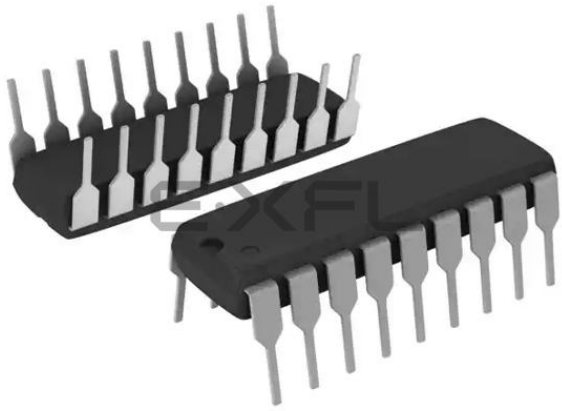


**Welcome to E-XFL.COM**

## 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 | 20MHz |
| Connectivity | - |
| Peripherals | POR, WDT |
| Number of I/O | 13 |
| Program Memory Size | 3.5KB (2K x 14) |
| Program Memory Type | OTP |
| EEPROM Size | - |
| RAM Size | 128 x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | External |
| Operating Temperature | 0°C ~ 70°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/pic16c558-20-p |

## Table of Contents



## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@mail.microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 792-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at **www.microchip.com/cn** to receive the most current information on all of our products.

**TABLE 4-1: SPECIAL REGISTERS FOR THE PIC16C55X**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR Reset | Detail on Page: |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bank 0** | | | | | | | | | | | |
| 00h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | 21 |
| 01h | TMR0 | Timer0 Module's Register | | | | | | | | xxxx xxxx | 47 |
| 02h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 21 |
| 03h | STATUS | IRP(2) | RP1(2) | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 17 |
| 04h | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | 21 |
| 05h | PORTA | — | — | — | RA4 | RA3 | RA2 | RA1 | RA0 | ---x xxxx | 23 |
| 06h | PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | 25 |
| 07h | PORTC(4) | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | 27 |
| 08h | — | Unimplemented | | | | | | | | — | — |
| 09h | — | Unimplemented | | | | | | | | — | — |
| 0Ah | PCLATH | — | — | — | Write buffer for upper 5 bits of program counter | | | | | ---0 0000 | 21 |
| 0Bh | INTCON | GIE | (3) | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 19 |
| 0Ch | — | Unimplemented | | | | | | | | — | — |
| 0Dh-1Eh | — | Unimplemented | | | | | | | | — | — |
| 1Fh | — | Unimplemented | | | | | | | | — | — |
| **Bank 1** | | | | | | | | | | | |
| 80h | INDF | Addressing this location uses contents of FSR to address data memory (not a physical register) | | | | | | | | xxxx xxxx | 21 |
| 81h | OPTION | $\overline{RBPU}$ | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 | 1111 1111 | 18 |
| 82h | PCL | Program Counter's (PC) Least Significant Byte | | | | | | | | 0000 0000 | 21 |
| 83h | STATUS | — | — | RP0 | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | 0001 1xxx | 17 |
| 84h | FSR | Indirect data memory address pointer | | | | | | | | xxxx xxxx | 21 |
| 85h | TRISA | — | — | — | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | ---1 1111 | 23 |
| 86h | TRISB | TRISB7 | TRISB6 | TRISB5 | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 1111 1111 | 25 |
| 87h | TRISC(4) | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 27 |
| 88h | — | Unimplemented | | | | | | | | — | — |
| 89h | — | Unimplemented | | | | | | | | — | — |
| 8Ah | PCLATH | — | — | — | Write buffer for upper 5 bits of program counter | | | | | ---0 0000 | 21 |
| 8Bh | INTCON | GIE | (3) | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 19 |
| 8Ch | — | Unimplemented | | | | | | | | — | — |
| 8Dh | — | Unimplemented | | | | | | | | — | — |
| 8Eh | PCON | — | — | — | — | — | — | $\overline{POR}$ | — | ---- --0- | 20 |
| 8Fh-9Eh | — | Unimplemented | | | | | | | | — | — |
| 9Fh | — | Unimplemented | | | | | | | | — | — |

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

**Note 1:** Other (non Power-up) Resets include $\overline{MCLR}$ Reset and Watchdog Timer Reset during normal operation.

**2:** IRP & RP1 bits are reserved, always maintain these bits clear.

**3:** Bit 6 of INTCON register is reserved for future use. Always maintain this bit as clear.

**4:** PIC16C557 only.

### 4.2.2.4 PCON Register

The PCON register contains a flag bit to differentiate between a Power-on Reset, an external $\overline{MCLR}$ Reset or WDT Reset. See Section 6.3 and Section 6.4 for detailed RESET operation.

**REGISTER 4-4: PCON REGISTER (ADDRESS 8Eh)**

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | $\overline{POR}$ | — |
| bit7 | | | | | | | bit0 |

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **$\overline{POR}$**: Power-on Reset status bit
1 = No Power-on Reset occurred
0 = Power-on Reset occurred

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

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

## 5.0 I/O PORTS

The PIC16C554 and PIC16C558 have two ports, PORTA and PORTB. The PIC16C557 has three ports, PORTA, PORTB and PORTC.

### 5.1 PORTA and TRISA Registers

PORTA is a 5-bit wide latch. RA4 is a Schmitt Trigger input and an open-drain output. Port RA4 is multiplexed with the T0CKI clock input. All other RA port pins have Schmitt Trigger input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers) which can configure these pins as input or output.

A '1' in the TRISA register puts the corresponding output driver in a Hi-impedance mode. A '0' in the TRISA register puts the contents of the output latch on the selected pin(s).

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. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

**Note 1:** On RESET, the TRISA register is set to all inputs.

**FIGURE 5-1: BLOCK DIAGRAM OF PORT PINS RA<3:0>**

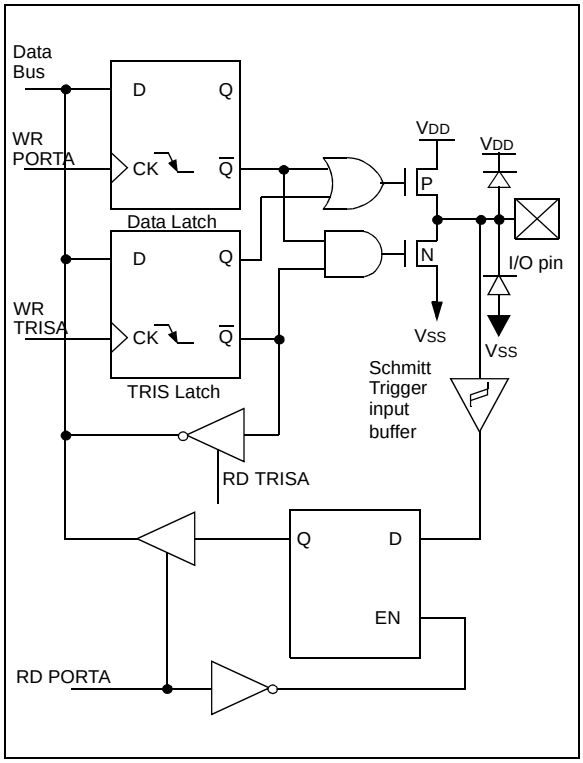
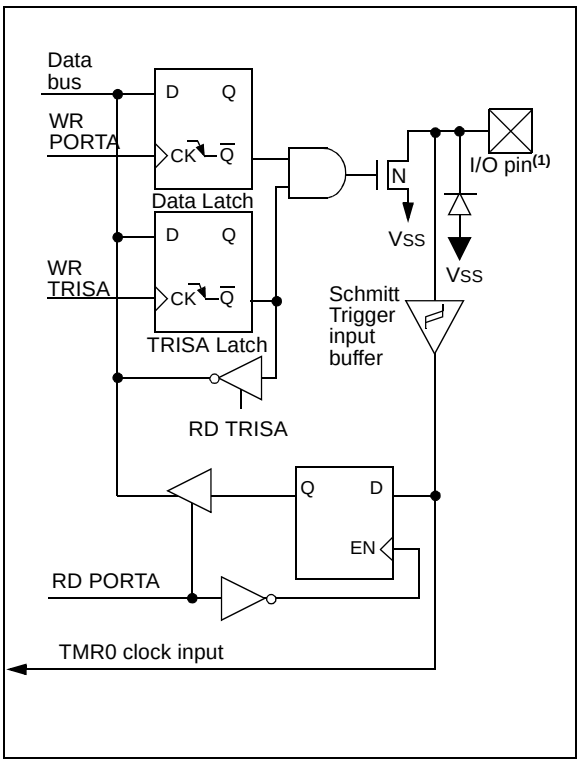



**FIGURE 5-2: BLOCK DIAGRAM OF RA4 PIN**



## 5.3 PORTC and TRISC Registers(1)

PORTC is a 8-bit wide latch. All pins have data direction bits (TRIS registers) which can configure these pins as input or output.

A '1' in the TRISC register puts the corresponding output driver in a Hi-impedance mode. A '0' in the TRISC register puts the contents of the output latch on the selected pin(s).

Reading the PORTC 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. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch

**FIGURE 5-5: BLOCK DIAGRAM OF PORT PINS RC<7:0>**

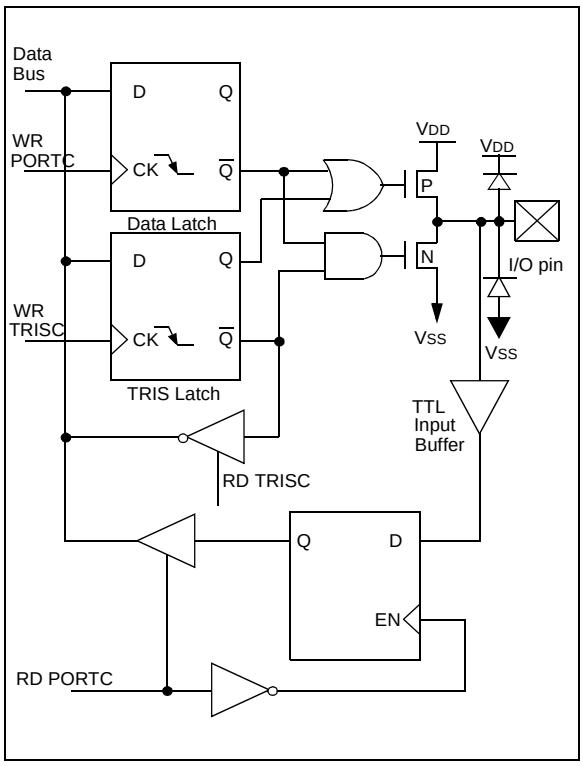



**TABLE 5-5: PORTC FUNCTIONS**

| Name | Bit # | Buffer Type | Function |
|---|---|---|---|
| RC0 | Bit 0 | TTL | Bi-directional I/O port. |
| RC1 | Bit 1 | TTL | Bi-directional I/O port. |
| RC2 | Bit 2 | TTL | Bi-directional I/O port. |
| RC3 | Bit 3 | TTL | Bi-directional I/O port. |
| RC4 | Bit 4 | TTL | Bi-directional I/O port. |
| RC5 | Bit 5 | TTL | Bi-directional I/O port. |
| RC6 | Bit 6 | TTL | Bi-directional I/O port. |
| RC7 | Bit 7 | TTL | Bi-directional I/O port. |

Legend: ST = Schmitt Trigger, TTL = TTL input

**TABLE 5-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC AND TRISC**

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR | Value on All Other RESETS |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 07h | PORTC | RC7 | RC6 | RC5 | RC4 | RC3 | RC2 | RC1 | RC0 | xxxx xxxx | uuuu uuuu |
| 87h | TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111 | 1111 1111 |

Legend: x = unknown, u = unchanged
**Note 1:** PIC16C557 ONLY.

## 6.2 Oscillator Configurations

### 6.2.1 OSCILLATOR TYPES

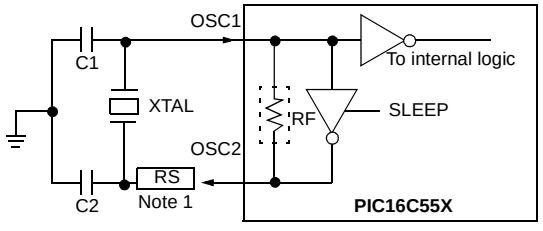
The PIC16C55X can be operated in four different oscillator options. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

### 6.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 6-1). The PIC16C55X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1 pin (Figure 6-2).

**FIGURE 6-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)**




**Note 1:** A series resistor may be required for AT strip cut crystals.
**2:** See Table 6-1 and Table 6-2 for recommended values of C1 and C2.

**FIGURE 6-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)**

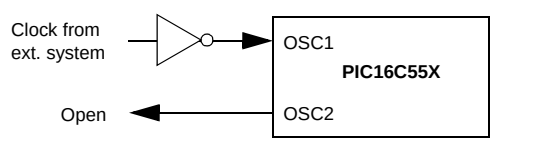



**TABLE 6-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS (PRELIMINARY)**

| Ranges Characterized: | | | |
|---|---|---|---|
| **Mode** | **Freq** | **OSC1(C1)** | **OSC2(C2)** |
| XT | 455 kHz<br>2.0 MHz<br>4.0 MHz | 22 - 100 pF<br>15 - 68 pF<br>15 - 68 pF | 22 - 100 pF<br>15 - 68 pF<br>15 - 68 pF |
| HS | 8.0 MHz<br>16.0 MHz | 10 - 68 pF<br>10 - 22 pF | 10 - 68 pF<br>10 - 22 pF |

**Note 1:** Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult with the resonator manufacturer for appropriate values of external components.

**TABLE 6-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR (PRELIMINARY)**

| Mode | Freq | OSC1(C1) | OSC2(C2) |
|---|---|---|---|
| LP | 32 kHz<br>200 kHz | 68 - 100 pF<br>15 - 30 pF | 68 - 100 pF<br>15 - 30 pF |
| XT | 100 kHz<br>2 MHz<br>4 MHz | 68 - 150 pF<br>15 - 30 pF<br>15 - 30 pF | 150 - 200 pF<br>15 - 30 pF<br>15 - 30 pF |
| HS | 8 MHz<br>10 MHz<br>20 MHz | 15 - 30 pF<br>15 - 30 pF<br>15 - 30 pF | 15 - 30 pF<br>15 - 30 pF<br>15 - 30 pF |

**Note 1:** Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low-drive level specification. Since each crystal has its own characteristics, the user should consult with the crystal manufacturer for appropriate values of external components.

### 6.4.5 POWER CONTROL/STATUS REGISTER (PCON)

Bit1 is $\overline{POR}$ (Power-on Reset). It is a '0' on Power-on Reset and unaffected otherwise. The user must write a '1' to this bit following a Power-on Reset. On a subsequent RESET if $\overline{POR}$ is '0', it will indicate that a Power-on Reset must have occurred ($V_{DD}$ may have gone too low).

**TABLE 6-3: TIMEOUT IN VARIOUS SITUATIONS**

| Oscillator Configuration | Power-up | | Wake-up from SLEEP |
|---|---|---|---|
| | $\overline{PWRTE}$ = 0 | $\overline{PWRTE}$ = 1 | |
| XT, HS, LP | 72 ms + 1024 TOSC | 1024 TOSC | 1024 TOSC |
| RC | 72 ms | — | — |

**TABLE 6-4: STATUS BITS AND THEIR SIGNIFICANCE**

| POR | TO | PD | |
|---|---|---|---|
| 0 | 1 | 1 | Power-on Reset |
| 0 | 0 | X | Illegal, $\overline{TO}$ is set on $\overline{POR}$ |
| 0 | X | 0 | Illegal, $\overline{PD}$ is set on $\overline{POR}$ |
| 1 | 0 | u | WDT Reset |
| 1 | 0 | 0 | WDT Wake-up |
| 1 | u | u | $\overline{MCLR}$ Reset during normal operation |
| 1 | 1 | 0 | $\overline{MCLR}$ Reset during SLEEP |

## 6.9 Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

| **Note:** | Microchip does not recommend code protecting windowed devices. |
|---|---|

## 6.10 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify.

## 6.11 In-Circuit Serial Programming™

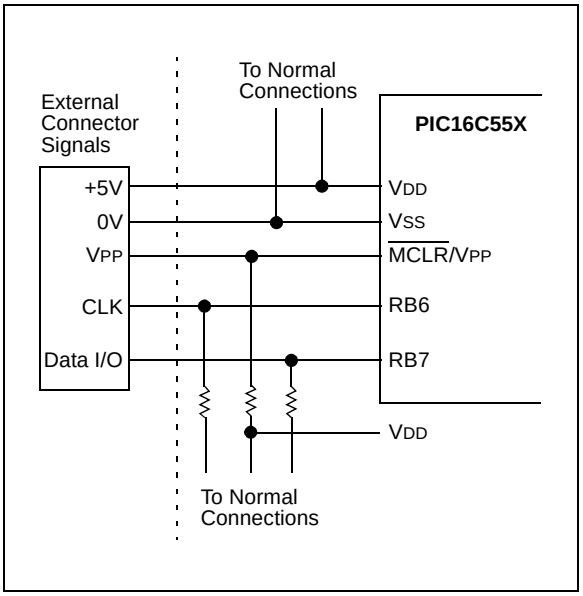
The PIC16C55X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a Program/Verify mode by holding the RB6 and RB7 pins low while raising the $\overline{MCLR}$ ($V_{PP}$) pin from $V_{IL}$ to $V_{IHH}$ (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After RESET, to place the device into Programming/ Verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14 bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X Programming Specifications (Literature #DS30228).

A typical in-circuit serial programming connection is shown in Figure 6-15.

**FIGURE 6-15: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION**



## 8.0 INSTRUCTION SET SUMMARY

Each PIC16C55X instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16C55X instruction set summary in Table 8-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 8-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the W register. If 'd' is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, 'b' represents a bit field designator which selects the number of the bit affected by the operation, while 'f' represents the number of the file in which the bit is located.

For **literal and control** operations, 'k' represents an eight or eleven bit constant or literal value.

**TABLE 8-1: OPCODE FIELD DESCRIPTIONS**

| Field | Description |
|---|---|
| f | Register file address (0x00 to 0x7F) |
| W | Working register (accumulator) |
| b | Bit address within an 8-bit file register |
| k | Literal field, constant data or label |
| x | Don't care location (= 0 or 1)<br>The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools. |
| d | Destination select; d = 0: store result in W,<br>d = 1: store result in file register f.<br>Default is d = 1 |
| label | Label name |
| TOS | Top of Stack |
| PC | Program Counter |
| PCLATH | Program Counter High Latch |
| GIE | Global Interrupt Enable bit |
| WDT | Watchdog Timer/Counter |
| TO | Timeout bit |
| PD | Power-down bit |
| dest | Destination either the W register or the specified register file location |
| [ ] | Options |
| ( ) | Contents |
| → | Assigned to |
| < > | Register bit field |
| ∈ | In the set of |
| italics | User defined term (font is courier) |

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Table 8-1 lists the instructions recognized by the MPASM™ assembler.

Figure 8-1 shows the three general formats that the instructions can have.

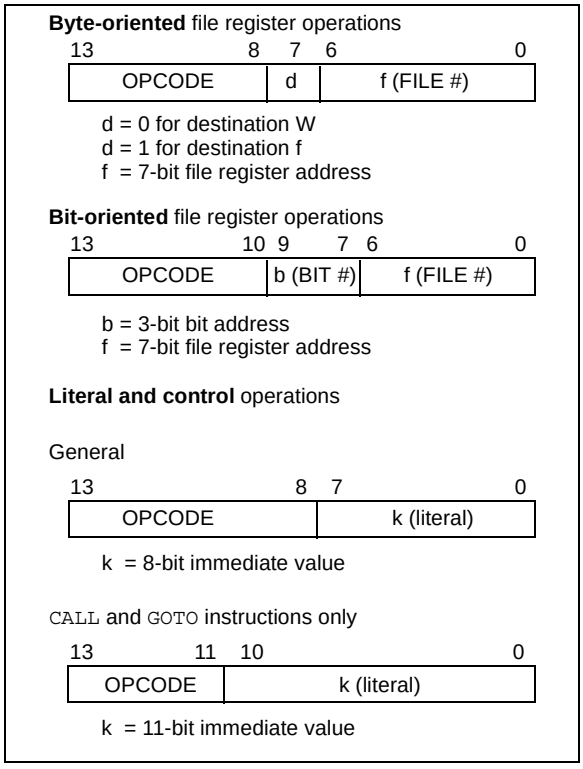
**Note:** To maintain upward compatibility with future PIC® MCU products, do not use the OPTION and TRIS instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

**FIGURE 8-1: GENERAL FORMAT FOR INSTRUCTIONS**



## 8.1 Instruction Descriptions

**ADDLW** **Add Literal and W**

Syntax: [ *label* ] ADDLW k

Operands: $0 \le k \le 255$

Operation: (W) + k → (W)

Status Affected: C, DC, Z

Encoding:

| | | | |
|---|---|---|---|
| 11 | 111x | kkkk | kkkk |

Description: The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.

Words: 1

Cycles: 1

Example
```
ADDLW   0x15
```
Before Instruction
W = 0x10
After Instruction
W = 0x25

**ADDWF** **Add W and f**

Syntax: [ *label* ] ADDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) + (f) → (dest)

Status Affected: C, DC, Z

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0111 | dfff | ffff |

Description: Add the contents of the W register with 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

Example
```
ADDWF   FSR,   0
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0xD9
FSR = 0xC2

**ANDLW** **AND Literal with W**

Syntax: [ *label* ] ANDLW k

Operands: $0 \le k \le 255$

Operation: (W) .AND. (k) → (W)

Status Affected: Z

Encoding:

| | | | |
|---|---|---|---|
| 11 | 1001 | kkkk | kkkk |

Description: The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example
```
ANDLW   0x5F
```
Before Instruction
W = 0xA3
After Instruction
W = 0x03

**ANDWF** **AND W with f**

Syntax: [ *label* ] ANDWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .AND. (f) → (dest)

Status Affected: Z

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0101 | dfff | ffff |

Description: AND the W register with 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

Example
```
ANDWF   FSR,   1
```
Before Instruction
W = 0x17
FSR = 0xC2
After Instruction
W = 0x17
FSR = 0x02

**MOVF** **Move f**

Syntax: [ *label* ] MOVF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Encoding:

| | | | |
|---|---|---|---|
| 00 | 1000 | dfff | ffff |

Description: The contents of register f is moved to a destination dependant upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example
```
MOVF    FSR,   0
```
After Instruction
W = value in FSR register
Z = 1

**MOVWF** **Move W to f**

Syntax: [ *label* ] MOVWF f

Operands: $0 \le f \le 127$

Operation: (W) → (f)

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0000 | 1fff | ffff |

Description: Move data from W register to register 'f'.

Words: 1

Cycles: 1

Example
```
MOVWF   OPTION
```
Before Instruction
OPTION = 0xFF
W = 0x4F
After Instruction
OPTION = 0x4F
W = 0x4F

**NOP** **No Operation**

Syntax: [ *label* ] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0000 | 0xx0 | 0000 |

Description: No operation.

Words: 1

Cycles: 1

Example
```
NOP
```

**OPTION** **Load Option Register**

Syntax: [ *label* ] OPTION

Operands: None

Operation: (W) → OPTION

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0000 | 0110 | 0010 |

Description: The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.

Words: 1

Cycles: 1

Example

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

**RETFIE** **Return from Interrupt**

Syntax: [ *label* ] RETFIE

Operands: None

Operation: TOS → PC,
1 → GIE

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0000 | 0000 | 1001 |

Description: Return from Interrupt. Stack is POPed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.

Words: 1

Cycles: 2

Example

```
RETFIE
```

After Interrupt
PC = TOS
GIE = 1

**RETURN** **Return from Subroutine**

Syntax: [ *label* ] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0000 | 0000 | 1000 |

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.

Words: 1

Cycles: 2

Example

```
RETURN
```

After Interrupt
PC = TOS

**RETLW** **Return with Literal in W**

Syntax: [ *label* ] RETLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W);
TOS → PC

Status Affected: None

Encoding:

| | | | |
|---|---|---|---|
| 11 | 01xx | kkkk | kkkk |

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.

Words: 1

Cycles: 2

Example

```
      CALL TABLE;W contains table
                ;offset value
      •         ;W now has table
      value
      •
      •
      ADDWF PC  ;W = offset
TABLE RETLW k1  ;Begin table
      RETLW k2  ;
      •
      •
      •
      RETLW kn  ; End of table
```

Before Instruction
W = 0x07
After Instruction
W = value of k8

**RLF** **Rotate Left f through Carry**

Syntax: [ *label* ] RLF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

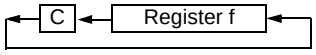
Operation: See description below

Status Affected: C

Encoding:

| | | | |
|---|---|---|---|
| 00 | 1101 | dfff | ffff |

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is stored back in register 'f'.




Words: 1

Cycles: 1

Example

```
RLF   REG1,0
```

Before Instruction
REG1 = 1110 0110
C = 0
After Instruction
REG1 = 1110 0110
W = 1100 1100
C = 1

**XORLW** **Exclusive OR Literal with W**

Syntax: [ *label* ] XORLW k

Operands: $0 \le k \le 255$

Operation: (W) .XOR. k → (W)

Status Affected: Z

Encoding:

| | | | |
|---|---|---|---|
| 11 | 1010 | kkkk | kkkk |

Description: The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.

Words: 1

Cycles: 1

Example:

```
XORLW   0xAF
```

Before Instruction

W = 0xB5

After Instruction

W = 0x1A

**XORWF** **Exclusive OR W with f**

Syntax: [ *label* ] XORWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (W) .XOR. (f) → (dest)

Status Affected: Z

Encoding:

| | | | |
|---|---|---|---|
| 00 | 0110 | dfff | ffff |

Description: Exclusive OR the contents of the W register with 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

Example

```
XORWF   REG     1
```

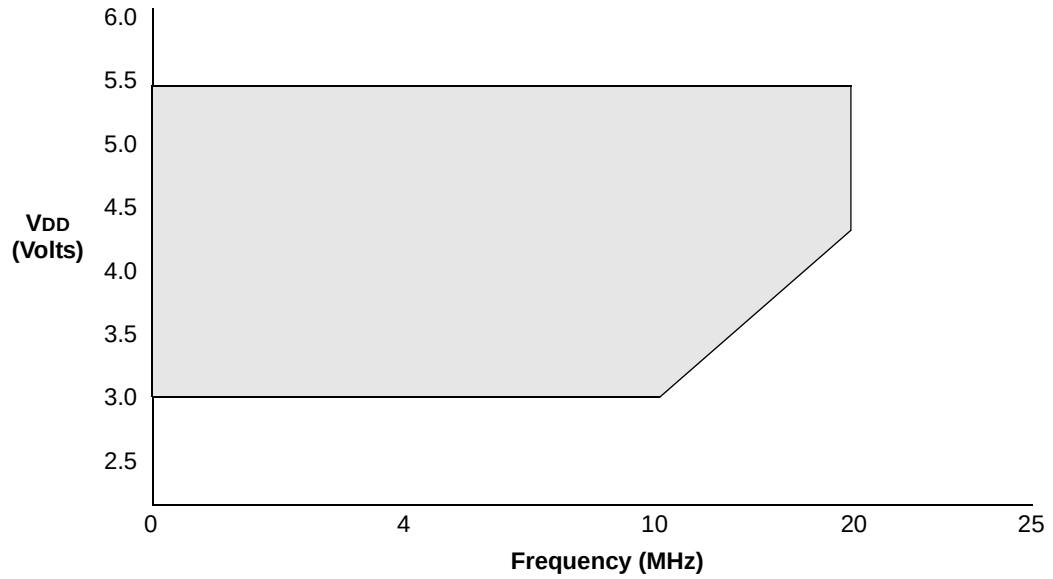
Before Instruction

REG = 0xAF
W = 0xB5

After Instruction

REG = 0x1A
W = 0xB5

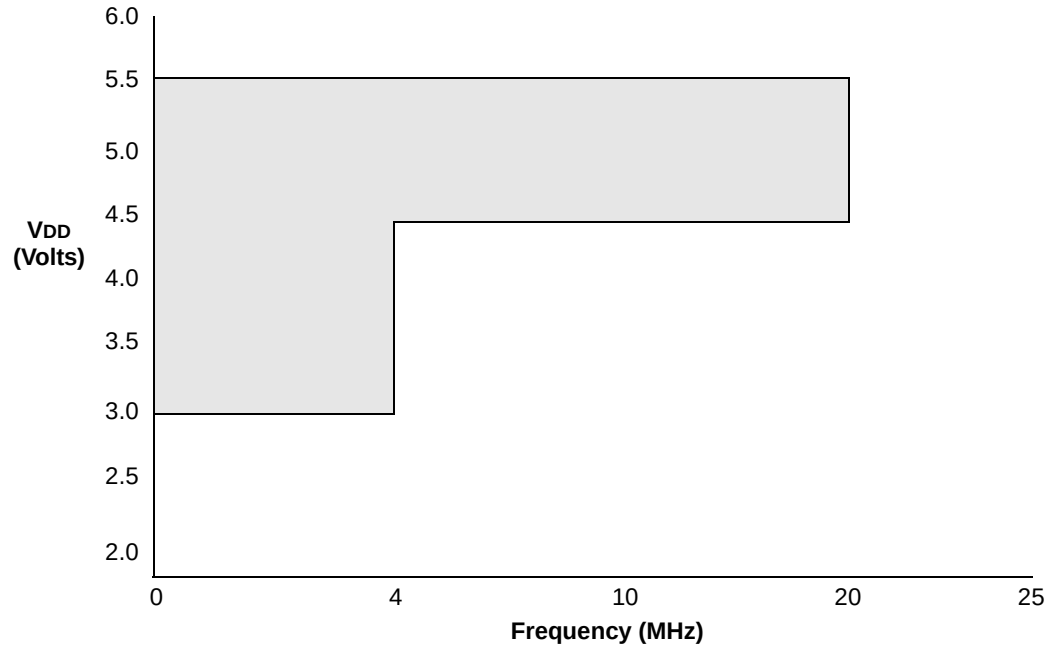
**FIGURE 10-1: VOLTAGE-FREQUENCY GRAPH, 0°C ≤ $T_A$ ≤ +70°C (COMMERCIAL TEMPS)**




**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.

**2:** The maximum rated speed of the part limits the permissible combinations of voltage and frequency. Please reference the Product Identification System section for the maximum rated speed of the parts.

**FIGURE 10-2: VOLTAGE-FREQUENCY GRAPH, -40°C ≤ $T_A$ < 0°C, +70°C < $T_A$ ≤ +125°C (OUTSIDE OF COMMERCIAL TEMPS)**

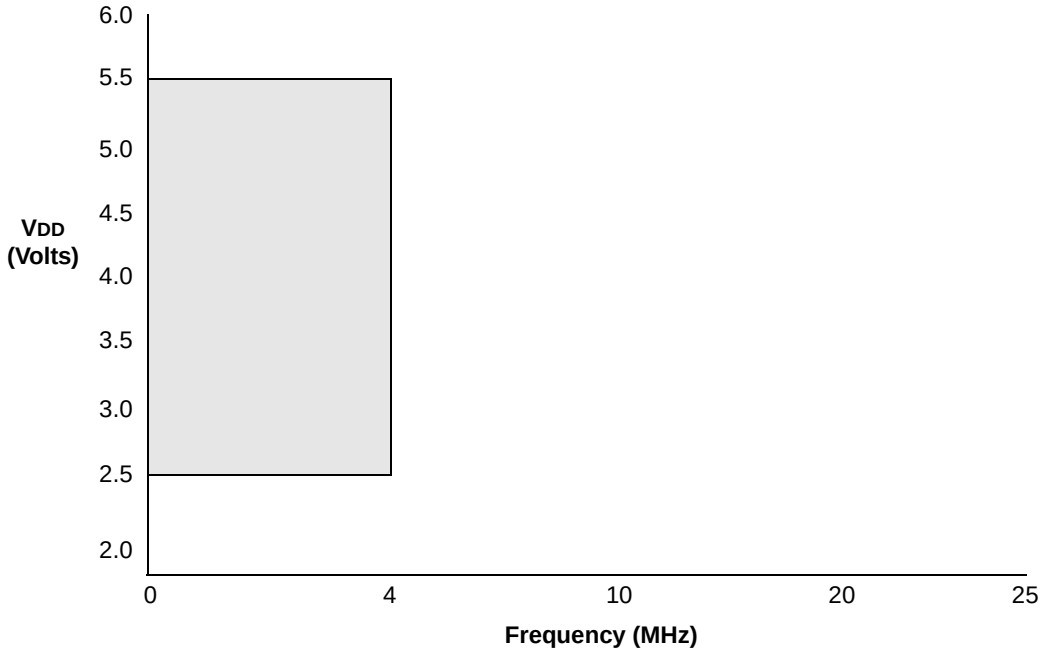



**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.

**2:** The maximum rated speed of the part limits the permissible combinations of voltage and frequency. Please reference the Product Identification System section for the maximum rated speed of the parts.

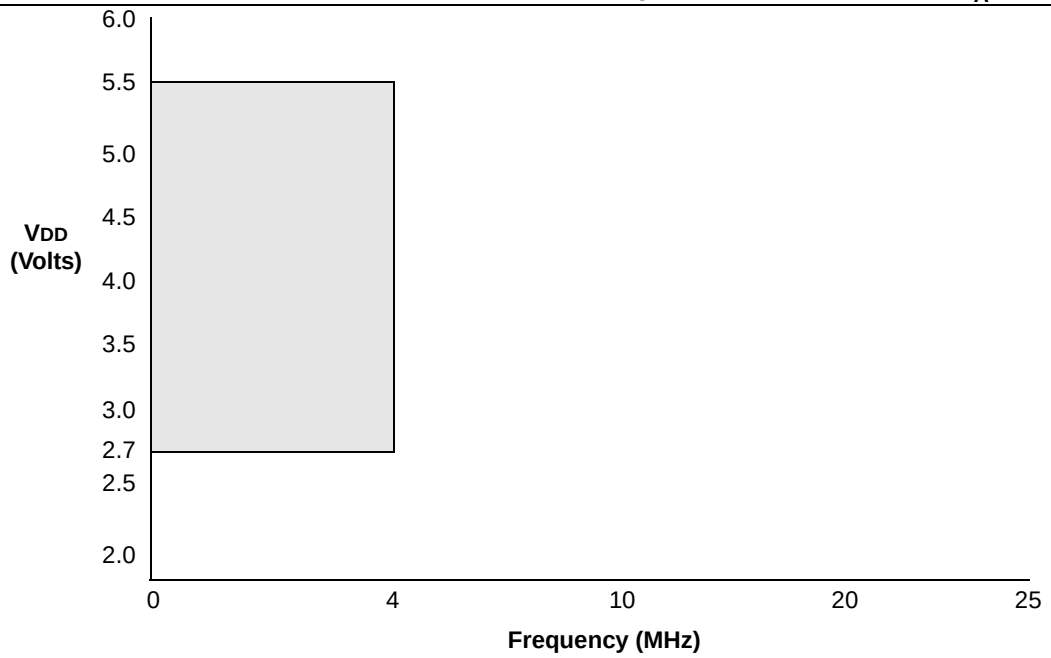
**FIGURE 10-3: VOLTAGE-FREQUENCY GRAPH, 0°C ≤ $T_A$ ≤ +85°C**




**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.

**2:** The maximum rated speed of the part limits the permissible combinations of voltage and frequency. Please reference the Product Identification System section for the maximum rated speed of the parts.

**FIGURE 10-4: PIC16LC554/557/558 VOLTAGE-FREQUENCY GRAPH, -40°C ≤ $T_A$ ≤ 0°C**




**Note 1:** The shaded region indicates the permissible combinations of voltage and frequency.

**2:** The maximum rated speed of the part limits the permissible combinations of voltage and frequency. Please reference the Product Identification System section for the maximum rated speed of the parts.

## 10.4 Timing Diagrams and Specifications

**FIGURE 10-6: EXTERNAL CLOCK TIMING**

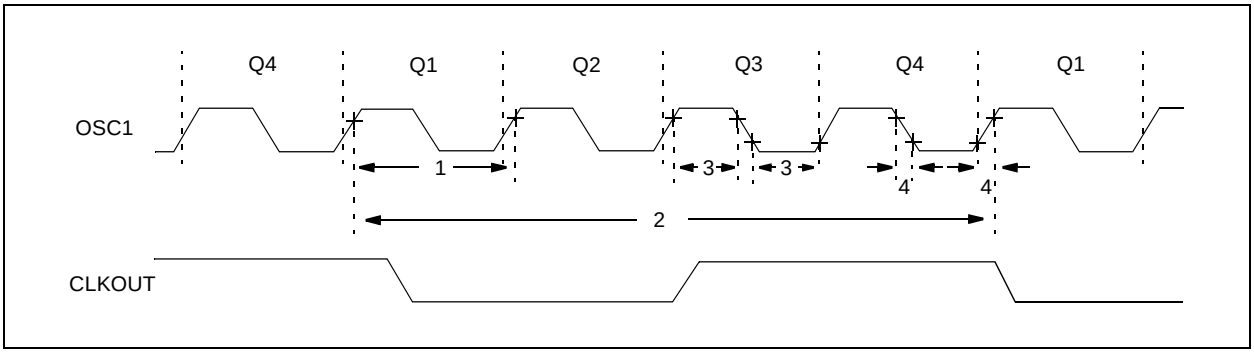



**TABLE 10-1: EXTERNAL CLOCK TIMING REQUIREMENTS**

| Parameter No. | Sym | Characteristic | Min | Typ† | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| | Fos | External CLKIN Frequency(1) | DC | — | 4 | MHz | XT and RC osc mode, VDD=5.0V |
| | | | DC | — | 20 | MHz | HS osc mode |
| | | | DC | — | 200 | kHz | LP osc mode |
| | | Oscillator Frequency(1) | DC | — | 4 | MHz | RC osc mode, VDD=5.0V |
| | | | 0.1 | — | 4 | MHz | XT osc mode |
| | | | 1 | — | 20 | MHz | HS osc mode |
| | | | DC | – | 200 | kHz | LP osc mode |
| 1 | Tosc | External CLKIN Period(1) | 250 | — | — | ns | XT and RC osc mode |
| | | | 50 | — | — | ns | HS osc mode |
| | | | 5 | — | — | µs | LP osc mode |
| | | Oscillator Period(1) | 250 | — | — | ns | RC osc mode |
| | | | 250 | — | 10,000 | ns | XT osc mode |
| | | | 50 | — | 1,000 | ns | HS osc mode |
| | | | 5 | — | — | µs | LP osc mode |
| 2 | Tcy | Instruction Cycle Time(1) | 1.0 | Fos/4 | DC | µs | TCY=FOS/4 |
| 3* | TosL, TosH | External Clock in (OSC1) High or Low Time | 100* | — | — | ns | XT osc mode |
| | | | 2* | — | — | µs | LP osc mode |
| | | | 20* | — | — | ns | HS osc mode |
| 4* | TosR, TosF | External Clock in (OSC1) Rise or Fall Time | 25* | — | — | ns | XT osc mode |
| | | | 50* | — | — | ns | LP osc mode |
| | | | 15* | — | — | ns | HS osc mode |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0 V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1 pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

**TABLE 10-2: CLKOUT AND I/O TIMING REQUIREMENTS**

| Parameter # | Sym | Characteristic | Min | Typ† | Max | Units |
|---|---|---|---|---|---|---|
| 10* | TosH2ckL | OSC1↑ to CLKOUT↓ (1) | —<br>— | 75<br>— | 200<br>400 | ns<br>ns |
| 11* | TosH2ckH | OSC1↑ to CLKOUT↑ (1) | —<br>— | 75<br>— | 200<br>400 | ns<br>ns |
| 12* | TckR | CLKOUT rise time(1) | —<br>— | 35<br>— | 100<br>200 | ns<br>ns |
| 13* | TckF | CLKOUT fall time(1) | —<br>— | 35<br>— | 100<br>200 | ns<br>ns |
| 14* | TckL2ioV | CLKOUT ↓ to Port out valid(1) | — | — | 20 | ns |
| 15* | TioV2ckH | Port in valid before CLKOUT ↑ (1) | Tosc +200 ns<br>Tosc +400 ns | —<br>— | —<br>— | ns<br>ns |
| 16* | TckH2ioI | Port in hold after CLKOUT ↑ (1) | 0 | — | — | ns |
| 17* | TosH2ioV | OSC1↑ (Q1 cycle) to Port out valid | —<br>— | 50 | 150<br>300 | ns<br>ns |
| 18* | TosH2ioI | OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time) | 100<br>200 | —<br>— | —<br>— | ns<br>ns |
| 19* | TioV2osH | Port input valid to OSC1↑ (I/O in setup time) | 0 | — | — | ns |
| 20* | TioR | Port output rise time | —<br>— | 10<br>— | 40<br>80 | ns<br>ns |
| 21* | TioF | Port output fall time | —<br>— | 10<br>— | 40<br>80 | ns<br>ns |
| 22* | Tinp | RB0/INT pin high or low time | 25<br>40 | —<br>— | —<br>— | ns<br>ns |
| 23* | Trbp | RB<7:4> change interrupt high or low time | Tcy | — | — | ns |

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

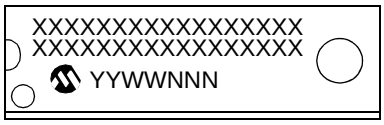
**NOTES:**

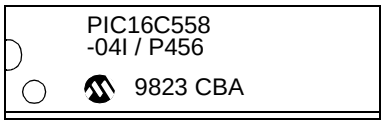
## 11.0 PACKAGING INFORMATION

### 11.1 Package Marking Information

18-Lead PDIP

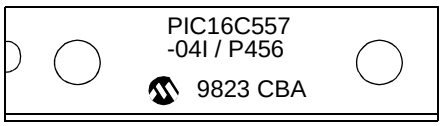



Example

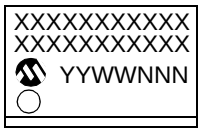



28-Lead PDIP

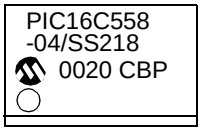

Example




20-Lead SSOP




Example




28-Lead SSOP

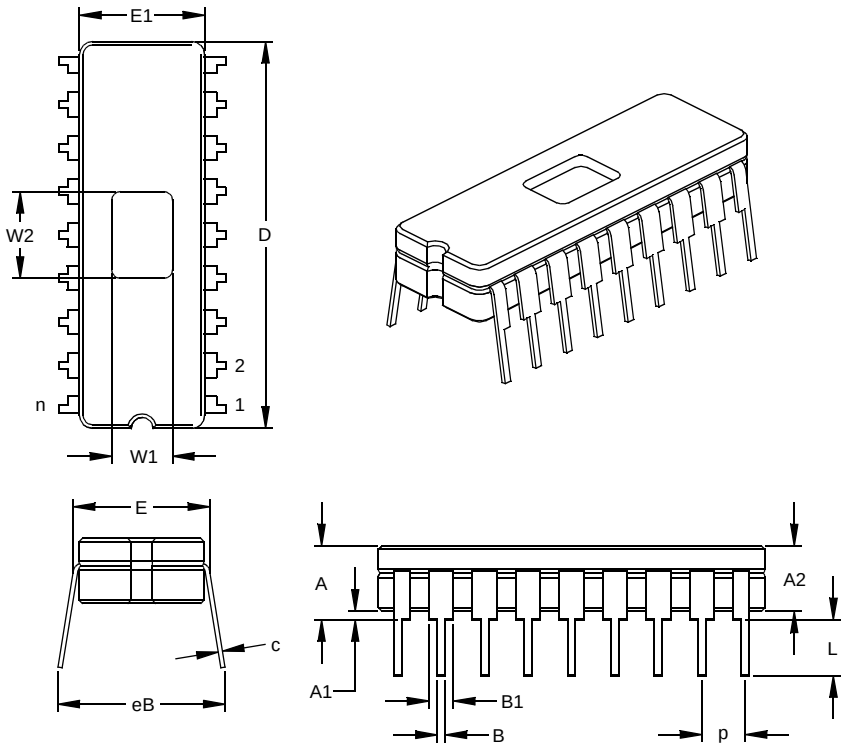

Example


| | | |
|---|---|---|
| **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.

### 18-Lead Ceramic Dual In-line with Window (JW) – 300 mil (CERDIP)

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




| Units | | INCHES* | | | MILLIMETERS | | |
|---|---|---|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX | MIN | NOM | MAX |
| Number of Pins | n | | 18 | | | 18 | |
| Pitch | p | | .100 | | | 2.54 | |
| Top to Seating Plane | A | .170 | .183 | .195 | 4.32 | 4.64 | 4.95 |
| Ceramic Package Height | A2 | .155 | .160 | .165 | 3.94 | 4.06 | 4.19 |
| Standoff | A1 | .015 | .023 | .030 | 0.38 | 0.57 | 0.76 |
| Shoulder to Shoulder Width | E | .300 | .313 | .325 | 7.62 | 7.94 | 8.26 |
| Ceramic Pkg. Width | E1 | .285 | .290 | .295 | 7.24 | 7.37 | 7.49 |
| Overall Length | D | .880 | .900 | .920 | 22.35 | 22.86 | 23.37 |
| Tip to Seating Plane | L | .125 | .138 | .150 | 3.18 | 3.49 | 3.81 |
| Lead Thickness | c | .008 | .010 | .012 | 0.20 | 0.25 | 0.30 |
| Upper Lead Width | B1 | .050 | .055 | .060 | 1.27 | 1.40 | 1.52 |
| Lower Lead Width | B | .016 | .019 | .021 | 0.41 | 0.47 | 0.53 |
| Overall Row Spacing § | eB | .345 | .385 | .425 | 8.76 | 9.78 | 10.80 |
| Window Width | W1 | .130 | .140 | .150 | 3.30 | 3.56 | 3.81 |
| Window Length | W2 | .190 | .200 | .210 | 4.83 | 5.08 | 5.33 |

* Controlling Parameter
§ Significant Characteristic
JEDEC Equivalent: MO-036
Drawing No. C04-010