmware Controlled Master mode (slave Idle)

1001 = Load SSPxMSK register at SSPxADD SFR address(3,4)

1000 = I^2C Master mode: Clock = FOSC/(4 * (SSPxADD + 1))

0111 = I^2C Slave mode: 10-bit address(3,4)

0110 = I^2C Slave mode: 7-bit address

Note 1: When enabled, the SDAx and SCLx pins must be configured as inputs.

2: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

3: When SSPM<3:0> = 1001, any reads or writes to the SSPxADD SFR address actually accesses the SSPxMSK register.

4: This mode is only available when 7-Bit Address Masking mode is selected (MSSPMSK Configuration bit is '1').

20.5.7 CLOCK STRETCHING

Both 7-Bit and 10-Bit Slave modes implement automatic clock stretching during a transmit sequence.

The SEN bit (SSPxCON2<0>) allows clock stretching to be enabled during receives. Setting SEN will cause the SCLx pin to be held low at the end of each data receive sequence.

20.5.7.1 Clock Stretching for 7-Bit Slave Receive Mode (SEN = 1)

In 7-Bit Slave Receive mode, on the falling edge of the ninth clock at the end of the $\overline{ACK}$ sequence, if the BF bit is set, the CKP bit in the SSPxCON1 register is automatically cleared, forcing the SCLx output to be held low. The CKP bit, being cleared to '0', will assert the SCLx line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCLx line low, the user has time to service the ISR and read the contents of the SSPxBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 20-15).

Note 1: If the user reads the contents of the SSPxBUF before the falling edge of the ninth clock, thus clearing the BF bit, the CKP bit will not be cleared and clock stretching will not occur.

2: The CKP bit can be set in software regardless of the state of the BF bit. The user should be careful to clear the BF bit in the ISR before the next receive sequence in order to prevent an overflow condition.

20.5.7.2 Clock Stretching for 10-Bit Slave Receive Mode (SEN = 1)

In 10-Bit Slave Receive mode, during the address sequence, clock stretching automatically takes place but CKP is not cleared. During this time, if the UA bit is set after the ninth clock, clock stretching is initiated. The UA bit is set after receiving the upper byte of the 10-bit address, and following the receive of the second byte of the 10-bit address, with the R/$\overline{W}$ bit cleared to '0'. The release of the clock line occurs upon updating SSPxADD. Clock stretching will occur on each data receive sequence as described in 7-bit mode.

Note: If the user polls the UA bit and clears it by updating the SSPxADD register before the falling edge of the ninth clock occurs, and if the user hasn't cleared the BF bit by reading the SSPxBUF register before that time, then the CKP bit will still NOT be asserted low. Clock stretching, on the basis of the state of the BF bit, only occurs during a data sequence, not an address sequence.

20.5.7.3 Clock Stretching for 7-Bit Slave Transmit Mode

The 7-Bit Slave Transmit mode implements clock stretching by clearing the CKP bit after the falling edge of the ninth clock if the BF bit is clear. This occurs regardless of the state of the SEN bit.

The user's ISR must set the CKP bit before transmission is allowed to continue. By holding the SCLx line low, the user has time to service the ISR and load the contents of the SSPxBUF before the master device can initiate another transmit sequence (see Figure 20-10).

Note 1: If the user loads the contents of SSPxBUF, setting the BF bit before the falling edge of the ninth clock, the CKP bit will not be cleared and clock stretching will not occur.

2: The CKP bit can be set in software, regardless of the state of the BF bit.

20.5.7.4 Clock Stretching for 10-Bit Slave Transmit Mode

In 10-Bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-Bit Slave Receive mode. The first two addresses are followed by a third address sequence, which contains the high-order bits of the 10-bit address and the R/$\overline{W}$ bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled by the BF flag as in 7-Bit Slave Transmit mode (see Figure 20-13).

20.5.8 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I^2C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an Acknowledge.

The general call address is one of eight addresses reserved for specific purposes by the I^2C protocol. It consists of all '0's with R/$\overline{W}$ = 0.

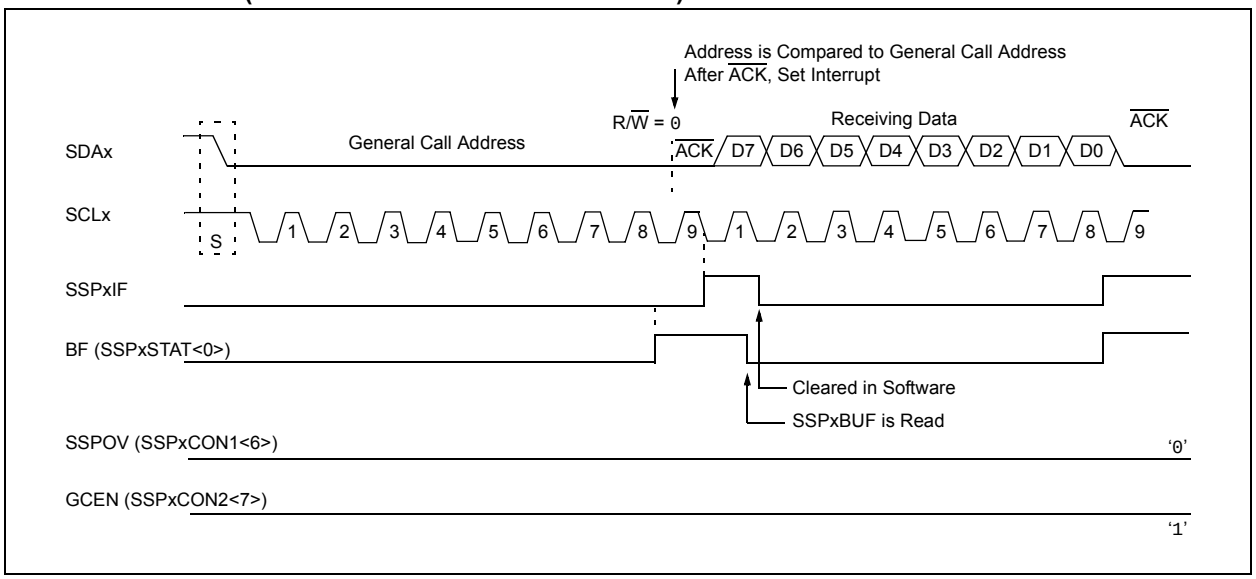
The general call address is recognized when the General Call Enable bit, GCEN, is enabled (SSPxCON2<7> set). Following a Start bit detect, eight bits are shifted into the SSPxSR and the address is compared against the SSPxADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPxSR is transferred to the SSPxBUF, the BF flag bit is set (eighth bit), and on the falling edge of the ninth bit ($\overline{ACK}$ bit), the SSPxIF interrupt flag bit is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPxBUF. The value can be used to determine if the address was device-specific or a general call address.

In 10-Bit Addressing mode, the SSPxADD is required to be updated for the second half of the address to match and the UA bit is set (SSPxSTAT<1>). If the general call address is sampled when the GCEN bit is set, while the slave is configured in 10-Bit Addressing mode, then the second half of the address is not necessary, the UA bit will not be set and the slave will begin receiving data after the Acknowledge (Figure 20-17).

FIGURE 20-17: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT ADDRESSING MODE)



REGISTER 22-12: ADCHS0H: A/D SAMPLE SELECT REGISTER 0 HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0
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 **CH0NB<2:0>:** Sample B Channel 0 Negative Input Select bits
1xx = Unimplemented
011 = Unimplemented
010 = AN1
001 = Unimplemented
000 = V_{REF-}/AV_{SS}

bit 4-0 **CH0SB<4:0>:** Sample B 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}) **(1,3)**
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

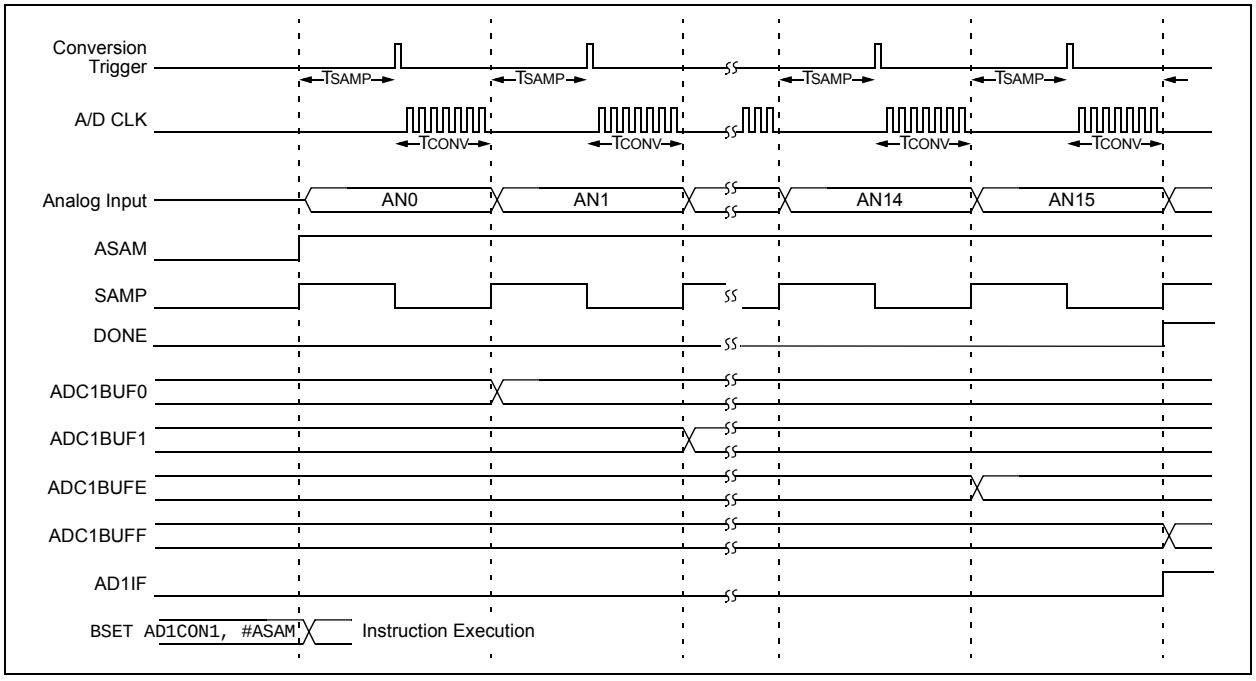
Note 1: These input channels do not have corresponding memory mapped result buffers.
2: These channels are implemented in 80-pin and 100-pin devices only.
3: For accurately sampling the band gap set SMPI bits in ADCON2L register to 0, so that the ADC samples the band gap only once on every trigger. When the band gap is sampled multiple times, a large capacitive load is connected to the output of the band gap multiple times, which could cause the output to become unstable for a while and an overshoot or undershoot could be sampled.

22.8.2.2 A/D Conversions While Scanning Through All Analog Inputs

Figure 22-17 and Example 22-5 illustrate a typical setup, where all available analog input channels are sampled and converted. In this instance, 16 analog inputs are assumed. The set CSCNA bit specifies scanning of the A/D inputs to the S/H positive input. Other conditions are similar to those located in Section **Section 22.8.2.1 "Sampling and Converting a Single Channel Multiple Times"**.

Initially, the AN0 input is sampled and converted. The result is stored in the ADCBUFn buffer. Then, the AN1 input is sampled and converted. This process of scanning the inputs repeats 16 times, until the buffer is full, and then the module generates an interrupt. The entire process will then repeat.

FIGURE 22-17: SCANNING ALL 16 INPUTS PER SINGLE INTERRUPT



23.0 COMPARATOR MODULE

The analog comparator module contains three comparators that can be independently configured in a variety of ways. The inputs can be selected from the analog inputs and two internal voltage references. The digital outputs are available at the pin level, via PPS-Lite, and can also be read through the control register. Multiple output and interrupt event generations are also available. A generic single comparator from the module is shown in Figure 23-1.

Key features of the module includes:

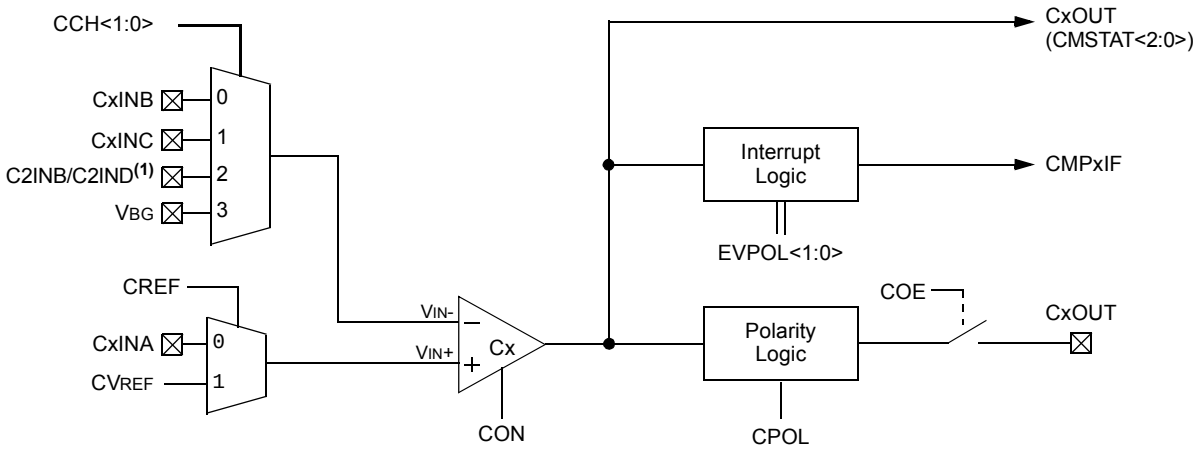
- Independent comparator control
- Programmable input configuration
- Output to both pin and register levels
- Programmable output polarity
- Independent interrupt generation for each comparator with configurable interrupt-on-change

23.1 Registers

The CMxCON registers (CM1CON, CM2CON and CM3CON) select the input and output configuration for each comparator, as well as the settings for interrupt generation (see Register 23-1).

The CMSTAT register (Register 23-2) provides the output results of the comparators. The bits in this register are read-only.

FIGURE 23-1: COMPARATOR SIMPLIFIED BLOCK DIAGRAM



Note 1: Comparator 1 and Comparator 3 use C2INB as an input to the inverted terminal. Comparator 2 uses C2IND as an input to the inverted terminal.

REGISTER 27-1: UCON: USB CONTROL REGISTER

U-0	R/W-0	R-x	R/C-0	R/W-0	R/W-0	R/W-0	U-0
—	PPBRST(2)	SE0	PKTDIS	USBEN(1)	RESUME	SUSPND	—
bit 7							bit 0

Legend:	C = Clearable 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 7 **Unimplemented:** Read as '0'

bit 6 **PPBRST:** Ping-Pong Buffers Reset bit(2)
- 1 = Reset all Ping-Pong Buffer Pointers to the Even Buffer Descriptor (BD) banks
- 0 = Ping-Pong Buffer Pointers are not being reset

bit 5 **SE0:** Live Single-Ended Zero Flag bit
- 1 = Single-ended zero is active on the USB bus
- 0 = No single-ended zero is detected

bit 4 **PKTDIS:** Packet Transfer Disable bit
- 1 = SIE token and packet processing are disabled, automatically set when a SETUP token is received
- 0 = SIE token and packet processing are enabled

bit 3 **USBEN:** USB Module Enable bit(1)
- 1 = USB module and supporting circuitry are enabled (device attached)
- 0 = USB module and supporting circuitry are disabled (device detached)

bit 2 **RESUME:** Resume Signaling Enable bit
- 1 = Resume signaling is activated
- 0 = Resume signaling is disabled

bit 1 **SUSPND:** Suspend USB bit
- 1 = USB module and supporting circuitry are in Power Conserve mode, SIE clock is inactive
- 0 = USB module and supporting circuitry are in normal operation, SIE is clocked at the configured rate

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

Note 1: Make sure the USB clock source is correctly configured before setting this bit.

2: There should be at least four cycles of delay between the setting and PPBRST.

XORWF	Exclusive OR W with f
Syntax:	XORWF f {,d {,a}}
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(W) .XOR. (f) $\rightarrow$ dest
Status Affected:	N, Z

Encoding:

0001	10da	ffff	ffff

Description: Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the register 'f'.

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

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 29.2.3 "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:

```
XORWF   REG, 1, 0
```

Before Instruction
REG = AFh
W = B5h
After Instruction
REG = 1Ah
W = B5h

TABLE 30-1: DC CHARACTERISTICS: SUPPLY VOLTAGE PIC18FXXJ94 (INDUSTRIAL)

PIC18FXXJ94 (Industrial)			**Standard Operating Conditions: 2V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**	2.0	—	3.6	V	
D001C	AVDD	**Analog Supply Voltage**	VDD – 0.3	—	VDD + 0.3	V	
D001D	AVSS	**Analog Ground Potential**	VSS – 0.3	—	VSS + 0.3	V	
D001E	VUSB3V3	**USB Supply Voltage**	3	3.3	3.6	V	USB module enabled(3)
D002	VDR	**RAM Data Retention Voltage(1)**	1.2	—	—	V	
D003	VPOR	**VDD/VBAT Start Voltage** to Ensure Internal Power-on Reset Signal	—	—	0.7	V	See **Section 5.2 "Power-on Reset (POR)"** for details
D004	SVDD	**VDD/VBAT Rise Rate** to Ensure Internal Power-on Reset Signal	0.05	—	—	V/ms	See **Section 5.2 "Power-on Reset (POR)"** for details
D005	BVDD	**Brown-out Reset Voltage** BORV = 1(2) BORV = 0	 1.8 2.0	 1.88 2.05	 1.95 2.20	 V V	
D006	VVDDBOR		1.4V		2.0	V	
D007	VVBATBOR		1.4V		1.95	V	
D008	VDSBOR				1.8		

Note 1: This is the limit to which VDD can be lowered in Sleep mode, or during a device Reset, without losing RAM data.

2: The device will operate normally until Brown-out Reset occurs, even though VDD may be below VDDMIN.

3: VUSB3V3 should be connected to VDD.