**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

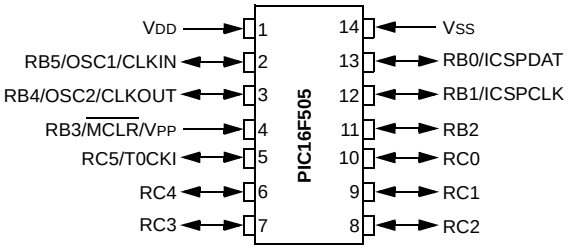
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded"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

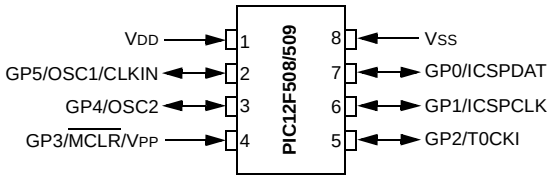
| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 4MHz |
| Connectivity | - |
| Peripherals | POR, WDT |
| Number of I/O | 5 |
| Program Memory Size | 768B (512 x 12) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 25 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 5.5V |
| Data Converters | - |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 8-SOIC (0.154", 3.90mm Width) |
| Supplier Device Package | 8-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic12f508t-i-sn |

## Pin Diagrams

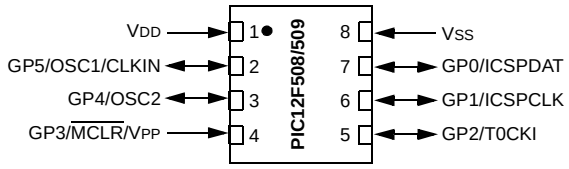
**PDIP, SOIC, TSSOP**



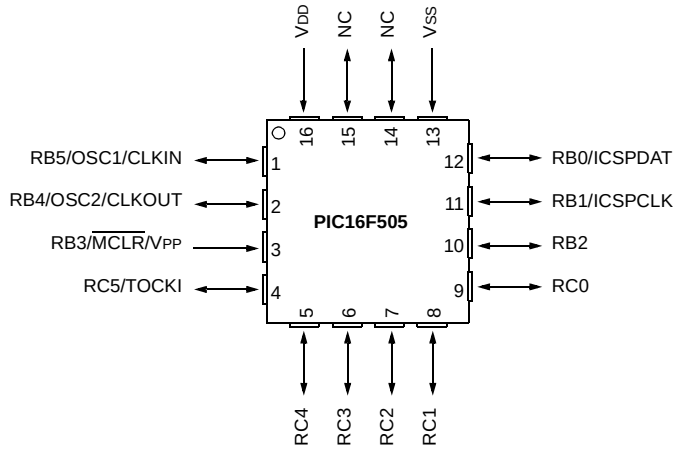

**PDIP, SOIC, MSOP**




**DFN**




## PIC16F505 16-Pin Diagram (QFN)



**NOTES:**

## 4.0 MEMORY ORGANIZATION

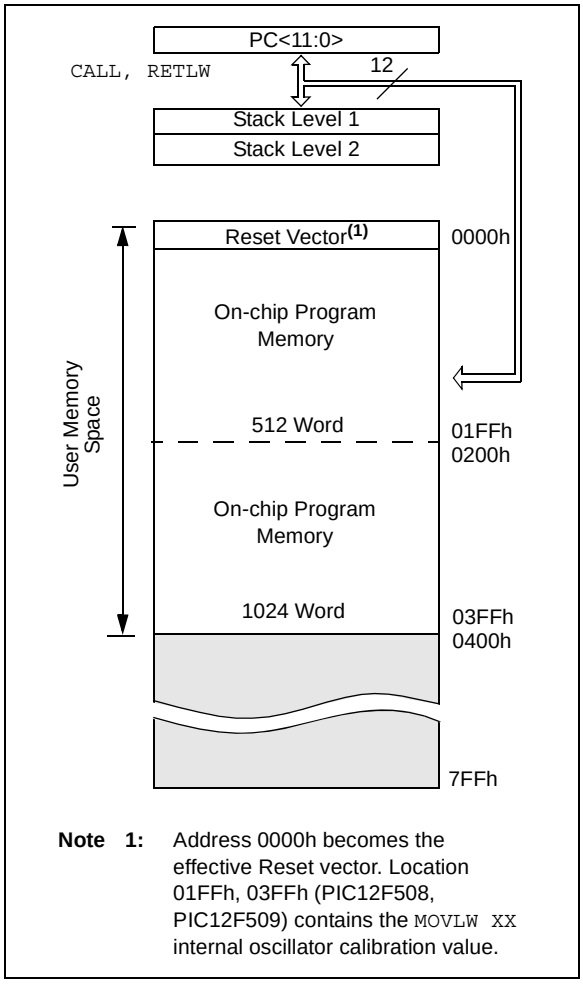
The PIC12F508/509/16F505 memories are organized into program memory and data memory. For devices with more than 512 bytes of program memory, a paging scheme is used. Program memory pages are accessed using one STATUS register bit. For the PIC12F509 and PIC16F505, with data memory register files of more than 32 registers, a banking scheme is used. Data memory banks are accessed using the File Select Register (FSR).

### 4.1 Program Memory Organization for the PIC12F508/509

The PIC12F508 device has a 10-bit Program Counter (PC) and PIC12F509 has a 11-bit Program Counter (PC) capable of addressing a 2K x 12 program memory space.

Only the first 512 x 12 (0000h-01FFh) for the PIC12F508, and 1K x 12 (0000h-03FFh) for the PIC12F509 are physically implemented (see Figure 4-1). Accessing a location above these boundaries will cause a wrap-around within the first 512 x 12 space (PIC12F508) or 1K x 12 space (PIC12F509). The effective Reset vector is a 0000h (see Figure 4-1). Location 01FFh (PIC12F508) and location 03FFh (PIC12F509) contain the internal clock oscillator calibration value. This value should never be overwritten.

**FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC12F508/509**




**Note 1:** Address 0000h becomes the effective Reset vector. Location 01FFh, 03FFh (PIC12F508, PIC12F509) contains the `MOVLW XX` internal oscillator calibration value.

## 4.9 Indirect Data Addressing: INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR register (FSR is a *pointer*). This is indirect addressing.

### 4.9.1 INDIRECT ADDRESSING

- Register file 07 contains the value 10h
- Register file 08 contains the value 0Ah
- Load the value 07 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 08)
- A read of the INDR register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no operation (although Status bits may be affected).

A simple program to clear RAM locations 10h-1Fh using indirect addressing is shown in Example 4-1.

**EXAMPLE 4-1: HOW TO CLEAR RAM USING INDIRECT ADDRESSING**

```
        MOVLW   0x10    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF
                        ;register
        INCF    FSR,F   ;inc pointer
        BTFSC   FSR,4   ;all done?
        GOTO    NEXT    ;NO, clear next
CONTINUE
            :           ;YES, continue
            :
```

The FSR is a 5-bit wide register. It is used in conjunction with the INDF register to indirectly address the data memory area.

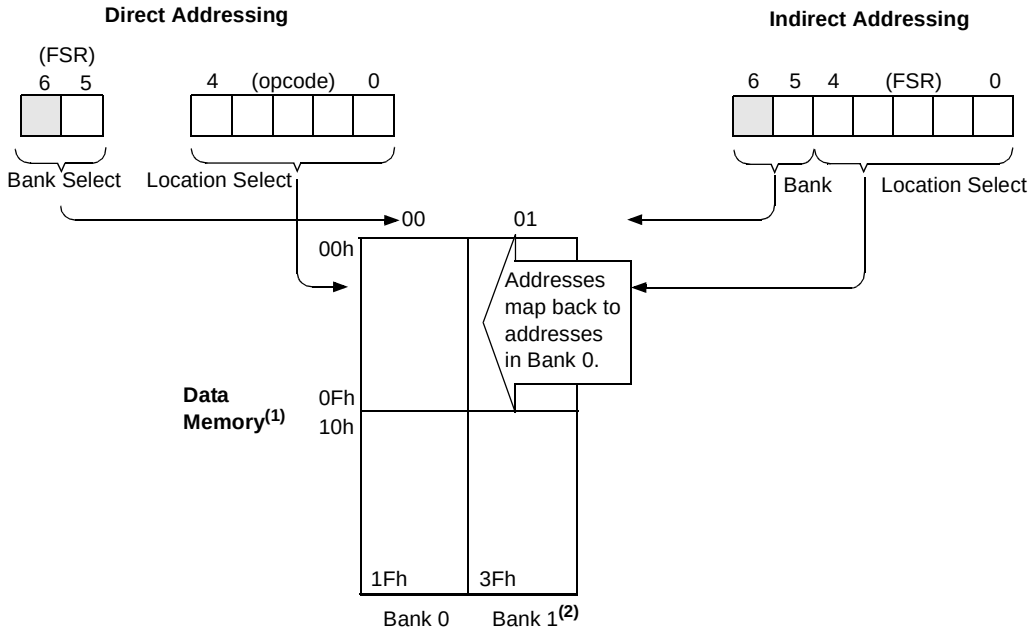
The FSR<4:0> bits are used to select data memory addresses 00h to 1Fh.

**PIC12F508** – Does not use banking. FSR <7:5> are unimplemented and read as '1's.

**PIC12F509** – Uses FSR<5>. Selects between bank 0 and bank 1. FSR<7:6> are unimplemented, read as '1'.

**PIC16F505** – Uses FSR<6:5>. Selects from bank 0 to bank 3. FSR<7> is unimplemented, read as '1'.

**FIGURE 4-7: DIRECT/INDIRECT ADDRESSING (PIC12F508/509)**




**Note 1:** For register map detail, see **Section 4.3 "Data Memory Organization"**.

**2:** PIC12F509.

**NOTES:**

## 5.0 I/O PORT

As with any other register, the I/O register(s) can be written and read under program control. However, read instructions (e.g., `MOVF PORTB,W`) always read the I/O pins independent of the pin's Input/Output modes. On Reset, all I/O ports are defined as input (inputs are at high-impedance) since the I/O control registers are all set.

| | |
|---|---|
| **Note:** | On the PIC12F508/509, I/O PORTB is referenced as GPIO. On the PIC16F505, I/O PORTB is referenced as PORTB. |

## 5.1 PORTB/GPIO

PORTB/GPIO is an 8-bit I/O register. Only the low-order 6 bits are used (RB/GP<5:0>). Bits 7 and 6 are unimplemented and read as '0's. Please note that RB3/GP3 is an input only pin. The Configuration Word can set several I/O's to alternate functions. When acting as alternate functions, the pins will read as '0' during a port read. Pins RB0/GP0, RB1/GP1, RB3/GP3 and RB4 can be configured with weak pull-ups and also for wake-up on change. The wake-up on change and weak pull-up functions are not pin selectable. If RB3/GP3/$\overline{MCLR}$ is configured as $\overline{MCLR}$, weak pull-up is always on and wake-up on change for this pin is not enabled.

## 5.2 PORTC (PIC16F505 Only)

PORTC is an 8-bit I/O register. Only the low-order 6 bits are used (RC<5:0>). Bits 7 and 6 are unimplemented and read as '0's.

| | |
|---|---|
| **Note:** | On power-up, TOCKI functionality is enabled in the OPTION register and must be disabled to allow RC5 to be used as general purpose I/O. |

## 5.3 TRIS Registers

The Output Driver Control register is loaded with the contents of the W register by executing the `TRIS f` instruction. A '1' from a TRIS register bit puts the corresponding output driver in a High-Impedance mode. A '0' puts the contents of the output data latch on the selected pins, enabling the output buffer. The exceptions are RB3/GP3, which is input only and the T0CKI pin, which may be controlled by the OPTION register. See Register 4-3 and Register 4-4.

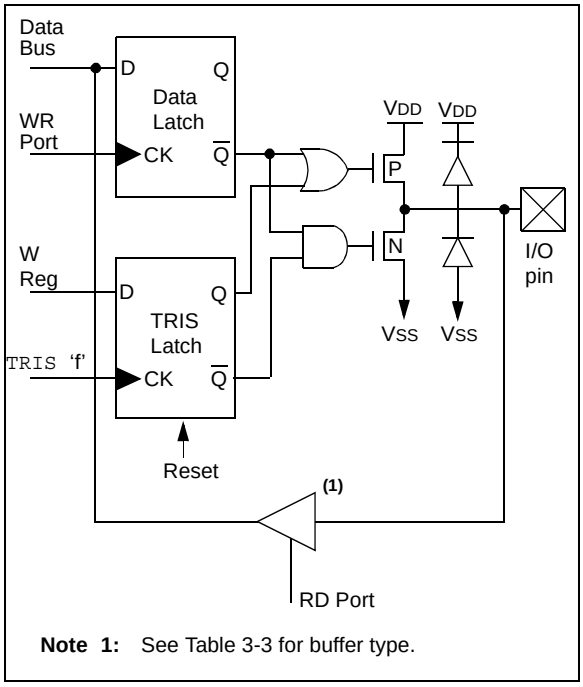
| | |
|---|---|
| **Note:** | A read of the ports reads the pins, not the output data latches. That is, if an output driver on a pin is enabled and driven high, but the external system is holding it low, a read of the port will indicate that the pin is low. |

The TRIS registers are "write-only" and are set (output drivers disabled) upon Reset.

## 5.4 I/O Interfacing

The equivalent circuit for an I/O port pin is shown in Figure 5-2. All port pins, except RB3/GP3 which is input only, may be used for both input and output operations. For input operations, these ports are non-latching. Any input must be present until read by an input instruction (e.g., `MOVF PORTB, W`). The outputs are latched and remain unchanged until the output latch is rewritten. To use a port pin as output, the corresponding direction control bit in TRIS must be cleared (= `0`). For use as an input, the corresponding TRIS bit must be set. Any I/O pin (except RB3/GP3) can be programmed individually as input or output.

**FIGURE 5-1: PIC12F508/509/16F505 EQUIVALENT CIRCUIT FOR A SINGLE I/O PIN**




**Note 1:** See Table 3-3 for buffer type.

**REGISTER 7-1: CONFIGURATION WORD FOR PIC12F508/509(1)**

| — | — | — | — | — | — | — | MCLRE | $\overline{CP}$ | WDTE | FOSC1 | FOSC0 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| bit 11 | | | | | | | | | | | bit 0 |

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

bit 11-5 **Unimplemented**: Read as '0'

bit 4 **MCLRE:** GP3/$\overline{MCLR}$ Pin Function Select bit
1 = GP3/$\overline{MCLR}$ pin function is $\overline{MCLR}$
0 = GP3/$\overline{MCLR}$ pin function is digital input, $\overline{MCLR}$ internally tied to $V_{DD}$

bit 3 **$\overline{CP}$:** Code Protection bit
1 = Code protection off
0 = Code protection on

bit 2 **WDTE:** Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 1-0 **FOSC<1:0>:** Oscillator Selection bits
11 = EXTRC = external RC oscillator
10 = INTRC = internal RC oscillator
01 = XT oscillator
00 = LP oscillator

**Note 1:** Refer to the "*PIC12F508/509 Memory Programming Specifications*" (DS41227) to determine how to access the Configuration Word. The Configuration Word is not user addressable during device operation.

**TABLE 7-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR – PIC12F508/509/16F505(2)**

| Osc Type | Resonator Freq. | Cap. Range C1 | Cap. Range C2 |
|---|---|---|---|
| LP | 32 kHz(1) | 15 pF | 15 pF |
| XT | 200 kHz<br>1 MHz<br>4 MHz | 47-68 pF<br>15 pF<br>15 pF | 47-68 pF<br>15 pF<br>15 pF |
| HS(3) | 20 MHz | 15-47 pF | 15-47 pF |

**Note 1:** For $V_{DD}$ > 4.5V, C1 = C2 ≈ 30 pF is recommended.

**2:** These values are for design guidance only. Rs may be required to avoid overdriving crystals with low drive level specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

**3:** PIC16F505 only.

### 7.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator or a simple oscillator circuit with TTL gates can be used as an external crystal oscillator circuit. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used: one with parallel resonance, or one with series resonance.

Figure 7-3 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180-degree phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This circuit could be used for external oscillator designs.

**FIGURE 7-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT**

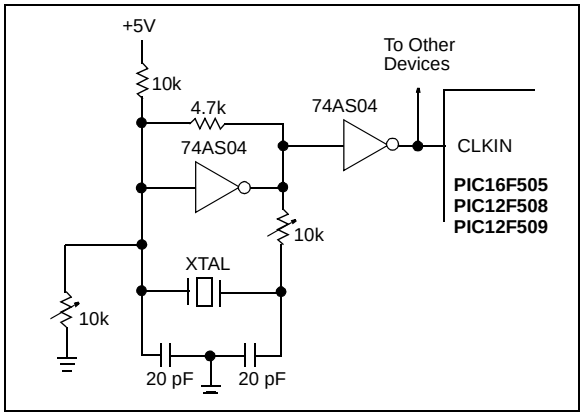
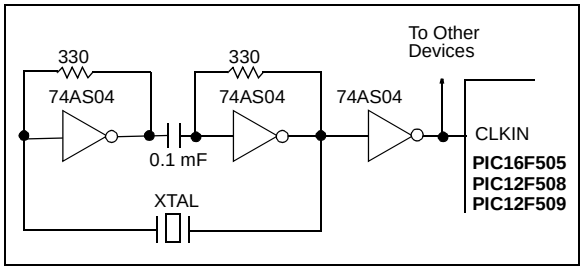



Figure 7-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 Ω resistors provide the negative feedback to bias the inverters in their linear region.

**FIGURE 7-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT**




### 7.2.4 EXTERNAL RC OSCILLATOR

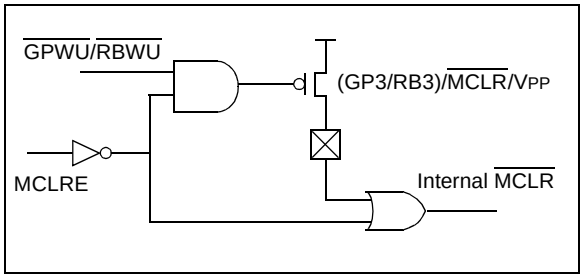
For timing insensitive applications, the RC device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor ($R_{EXT}$) and capacitor ($C_{EXT}$) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit-to-unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low $C_{EXT}$ values. The user also needs to take into account variation due to tolerance of external R and C components used.

Figure 7-5 shows how the R/C combination is connected to the PIC12F508/509/16F505 devices. For $R_{EXT}$ values below 3.0 kΩ, the oscillator operation may become unstable, or stop completely. For very high $R_{EXT}$ values (e.g., 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend keeping $R_{EXT}$ between 5.0 kΩ and 100 kΩ.

### 7.3.1 MCLR ENABLE

This Configuration bit, when unprogrammed (left in the '1' state), enables the external MCLR function. When programmed, the MCLR function is tied to the internal VDD and the pin is assigned to be an input only. See Figure 7-6.

**FIGURE 7-6: MCLR SELECT**




## 7.4 Power-on Reset (POR)

The PIC12F508/509/16F505 devices incorporate an on-chip Power-on Reset (POR) circuitry, which provides an internal chip Reset for most power-up situations.

The on-chip POR circuit holds the chip in Reset until VDD has reached a high enough level for proper operation. To take advantage of the internal POR, program the (GP3/RB3)/MCLR/VPP pin as MCLR and tie through a resistor to VDD, or program the pin as (GP3/RB3). An internal weak pull-up resistor is implemented using a transistor (refer to Table 10-2 for the pull-up resistor ranges). This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for VDD is specified. See **Section 10.0 "Electrical Characteristics"** for details.

When the devices start normal operation (exit the Reset condition), device operating parameters (voltage, frequency, temperature,...) must be met to ensure operation. If these conditions are not met, the devices must be held in Reset until the operating parameters are met.

A simplified block diagram of the on-chip Power-on Reset circuit is shown in Figure 7-7.

The Power-on Reset circuit and the Device Reset Timer (see **Section 7.5 "Device Reset Timer (DRT)"**) circuit are closely related. On power-up, the Reset latch is set and the DRT is reset. The DRT timer begins counting once it detects MCLR to be high. After the time-out period, which is typically 18 ms, it will reset the Reset latch and thus end the on-chip Reset signal.

A power-up example where MCLR is held low is shown in Figure 7-8. VDD is allowed to rise and stabilize before bringing MCLR high. The chip will actually come out of Reset TDRT msec after MCLR goes high.

In Figure 7-9, the on-chip Power-on Reset feature is being used (MCLR and VDD are tied together or the pin is programmed to be (GP3/RB3). The VDD is stable before the start-up timer times out and there is no problem in getting a proper Reset. However, Figure 7-10 depicts a problem situation where VDD rises too slowly. The time between when the DRT senses that MCLR is high and when MCLR and VDD actually reach their full value, is too long. In this situation, when the start-up timer times out, VDD has not reached the VDD (min) value and the chip may not function correctly. For such situations, we recommend that external RC circuits be used to achieve longer POR delay times (Figure 7-9).

| | |
|---|---|
| **Note:** | When the devices start normal operation (exit the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met. |

For additional information, refer to Application Notes AN522 *"Power-Up Considerations"* (DS00522) and AN607 *"Power-up Trouble Shooting"* (DS00607).

## 7.9 Power-down Mode (Sleep)

A device may be powered down (Sleep) and later powered up (wake-up from Sleep).

### 7.9.1 SLEEP

The Power-Down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{TO}$ bit (STATUS<4>) is set, the $\overline{PD}$ bit (STATUS<3>) is cleared and the oscillator driver is turned off. The I/O ports maintain the status they had before the SLEEP instruction was executed (driving high, driving low or high-impedance).

| | |
|---|---|
| **Note:** | A Reset generated by a WDT time-out does not drive the $\overline{MCLR}$ pin low. |

For lowest current consumption while powered down, the T0CKI input should be at $V_{DD}$ or $V_{SS}$ and the (GP3/RB3)/$\overline{MCLR}$/$V_{PP}$ pin must be at a logic high level if $\overline{MCLR}$ is enabled.

### 7.9.2 WAKE-UP FROM SLEEP

The device can wake-up from Sleep through one of the following events:

1. An external Reset input on (GP3/RB3)/$\overline{MCLR}$/$V_{PP}$ pin, when configured as $\overline{MCLR}$.
2. A Watchdog Timer time-out Reset (if WDT was enabled).
3. A change on input pin GP0/RB0, GP1/RB1, GP3/RB3 or RB4 when wake-up on change is enabled.

These events cause a device Reset. The $\overline{TO}$, $\overline{PD}$ and GPWUF/RBWUF bits can be used to determine the cause of device Reset. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up). The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The GPWUF/RBWUF bit indicates a change in state while in Sleep at pins GP0/RB0, GP1/RB1, GP3/RB3 or RB4 (since the last file or bit operation on GP/RB port).

| | |
|---|---|
| **Note:** | **Caution:** Right before entering Sleep, read the input pins. When in Sleep, wake-up occurs when the values at the pins change from the state they were in at the last reading. If a wake-up on change occurs and the pins are not read before re-entering Sleep, a wake-up will occur immediately even if no pins change while in Sleep mode. |

The WDT is cleared when the device wakes from Sleep, regardless of the wake-up source.

## 7.10 Program Verification/Code Protection

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

The first 64 locations and the last location (OSCCAL) can be read, regardless of the code protection bit setting.

The last memory location can be read regardless of the code protection bit setting on the PIC12F508/509/16F505 devices.

## 7.11 ID Locations

Four memory locations 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.

Use only the lower 4 bits of the ID locations and always program the upper 8 bits as '0's.

## 7.12 In-Circuit Serial Programming™

The PIC12F508/509/16F505 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 devices are placed into a Program/Verify mode by holding the GP1/RB1 and GP0/RB0 pins low while raising the $\overline{MCLR}$ ($V_{PP}$) pin from $V_{IL}$ to $V_{IHH}$ (see programming specification). GP1/RB1 becomes the programming clock and GP0/RB0 becomes the programming data. Both GP1/RB1 and GP0/RB0 are Schmitt Trigger inputs in this mode.

After Reset, 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 PIC12F508/509/16F505 Programming Specifications.

A typical In-Circuit Serial Programming connection is shown in Figure 7-15.

## 8.0 INSTRUCTION SET SUMMARY

The PIC16 instruction set is highly orthogonal and is comprised of three basic categories.

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

Each PIC16 instruction is a 12-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 formats for each of the categories is presented in Figure 8-1, while the various opcode fields are summarized in Table 8-1.

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 '0', the result is placed in the W register. If 'd' is '1', 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 8 or 9-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;<br>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 |
| WDT | Watchdog Timer counter |
| $\overline{TO}$ | Time-out bit |
| $\overline{PD}$ | 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) |

All instructions are executed within a 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. 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.

Figure 8-1 shows the three general formats that the instructions can have. All examples in the figure use the following format to represent a hexadecimal number:

0xhhh

where 'h' signifies a hexadecimal digit.

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

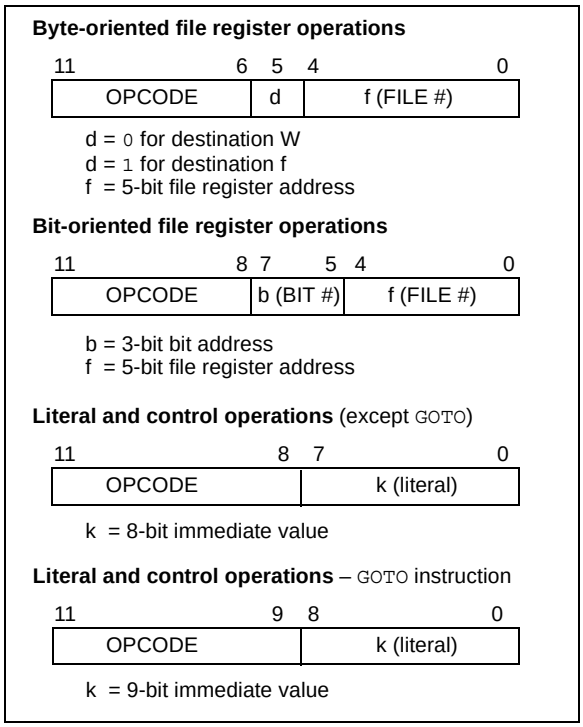


**TABLE 8-2: INSTRUCTION SET SUMMARY**

| Mnemonic, Operands | | Description | Cycles | 12-Bit Opcode | | | Status Affected | Notes |
|---|---|---|---|---|---|---|---|---|
| | | | | MSb | | LSb | | |
| ADDWF | f, d | Add W and f | 1 | 0001 | 11df | ffff | C, DC, Z | **1, 2, 4** |
| ANDWF | f, d | AND W with f | 1 | 0001 | 01df | ffff | Z | **2, 4** |
| CLRF | f | Clear f | 1 | 0000 | 011f | ffff | Z | **4** |
| CLRW | — | Clear W | 1 | 0000 | 0100 | 0000 | Z | |
| COMF | f, d | Complement f | 1 | 0010 | 01df | ffff | Z | |
| DECF | f, d | Decrement f | 1 | 0000 | 11df | ffff | Z | **2, 4** |
| DECFSZ | f, d | Decrement f, Skip if 0 | 1**(2)** | 0010 | 11df | ffff | None | **2, 4** |
| INCF | f, d | Increment f | 1 | 0010 | 10df | ffff | Z | **2, 4** |
| INCFSZ | f, d | Increment f, Skip if 0 | 1**(2)** | 0011 | 11df | ffff | None | **2, 4** |
| IORWF | f, d | Inclusive OR W with f | 1 | 0001 | 00df | ffff | Z | **2, 4** |
| MOVF | f, d | Move f | 1 | 0010 | 00df | ffff | Z | **2, 4** |
| MOVWF | f | Move W to f | 1 | 0000 | 001f | ffff | None | **1, 4** |
| NOP | — | No Operation | 1 | 0000 | 0000 | 0000 | None | |
| RLF | f, d | Rotate left f through Carry | 1 | 0011 | 01df | ffff | C | **2, 4** |
| RRF | f, d | Rotate right f through Carry | 1 | 0011 | 00df | ffff | C | **2, 4** |
| SUBWF | f, d | Subtract W from f | 1 | 0000 | 10df | ffff | C, DC, Z | **1, 2, 4** |
| SWAPF | f, d | Swap f | 1 | 0011 | 10df | ffff | None | **2, 4** |
| XORWF | f, d | Exclusive OR W with f | 1 | 0001 | 10df | ffff | Z | **2, 4** |
| **BIT-ORIENTED FILE REGISTER OPERATIONS** | | | | | | | | |
| BCF | f, b | Bit Clear f | 1 | 0100 | bbbf | ffff | None | **2, 4** |
| BSF | f, b | Bit Set f | 1 | 0101 | bbbf | ffff | None | **2, 4** |
| BTFSC | f, b | Bit Test f, Skip if Clear | 1**(2)** | 0110 | bbbf | ffff | None | |
| BTFSS | f, b | Bit Test f, Skip if Set | 1**(2)** | 0111 | bbbf | ffff | None | |
| **LITERAL AND CONTROL OPERATIONS** | | | | | | | | |
| ANDLW | k | AND literal with W | 1 | 1110 | kkkk | kkkk | Z | |
| CALL | k | Call Subroutine | 2 | 1001 | kkkk | kkkk | None | **1** |
| CLRWDT | — | Clear Watchdog Timer | 1 | 0000 | 0000 | 0100 | $\overline{TO}$, $\overline{PD}$ | |
| GOTO | k | Unconditional branch | 2 | 101k | kkkk | kkkk | None | |
| IORLW | k | Inclusive OR literal with W | 1 | 1101 | kkkk | kkkk | Z | |
| MOVLW | k | Move literal to W | 1 | 1100 | kkkk | kkkk | None | |
| OPTION | — | Load OPTION register | 1 | 0000 | 0000 | 0010 | None | |
| RETLW | k | Return, place literal in W | 2 | 1000 | kkkk | kkkk | None | |
| SLEEP | — | Go into Standby mode | 1 | 0000 | 0000 | 0011 | $\overline{TO}$, $\overline{PD}$ | |
| TRIS | f | Load TRIS register | 1 | 0000 | 0000 | 0fff | None | **3** |
| XORLW | k | Exclusive OR literal to W | 1 | 1111 | kkkk | kkkk | Z | |

**Note 1:** The 9th bit of the program counter will be forced to a '0' by any instruction that writes to the PC except for GOTO. See **Section 4.7 "Program Counter"**.

**2:** 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'.

**3:** The instruction TRIS f, where f = 6, causes the contents of the W register to be written to the tri-state latches of PORTB. A '1' forces the pin to a high-impedance state and disables the output buffers.

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

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

| | |
|---|---|
| Syntax: | [ *label* ] ADDWF f,d |
| Operands: | 0 ≤ f ≤ 31<br>d ∈ [0,1] |
| Operation: | (W) + (f) → (dest) |
| Status Affected: | C, DC, Z |
| Description: | Add the contents of the W register and 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'. |

**ANDLW** **AND literal with W**

| | |
|---|---|
| Syntax: | [ *label* ] ANDLW k |
| Operands: | 0 ≤ k ≤ 255 |
| Operation: | (W).AND. (k) → (W) |
| Status Affected: | Z |
| Description: | The contents of the W register are AND'ed with the eight-bit literal 'k'. The result is placed in the W register. |

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

| | |
|---|---|
| Syntax: | [ *label* ] ANDWF f,d |
| Operands: | 0 ≤ f ≤ 31<br>d ∈ [0,1] |
| Operation: | (W) .AND. (f) → (dest) |
| Status Affected: | Z |
| Description: | The contents of the W register are AND'ed 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'. |

**BCF** **Bit Clear f**

| | |
|---|---|
| Syntax: | [ *label* ] BCF f,b |
| Operands: | 0 ≤ f ≤ 31<br>0 ≤ b ≤ 7 |
| Operation: | 0 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is cleared. |

**BSF** **Bit Set f**

| | |
|---|---|
| Syntax: | [ *label* ] BSF f,b |
| Operands: | 0 ≤ f ≤ 31<br>0 ≤ b ≤ 7 |
| Operation: | 1 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is set. |

**BTFSC** **Bit Test f, Skip if Clear**

| | |
|---|---|
| Syntax: | [ *label* ] BTFSC f,b |
| Operands: | 0 ≤ f ≤ 31<br>0 ≤ b ≤ 7 |
| Operation: | skip if (f<b>) = 0 |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '0', then the next instruction is skipped.<br>If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a two-cycle instruction. |

## 9.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C18 and MPLAB C30 C Compilers
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
  - PICkit™ 2 Development Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

### 9.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Visual device initializer for easy register initialization
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (assembly or C)
  - Mixed assembly and C
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

**TABLE 10-3: EXTERNAL CLOCK TIMING REQUIREMENTS – PIC12F508/509/16F505**

| AC CHARACTERISTICS | | | Standard Operating Conditions (unless otherwise specified)<br>Operating Temperature -40°C ≤ TA ≤ +85°C (industrial),<br>-40°C ≤ TA ≤ +125°C (extended)<br>Operating Voltage VDD range is described in **Section 10.1 "Power-on Reset (POR)"** | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym.** | **Characteristic** | **Min.** | **Typ(1)** | **Max.** | **Units** | **Conditions** |
| 1A | FOSC | External CLKIN Frequency(2) | DC | — | 4 | MHz | XT Oscillator mode |
| | | | DC | — | 20 | MHz | EC, HS Oscillator mode (PIC16F505 only) |
| | | | DC | — | 200 | kHz | LP Oscillator mode |
| | | Oscillator Frequency(2) | — | — | 4 | MHz | EXTRC Oscillator mode |
| | | | 0.1 | — | 4 | MHz | XT Oscillator mode |
| | | | 4 | — | 20 | MHz | HS Oscillator mode (PIC16F505 only) |
| | | | — | — | 200 | kHz | LP Oscillator mode |
| 1 | TOSC | External CLKIN Period(2) | 250 | — | — | ns | XT Oscillator mode |
| | | | 50 | — | — | ns | EC, HS Oscillator mode (PIC16F505 only) |
| | | | 5 | — | — | µs | LP Oscillator mode |
| | | Oscillator Period(2) | 250 | — | — | ns | EXTRC Oscillator mode |
| | | | 250 | — | 10,000 | ns | XT Oscillator mode |
| | | | 50 | — | 250 | ns | HS Oscillator mode (PIC16F505 only) |
| | | | 5 | — | — | µs | LP Oscillator mode |
| 2 | TCY | Instruction Cycle Time | 200 | 4/FOSC | — | ns | |
| 3 | TosL, TosH | Clock in (OSC1) Low or High Time | 50* | — | — | ns | XT Oscillator |
| | | | 2* | — | — | µs | LP Oscillator |
| | | | 10* | — | — | ns | EC, HS Oscillator (PIC16F505 only) |
| 4 | TosR, TosF | Clock in (OSC1) Rise or Fall Time | — | — | 25* | ns | XT Oscillator |
| | | | — | — | 50* | ns | LP Oscillator |
| | | | — | — | 15* | ns | EC, HS Oscillator (PIC16F505 only) |

* These parameters are characterized but not tested.

**Note 1:** Data in the Typical ("Typ") column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**2:** 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. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

**FIGURE 11-4: MAXIMUM IPD VS. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)**

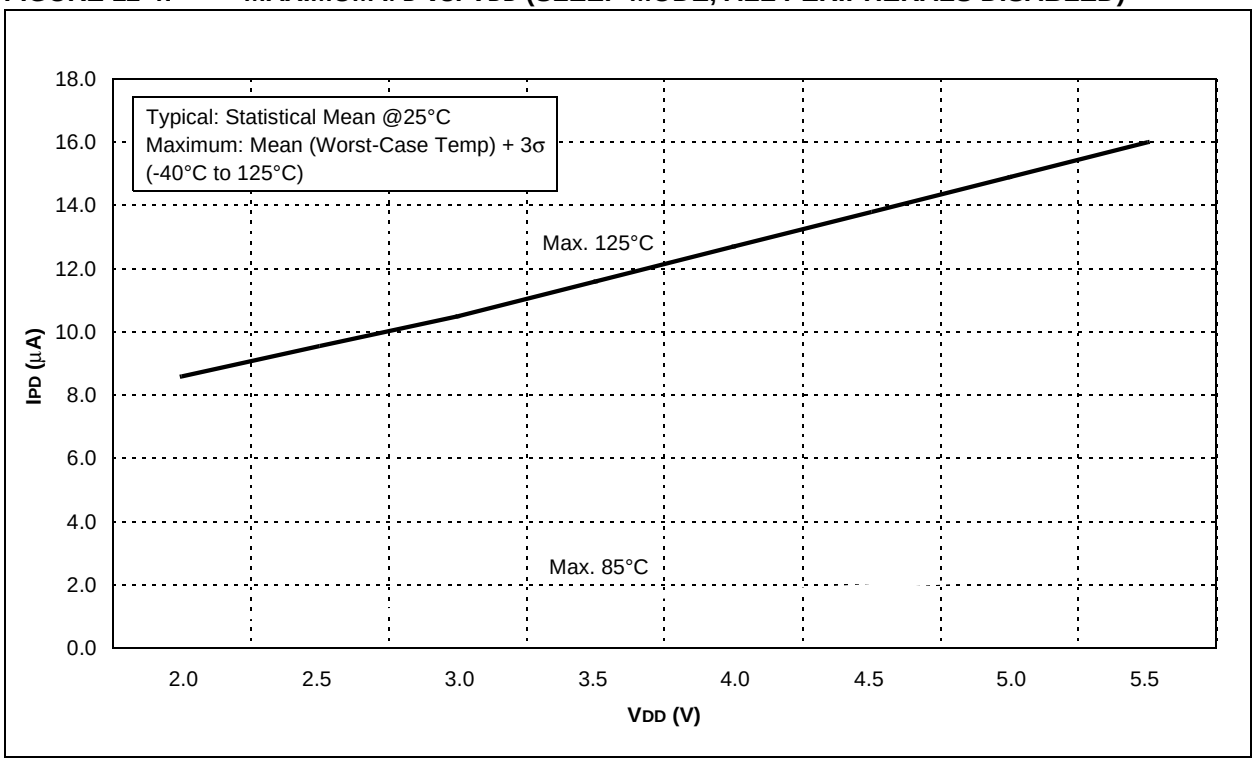



**FIGURE 11-5: TYPICAL WDT IPD VS. VDD**

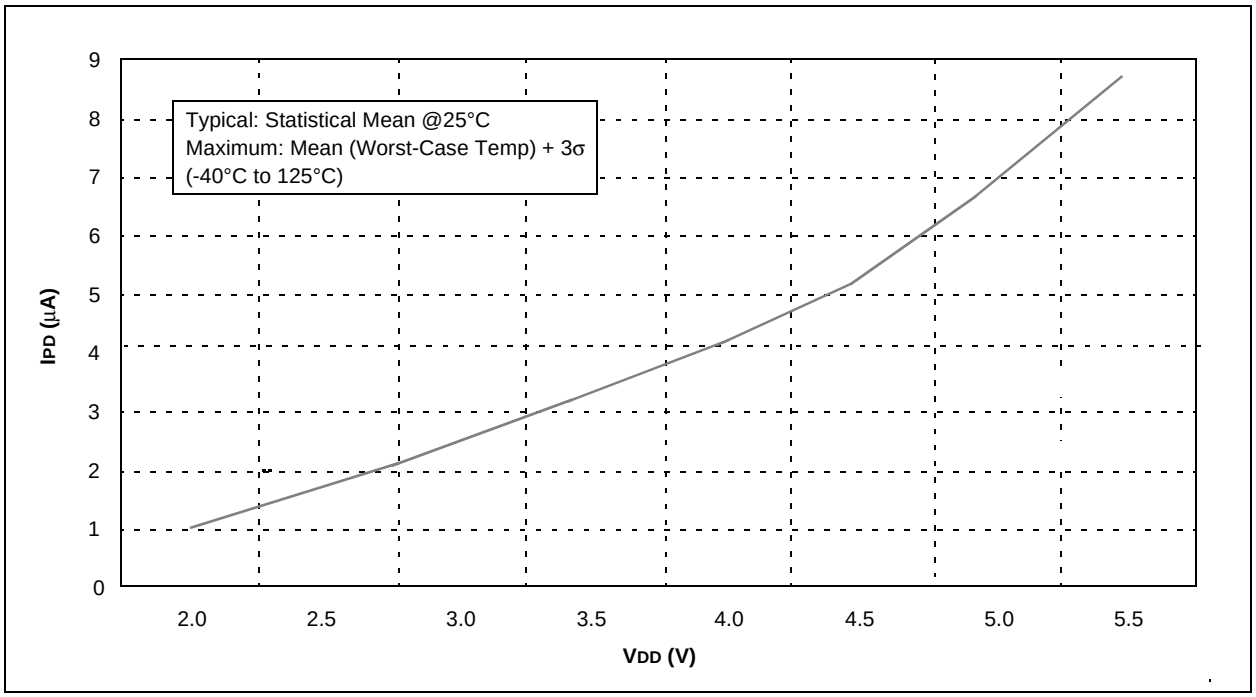


**FIGURE 11-8: VOL vs. IOL OVER TEMPERATURE (VDD = 3.0V)**

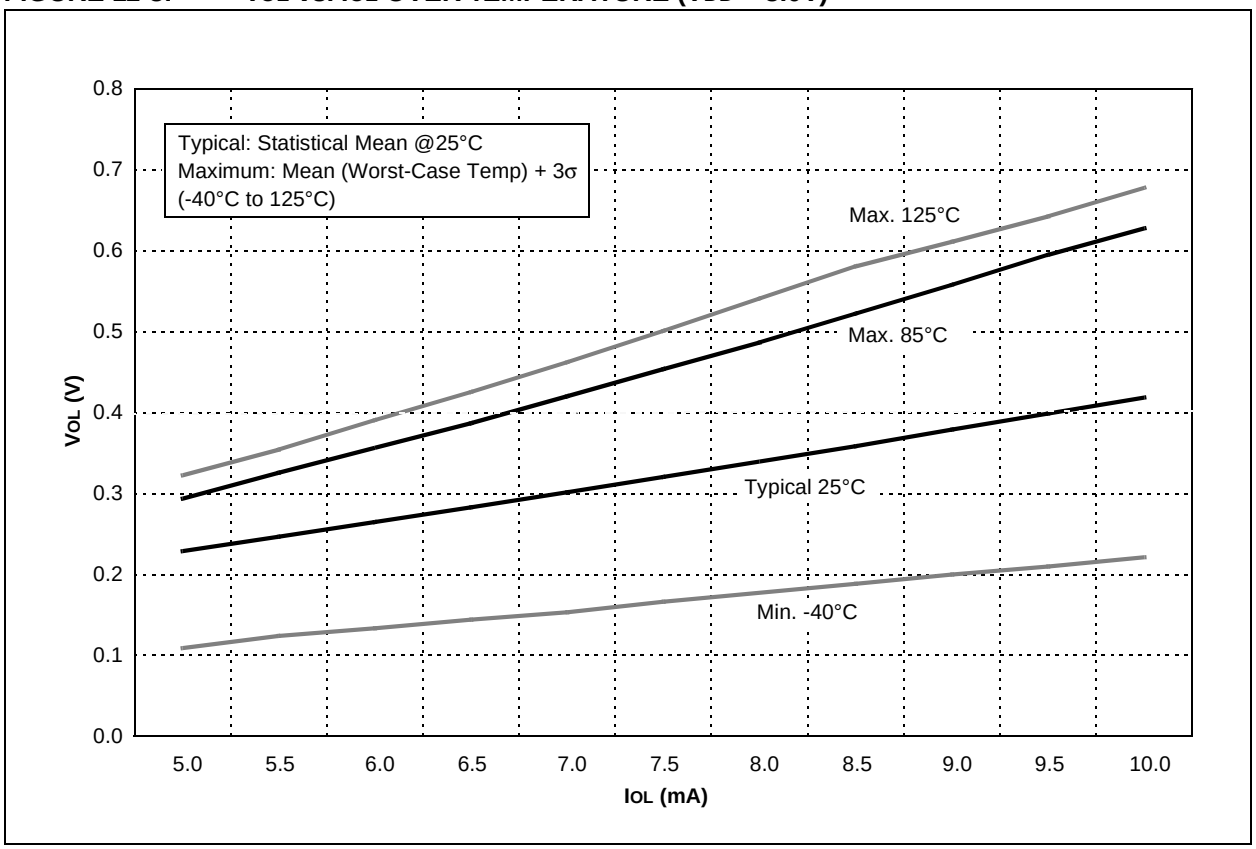
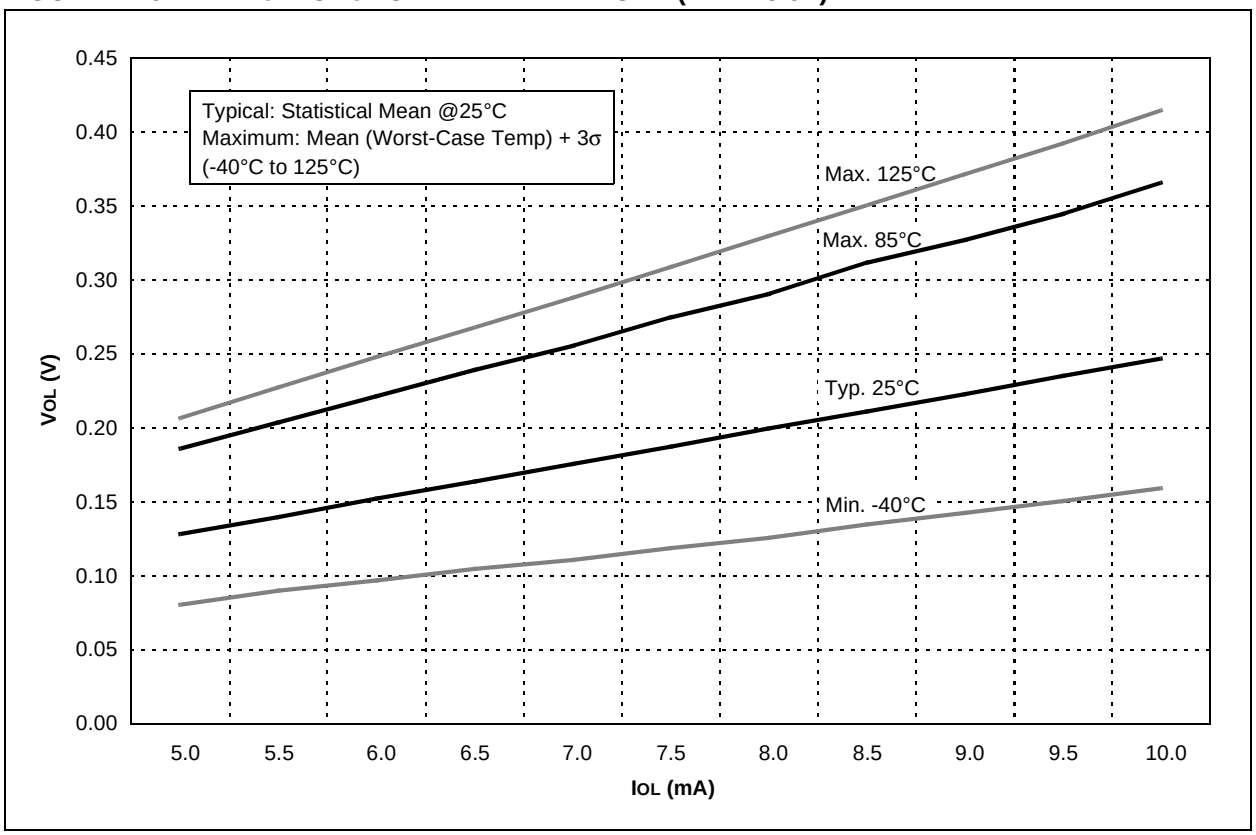
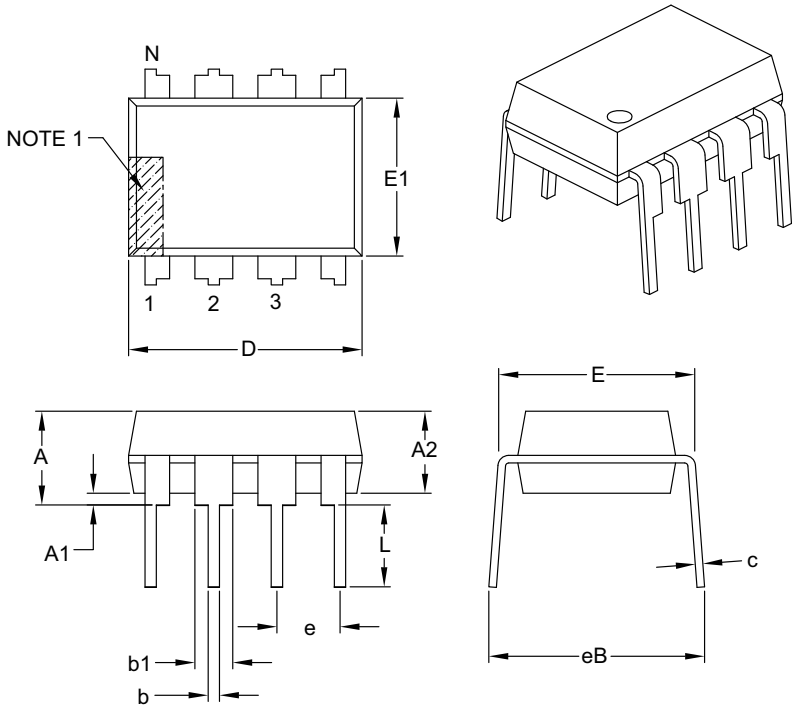



**FIGURE 11-9: VOL vs. IOL OVER TEMPERATURE (VDD = 5.0V)**



## 8-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

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




| Units | | INCHES | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 8 | |
| Pitch | e | | .100 BSC | |
| Top to Seating Plane | A | – | – | .210 |
| Molded Package Thickness | A2 | .115 | .130 | .195 |
| Base to Seating Plane | A1 | .015 | – | – |
| Shoulder to Shoulder Width | E | .290 | .310 | .325 |
| Molded Package Width | E1 | .240 | .250 | .280 |
| Overall Length | D | .348 | .365 | .400 |
| Tip to Seating Plane | L | .115 | .130 | .150 |
| Lead Thickness | c | .008 | .010 | .015 |
| Upper Lead Width | b1 | .040 | .060 | .070 |
| Lower Lead Width | b | .014 | .018 | .022 |
| Overall Row Spacing § | eB | – | – | .430 |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-018B

## APPENDIX A: REVISION HISTORY

**Revision A (April 2004)**

Original data sheet for PIC12F508/509/16F505 devices

**Revision B (June 2005)**

Update packages

**Revision C (03/2007)**

Revised Table 3-2 Legend; Revised Table 3-3 RB3 and Legend; Revised Table 10-4 F10; Replaced Package Drawings (Rev. AN); Added DFN package; Replaced Development Support Section; Revised Product ID System.

**Revision D (12/2007)**

Revised Title; Operating Current; Table 1-1 added DFN and revised note; Revised Section 3.0, last paragraph; Revised Figure 4-4; Revised Table 4-2 (FSR); Revised Register 7-1 and Register 7-2; Revised Section 7.2.2; Revised Table 7-3, Note 2; Revised Table 7-4 (FSR) and Note 2; Deleted Section 7.3.1: External Clock In and Figure 7-6; Revised new Section 7.3.1; Replaced TBD with new data in Tables 10-4 and 10-5; Revised Tables 10-1 (Industrial), 10-2 (Extended), and Tables 10-1 (Industrial, Extended) and 10-2 (Pull-up Resistor Ranges), 10-3, 10-4 and 10-6; Revised Figure 10-1, Figure 10-2; Section 11.0, Added Char data; Revised Package Marking Information; Revised Product ID System.

**Revision E (08/2009)**

Added PIC16F505 16-Pin diagram (QFN); Added Note after subsection 5.2 PORTC; Updated Note 4 and deleted Note 5, Table 10-1; Deleted Param. No. D061 (Table 10-1) and Param. No. D061A becomes D061; Added QFN Package Information; Revised Product Identification System; Added Figures 11-14, 11-15, 11-16, 11-7 to Char Data section; Other minor corrections; Removed Preliminary status.

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| PART NO. | X | /XX | XXX |
|---|---|---|---|
| Device | Temperature Range | Package | Pattern |

| | | |
|---|---|---|
| **Device:** | PIC16F505<br>PIC12F508<br>PIC12F509<br>PIC16F505T(1)<br>PIC12F508T(2)<br>PIC12F509T(2) | |
| **Temperature Range:** | I = -40°C to +85°C (Industrial)<br>E = -40°C to +125°C (Extended) | |
| **Package:** | MC = 8L DFN 2x3 (DUAL Flatpack No-Leads)(3, 4)<br>MS = Micro-Small Outline Package (MSOP)(3, 4)<br>P = Plastic (PDIP)(4)<br>SL = 14L Small Outline, 3.90 mm (SOIC)(4)<br>SN = 8L Small Outline, 3.90 mm Narrow (SOIC)(4)<br>ST = Thin Shrink Small Outline (TSSOP)(4)<br>MG = 16L QFN (3x3x0.9)(5) | |
| **Pattern:** | Special Requirements | |

**Note:** Tape and Reel available for only the following packages: SOIC, MSOP and TSSOP.

**Examples:**

a) PIC12F508-E/P 301 = Extended Temp., PDIP package, QTP pattern #301

b) PIC12F508-I/SN = Industrial Temp., SOIC package

c) PIC12F508T-E/P = Extended Temp., PDIP package, Tape and Reel

**Note 1:** T = in tape and reel SOIC, TSSOP and QFN packages only

**2:** T = in tape and reel SOIC and MSOP packages only.

**3:** PIC12F508/PIC12F509 only.

**4:** Pb-free.

**5:** PIC16F505 only.