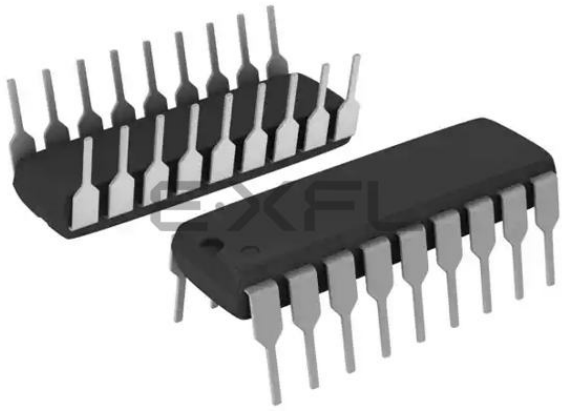


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

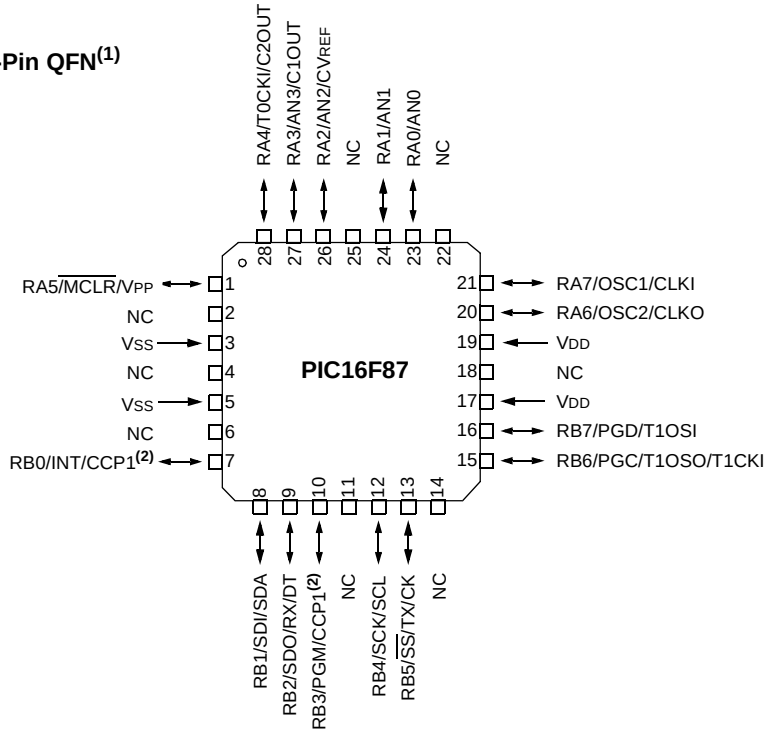
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

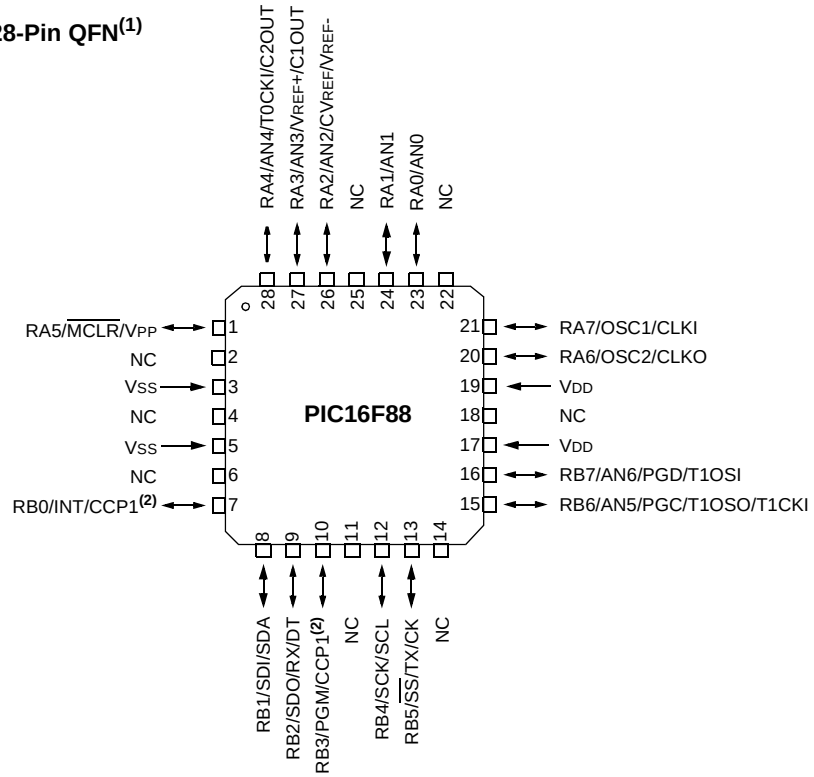
| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 10MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 16 |
| Program Memory Size | 7KB (4K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 256 x 8 |
| RAM Size | 368 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 5.5V |
| Data Converters | A/D 7x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Through Hole |
| Package / Case | 18-DIP (0.300", 7.62mm) |
| Supplier Device Package | 18-PDIP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16lf88-i-p |

## Pin Diagrams (Cont'd)

**28-Pin QFN(1)**




**28-Pin QFN(1)**




**Note 1:** For the QFN package, it is recommended that the bottom pad be connected to VSS.

**2:** The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

## 1.0 DEVICE OVERVIEW

This document contains device specific information for the operation of the PIC16F87/88 devices. Additional information may be found in the "*PIC® Mid-Range MCU Family Reference Manual*" (DS33023) which may be downloaded from the Microchip web site. This Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

The PIC16F87/88 belongs to the Mid-Range family of the PIC® devices. Block diagrams of the devices are shown in Figure 1-1 and Figure 1-2. These devices contain features that are new to the PIC16 product line:

- Low-power modes: RC_RUN allows the core and peripherals to be clocked from the INTRC, while SEC_RUN allows the core and peripherals to be clocked from the low-power Timer1. Refer to **Section 4.7 "Power-Managed Modes"** for further details.
- Internal RC oscillator with eight selectable frequencies, including 31.25 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz and 8 MHz. The INTRC can be configured as a primary or secondary clock source. Refer to **Section 4.5 "Internal Oscillator Block"** for further details.
- The Timer1 module current consumption has been greatly reduced from 20 μA (previous PIC16 devices) to 1.8 μA typical (32 kHz at 2V), which is ideal for real-time clock applications. Refer to **Section 7.0 "Timer1 Module"** for further details.
- Extended Watchdog Timer (WDT) that can have a programmable period from 1 ms to 268s. The WDT has its own 16-bit prescaler. Refer to **Section 15.12 "Watchdog Timer (WDT)"** for further details.
- Two-Speed Start-up: When the oscillator is configured for LP, XT or HS Oscillator mode, this feature will clock the device from the INTRC while the oscillator is warming up. This, in turn, will enable almost immediate code execution. Refer to **Section 15.12.3 "Two-Speed Clock Start-up Mode"** for further details.
- Fail-Safe Clock Monitor: This feature will allow the device to continue operation if the primary or secondary clock source fails by switching over to the INTRC.
- The A/D module has a new register for PIC16 devices named ANSEL. This register allows easier configuration of analog or digital I/O pins.

**TABLE 1-1: AVAILABLE MEMORY IN PIC16F87/88 DEVICES**

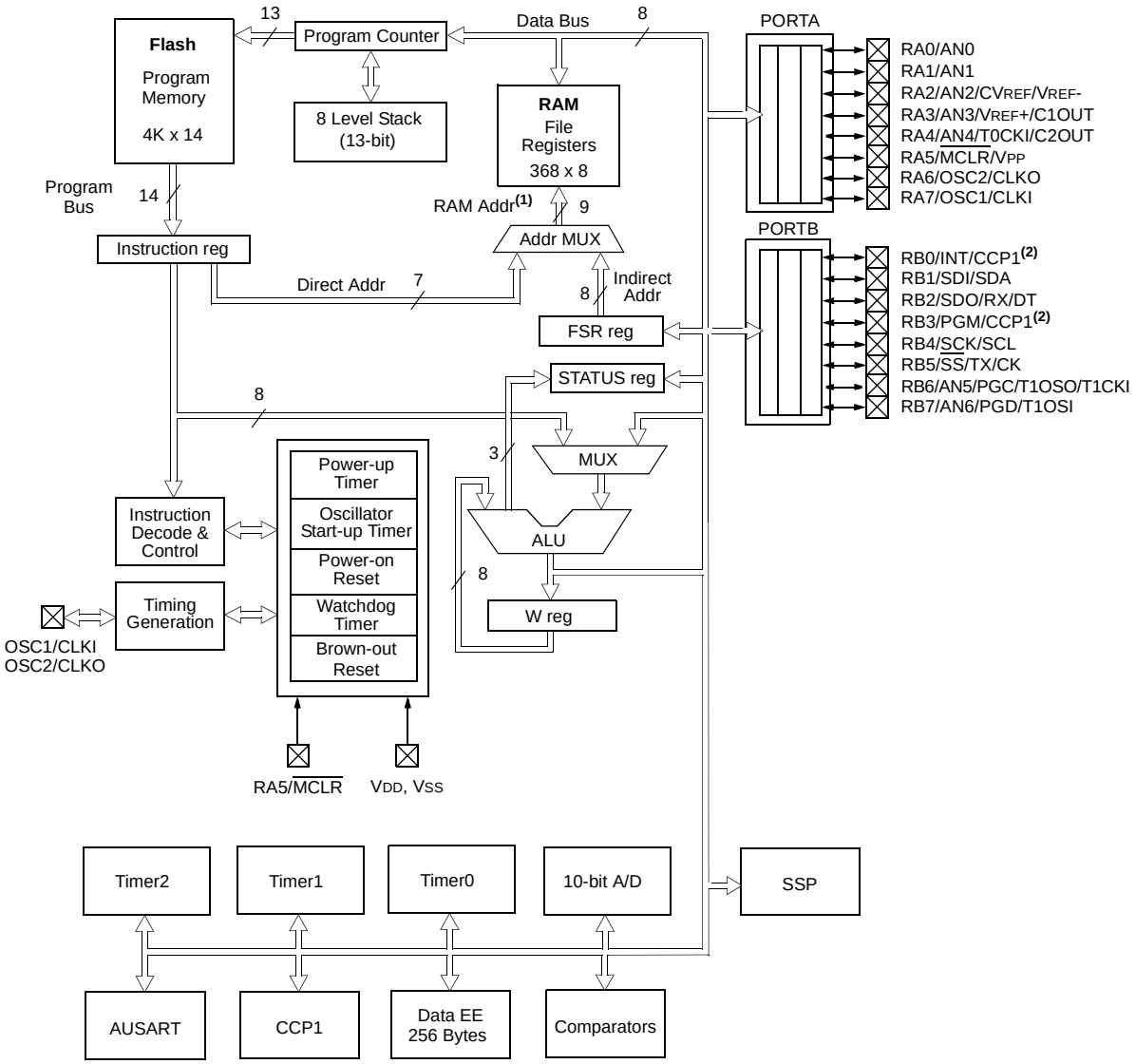
| Device | Program Flash | Data Memory | Data EEPROM |
|---|---|---|---|
| PIC16F87/88 | 4K x 14 | 368 x 8 | 256 x 8 |

There are 16 I/O pins that are user configurable on a pin-to-pin basis. Some pins are multiplexed with other device functions. These functions include:

- External Interrupt
- Change on PORTB Interrupt
- Timer0 Clock Input
- Low-Power Timer1 Clock/Oscillator
- Capture/Compare/PWM
- 10-bit, 7-channel A/D Converter (PIC16F88 only)
- SPI/$I^2C$™
- Two Analog Comparators
- AUSART
- $\overline{MCLR}$ (RA5) can be configured as an input

Table 1-2 details the pinout of the devices with descriptions and details for each pin.

**FIGURE 1-2: PIC16F88 DEVICE BLOCK DIAGRAM**




**Note 1:** Higher order bits are from the STATUS register.

**2:** The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

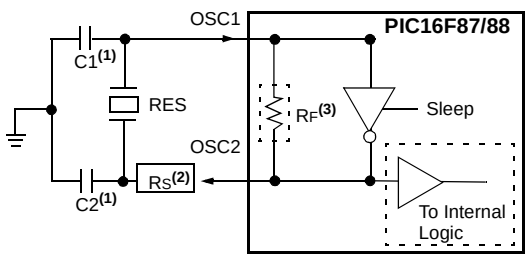
**TABLE 1-2: PIC16F87/88 PINOUT DESCRIPTION**

| Pin Name | PDIP/ SOIC Pin# | SSOP Pin# | QFN Pin# | I/O/P Type | Buffer Type | Description |
|---|---|---|---|---|---|---|
| | | | | | | PORTA is a bidirectional I/O port. |
| RA0/AN0 | 17 | 19 | 23 | | | |
| RA0 | | | | I/O | TTL | Bidirectional I/O pin. |
| AN0 | | | | I | Analog | Analog input channel 0. |
| RA1/AN1 | 18 | 20 | 24 | | | |
| RA1 | | | | I/O | TTL | Bidirectional I/O pin. |
| AN1 | | | | I | Analog | Analog input channel 1. |
| RA2/AN2/CVREF/VREF- | 1 | 1 | 26 | | | |
| RA2 | | | | I/O | TTL | Bidirectional I/O pin. |
| AN2 | | | | I | Analog | Analog input channel 2. |
| CVREF | | | | O | | Comparator VREF output. |
| VREF-(4) | | | | I | Analog | A/D reference voltage (Low) input. |
| RA3/AN3/VREF+/C1OUT | 2 | 2 | 27 | | | |
| RA3 | | | | I/O | TTL | Bidirectional I/O pin. |
| AN3 | | | | I | Analog | Analog input channel 3. |
| VREF+(4) | | | | I | Analog | A/D reference voltage (High) input. |
| C1OUT | | | | O | | Comparator 1 output. |
| RA4/AN4/T0CKI/C2OUT | 3 | 3 | 28 | | | |
| RA4 | | | | I/O | ST | Bidirectional I/O pin. |
| AN4(4) | | | | I | Analog | Analog input channel 4. |
| T0CKI | | | | I | ST | Clock input to the TMR0 timer/counter. |
| C2OUT | | | | O | | Comparator 2 output. |
| RA5/$\overline{\text{MCLR}}$/VPP | 4 | 4 | 1 | | | |
| RA5 | | | | I | ST | Input pin. |
| $\overline{\text{MCLR}}$ | | | | I | ST | Master Clear (Reset). Input/programming voltage input. This pin is an active-low Reset to the device. |
| VPP | | | | P | – | Programming voltage input. |
| RA6/OSC2/CLKO | 15 | 17 | 20 | | | |
| RA6 | | | | I/O | ST | Bidirectional I/O pin. |
| OSC2 | | | | O | – | Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. |
| CLKO | | | | O | – | In RC mode, this pin outputs CLKO signal which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. |
| RA7/OSC1/CLKI | 16 | 18 | 21 | | | |
| RA7 | | | | I/O | ST | Bidirectional I/O pin. |
| OSC1 | | | | I | ST/CMOS(3) | Oscillator crystal input. |
| CLKI | | | | I | – | External clock source input. |

**Legend:** I = Input O = Output I/O = Input/Output P = Power
– = Not used TTL = TTL Input ST = Schmitt Trigger Input

**Note 1:** This buffer is a Schmitt Trigger input when configured as the external interrupt.
**2:** This buffer is a Schmitt Trigger input when used in Serial Programming mode.
**3:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.
**4:** PIC16F88 devices only.
**5:** The CCP1 pin is determined by the CCPMX bit in Configuration Word 1 register.

**FIGURE 4-2: CERAMIC RESONATOR OPERATION (HS OR XT OSC CONFIGURATION)**




**Note 1:** See Table 4-2 for typical values of C1 and C2.

**2:** A series resistor ($R_S$) may be required.

**3:** $R_F$ varies with the resonator chosen (typically between 2 MΩ to 10 MΩ).

**TABLE 4-2: CERAMIC RESONATORS (FOR DESIGN GUIDANCE ONLY)**

| Typical Capacitor Values Used: | | | |
|---|---|---|---|
| **Mode** | **Freq** | **OSC1** | **OSC2** |
| XT | 455 kHz<br>2.0 MHz<br>4.0 MHz | 56 pF<br>47 pF<br>33 pF | 56 pF<br>47 pF<br>33 pF |
| HS | 8.0 MHz<br>16.0 MHz | 27 pF<br>22 pF | 27 pF<br>22 pF |

**Capacitor values are for design guidance only.**

These capacitors were tested with the resonators listed below for basic start-up and operation. These values were not optimized.

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected $V_{DD}$ and temperature range for the application.

See the notes following this table for additional information.

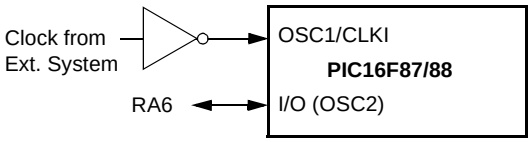
**Note:** When using resonators with frequencies above 3.5 MHz, the use of HS mode, rather than XT mode, is recommended. HS mode may be used at any $V_{DD}$ for which the controller is rated. If HS is selected, it is possible that the gain of the oscillator will overdrive the resonator. Therefore, a series resistor should be placed between the OSC2 pin and the resonator. As a good starting point, the recommended value of $R_S$ is 330Ω.

## 4.3 External Clock Input

The ECIO Oscillator mode requires an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset, or after an exit from Sleep mode.

In the ECIO Oscillator mode, the OSC2 pin becomes an additional general purpose I/O pin. The I/O pin becomes bit 6 of PORTA (RA6). Figure 4-3 shows the pin connections for the ECIO Oscillator mode.

**FIGURE 4-3: EXTERNAL CLOCK INPUT OPERATION (ECIO CONFIGURATION)**



## 5.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the "*PIC® Mid-Range MCU Family Reference Manual*" (DS33023).

### 5.1 PORTA and the TRISA Register

PORTA is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

| **Note:** | On a Power-on Reset, the pins PORTA<4:0> are configured as analog inputs and read as '0'. |
|---|---|

Reading the PORTA register, reads the status of the pins, whereas writing to it, will write to the port latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input. On PIC16F88 devices, it is also multiplexed with an analog input to become the RA4/AN4/T0CKI/C2OUT pin. The RA4/AN4/T0CKI/C2OUT pin is a Schmitt Trigger input and full CMOS output driver.

Pin RA5 is multiplexed with the Master Clear module input. The RA5/MCLR/VPP pin is a Schmitt Trigger input.

Pin RA6 is multiplexed with the oscillator module input and external oscillator output. Pin RA7 is multiplexed with the oscillator module input and external oscillator input. Pin RA6/OSC2/CLKO and pin RA7/OSC1/CLKI are Schmitt Trigger inputs and full CMOS output drivers.

Pins RA<1:0> are multiplexed with analog inputs. Pins RA<3:2> are multiplexed with analog inputs and comparator outputs. On PIC16F88 devices, pins RA<3:2> are also multiplexed with the VREF inputs. Pins RA<3:0> have TTL inputs and full CMOS output drivers.

**EXAMPLE 5-1: INITIALIZING PORTA**

```
BANKSEL PORTA     ; select bank of PORTA
CLRF    PORTA     ; Initialize PORTA by
                  ; clearing output
                  ; data latches
BANKSEL ANSEL     ; Select Bank of ANSEL
MOVLW   0x00      ; Configure all pins
MOVWF   ANSEL     ; as digital inputs

MOVLW   0xFF      ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISA     ; Set RA<7:0> as inputs
```

**TABLE 5-1: PORTA FUNCTIONS**

| Name | Bit# | Buffer | Function |
|---|---|---|---|
| RA0/AN0 | bit 0 | TTL | Input/output or analog input. |
| RA1/AN1 | bit 1 | TTL | Input/output or analog input. |
| RA2/AN2/CVREF/VREF-(2) | bit 2 | TTL | Input/output, analog input, VREF- or comparator VREF output. |
| RA3/AN3/VREF+(2)/C1OUT | bit 3 | TTL | Input/output, analog input, VREF+ or comparator output. |
| RA4/AN4(2)/T0CKI/C2OUT | bit 4 | ST | Input/output, analog input, TMR0 external input or comparator output. |
| RA5/$\overline{\text{MCLR}}$/VPP | bit 5 | ST | Input, Master Clear (Reset) or programming voltage input. |
| RA6/OSC2/CLKO | bit 6 | ST | Input/output, connects to crystal or resonator, oscillator output or 1/4 the frequency of OSC1 and denotes the instruction cycle in RC mode. |
| RA7/OSC1/CLKI | bit 7 | ST/CMOS(1) | Input/output, connects to crystal or resonator or oscillator input. |

**Legend:** TTL = TTL input, ST = Schmitt Trigger input

**Note 1:** This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

**2:** PIC16F88 only.

**REGISTER 10-2: SSPCON: SYNCHRONOUS SERIAL PORT CONTROL REGISTER (ADDRESS 14h)**

| 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 | SSPM2 | SSPM1 | SSPM0 |
| bit 7 | | | | | | | bit 0 |

bit 7 **WCOL:** Write Collision Detect bit

1 = An attempt to write the SSPBUF register failed because the SSP module is busy (must be cleared in software)

0 = No collision

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

In SPI mode:

1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

0 = No overflow

In $I^2C$ mode:

1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode. SSPOV must be cleared in software in either mode.

0 = No overflow

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

In SPI mode:

1 = Enables serial port and configures SCK, SDO and SDI as serial port pins

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

In $I^2C$ mode:

1 = Enables the serial port and configures the SDA and SCL pins as serial port pins

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

**Note 1:** In both modes, when enabled, these pins must be properly configured as input or output.

bit 4 **CKP:** Clock Polarity Select bit

In SPI mode:

1 = Transmit happens on falling edge, receive on rising edge. Idle state for clock is a high level.

0 = Transmit happens on rising edge, receive on falling edge. Idle state for clock is a low level.

In $I^2C$ Slave mode:

SCK release control

1 = Enable clock

0 = Holds clock low (clock stretch). (Used to ensure data setup time.)

bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits

0000 = SPI Master mode, clock = OSC/4

0001 = SPI Master mode, clock = OSC/16

0010 = SPI Master mode, clock = OSC/64

0011 = SPI Master mode, clock = TMR2 output/2

0100 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control enabled.

0101 = SPI Slave mode, clock = SCK pin. $\overline{SS}$ pin control disabled. $\overline{SS}$ can be used as I/O pin.

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

0111 = $I^2C$ Slave mode, 10-bit address

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

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

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

1000, 1001, 1010, 1100, 1101 = Reserved

| **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 |

An SSP interrupt is generated for each data transfer byte. Flag bit, SSPIF, must be cleared in software and the SSPSTAT register is used to determine the status of the byte. Flag bit, SSPIF, is set on the falling edge of the ninth clock pulse.

As a slave transmitter, the $\overline{ACK}$ pulse from the master receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not $\overline{ACK}$), then the data transfer is complete. When the $\overline{ACK}$ is latched by the slave device, the slave logic is reset (resets SSPSTAT register) and the slave device then monitors for another occurrence of the Start bit. If the SDA line was low ($\overline{ACK}$), the transmit data must be loaded into the SSPBUF register which also loads the SSPSR register. Then, pin RB4/SCK/SCL should be enabled by setting bit CKP.

**TABLE 10-2: DATA TRANSFER RECEIVED BYTE ACTIONS**

| Status Bits as Data Transfer is Received | | SSPSR → SSPBUF | Generate $\overline{ACK}$ Pulse | Set SSPIF Bit (SSP Interrupt Occurs if Enabled) |
|---|---|---|---|---|
| BF | SSPOV | | | |
| 0 | 0 | Yes | Yes | Yes |
| 1 | 0 | No | No | Yes |
| 1 | 1 | No | No | Yes |
| 0 | 1 | No | No | Yes |

**Note 1:** Shaded cells show the conditions where the user software did not properly clear the overflow condition.

**FIGURE 10-6: I²C™ WAVEFORMS FOR RECEPTION (7-BIT ADDRESS)**

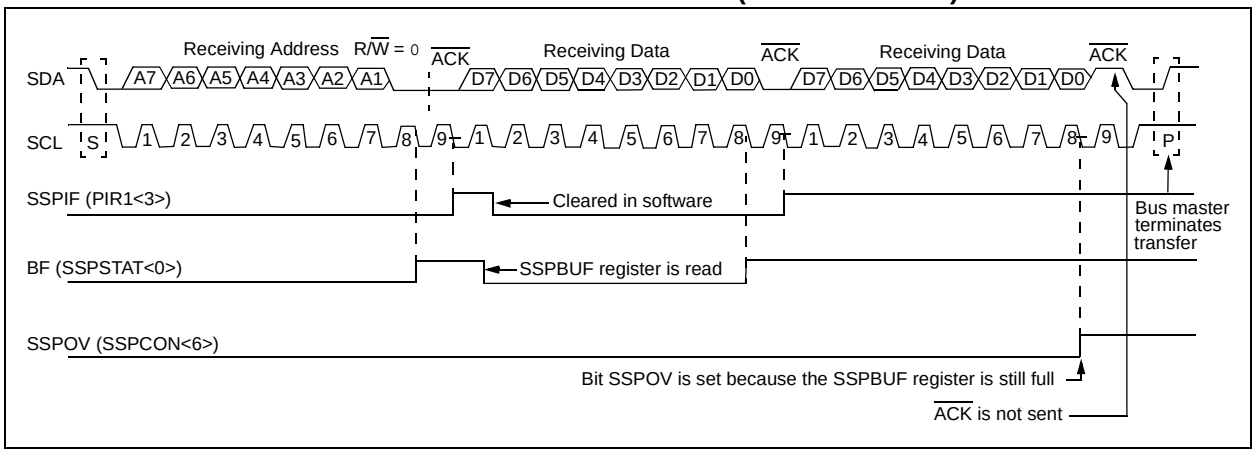



**FIGURE 10-7: I²C™ WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)**

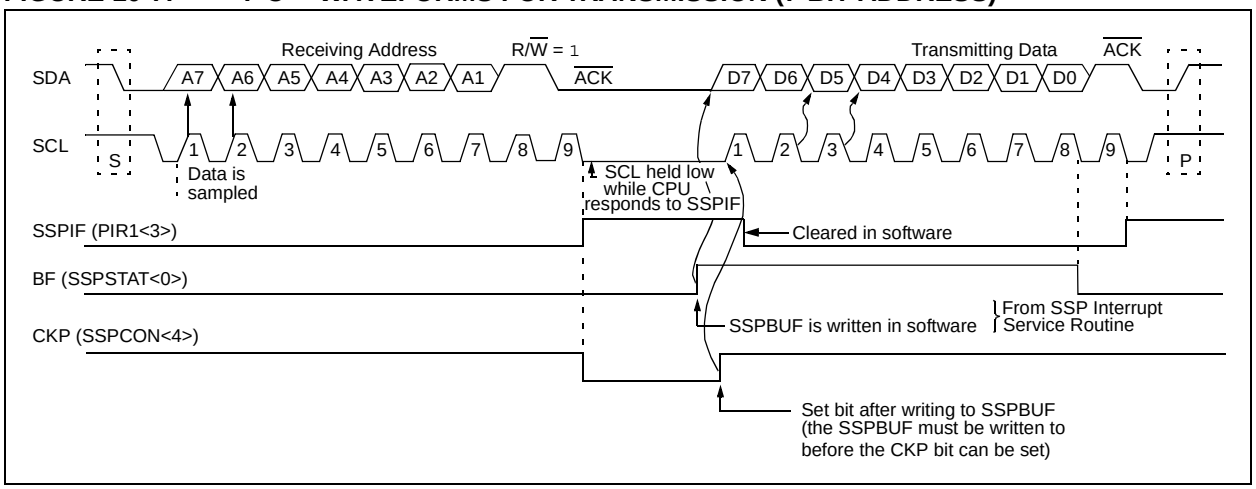


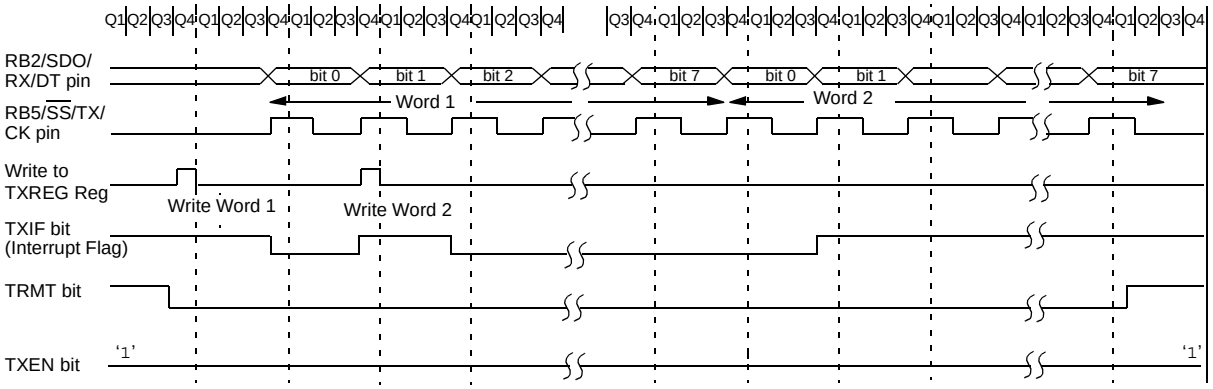
**TABLE 11-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh, 8Bh, 10Bh,18Bh | INTCON | GIE | PEIE | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF(1) | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 19h | TXREG | AUSART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | — | ADIE(1) | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

**Legend:** x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

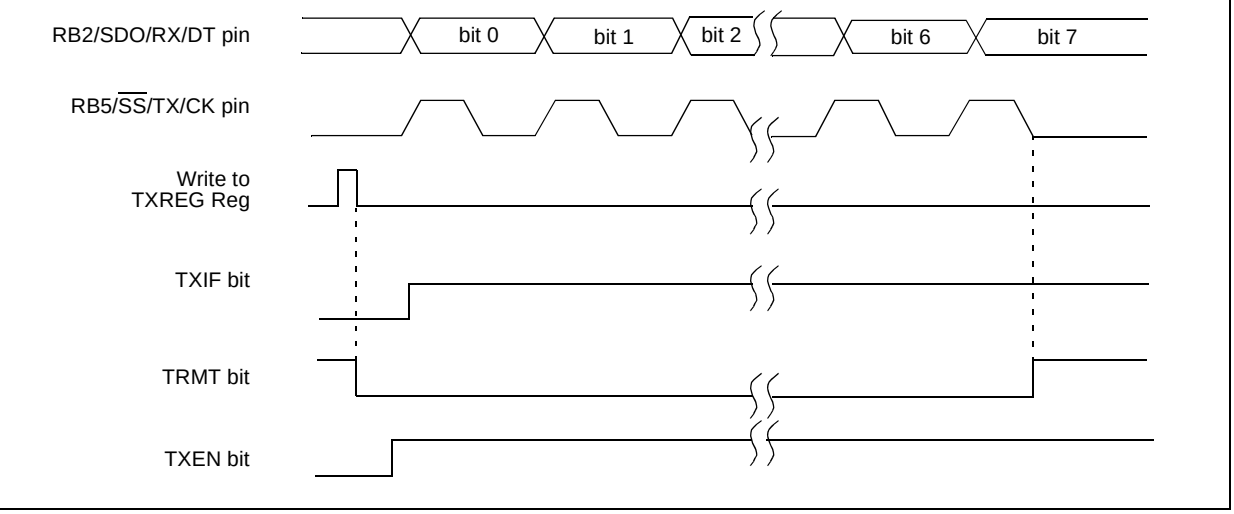
**Note 1:** This bit is only implemented on the PIC16F88. The bit will read '0' on the PIC16F87.

**FIGURE 11-9: SYNCHRONOUS TRANSMISSION**




**Note:** Sync Master mode; SPBRG = 0. Continuous transmission of two 8-bit words.

**FIGURE 11-10: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**



### 11.4.2 AUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of the Sleep mode. Bit SREN is a “don’t care” in Slave mode.

If receive is enabled by setting bit CREN prior to the SLEEP instruction, then a word may be received during Sleep. On completely receiving the word, the RSR register will transfer the data to the RCREG register and if enable bit RCIE bit is set, the interrupt generated will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

When setting up a synchronous slave reception, follow these steps:

1. Enable the synchronous master serial port by setting bits SYNC and SPEN and clearing bit CSRC.
2. If interrupts are desired, set enable bit RCIE.
3. If 9-bit reception is desired, set bit RX9.
4. To enable reception, set enable bit CREN.
5. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
6. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG register.
8. If any error occurred, clear the error by clearing bit CREN.
9. If using interrupts, ensure that GIE and PEIE (bits 7 and 6) of the INTCON register are set.

**TABLE 11-13: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Bh, 8Bh, 10Bh,18Bh | INTCON | GIE | PEIE | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | — | ADIF(1) | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | -000 0000 | -000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D | 0000 000x | 0000 000x |
| 1Ah | RCREG | AUSART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | — | ADIE(1) | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | -000 0000 | -000 0000 |
| 98h | TXSTA | CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D | 0000 -010 | 0000 -010 |
| 99h | SPBRG | Baud Rate Generator Register | | | | | | | | 0000 0000 | 0000 0000 |

**Legend:** x = unknown, - = unimplemented, read as ‘0’. Shaded cells are not used for synchronous slave reception.

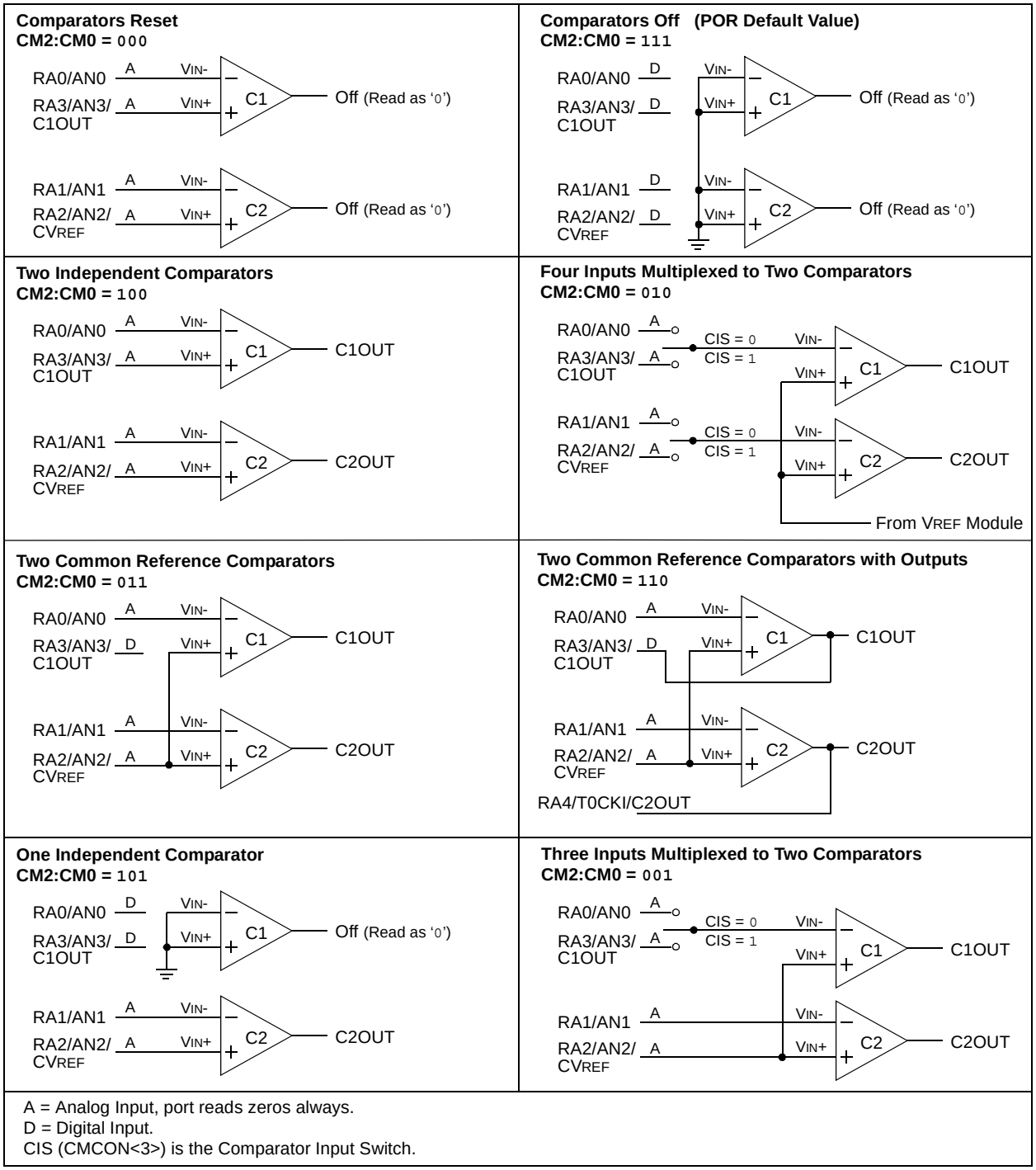
**Note 1:** This bit is only implemented on the PIC16F88. The bit will read ‘0’ on the PIC16F87.

## 13.1 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select these modes. Figure 13-1 shows the eight possible modes. The TRISA register controls the data direction of the comparator pins for each mode. If the Comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in **Section 18.0 "Electrical Characteristics"**.

| **Note:** | Comparator interrupts should be disabled during a Comparator mode change; otherwise, a false interrupt may occur. |
|---|---|

**FIGURE 13-1: COMPARATOR I/O OPERATING MODES**




A = Analog Input, port reads zeros always.
D = Digital Input.
CIS (CMCON<3>) is the Comparator Input Switch.

## 15.2 Reset

The PIC16F87/88 differentiates between various kinds of Reset:

- Power-on Reset (POR)
- $\overline{MCLR}$ Reset during normal operation
- $\overline{MCLR}$ Reset during Sleep
- WDT Reset during normal operation
- WDT wake-up during Sleep
- Brown-out Reset (BOR)

Some registers are not affected in any Reset condition. Their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on Power-on Reset (POR), on the $\overline{MCLR}$ and WDT Reset, on $\overline{MCLR}$ Reset during Sleep and Brown-out Reset (BOR). They are not affected by a WDT wake-up which is viewed as the resumption of normal operation. The $\overline{TO}$ and $\overline{PD}$ bits are set or cleared differently in different Reset situations, as indicated in Table 15-3. These bits are used in software to determine the nature of the Reset. Upon a POR, BOR or wake-up from Sleep, the CPU requires approximately 5-10 μs to become ready for code execution. This delay runs in parallel with any other timers. See Table 15-4 for a full description of Reset states of all registers.

A simplified block diagram of the On-Chip Reset Circuit is shown in Figure 15-1.

**FIGURE 15-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**

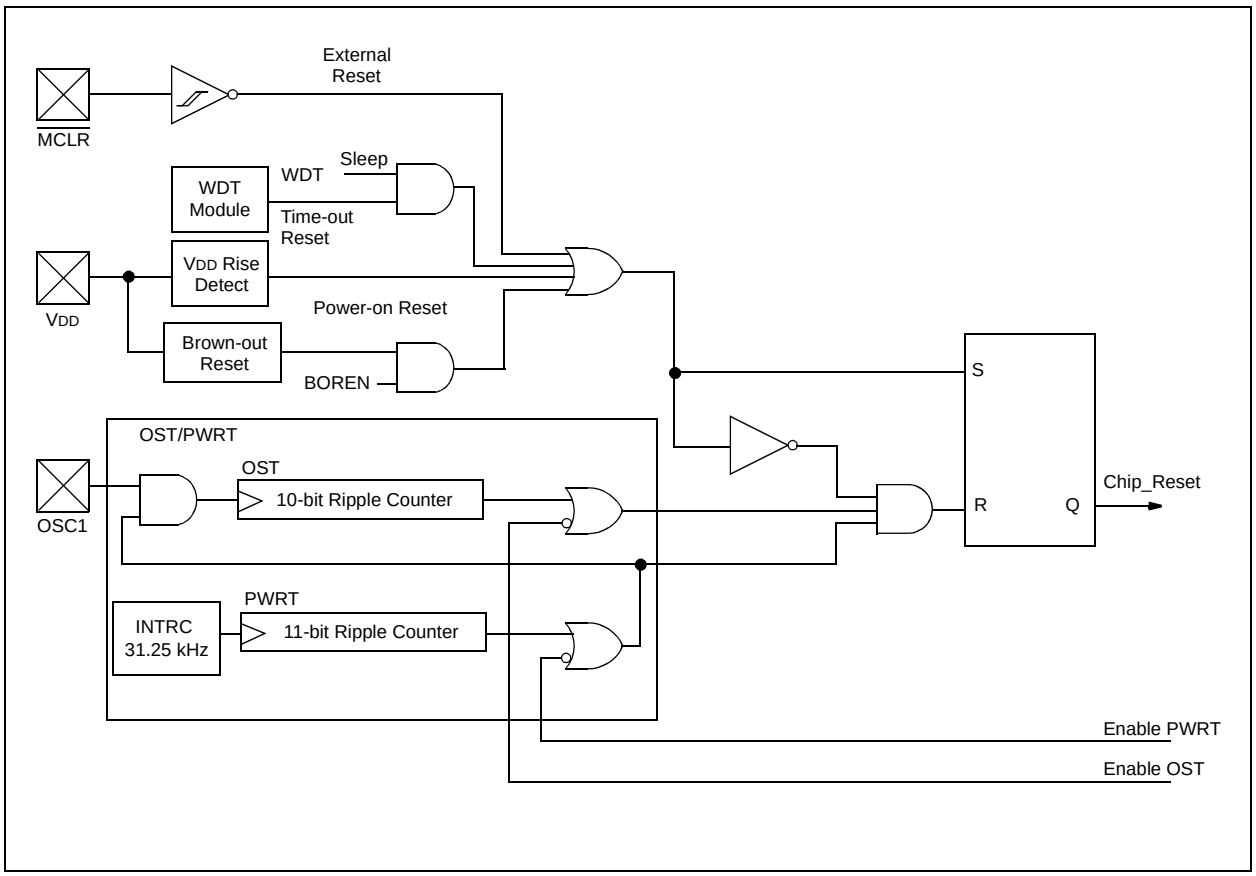


**TABLE 15-3: RESET CONDITION FOR SPECIAL REGISTERS**

| Condition | Program Counter | STATUS Register | PCON Register |
|---|---|---|---|
| Power-on Reset | 000h | 0001 1xxx | ---- --0x |
| $\overline{\text{MCLR}}$ Reset during normal operation | 000h | 000u uuuu | ---- --uu |
| $\overline{\text{MCLR}}$ Reset during Sleep | 000h | 0001 0uuu | ---- --uu |
| WDT Reset | 000h | 0000 1uuu | ---- --uu |
| WDT Wake-up | PC + 1 | uuu0 0uuu | ---- --uu |
| Brown-out Reset | 000h | 0001 1uuu | ---- --u0 |
| Interrupt Wake-up from Sleep | PC + 1(1) | uuu1 0uuu | ---- --uu |

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, read as '0'

**Note 1:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**TABLE 15-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS**

| Register | Power-on Reset, Brown-out Reset | $\overline{\text{MCLR}}$ Reset, WDT Reset | Wake-up via WDT or Interrupt |
|---|---|---|---|
| W | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| INDF | N/A | N/A | N/A |
| TMR0 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PCL | 0000h | 0000h | PC + 1(2) |
| STATUS | 0001 1xxx | 000q quuu(3) | uuuq quuu(3) |
| FSR | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| PORTA (PIC16F87)<br>PORTA (PIC16F88) | xxxx 0000<br>xxx0 0000 | uuuu 0000<br>uuu0 0000 | uuuu uuuu<br>uuuu uuuu |
| PORTB (PIC16F87)<br>PORTB (PIC16F87) | xxxx xxxx<br>00xx xxxx | uuuu uuuu<br>00uu uuuu | uuuu uuuu<br>uuuu uuuu |
| PCLATH | ---0 0000 | ---0 0000 | ---u uuuu |
| INTCON | 0000 000x | 0000 000u | uuuu uuuu(1) |
| PIR1 | -000 0000 | -000 0000 | -uuu uuuu(1) |
| PIR2 | 00-0 ---- | 00-0 ---- | uu-u ----(1) |
| TMR1L | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| TMR1H | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| T1CON | -000 0000 | -uuu uuuu | -uuu uuuu |
| TMR2 | 0000 0000 | 0000 0000 | uuuu uuuu |
| T2CON | -000 0000 | -000 0000 | -uuu uuuu |
| SSPBUF | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| SSPCON | 0000 0000 | 0000 0000 | uuuu uuuu |
| CCPR1L | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCPR1H | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| CCP1CON | --00 0000 | --00 0000 | --uu uuuu |
| RCSTA | 0000 000x | 0000 000x | uuuu uuuu |

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition

**Note 1:** One or more bits in INTCON, PIR1 and PR2 will be affected (to cause wake-up).

**2:** When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

**3:** See Table 15-3 for Reset value for specific condition.

## 15.12 Watchdog Timer (WDT)

For PIC16F87/88 devices, the WDT has been modified from previous PIC16 devices. The new WDT is code and functionally backward compatible with previous PIC16 WDT modules and allows the user to have a scaler value for the WDT and TMR0 at the same time. In addition, the WDT time-out value can be extended to 268 seconds, using the prescaler with the postscaler when PSA is set to '1'.

### 15.12.1 WDT OSCILLATOR

The WDT derives its time base from the 31.25 kHz INTRC. The value of WDTCON is '---0 1000' on all Resets. This gives a nominal time base of 16.38 ms, which is compatible with the time base generated with previous PIC16 microcontroller versions.

| **Note:** | When the OST is invoked, the WDT is held in Reset because the WDT ripple counter is used by the OST to perform the oscillator delay count. When the OST count has expired, the WDT will begin counting (if enabled). |
|---|---|

A new prescaler has been added to the path between the internal RC and the multiplexors used to select the path for the WDT. This prescaler is 16 bits and can be programmed to divide the internal RC by 32 to 65536, giving the time base used for the WDT a nominal range of 1 ms to 2.097s.

### 15.12.2 WDT CONTROL

The WDTEN bit is located in Configuration Word 1 and when this bit is set, the WDT runs continuously.

The SWDTEN bit is in the WDTCON register. When the WDTEN bit in the Configuration Word 1 register is set, the SWDTEN bit has no effect. If WDTEN is clear, then the SWDTEN bit can be used to enable and disable the WDT. Setting the bit will enable it and clearing the bit will disable it.

The PSA and PS<2:0> bits (OPTION_REG register) have the same function as in previous versions of the PIC16 family of microcontrollers.

**FIGURE 15-8: WATCHDOG TIMER BLOCK DIAGRAM**

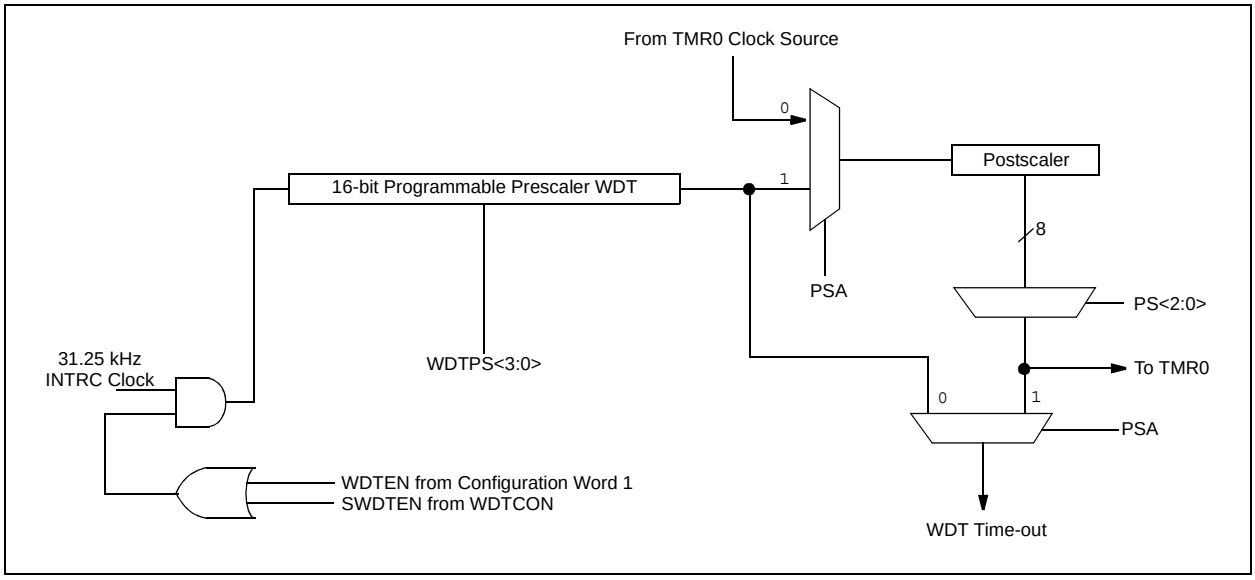


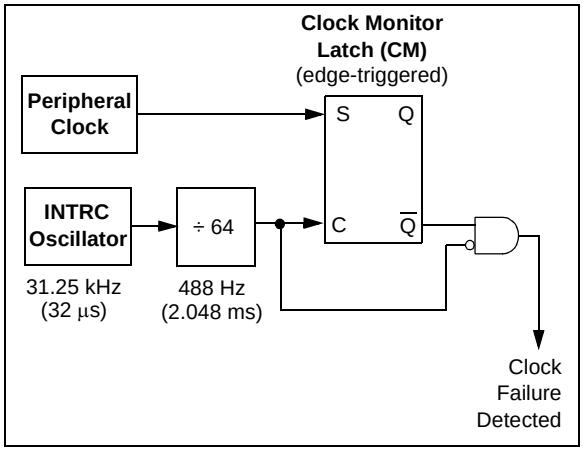

**TABLE 15-5: PRESCALER/POSTSCALER BIT STATUS**

| Conditions | Prescaler | Postscaler (PSA = 1) |
|---|---|---|
| WDTEN = 0 | Cleared | Cleared |
| CLRWDT command | | |
| Oscillator fail detected | | |
| Exit Sleep + System Clock = T1OSC, EXTRC, INTRC, ECIO | | |
| Exit Sleep + System Clock = XT, HS, LP | Cleared at end of OST | Cleared at end of OST |

### 15.12.4 FAIL-SAFE OPTION

The Fail-Safe Clock Monitor (FSCM) is designed to allow the device to continue to operate even in the event of an oscillator failure.

**FIGURE 15-10: FSCM BLOCK DIAGRAM**




The FSCM function is enabled by setting the FCMEN bit in Configuration Word 2.

In the event of an oscillator failure, the FSCM will generate an oscillator fail interrupt and will switch the system clock over to the internal oscillator. The system will continue to come from the internal oscillator until the fail-safe condition is exited. The fail-safe condition is exited with either a Reset, the execution of a SLEEP instruction or a write to the OSCCON register.

The frequency of the internal oscillator will depend upon the value contained in the IRCF bits. Another clock source can be selected via the IRCF and the SCS bits of the OSCCON register.

The FSCM sample clock is generated by dividing the INTRC clock by 64. This will allow enough time between FSCM sample clocks for a system clock edge to occur.

On the rising edge of the postscaled clock, the monitoring latch (CM = 0) will be cleared. On a falling edge of the primary or secondary system clock, the monitoring latch will be set (CM = 1). In the event that a falling edge of the postscaled clock occurs and the monitoring latch is not set, a clock failure has been detected.

While in Fail-Safe mode, a Reset will exit the fail-safe condition. If the primary clock source is configured for a crystal, the OST timer will wait for the 1024 clock cycles for the OST time-out and the device will continue running from the internal oscillator until the OST is complete. A SLEEP instruction, or a write to the SCS bits (where SCS bits do not = 00), can be performed to put the device into a low-power mode.

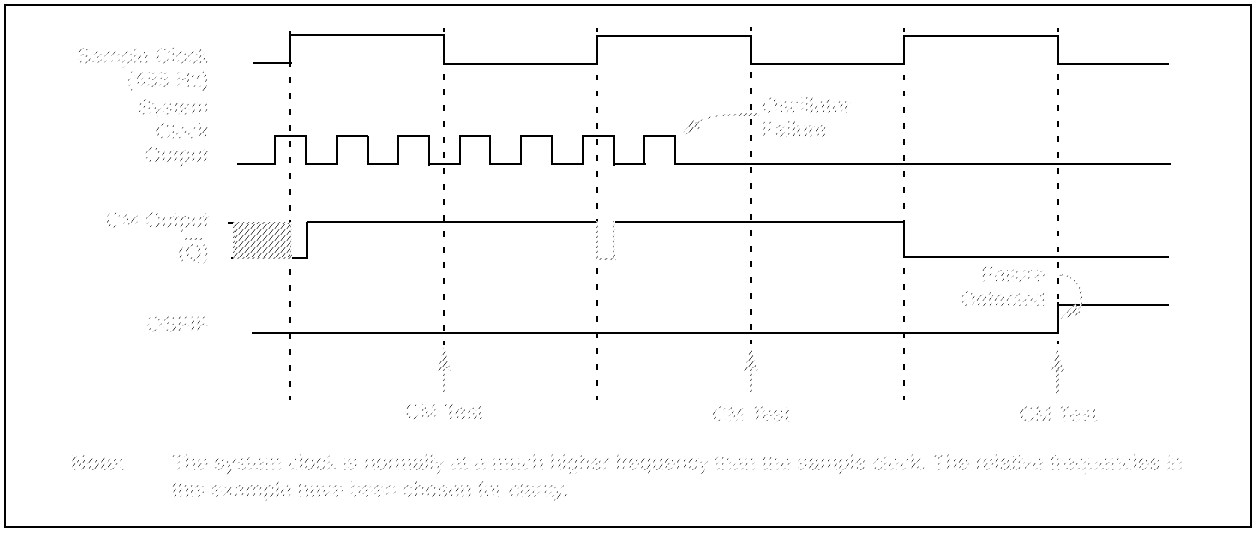
| **Note:** | Two-Speed Start-up mode is automatically enabled when the fail-safe option is enabled. |
|---|---|

If Reset occurs while in Fail-Safe mode and the primary clock source is EC or RC, then the device will immediately switch back to EC or RC mode.

#### 15.12.4.1 Fail-Safe in Low-Power Mode

A write to the OSCCON register, or SLEEP instruction, will end the fail-safe condition. The system clock will default to the source selected by the SCS bits, which is either T1OSC, INTRC or none (Sleep mode). However, the FSCM will continue to monitor the system clock. If the secondary clock fails, the device will immediately switch to the internal oscillator clock. If OSFIE is set, an interrupt will be generated.

**FIGURE 15-11: FSCM TIMING DIAGRAM**



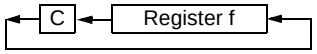

Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

| RETFIE | Return from Interrupt |
|---|---|
| Syntax: | [ *label* ] RETFIE |
| Operands: | None |
| Operation: | TOS → PC,<br>1 → GIE |
| Status Affected: | None |

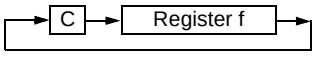
| RETLW | Return with Literal in W |
|---|---|
| Syntax: | [ *label* ] RETLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | k → (W);<br>TOS → PC |
| Status Affected: | None |
| Description: | The W register is loaded with the eight-bit literal ‘k’. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction. |

| RETURN | Return from Subroutine |
|---|---|
| Syntax: | [ *label* ] RETURN |
| Operands: | None |
| Operation: | TOS → PC |
| Status Affected: | None |
| Description: | Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction. |

| RLF | Rotate Left f through Carry |
|---|---|
| Syntax: | [ *label* ] RLF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | See description below |
| Status Affected: | C |
| Description: | The contents of register ‘f’ are rotated one bit to the left through the Carry flag. If ‘d’ = 0, the result is placed in the W register. If ‘d’ = 1, the result is stored back in register ‘f’. |




| RRF | Rotate Right f through Carry |
|---|---|
| Syntax: | [ *label* ] RRF f,d |
| Operands: | 0 ≤ f ≤ 127<br>d ∈ [0,1] |
| Operation: | See description below |
| Status Affected: | C |
| Description: | The contents of register ‘f’ are rotated one bit to the right through the Carry flag. If ‘d’ = 0, the result is placed in the W register. If ‘d’ = 1, the result is placed back in register ‘f’. |




| SLEEP | Sleep |
|---|---|
| Syntax: | [ *label* ] SLEEP |
| Operands: | None |
| Operation: | 00h → WDT,<br>0 → WDT prescaler,<br>1 → $\overline{TO}$,<br>0 → $\overline{PD}$ |
| Status Affected: | $\overline{TO}$, $\overline{PD}$ |
| Description: | The Power-Down status bit, $\overline{PD}$, is cleared. Time-out status bit, $\overline{TO}$, is set. Watchdog Timer and its prescaler are cleared.<br>The processor is put into Sleep mode with the oscillator stopped. |

**NOTES:**

**FIGURE 19-17: IPD A/D, -40°C TO +125°C, SLEEP MODE, A/D ENABLED (NOT CONVERTING)**

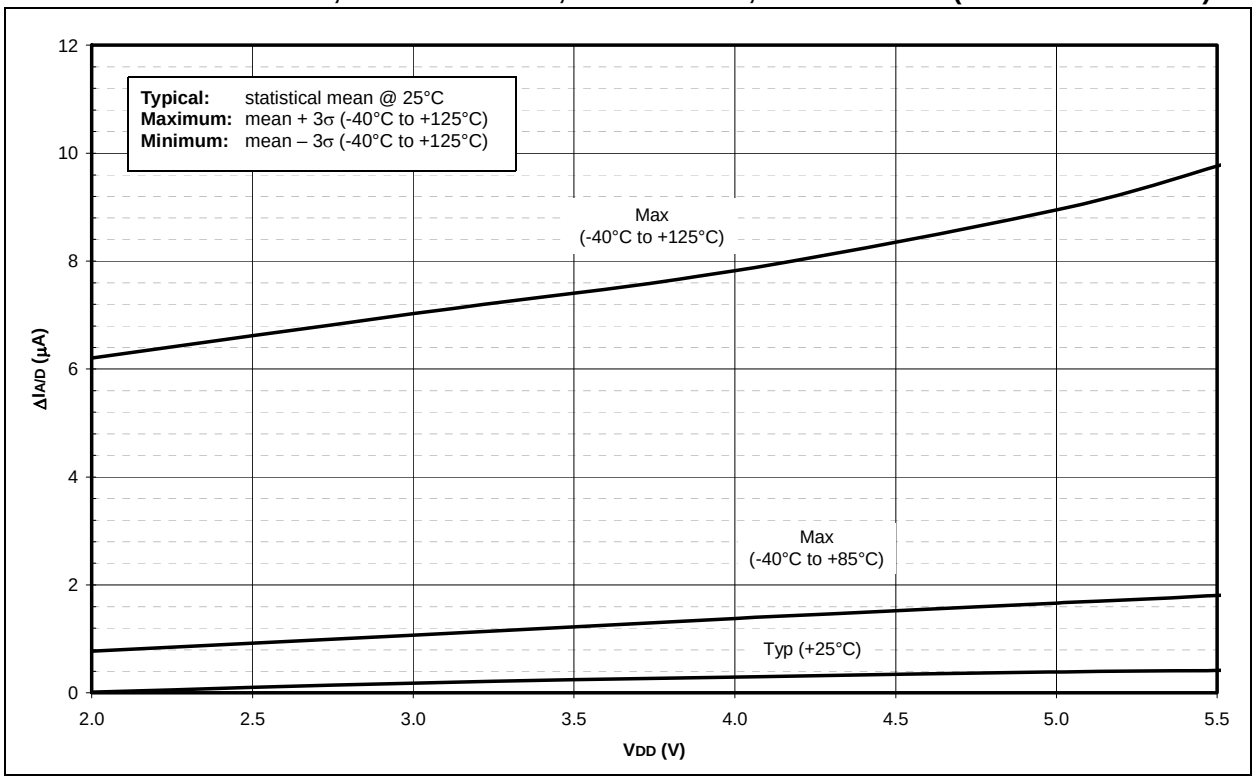



**FIGURE 19-18: TYPICAL, MINIMUM AND MAXIMUM VOH vs. IOH (VDD = 5V, -40°C TO +125°C)**

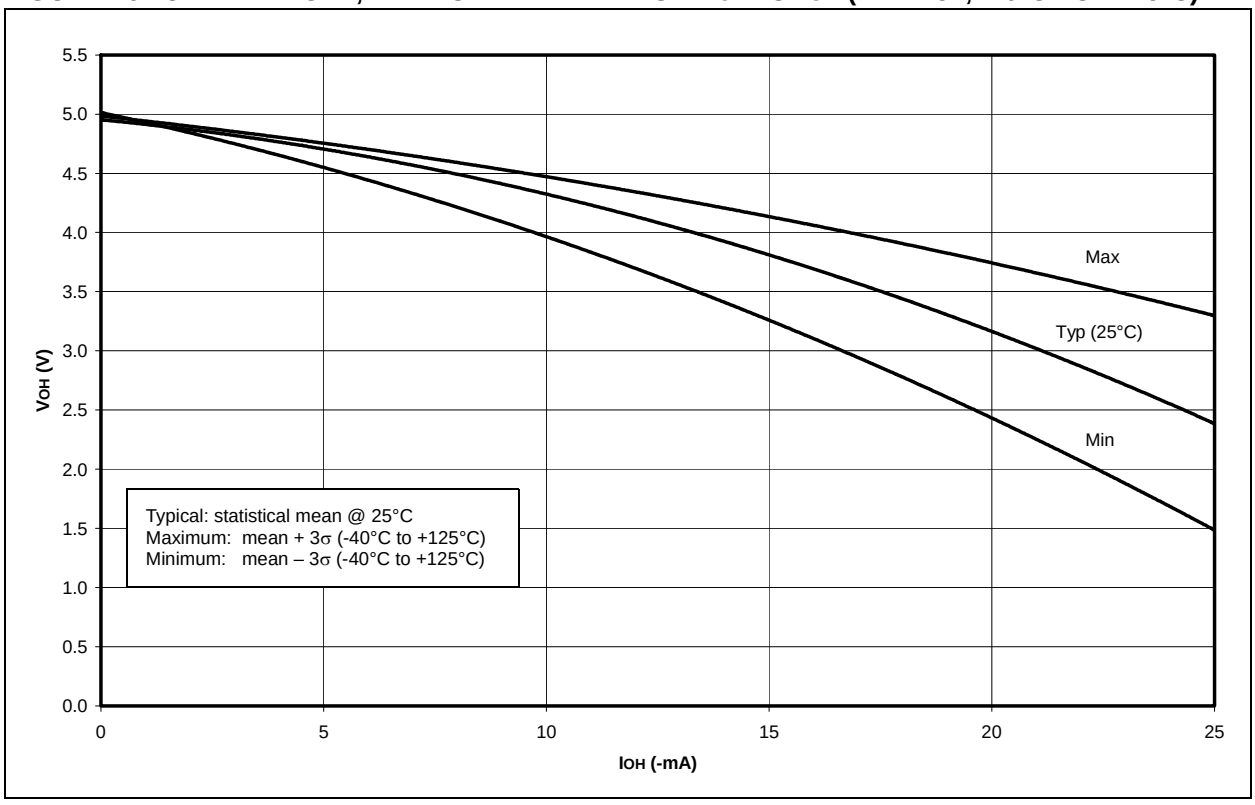


**FIGURE 19-21: TYPICAL, MINIMUM AND MAXIMUM $V_{OL}$ vs. $I_{OL}$ ($V_{DD}$ = 3V, -40°C TO +125°C)**

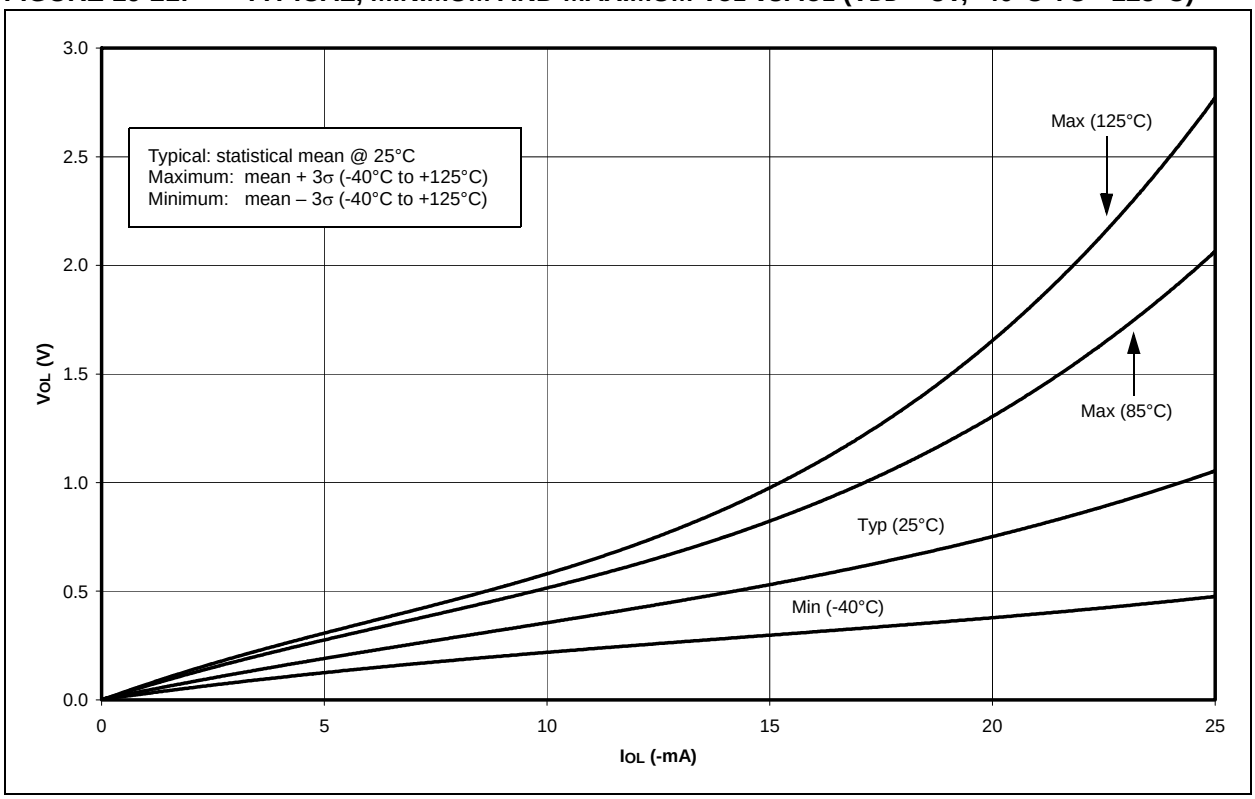
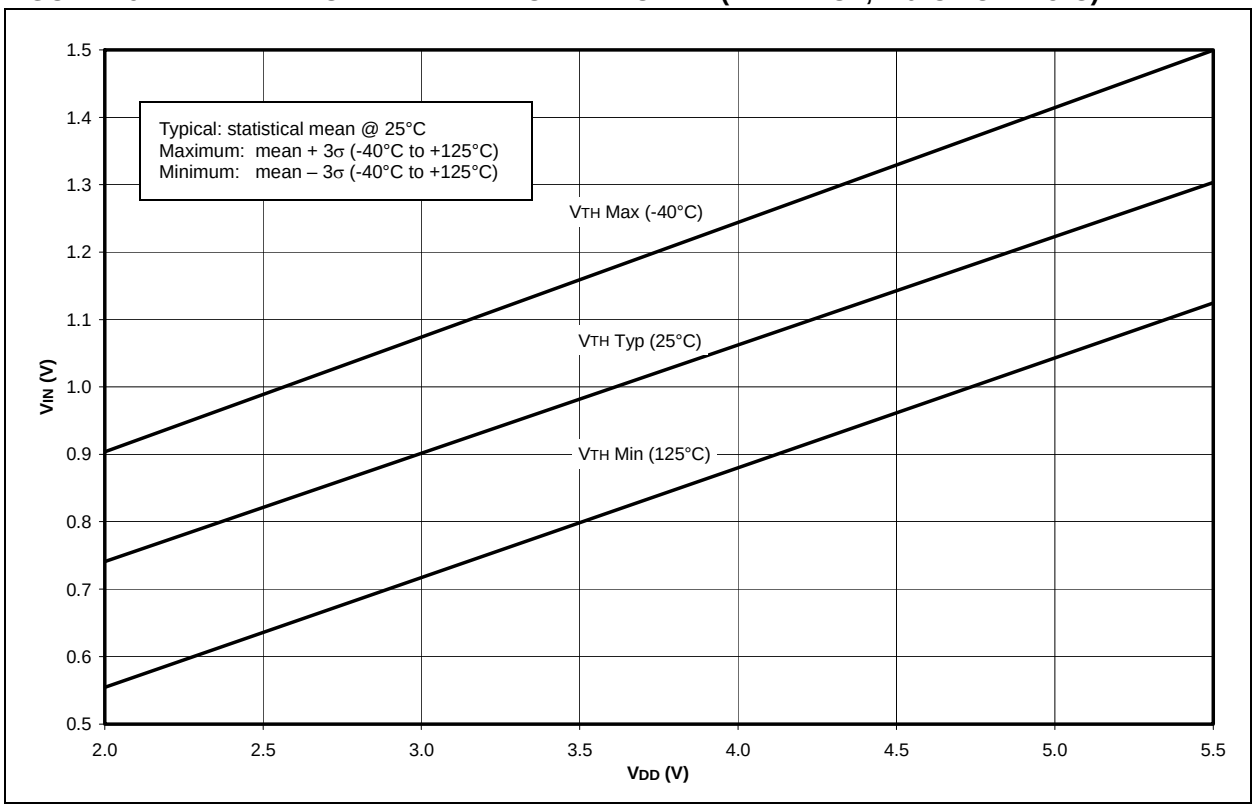


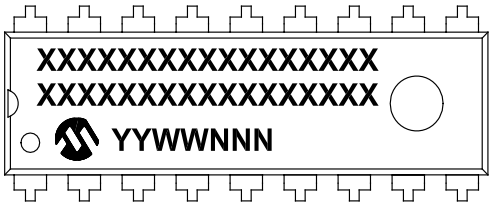

**FIGURE 19-22: MINIMUM AND MAXIMUM $V_{IN}$ vs. $V_{DD}$ (TTL INPUT, -40°C TO +125°C)**



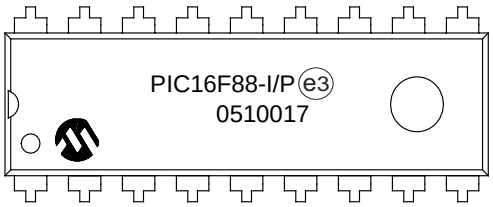
## 20.0 PACKAGING INFORMATION

### 20.1 Package Marking Information

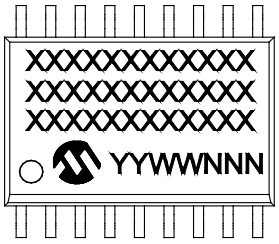
18-Lead PDIP (300 mil)



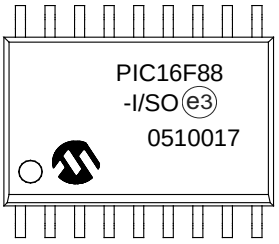

Example



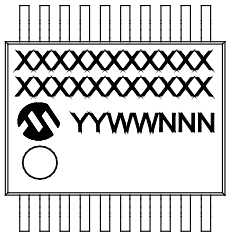

18-Lead SOIC (7.50 mm)



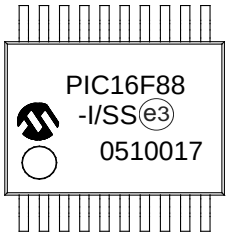

Example



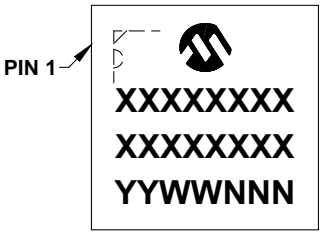

20-Lead SSOP (5.30 mm)



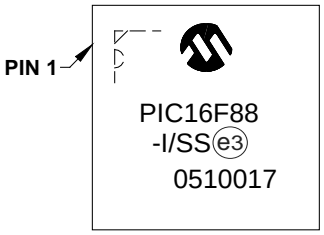

Example




28-Lead QFN (6x6 mm)




Example




| | | |
|---|---|---|
| **Legend:** | XX...X | Customer-specific information |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week '01') |
| | NNN | Alphanumeric traceability code |
| | (e3) | Pb-free JEDEC designator for Matte Tin (Sn) |
| | * | This package is Pb-free. The Pb-free JEDEC designator ( e3 ) can be found on the outer packaging for this package. |

**Note**: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.