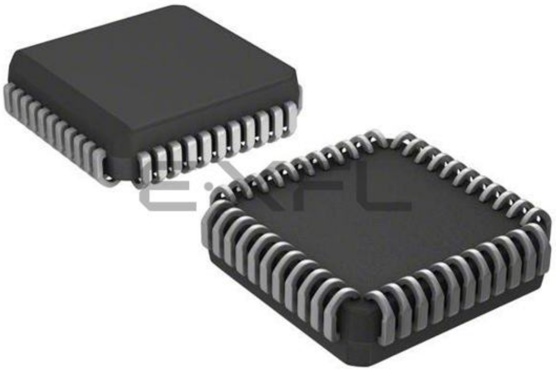





## What is "Embedded - Microcontrollers"?

"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 | 4MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 33 |
| Program Memory Size | 7KB (4K x 14) |
| Program Memory Type | OTP |
| EEPROM Size | - |
| RAM Size | 192 x 8 |
| Voltage - Supply (Vcc/Vdd) | 4V ~ 6V |
| Data Converters | - |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-LCC (J-Lead) |
| Supplier Device Package | 44-PLCC (16.59x16.59) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16c65a-04i-l |

**NOTES:**

**TABLE 4-6: SPECIAL FUNCTION REGISTERS FOR THE PIC16C66/67**

| 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[3] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0** | | | | | | | | | | | |
| 00h[1] | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | 0000 0000 | 0000 0000 |
| 01h | TMR0 | Timer0 module's register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 02h[1] | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 0000 0000 |
| 03h[1] | STATUS | IRP | RP1 | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 000q quuu |
| 04h[1] | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | uuuu uuuu |
| 05h | PORTA | — | — | PORTA Data Latch when written: PORTA pins when read | | | | | | --xx xxxx | --uu uuuu |
| 06h | PORTB | PORTB Data Latch when written: PORTB pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 07h | PORTC | PORTC Data Latch when written: PORTC pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 08h[5] | PORTD | PORTD Data Latch when written: PORTD pins when read | | | | | | | | xxxx xxxx | uuuu uuuu |
| 09h[5] | PORTE | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 0Ah[1,2] | PCLATH | — | — | — | Write Buffer for the upper 5 bits of the Program Counter | | | | | ---0 0000 | ---0 0000 |
| 0Bh[1] | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF[6] | (4) | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 0Dh | PIR2 | — | — | — | — | — | — | — | CCP2IF | ---- ---0 | ---- ---0 |
| 0Eh | TMR1L | Holding register for the Least Significant Byte of the 16-bit TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 0Fh | TMR1H | Holding register for the Most Significant Byte of the 16-bit TMR1 register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 10h | T1CON | — | — | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{T1SYNC}$ | TMR1CS | TMR1ON | --00 0000 | --uu uuuu |
| 11h | TMR2 | Timer2 module's register | | | | | | | | 0000 0000 | 0000 0000 |
| 12h | T2CON | — | TOUTPS3 | TOUTPS2 | TOUTPS1 | TOUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | -000 0000 | -000 0000 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 15h | CCPR1L | Capture/Compare/PWM1 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 16h | CCPR1H | Capture/Compare/PWM1 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 17h | CCP1CON | — | — | CCP1X | CCP1Y | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000 | --00 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 19h | TXREG | USART Transmit Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Ah | RCREG | USART Receive Data Register | | | | | | | | 0000 0000 | 0000 0000 |
| 1Bh | CCPR2L | Capture/Compare/PWM2 (LSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Ch | CCPR2H | Capture/Compare/PWM2 (MSB) | | | | | | | | xxxx xxxx | uuuu uuuu |
| 1Dh | CCP2CON | — | — | CCP2X | CCP2Y | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | --00 0000 | --00 0000 |
| 1Eh-1Fh | — | Unimplemented | | | | | | | | — | — |

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented location read as '0'.
Shaded locations are unimplemented, read as '0'.

Note 1: These registers can be addressed from any bank.
2: The upper byte of the Program Counter (PC) is not directly accessible. PCLATH is a holding register for the PC whose contents are transferred to the upper byte of the program counter. (PC<12:8>)
3: Other (non power-up) resets include external reset through $\overline{MCLR}$ and the Watchdog Timer reset.
4: PIE1<6> and PIR1<6> are reserved on the PIC16C66/67, always maintain these bits clear.
5: PORTD, PORTE, TRISD, and TRISE are not implemented on the PIC16C66, read as '0'.
6: PSPIF (PIR1<7>) and PSPIE (PIE1<7>) are reserved on the PIC16C66, maintain these bits clear.

## 5.3 PORTC and TRISC Register

| Applicable Devices | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 | |

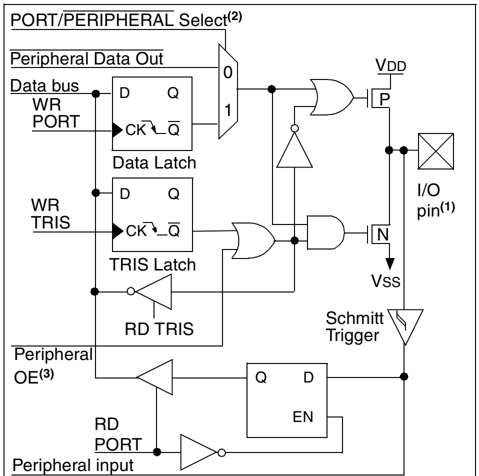
PORTC is an 8-bit wide bi-directional port. Each pin is individually configurable as an input or output through the TRISC register. PORTC is multiplexed with several peripheral functions (Table 5-5). PORTC pins have Schmitt Trigger input buffers.

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (`BSF, BCF, XORWF`) with TRISC as destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

### EXAMPLE 5-3: INITIALIZING PORTC

```
BCF    STATUS, RP0  ;
BCF    STATUS, RP1  ; PIC16C66/67 only
CLRF   PORTC        ; Initialize PORTC by
                    ; clearing output
                    ; data latches
BSF    STATUS, RP0  ; Select Bank 1
MOVLW  0xCF         ; Value used to
                    ; initialize data
                    ; direction
MOVWF  TRISC        ; Set RC<3:0> as inputs
                    ; RC<5:4> as outputs
                    ; RC<7:6> as inputs
```

### FIGURE 5-6: PORTC BLOCK DIAGRAM




Note 1: I/O pins have diode protection to VDD and VSS.
2: Port/Peripheral select signal selects between port data and peripheral output.
3: Peripheral OE (output enable) is only activated if peripheral select is active.

### TABLE 5-5: PORTC FUNCTIONS FOR PIC16C62/64

| Name | Bit# | Buffer Type | Function |
|---|---|---|---|
| RC0/T1OSI/T1CKI | bit0 | ST | Input/output port pin or Timer1 oscillator input or Timer1 clock input |
| RC1/T1OSO | bit1 | ST | Input/output port pin or Timer1 oscillator output |
| RC2/CCP1 | bit2 | ST | Input/output port pin or Capture1 input/Compare1 output/PWM1 output |
| RC3/SCK/SCL | bit3 | ST | RC3 can also be the synchronous serial clock for both SPI and $I^2C$ modes. |
| RC4/SDI/SDA | bit4 | ST | RC4 can also be the SPI Data In (SPI mode) or data I/O ($I^2C$ mode). |
| RC5/SDO | bit5 | ST | Input/output port pin or synchronous serial port data output |
| RC6 | bit6 | ST | Input/output port pin |
| RC7 | bit7 | ST | Input/output port pin |

Legend: ST = Schmitt Trigger input

**TABLE 5-6: PORTC FUNCTIONS FOR PIC16C62A/R62/64A/R64**

| Name | Bit# | Buffer Type | Function |
|---|---|---|---|
| RC0/T1OSO/T1CKI | bit0 | ST | Input/output port pin or Timer1 oscillator output or Timer1 clock input |
| RC1/T1OSI | bit1 | ST | Input/output port pin or Timer1 oscillator input |
| RC2/CCP1 | bit2 | ST | Input/output port pin or Capture input/Compare output/PWM1 output |
| RC3/SCK/SCL | bit3 | ST | RC3 can also be the synchronous serial clock for both SPI and $I^2C$ modes. |
| RC4/SDI/SDA | bit4 | ST | RC4 can also be the SPI Data In (SPI mode) or data I/O ($I^2C$ mode). |
| RC5/SDO | bit5 | ST | Input/output port pin or synchronous serial port data output |
| RC6 | bit6 | ST | Input/output port pin |
| RC7 | bit7 | ST | Input/output port pin |

Legend: ST = Schmitt Trigger input

**TABLE 5-7: PORTC FUNCTIONS FOR PIC16C63/R63/65/65A/R65/66/67**

| Name | Bit# | Buffer Type | Function |
|---|---|---|---|
| RC0/T1OSO/T1CKI | bit0 | ST | Input/output port pin or Timer1 oscillator output or Timer1 clock input |
| RC1/T1OSI/CCP2 | bit1 | ST | Input/output port pin or Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output |
| RC2/CCP1 | bit2 | ST | Input/output port pin or Capture1 input/Compare1 output/PWM1 output |
| RC3/SCK/SCL | bit3 | ST | RC3 can also be the synchronous serial clock for both SPI and $I^2C$ modes. |
| RC4/SDI/SDA | bit4 | ST | RC4 can also be the SPI Data In (SPI mode) or data I/O ($I^2C$ mode). |
| RC5/SDO | bit5 | ST | Input/output port pin or synchronous serial port data output |
| RC6/TX/CK | bit6 | ST | Input/output port pin or USART Asynchronous Transmit, or USART Synchronous Clock |
| RC7/RX/DT | bit7 | ST | Input/output port pin or USART Asynchronous Receive, or USART Synchronous Data |

Legend: ST = Schmitt Trigger input

**TABLE 5-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 07h | PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | uuuu uuuu |
| 87h | TRISC | PORTC Data Direction Register | | | | | | | | 1111 1111 | 1111 1111 |

Legend: x = unknown, u = unchanged.

**FIGURE 5-12: PARALLEL SLAVE PORT WRITE WAVEFORMS**

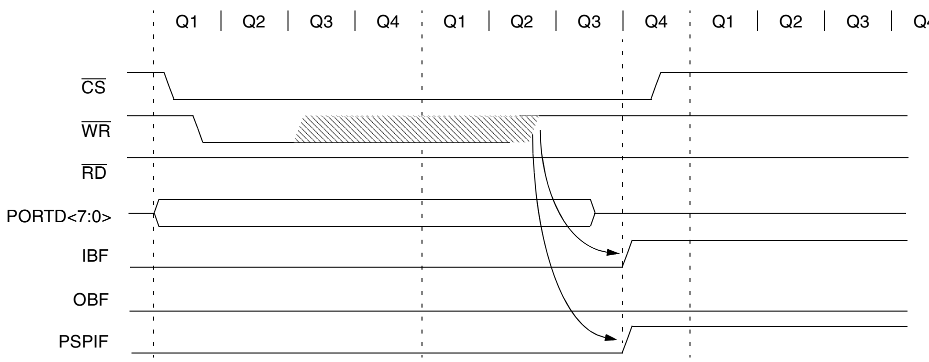



**FIGURE 5-13: PARALLEL SLAVE PORT READ WAVEFORMS**

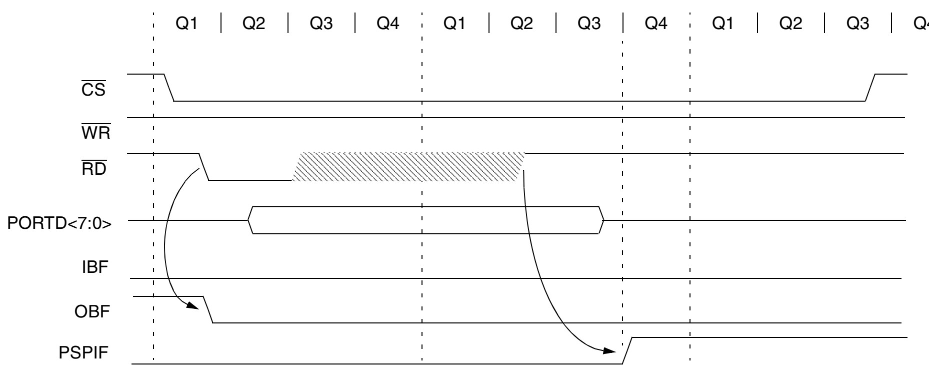



**TABLE 5-13: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT**

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 08h | PORTD | PSP7 | PSP6 | PSP5 | PSP4 | PSP3 | PSP2 | PSP1 | PSP0 | xxxx xxxx | uuuu uuuu |
| 09h | PORTE | — | — | — | — | — | RE2 | RE1 | RE0 | ---- -xxx | ---- -uuu |
| 89h | TRISE | IBF | OBF | IBOV | PSPMODE | — | PORTE Data Direction Bits | | | 0000 -111 | 0000 -111 |
| 0Ch | PIR1 | PSPIF | (1) | RCIF(2) | TXIF(2) | SSPIF | CCP1IF | TMR2IF | TRM1IF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE | (1) | RCIE(2) | TXIE(2) | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by the PSP.
Note 1: These bits are reserved, always maintain these bits clear.
2: These bits are implemented on the PIC16C65/65A/R65/67 only.

# 6.0 OVERVIEW OF TIMER MODULES

| Applicable Devices | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 |

All PIC16C6X devices have three timer modules except for the PIC16C61, which has one timer module. Each module can generate an interrupt to indicate that an event has occurred (i.e., timer overflow). Each of these modules are detailed in the following sections. The timer modules are:

- Timer0 module (Section 7.0)
- Timer1 module (Section 8.0)
- Timer2 module (Section 9.0)

## 6.1 Timer0 Overview

| Applicable Devices | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 |

The Timer0 module is a simple 8-bit overflow counter. The clock source can be either the internal system clock (Fosc/4) or an external clock. When the clock source is an external clock, the Timer0 module can be selected to increment on either the rising or falling edge.

The Timer0 module also has a programmable prescaler option. This prescaler can be assigned to either the Timer0 module or the Watchdog Timer. Bit PSA (OPTION<3>) assigns the prescaler, and bits PS2:PS0 (OPTION<2:0>) determine the prescaler value. TMR0 can increment at the following rates: 1:1 when the prescaler is assigned to Watchdog Timer, 1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128, and 1:256.

Synchronization of the external clock occurs after the prescaler. When the prescaler is used, the external clock frequency may be higher then the device's frequency. The maximum frequency is 50 MHz, given the high and low time requirements of the clock.

## 6.2 Timer1 Overview

| Applicable Devices | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 |

Timer1 is a 16-bit timer/counter. The clock source can be either the internal system clock (Fosc/4), an external clock, or an external crystal. Timer1 can operate as either a timer or a counter. When operating as a counter (external clock source), the counter can either operate synchronized to the device or asynchronously to the device. Asynchronous operation allows Timer1 to operate during sleep, which is useful for applications that require a real-time clock as well as the power savings of SLEEP mode.

TImer1 also has a prescaler option which allows TMR1 to increment at the following rates: 1:1, 1:2, 1:4, and 1:8. TMR1 can be used in conjunction with the Capture/Compare/PWM module. When used with a CCP module, Timer1 is the time-base for 16-bit capture or 16-bit compare and must be synchronized to the device.

## 6.3 Timer2 Overview

| Applicable Devices | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 |

Timer2 is an 8-bit timer with a programmable prescaler and a programmable postscaler, as well as an 8-bit Period Register (PR2). Timer2 can be used with the CCP module (in PWM mode) as well as the Baud Rate Generator for the Synchronous Serial Port (SSP). The prescaler option allows Timer2 to increment at the following rates: 1:1, 1:4, and 1:16.

The postscaler allows TMR2 register to match the period register (PR2) a programmable number of times before generating an interrupt. The postscaler can be programmed from 1:1 to 1:16 (inclusive).

## 6.4 CCP Overview

| Applicable Devices | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 |

The CCP module(s) can operate in one of three modes: 16-bit capture, 16-bit compare, or up to 10-bit Pulse Width Modulation (PWM).

Capture mode captures the 16-bit value of TMR1 into the CCPRxH:CCPRxL register pair. The capture event can be programmed for either the falling edge, rising edge, fourth rising edge, or sixteenth rising edge of the CCPx pin.

Compare mode compares the TMR1H:TMR1L register pair to the CCPRxH:CCPRxL register pair. When a match occurs, an interrupt can be generated and the output pin CCPx can be forced to a given state (High or Low) and Timer1 can be reset. This depends on control bits CCPxM3:CCPxM0.

PWM mode compares the TMR2 register to a 10-bit duty cycle register (CCPRxH:CCPRxL<5:4>) as well as to an 8-bit period register (PR2). When the TMR2 register = Duty Cycle register, the CCPx pin will be forced low. When TMR2 = PR2, TMR2 is cleared to 00h, an interrupt can be generated, and the CCPx pin (if an output) will be forced high.

## 11.2 SPI Mode for PIC16C62/62A/R62/63/R63/64/64A/R64/65/65A/R65

This section contains register definitions and operational characteristics of the SPI module for the PIC16C62, PIC16C62A, PIC16CR62, PIC16C63, PIC16CR63, PIC16C64, PIC16C64A, PIC16CR64, PIC16C65, PIC16C65A, PIC16CR65.

### FIGURE 11-1: SSPSTAT: SYNC SERIAL PORT STATUS REGISTER (ADDRESS 94h)

| U-0 | U-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| — | — | D/$\overline{\text{A}}$ | P | S | R/$\overline{\text{W}}$ | UA | BF |
| bit7 | | | | | | | bit0 |

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n =Value at POR reset

bit 7-6: **Unimplemented:** Read as '0'

bit 5: **D/$\overline{\text{A}}$**: Data/Address bit ($I^2C$ mode only)
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4: **P**: Stop bit ($I^2C$ mode only. This bit is cleared when the SSP module is disabled, SSPEN is cleared)
1 = Indicates that a stop bit has been detected last (this bit is '0' on RESET)
0 = Stop bit was not detected last

bit 3: **S**: Start bit ($I^2C$ mode only. This bit is cleared when the SSP module is disabled, SSPEN is cleared)
1 = Indicates that a start bit has been detected last (this bit is '0' on RESET)
0 = Start bit was not detected last

bit 2: **R/$\overline{\text{W}}$**: Read/Write bit information ($I^2C$ mode only)
This bit holds the R/W bit information following the last address match. This bit is valid from the address match to the next start bit, stop bit, or $\overline{\text{ACK}}$ bit.
1 = Read
0 = Write

bit 1: **UA**: Update Address (10-bit $I^2C$ mode only)
1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0: **BF**: Buffer Full Status bit
Receive (SPI and $I^2C$ modes)
1 = Receive complete, SSPBUF is full
0 = Receive not complete, SSPBUF is empty
Transmit ($I^2C$ mode only)
1 = Transmit in progress, SSPBUF is full
0 = Transmit complete, SSPBUF is empty

To enable the serial port, SSP Enable bit, SSPEN (SSPCON<5>) must be set. To reset or reconfigure SPI mode, clear bit SSPEN, re-initialize the SSPCON register, and then set bit SSPEN. This configures the SDI, SDO, SCK, and $\overline{\text{SS}}$ pins as serial port pins. For the pins to behave as the serial port function, they must have their data direction bits (in the TRISC register) appropriately programmed. That is:

- SDI must have TRISC<4> set
- SDO must have TRISC<5> cleared
- SCK (Master mode) must have TRISC<3> cleared
- SCK (Slave mode) must have TRISC<3> set
- $\overline{\text{SS}}$ must have TRISA<5> set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value. An example would be in master mode where you are only sending data (to a display driver), then both SDI and $\overline{\text{SS}}$ could be used as general purpose outputs by clearing their corresponding TRIS register bits.

Figure 11-10 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge, and latched on the opposite edge of the clock. Both processors should be programmed to same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application firmware. This leads to three scenarios for data transmission:

- Master sends data — Slave sends dummy data
- Master sends data — Slave sends data
- Master sends dummy data — Slave sends data

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2) is to broadcast data by the firmware protocol.

In master mode the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SCK output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "line activity monitor" mode.

In slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched the interrupt flag bit SSPIF (PIR1<3>) is set.

The clock polarity is selected by appropriately programming bit CKP (SSPCON<4>). This then would give waveforms for SPI communication as shown in Figure 11-11, Figure 11-12, and Figure 11-13 where the MSB is transmitted first. In master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- $F_{OSC}/4$ (or $T_{CY}$)
- $F_{OSC}/16$ (or 4 • $T_{CY}$)
- $F_{OSC}/64$ (or 16 • $T_{CY}$)
- Timer2 output/2

This allows a maximum bit clock frequency (at 20 MHz) of 5 MHz. When in slave mode the external clock must meet the minimum high and low times.

In sleep mode, the slave can transmit and receive data and wake the device from sleep.

**FIGURE 11-10: SPI MASTER/SLAVE CONNECTION (PIC16C66/67)**

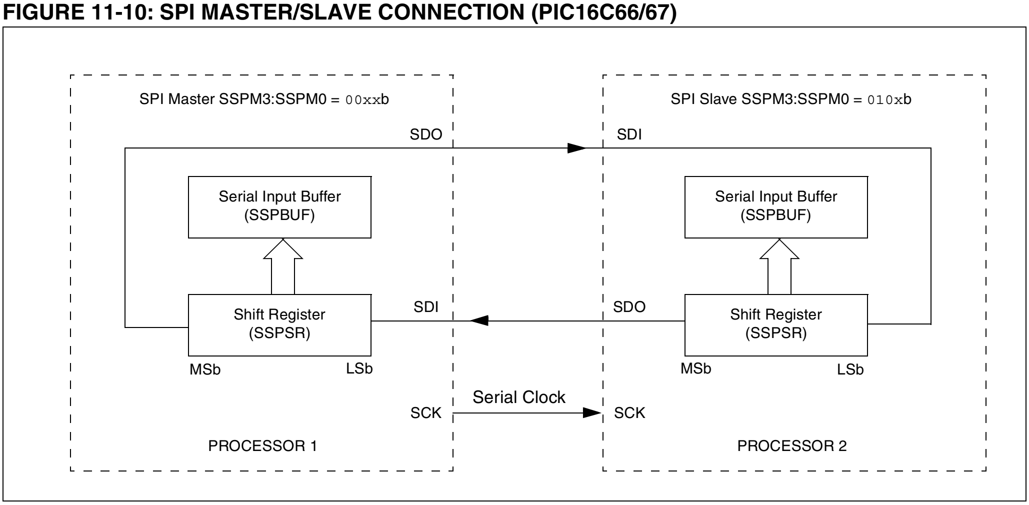


**FIGURE 12-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R-0 | R-0 | R-x |
|---|---|---|---|---|---|---|---|
| SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D |
| bit7 | | | | | | | bit0 |

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n = Value at POR reset
x = unknown

bit 7: **SPEN**: Serial Port Enable bit
(Configures RC7/RX/DT and RC6/TX/CK pins as serial port pins when bits TRISC<7:6> are set)
1 = Serial port enabled
0 = Serial port disabled

bit 6: **RX9**: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5: **SREN**: Single Receive Enable bit

Asynchronous mode
Don’t care

Synchronous mode - master
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.

Synchronous mode - slave
Unused in this mode

bit 4: **CREN**: Continuous Receive Enable bit

Asynchronous mode
1 = Enables continuous receive
0 = Disables continuous receive

Synchronous mode
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

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

bit 2: **FERR**: Framing Error bit
1 = Framing error (Can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1: **OERR**: Overrun Error bit
1 = Overrun error (Can be cleared by clearing bit CREN)
0 = No overrun error

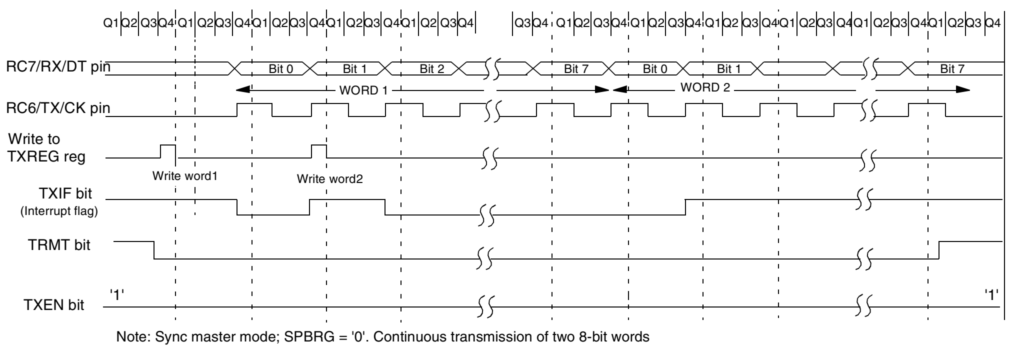
bit 0: **RX9D**: 9th bit of received data (Can be parity bit)

### TABLE 12-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

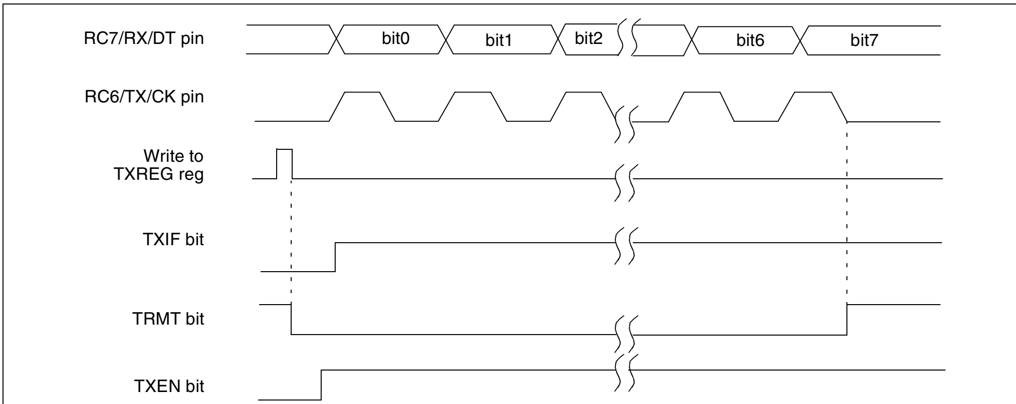
| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Value on all other Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0Ch | PIR1 | PSPIF(1) | (2) | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 18h | RCSTA | SPEN | RX9 | SREN | CREN | — | FERR | OERR | RX9D | 0000 -00x | 0000 -00x |
| 19h | TXREG | USART Transmit Register | | | | | | | | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE(1) | (2) | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 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 locations read as '0'. Shaded cells are not used for Synchronous Master Transmission.
Note 1: PSPIE and PSPIF are reserved on the PIC16C63/R63/66, always maintain these bits clear.
2: PIE1<6> and PIR1<6> are reserved, always maintain these bits clear.

### FIGURE 12-12: SYNCHRONOUS TRANSMISSION




### FIGURE 12-13: SYNCHRONOUS TRANSMISSION THROUGH TXEN



## TABLE 14-2: PIC16CXX INSTRUCTION SET

| Mnemonic, Operands | | Description | Cycles | 14-Bit Opcode MSb | | | LSb | Status Affected | Notes |
|---|---|---|---|---|---|---|---|---|---|
| **BYTE-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | | |
| **ADDWF** | **f, d** | Add W and f | 1 | 00 | 0111 | dfff | ffff | C,DC,Z | 1,2 |
| **ANDWF** | **f, d** | AND W with f | 1 | 00 | 0101 | dfff | ffff | Z | 1,2 |
| **CLRF** | **f** | Clear f | 1 | 00 | 0001 | lfff | ffff | Z | 2 |
| **CLRW** | **-** | Clear W | 1 | 00 | 0001 | 0xxx | xxxx | Z | |
| **COMF** | **f, d** | Complement f | 1 | 00 | 1001 | dfff | ffff | Z | 1,2 |
| **DECF** | **f, d** | Decrement f | 1 | 00 | 0011 | dfff | ffff | Z | 1,2 |
| **DECFSZ** | **f, d** | Decrement f, Skip if 0 | 1(2) | 00 | 1011 | dfff | ffff | | 1,2,3 |
| **INCF** | **f, d** | Increment f | 1 | 00 | 1010 | dfff | ffff | Z | 1,2 |
| **INCFSZ** | **f, d** | Increment f, Skip if 0 | 1(2) | 00 | 1111 | dfff | ffff | | 1,2,3 |
| **IORWF** | **f, d** | Inclusive OR W with f | 1 | 00 | 0100 | dfff | ffff | Z | 1,2 |
| **MOVF** | **f, d** | Move f | 1 | 00 | 1000 | dfff | ffff | Z | 1,2 |
| **MOVWF** | **f** | Move W to f | 1 | 00 | 0000 | lfff | ffff | | |
| **NOP** | **-** | No Operation | 1 | 00 | 0000 | 0xx0 | 0000 | | |
| **RLF** | **f, d** | Rotate Left f through Carry | 1 | 00 | 1101 | dfff | ffff | C | 1,2 |
| **RRF** | **f, d** | Rotate Right f through Carry | 1 | 00 | 1100 | dfff | ffff | C | 1,2 |
| **SUBWF** | **f, d** | Subtract W from f | 1 | 00 | 0010 | dfff | ffff | C,DC,Z | 1,2 |
| **SWAPF** | **f, d** | Swap nibbles in f | 1 | 00 | 1110 | dfff | ffff | | 1,2 |
| **XORWF** | **f, d** | Exclusive OR W with f | 1 | 00 | 0110 | dfff | ffff | Z | 1,2 |
| **BIT-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | | |
| **BCF** | **f, b** | Bit Clear f | 1 | 01 | 00bb | bfff | ffff | | 1,2 |
| **BSF** | **f, b** | Bit Set f | 1 | 01 | 01bb | bfff | ffff | | 1,2 |
| **BTFSC** | **f, b** | Bit Test f, Skip if Clear | 1 (2) | 01 | 10bb | bfff | ffff | | 3 |
| **BTFSS** | **f, b** | Bit Test f, Skip if Set | 1 (2) | 01 | 11bb | bfff | ffff | | 3 |
| **LITERAL AND CONTROL OPERATIONS** | | | | | | | | | |
| **ADDLW** | **k** | Add literal and W | 1 | 11 | 111x | kkkk | kkkk | C,DC,Z | |
| **ANDLW** | **k** | AND literal with W | 1 | 11 | 1001 | kkkk | kkkk | Z | |
| **CALL** | **k** | Call subroutine | 2 | 10 | 0kkk | kkkk | kkkk | | |
| **CLRWDT** | **-** | Clear Watchdog Timer | 1 | 00 | 0000 | 0110 | 0100 | $\overline{TO}$,$\overline{PD}$ | |
| **GOTO** | **k** | Go to address | 2 | 10 | 1kkk | kkkk | kkkk | | |
| **IORLW** | **k** | Inclusive OR literal with W | 1 | 11 | 1000 | kkkk | kkkk | Z | |
| **MOVLW** | **k** | Move literal to W | 1 | 11 | 00xx | kkkk | kkkk | | |
| **RETFIE** | **-** | Return from interrupt | 2 | 00 | 0000 | 0000 | 1001 | | |
| **RETLW** | **k** | Return with literal in W | 2 | 11 | 01xx | kkkk | kkkk | | |
| **RETURN** | **-** | Return from Subroutine | 2 | 00 | 0000 | 0000 | 1000 | | |
| **SLEEP** | **-** | Go into standby mode | 1 | 00 | 0000 | 0110 | 0011 | $\overline{TO}$,$\overline{PD}$ | |
| **SUBLW** | **k** | Subtract W from literal | 1 | 11 | 110x | kkkk | kkkk | C,DC,Z | |
| **XORLW** | **k** | Exclusive OR literal with W | 1 | 11 | 1010 | kkkk | kkkk | Z | |

Note 1: When an I/O register is modified as a function of itself ( e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

## SUBWF — Subtract W from f

| | |
|---|---|
| Syntax: | [ *label* ] SUBWF f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (f) - (W) → (destination) |
| Status Affected: | C, DC, Z |
| Encoding: | `00` `0010` `dfff` `ffff` |
| Description: | Subtract (2's complement method) W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'. |
| Words: | 1 |
| Cycles: | 1 |

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process data | Write to destination |

Example 1: `SUBWF REG1,1`

Before Instruction

REG1 = 3
W = 2
C = ?
Z = ?

After Instruction

REG1 = 1
W = 2
C = 1; result is positive
Z = 0

Example 2: Before Instruction

REG1 = 2
W = 2
C = ?
Z = ?

After Instruction

REG1 = 0
W = 2
C = 1; result is zero
Z = 1

Example 3: Before Instruction

REG1 = 1
W = 2
C = ?
Z = ?

After Instruction

REG1 = 0xFF
W = 2
C = 0; result is negative
Z = 0

## SWAPF — Swap Nibbles in f

| | |
|---|---|
| Syntax: | [ *label* ] SWAPF f,d |
| Operands: | $0 \leq f \leq 127$<br>$d \in [0,1]$ |
| Operation: | (f<3:0>) → (destination<7:4>),<br>(f<7:4>) → (destination<3:0>) |
| Status Affected: | None |
| Encoding: | `00` `1110` `dfff` `ffff` |
| Description: | The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'. |
| Words: | 1 |
| Cycles: | 1 |

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process data | Write to destination |

Example `SWAPF REG, 0`

Before Instruction

REG1 = 0xA5

After Instruction

REG1 = 0xA5
W = 0x5A

## TRIS — Load TRIS Register

| | |
|---|---|
| Syntax: | [*label*] TRIS f |
| Operands: | $5 \leq f \leq 7$ |
| Operation: | (W) → TRIS register f; |
| Status Affected: | None |
| Encoding: | `00` `0000` `0110` `0fff` |
| Description: | The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them. |
| Words: | 1 |
| Cycles: | 1 |
| Example | |

**To maintain upward compatibility with future PIC16CXX products, do not use this instruction.**

| Applicable Devices | 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**FIGURE 16-6: TYPICAL IPD VS. VDD WATCHDOG TIMER ENABLED 25°C**

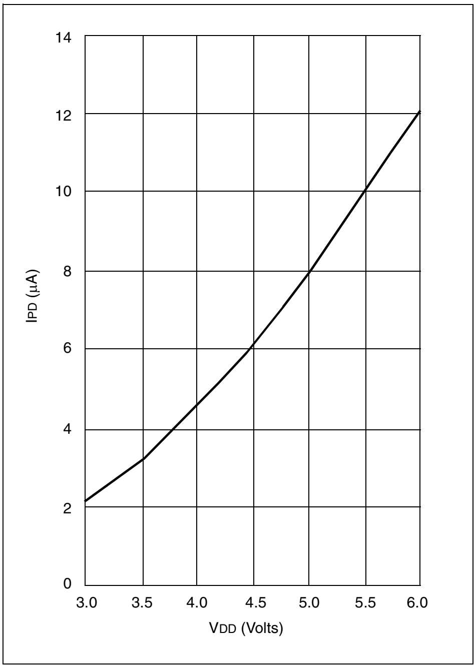
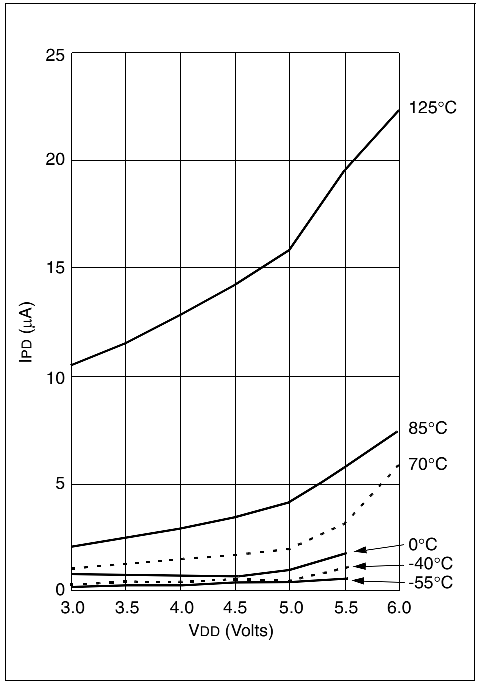



**FIGURE 16-7: MAXIMUM IPD VS. VDD WATCHDOG DISABLED**




Data based on matrix samples. See first page of this section for details.

| Applicable Devices | 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Data based on matrix samples. See first page of this section for details.

**FIGURE 16-17: TRANSCONDUCTANCE (gm) OF LP OSCILLATOR VS. VDD**

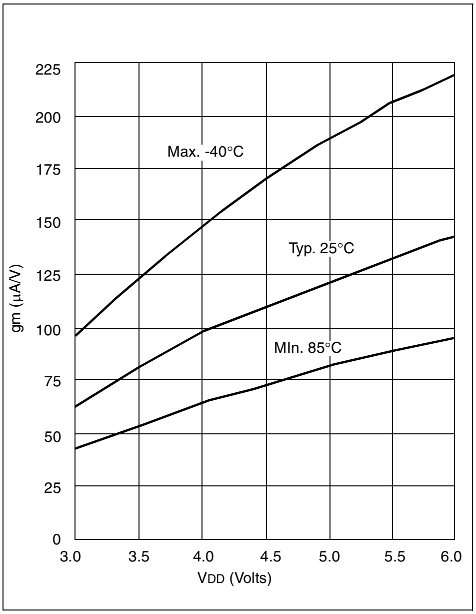



**FIGURE 16-18: TRANSCONDUCTANCE (gm) OF XT OSCILLATOR VS. VDD**

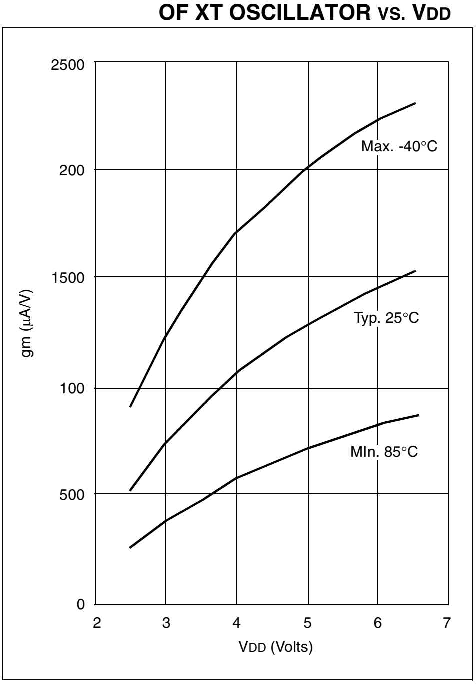



**FIGURE 16-19: IOH VS. VOH, VDD = 3V**

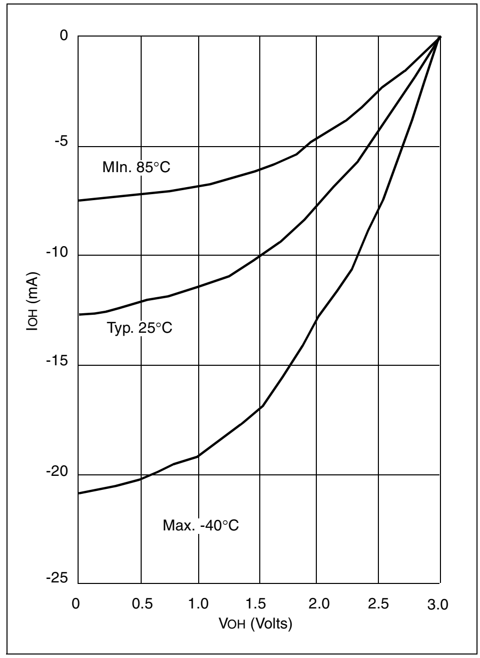
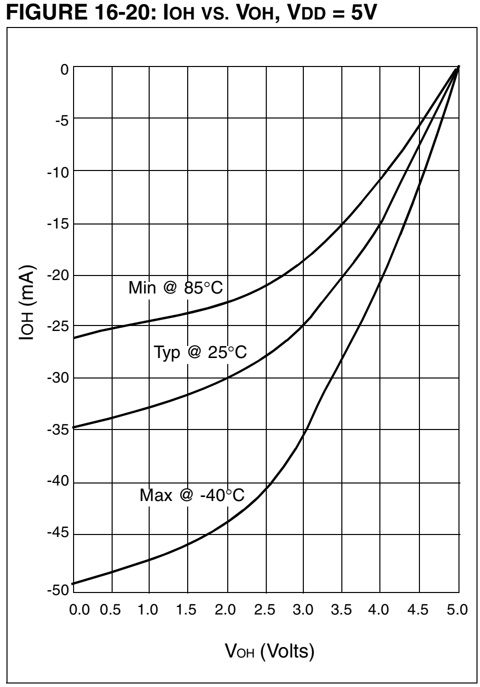



**FIGURE 16-20: IOH VS. VOH, VDD = 5V**



| Applicable Devices | 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

## 17.1 DC Characteristics: PIC16C62/64-04 (Commercial, Industrial) PIC16C62/64-10 (Commercial, Industrial) PIC16C62/64-20 (Commercial, Industrial)

| DC CHARACTERISTICS | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial | | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Characteristic | Sym | Min | Typ† | Max | Units | Conditions |
| D001<br>D001A | Supply Voltage | VDD | 4.0<br>4.5 | -<br>- | 6.0<br>5.5 | V<br>V | XT, RC and LP osc configuration<br>HS osc configuration |
| D002* | RAM Data Retention Voltage (Note 1) | VDR | - | 1.5 | - | V | |
| D003 | VDD start voltage to ensure internal Power-on Reset signal | VPOR | - | VSS | - | V | See section on Power-on Reset for details |
| D004* | VDD rise rate to ensure internal Power-on Reset signal | SVDD | 0.05 | - | - | V/ms | See section on Power-on Reset for details |
| D010 | Supply Current (Note 2, 5) | IDD | - | 2.7 | 5.0 | mA | XT, RC, osc configuration<br>FOSC = 4 MHz, VDD = 5.5V (Note 4) |
| D013 | | | - | 13.5 | 30 | mA | HS osc configuration<br>FOSC = 20 MHz, VDD = 5.5V |
| D020<br>D021<br>D021A | Power-down Current (Note 3, 5) | IPD | -<br>-<br>- | 10.5<br>1.5<br>1.5 | 42<br>21<br>24 | μA<br>μA<br>μA | VDD = 4.0V, WDT enabled, -40°C to +85°C<br>VDD = 4.0V, WDT disabled, -0°C to +70°C<br>VDD = 4.0V, WDT disabled, -40°C to +85°C |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail to rail; all I/O pins tristated, pulled to VDD
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

3: The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

5: Timer1 oscillator (when enabled) adds approximately 20 μA to the specification. This value is from characterization and is for design guidance only. This is not tested.

| Applicable Devices | 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

## 19.3 DC Characteristics: PIC16C65-04 (Commercial, Industrial) PIC16C65-10 (Commercial, Industrial) PIC16C65-20 (Commercial, Industrial) PIC16LC65-04 (Commercial, Industrial)

| DC CHARACTERISTICS | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial and<br>0°C ≤ TA ≤ +70°C for commercial<br>Operating voltage VDD range as described in DC spec Section 19.1 and Section 19.2 | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Characteristic** | **Sym** | **Min** | **Typ †** | **Max** | **Units** | **Conditions** |
| | **Input Low Voltage** | | | | | | |
| | I/O ports | VIL | | | | | |
| D030 | with TTL buffer | | VSS | - | 0.15VDD | V | For entire VDD range |
| D030A | | | VSS | - | 0.8V | V | 4.5V ≤ VDD ≤ 5.5V |
| D031 | with Schmitt Trigger buffer | | VSS | - | 0.2VDD | V | |
| D032 | $\overline{\text{MCLR}}$, OSC1(in RC mode) | | Vss | - | 0.2VDD | V | |
| D033 | OSC1 (in XT, HS and LP) | | Vss | - | 0.3VDD | V | Note1 |
| | **Input High Voltage** | | | | | | |
| | I/O ports | VIH | | - | | | |
| D040 | with TTL buffer | | 2.0 | - | VDD | V | 4.5V ≤ VDD ≤ 5.5V |
| D040A | | | 0.25VDD+ 0.8V | - | VDD | V | For entire VDD range |
| D041 | with Schmitt Trigger buffer | | 0.8VDD | - | VDD | | For entire VDD range |
| D042 | $\overline{\text{MCLR}}$ | | 0.8VDD | - | VDD | V | |
| D042A | OSC1 (XT, HS and LP) | | 0.7 VDD | - | VDD | V | Note1 |
| D043 | OSC1 (in RC mode) | | 0.9VDD | - | VDD | V | |
| D070 | PORTB weak pull-up current | IPURB | 50 | 250 | 400 | μA | VDD = 5V, VPIN = VSS |
| | **Input Leakage Current** (Notes 2, 3) | | | | | | |
| D060 | I/O ports | IIL | - | - | ±1 | μA | Vss ≤ VPIN ≤ VDD, Pin at hi-impedance |
| D061 | $\overline{\text{MCLR}}$, RA4/T0CKI | | - | - | ±5 | μA | Vss ≤ VPIN ≤ VDD |
| D063 | OSC1 | | - | - | ±5 | μA | Vss ≤ VPIN ≤ VDD, XT, HS, and LP osc configuration |
| | **Output Low Voltage** | | | | | | |
| D080 | I/O ports | VOL | - | - | 0.6 | V | IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C |
| D083 | OSC2/CLKOUT (RC osc config) | | - | - | 0.6 | V | IOL = 1.6 mA, VDD = 4.5V, -40°C to +85°C |
| | **Output High Voltage** | | | | | | |
| D090 | I/O ports (Note 3) | VOH | VDD-0.7 | - | - | V | IOH = -3.0 mA, VDD = 4.5V, -40°C to +85°C |
| D092 | OSC2/CLKOUT (RC osc config) | | VDD-0.7 | - | - | V | IOH = -1.3 mA, VDD = 4.5V, -40°C to +85°C |
| D150* | **Open-Drain High Voltage** | VOD | - | - | 14 | V | RA4 pin |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C6X be driven with external clock in RC mode.

2: The leakage current on the $\overline{\text{MCLR}}$/VPP pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

| Applicable Devices | 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

## 21.1 DC Characteristics: PIC16CR63/R65-04 (Commercial, Industrial) PIC16CR63/R65-10 (Commercial, Industrial) PIC16CR63/R65-20 (Commercial, Industrial)

**DC CHARACTERISTICS**

**Standard Operating Conditions (unless otherwise stated)**
Operating temperature -40°C ≤ TA ≤ +85°C for industrial and
0°C ≤ TA ≤ +70°C for commercial

| Param No. | Characteristic | Sym | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| D001 | Supply Voltage | VDD | 4.0 | - | 5.5 | V | XT, RC and LP osc configuration |
| D001A | | | 4.5 | - | 5.5 | V | HS osc configuration |
| D002* | RAM Data Retention Voltage (Note 1) | VDR | - | 1.5 | - | V | |
| D003 | VDD start voltage to ensure internal Power-on Reset signal | VPOR | - | VSS | - | V | See section on Power-on Reset for details |
| D004* | VDD rise rate to ensure internal Power-on Reset signal | SVDD | 0.05 | - | - | V/ms | See section on Power-on Reset for details |
| D005 | Brown-out Reset Voltage | BVDD | 3.7 | 4.0 | 4.3 | V | BODEN configuration bit is enabled |
| D010 | Supply Current (Note 2, 5) | IDD | - | 2.7 | 5 | mA | XT, RC, osc config FOSC = 4 MHz, VDD = 5.5V (Note 4) |
| D013 | | | - | 10 | 20 | mA | HS osc config FOSC = 20 MHz, VDD = 5.5V |
| D015* | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | μA | BOR enabled, VDD = 5.0V |
| D020 | Power-down Current (Note 3, 5) | IPD | - | 10.5 | 42 | μA | VDD = 4.0V, WDT enabled,-40°C to +85°C |
| D021 | | | - | 1.5 | 16 | μA | VDD = 4.0V, WDT disabled,-0°C to +70°C |
| D021A | | | - | 1.5 | 19 | μA | VDD = 4.0V, WDT disabled,-40°C to +85°C |
| D023* | Brown-out Reset Current (Note 6) | ΔIBOR | - | 350 | 425 | μA | BOR enabled, VDD = 5.0V |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail to rail; all I/O pins tristated, pulled to VDD,
MCLR = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

5: Timer1 oscillator (when enabled) adds approximately 20 μA to the specification. This value is from characterization and is for design guidance only. This is not tested.

6: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

| Applicable Devices | 61 | 62 | 62A | R62 | 63 | R63 | 64 | 64A | R64 | 65 | 65A | R65 | 66 | 67 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**FIGURE 21-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**

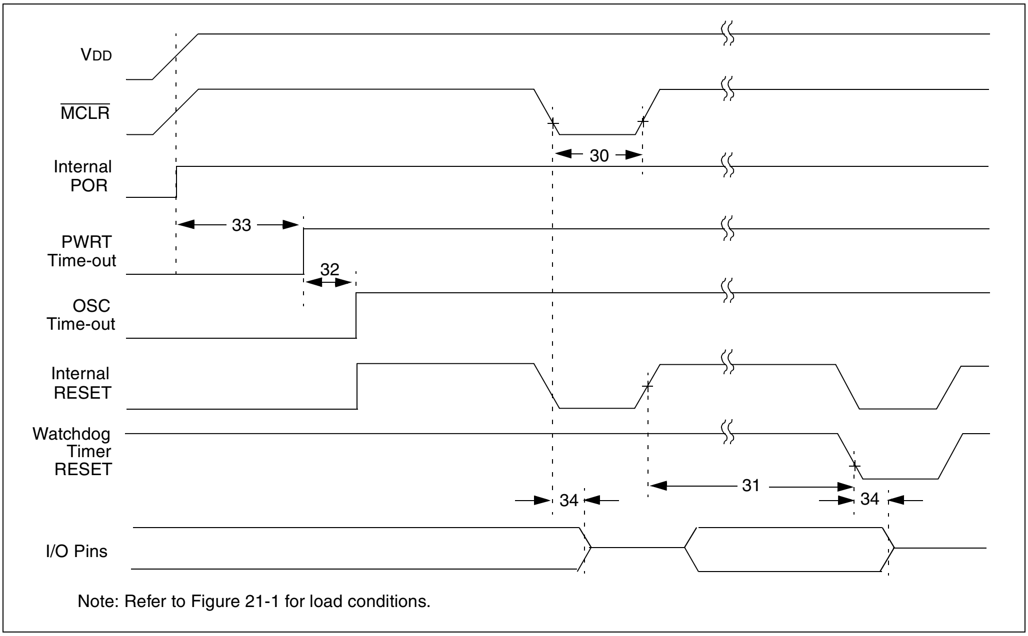



**FIGURE 21-5: BROWN-OUT RESET TIMING**

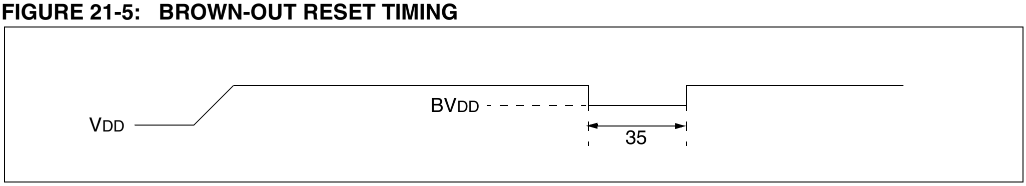



**TABLE 21-4: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET REQUIREMENTS**

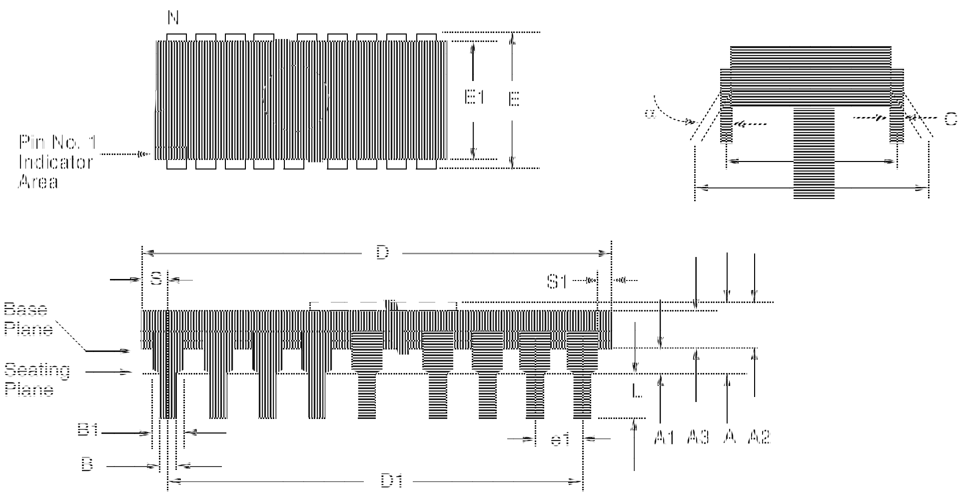
| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| 30 | TmcL | $\overline{\text{MCLR}}$ Pulse Width (low) | 2 | — | — | µs | VDD = 5V, -40°C to +125°C |
| 31* | Twdt | Watchdog Timer Time-out Period (No Prescaler) | 7 | 18 | 33 | ms | VDD = 5V, -40°C to +125°C |
| 32 | Tost | Oscillation Start-up Timer Period | — | 1024 TOSC | — | — | TOSC = OSC1 period |
| 33* | Tpwrt | Power-up Timer Period | 28 | 72 | 132 | ms | VDD = 5V, -40°C to +125°C |
| 34 | TIOZ | I/O Hi-impedance from $\overline{\text{MCLR}}$ Low or WDT reset | — | — | 2.1 | µs | |
| 35 | TBOR | Brown-out Reset Pulse Width | 100 | — | — | µs | VDD ≤ BVDD (D005) |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

## 24.8 40-Lead Ceramic CERDIP Dual In-line with Window (600 mil) (JW)

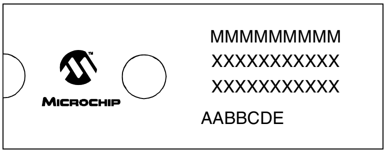
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



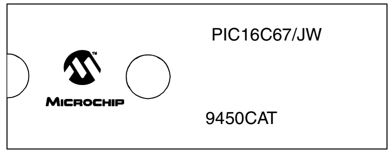

| Package Group: Ceramic CERDIP Dual In-Line (CDP) | | | | | | |
|---|---|---|---|---|---|---|
| | Millimeters | | | Inches | | |
| Symbol | Min | Max | Notes | Min | Max | Notes |
| α | 0° | 10° | | 0° | 10° | |
| A | 4.318 | 5.715 | | 0.170 | 0.225 | |
| A1 | 0.381 | 1.778 | | 0.015 | 0.070 | |
| A2 | 3.810 | 4.699 | | 0.150 | 0.185 | |
| A3 | 3.810 | 4.445 | | 0.150 | 0.175 | |
| B | 0.355 | 0.585 | | 0.014 | 0.023 | |
| B1 | 1.270 | 1.651 | **Typical** | 0.050 | 0.065 | **Typical** |
| C | 0.203 | 0.381 | **Typical** | 0.008 | 0.015 | **Typical** |
| D | 51.435 | 52.705 | | 2.025 | 2.075 | |
| D1 | 48.260 | 48.260 | **Reference** | 1.900 | 1.900 | **Reference** |
| E | 15.240 | 15.875 | | 0.600 | 0.625 | |
| E1 | 12.954 | 15.240 | | 0.510 | 0.600 | |
| e1 | 2.540 | 2.540 | **Reference** | 0.100 | 0.100 | **Reference** |
| eA | 14.986 | 16.002 | **Typical** | 0.590 | 0.630 | **Typical** |
| eB | 15.240 | 18.034 | | 0.600 | 0.710 | |
| L | 3.175 | 3.810 | | 0.125 | 0.150 | |
| N | 40 | 40 | | 40 | 40 | |
| S | 1.016 | 2.286 | | 0.040 | 0.090 | |
| S1 | 0.381 | 1.778 | | 0.015 | 0.070 | |

## Package Marking Information (Cont'd)

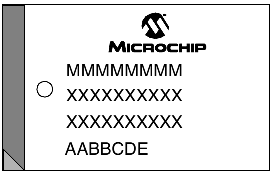
40-Lead CERDIP Windowed



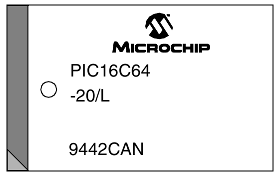

Example



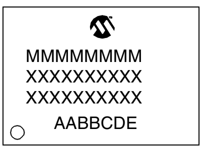

44-Lead PLCC



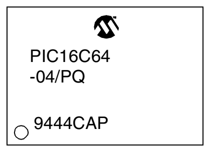

Example



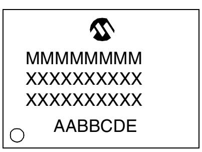

44-Lead MQFP



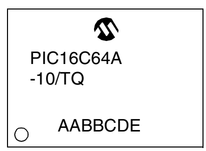

Example




44-Lead TQFP




Example




| | | |
|---|---|---|
| Legend: | MM...M | Microchip part number information |
| | XX...X | Customer specific information* |
| | AA | Year code (last 2 digits of calender year) |
| | BB | Week code (week of January 1 is week '01') |
| | C | Facility code of the plant at which wafer is manufactured.<br>C = Chandler, Arizona, U.S.A.<br>S = Tempe, Arizona, U.S.A. |
| | $D_1$ | Mask revision number for microcontroller |
| | E | Assembly code of the plant or country of origin in which part was assembled. |
| 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. | |

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask revision number, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.