PIC10F200/202/204/206

6-Pin, 8-Bit Flash Microcontrollers

Devices Included In This Data Sheet:
- PIC10F200
- PIC10F202
- PIC10F204
- PIC10F206

High-Performance RISC CPU:
- Only 33 Single-Word Instructions to Learn
- All Single-Cycle Instructions except for Program Branches, which are Two-Cycle
- 12-Bit Wide Instructions
- 2-Level Deep Hardware Stack
- Direct, Indirect and Relative Addressing modes for Data and Instructions
- 8-Bit Wide Data Path
- Eight Special Function Hardware Registers
- Operating Speed:
  - 4 MHz internal clock
  - 1 μs instruction cycle

Special Microcontroller Features:
- 4 MHz Precision Internal Oscillator:
  - Factory calibrated to ±1%
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit Debugging (ICD) Support
- Power-on Reset (POR)
- Device Reset Timer (DRT)
- Watchdog Timer (WDT) with Dedicated On-Chip RC Oscillator for Reliable Operation
- Programmable Code Protection
- Multiplexed MCLR Input Pin
- Internal Weak Pull-ups on I/O Pins
- Power-Saving Sleep mode
- Wake-up from Sleep on Pin Change

Low-Power Features/CMOS Technology:
- Operating Current:
  - < 175 µA @ 2V, 4 MHz, typical
- Standby Current:
  - 100 nA @ 2V, typical
- Low-Power, High-Speed Flash Technology:
  - 100,000 Flash endurance
  - > 40 year retention
- Fully Static Design
- Wide Operating Voltage Range: 2.0V to 5.5V
- Wide Temperature Range:
  - Industrial: -40°C to +85°C
  - Extended: -40°C to +125°C

Peripheral Features (PIC10F200/202):
- Four I/O Pins:
  - Three I/O pins with individual direction control
  - One input-only pin
  - High current sink/source for direct LED drive
  - Wake-on-change
  - Weak pull-ups
- 8-Bit Real-Time Clock/Counter (TMR0) with 8-Bit Programmable Prescaler

Peripheral Features (PIC10F204/206):
- Four I/O Pins:
  - Three I/O pins with individual direction control
  - One input-only pin
  - High current sink/source for direct LED drive
  - Wake-on-change
  - Weak pull-ups
- 8-Bit Real-Time Clock/Counter (TMR0) with 8-Bit Programmable Prescaler
- One Comparator:
  - Internal absolute voltage reference
  - Both comparator inputs visible externally
  - Comparator output visible externally

<table>
<thead>
<tr>
<th>Device</th>
<th>Program Memory</th>
<th>Data Memory</th>
<th>I/O</th>
<th>Timers 8-bit</th>
<th>Comparator</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Flash (words)</td>
<td>SRAM (bytes)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIC10F200</td>
<td>256</td>
<td>16</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>PIC10F202</td>
<td>512</td>
<td>24</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>PIC10F204</td>
<td>256</td>
<td>16</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>PIC10F206</td>
<td>512</td>
<td>24</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
## Table of Contents

1.0 General Description .................................................................................................................................................. 4
2.0 PIC10F200/202/204/206 Device Varieties .................................................................................................................. 5
3.0 Architectural Overview ............................................................................................................................................... 6
4.0 Memory Organization .................................................................................................................................................. 11
5.0 I/O Port .................................................................................................................................................................... 20
6.0 Timer0 Module and TMR0 Register (PIC10F200/202) ............................................................................................... 23
7.0 Timer0 Module and TMR0 Register (PIC10F204/206) ............................................................................................... 27
8.0 Comparator Module .................................................................................................................................................... 31
9.0 Special Features of the CPU ..................................................................................................................................... 35
10.0 Instruction Set Summary ......................................................................................................................................... 45
11.0 Development Support .............................................................................................................................................. 53
12.0 Electrical Characteristics ......................................................................................................................................... 57
13.0 DC and AC Characteristics Graphs and Tables ........................................................................................................ 67
14.0 Packaging Information ............................................................................................................................................. 75
The Microchip Web Site .................................................................................................................................................. 85
Customer Change Notification Service .......................................................................................................................... 85
Customer Support ........................................................................................................................................................... 85
Product Identification System ......................................................................................................................................... 86

---

### 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@microchip.com. 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](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., DS30000000A is version A of document DS30000000).

### 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](http://www.microchip.com)
- Your local Microchip sales office (see last page)

When contacting a sales office, 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](http://www.microchip.com) to receive the most current information on all of our products.
1.0 GENERAL DESCRIPTION

The PIC10F200/202/204/206 devices from Microchip Technology are low-cost, high-performance, 8-bit, fully-static, Flash-based CMOS microcontrollers. They employ a RISC architecture with only 33 single-word/single-cycle instructions. All instructions are single cycle (1 μs) except for program branches, which take two cycles. The PIC10F200/202/204/206 devices deliver performance in an order of magnitude higher than their competitors in the same price category. The 12-bit wide instructions are highly symmetrical, resulting in a typical 2:1 code compression over other 8-bit microcontrollers in its class. The easy-to-use and easy to remember instruction set reduces development time significantly.

The PIC10F200/202/204/206 products are equipped with special features that reduce system cost and power requirements. The Power-on Reset (POR) and Device Reset Timer (DRT) eliminate the need for external Reset circuitry. INTRC Internal Oscillator mode is provided, thereby preserving the limited number of I/O available. Power-Saving Sleep mode, Watchdog Timer and code protection features improve system cost, power and reliability.

The PIC10F200/202/204/206 devices are available in cost-effective Flash, which is suitable for production in any volume. The customer can take full advantage of Microchip’s price leadership in Flash programmable microcontrollers, while benefiting from the Flash programmable flexibility.

The PIC10F200/202/204/206 products are supported by a full-featured macro assembler, a software simulator, an in-circuit debugger, a ‘C’ compiler, a low-cost development programmer and a full featured programmer. All the tools are supported on IBM® PC and compatible machines.

### TABLE 1-1: PIC10F200/202/204/206 DEVICES

<table>
<thead>
<tr>
<th>Feature</th>
<th>PIC10F200</th>
<th>PIC10F202</th>
<th>PIC10F204</th>
<th>PIC10F206</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Memory</td>
<td>256</td>
<td>512</td>
<td>256</td>
<td>512</td>
</tr>
<tr>
<td>Peripherals</td>
<td>TMR0</td>
<td>TMR0</td>
<td>TMR0</td>
<td>TMR0</td>
</tr>
<tr>
<td>Wake-up from Sleep on Pin Change</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Comparators</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>I/O Pins</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Input-Only Pins</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Internal Pull-ups</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>In-Circuit Serial Programming™</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Number of Instructions</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>Packages</td>
<td>6-pin SOT-23 8-pin PDIP, DFN</td>
<td>6-pin SOT-23 8-pin PDIP, DFN</td>
<td>6-pin SOT-23 8-pin PDIP, DFN</td>
<td>6-pin SOT-23 8-pin PDIP, DFN</td>
</tr>
</tbody>
</table>

The PIC10F200/202/204/206 devices have Power-on Reset, selectable Watchdog Timer, selectable code-protect, high I/O current capability and precision internal oscillator.

The PIC10F200/202/204/206 devices use serial programming with data pin GP0 and clock pin GP1.

1.1 Applications

The PIC10F200/202/204/206 devices fit in applications ranging from personal care appliances and security systems to low-power remote transmitters/receivers. The Flash technology makes customizing application programs (transmitter codes, appliance settings, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages, for through hole or surface mounting, make these microcontrollers well suited for applications with space limitations. Low cost, low power, high performance, ease-of-use and I/O flexibility make the PIC10F200/202/204/206 devices very versatile even in areas where no microcontroller use has been considered before (e.g., timer functions, logic and PLDs in larger systems and coprocessor applications).
2.0 PIC10F200/202/204/206 DEVICE VARIETIES

A variety of packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in this section. When placing orders, please use the PIC10F200/202/204/206 Product Identification System at the back of this data sheet to specify the correct part number.

2.1 Quick Turn Programming (QTP) Devices

Microchip offers a QTP programming service for factory production orders. This service is made available for users who choose not to program medium-to-high quantity units and whose code patterns have stabilized. The devices are identical to the Flash devices but with all Flash locations and fuse options already programmed by the factory. Certain code and prototype verification procedures do apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

2.2 Serialized Quick Turn Programming℠ (SQTP℠) Devices

Microchip offers a unique programming service, where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number, which can serve as an entry code, password or ID number.
3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC10F200/202/204/206 devices can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC10F200/202/204/206 devices use a Harvard architecture in which program and data are accessed on separate buses. This improves bandwidth over traditional von Neumann architectures where program and data are fetched on the same bus. Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 12 bits wide, making it possible to have all single-word instructions. A 12-bit wide program memory access bus fetches a 12-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (33) execute in a single cycle (1 \( \mu \)s @ 4 MHz) except for program branches.

The table below lists program memory (Flash) and data memory (RAM) for the PIC10F200/202/204/206 devices.

<table>
<thead>
<tr>
<th>Device</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Program</td>
</tr>
<tr>
<td>PIC10F200</td>
<td>256 x 12</td>
</tr>
<tr>
<td>PIC10F202</td>
<td>512 x 12</td>
</tr>
<tr>
<td>PIC10F204</td>
<td>256 x 12</td>
</tr>
<tr>
<td>PIC10F206</td>
<td>512 x 12</td>
</tr>
</tbody>
</table>

The PIC10F200/202/204/206 devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8 bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two’s complement in nature. In two-operand instructions, one operand is typically the W (working) register. The other operand is either a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC) and Zero (Z) bits in the STATUS register. The C and DC bits operate as a borrow and digit borrow out bit, respectively, in subtraction. See the SUBWF and ADDWF instructions for examples.

A simplified block diagram is shown in Figure 3-1 and Figure 3-2, with the corresponding device pins described in Table 3-2.
FIGURE 3-1: PIC10F200/202 BLOCK DIAGRAM

- Flash: 512 x12 or 256 x12 Program Memory
- Program Bus
- Instruction Reg
- 9-10 Program Counter
- 12 Program Bus
- Data Bus 8
- RAM Addr 9
- Indirect Addr
- Status Reg
- MUX
- 3 Timer
- 8 ALU
- MCLR Vdd, Vss
- GP0/ICSPDAT
- GP1/ICSPCLK
- GP2/T0CKI/FOSC4
- GP3/MCLR/Vpp
- GPIO
- GP0/ICSPDAT
- GP1/ICSPCLK
- GP2/T0CKI/FOSC4
- GP3/MCLR/Vpp
FIGURE 3-2: PIC10F204/206 BLOCK DIAGRAM

- Flash 512 x12 or 256 x12 Program Memory
- Data Bus 8
- Program Counter 9-10
- Instruction Register
- Program Bus 12
- RAM Address 9
- RAM 24 or 16 bytes
- File Registers
- Addr MUX 5-7
- Direct Address 5
- Indirect Address
- Status Register
- Address MUX
- ALU
- MUX
- Comparator
- GPIO
- GP0/ICSPDAT/CIN+
- GP1/ICSPCLK/CIN-
- GP2/T0CKI/COUT/FOSC4
- GP3/MCLR/VPP
- MCLR
- Vdd, Vss
- Internal RC Clock
- Timer0
- W Register
### TABLE 3-2: PIC10F200/202/204/206 PINOUT DESCRIPTION

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Input Type</th>
<th>Output Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP0/ICSPDAT/CIN+</td>
<td>GP0</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O pin. Can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.</td>
</tr>
<tr>
<td>ICSPDAT</td>
<td>ST</td>
<td>CMOS</td>
<td>In-Circuit Serial Programming™ data pin.</td>
<td></td>
</tr>
<tr>
<td>CIN+</td>
<td>AN</td>
<td>—</td>
<td>Comparator input (PIC10F204/206 only).</td>
<td></td>
</tr>
<tr>
<td>GP1/ICSPCLK/CIN-</td>
<td>GP1</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O pin. Can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.</td>
</tr>
<tr>
<td>ICSPCLK</td>
<td>ST</td>
<td>CMOS</td>
<td>In-Circuit Serial Programming clock pin.</td>
<td></td>
</tr>
<tr>
<td>CIN-</td>
<td>AN</td>
<td>—</td>
<td>Comparator input (PIC10F204/206 only).</td>
<td></td>
</tr>
<tr>
<td>GP2/T0CKI/COUT/FOSC4</td>
<td>GP2</td>
<td>TTL</td>
<td>CMOS</td>
<td>Bidirectional I/O pin.</td>
</tr>
<tr>
<td>T0CKI</td>
<td>ST</td>
<td>—</td>
<td>Clock input to TMR0.</td>
<td></td>
</tr>
<tr>
<td>COUT</td>
<td>—</td>
<td>CMOS</td>
<td>Comparator output (PIC10F204/206 only).</td>
<td></td>
</tr>
<tr>
<td>FOSC4</td>
<td>—</td>
<td>CMOS</td>
<td>Oscillator/4 output.</td>
<td></td>
</tr>
<tr>
<td>GP3/MCLR/VPP</td>
<td>GP3</td>
<td>TTL</td>
<td>—</td>
<td>Input pin. Can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.</td>
</tr>
<tr>
<td>MCLR</td>
<td>ST</td>
<td>—</td>
<td>Master Clear (Reset). When configured as MCLR, this pin is an active-low Reset to the device. Voltage on GP3/MCLR/VPP must not exceed VDD during normal device operation or the device will enter Programming mode. Weak pull-up always on if configured as MCLR.</td>
<td></td>
</tr>
<tr>
<td>VPP</td>
<td>HV</td>
<td>—</td>
<td>Programming voltage input.</td>
<td></td>
</tr>
<tr>
<td>VDD</td>
<td>VDD</td>
<td>P</td>
<td>Positive supply for logic and I/O pins.</td>
<td></td>
</tr>
<tr>
<td>VSS</td>
<td>VSS</td>
<td>P</td>
<td>Ground reference for logic and I/O pins.</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:** I = Input, O = Output, I/O = Input/Output, P = Power, — = Not used, TTL = TTL input, ST = Schmitt Trigger input, AN = Analog input
3.1 Clocking Scheme/Instruction Cycle

The clock is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. Internally, the PC is incremented every Q1 and the instruction is fetched from program memory and latched into the instruction register in Q4. It is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-3 and Example 3-1.

3.2 Instruction Flow/Pipelining

An instruction cycle consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle, while decode and execute take another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the PC to change (e.g., GOTO), then two cycles are required to complete the instruction (Example 3-1).

A fetch cycle begins with the PC incrementing in Q1. In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-3: CLOCK/INSTRUCTION CYCLE

EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW

1. MOVLW 03H
   Fetch 1  Execute 1
2. MOVWF GPIO
   Fetch 2  Execute 2
3. CALL SUB_1
   Fetch 3  Execute 3
   Fetch 4  Flush
4. BSF GPIO, BIT1
   Fetch SUB_1  Execute SUB_1

All instructions are single cycle, except for any program branches. These take two cycles, since the fetch instruction is “flushed” from the pipeline, while the new instruction is being fetched and then executed.
4.0 MEMORY ORGANIZATION

The PIC10F200/202/204/206 memories are organized into program memory and data memory. Data memory banks are accessed using the File Select Register (FSR).

4.1 Program Memory Organization for the PIC10F200/204

The PIC10F200/204 devices have a 9-bit Program Counter (PC) capable of addressing a 512 x 12 program memory space.

Only the first 256 x 12 (0000h-00FFh) for the PIC10F200/204 are physically implemented (see Figure 4-1). Accessing a location above these boundaries will cause a wraparound within the first 256 x 12 space (PIC10F200/204). The effective Reset vector is at 0000h (see Figure 4-1). Location 00FFh (PIC10F200/204) contains the internal clock oscillator calibration value. This value should never be overwritten.

Note 1: Address 0000h becomes the effective Reset vector. Location 00FFh contains the internal oscillator calibration value.
4.2 Program Memory Organization for the PIC10F202/206

The PIC10F202/206 devices have a 10-bit Program Counter (PC) capable of addressing a 1024 x 12 program memory space.

Only the first 512 x 12 (0000h-01FFh) for the PIC10F202/206 are physically implemented (see Figure 4-2). Accessing a location above these boundaries will cause a wraparound within the first 512 x 12 space (PIC10F202/206). The effective Reset vector is at 0000h (see Figure 4-2). Location 01FFh (PIC10F202/206) contains the internal clock oscillator calibration value. This value should never be overwritten.

FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC10F202/206

4.3 Data Memory Organization

Data memory is composed of registers or bytes of RAM. Therefore, data memory for a device is specified by its register file. The register file is divided into two functional groups: Special Function Registers (SFR) and General Purpose Registers (GPR).

The Special Function Registers include the TMR0 register, the Program Counter (PCL), the STATUS register, the I/O register (GPIO) and the File Select Register (FSR). In addition, Special Function Registers are used to control the I/O port configuration and prescaler options.

The General Purpose registers are used for data and control information under command of the instructions. For the PIC10F200/204, the register file is composed of seven Special Function registers and 16 General Purpose registers (see Figure 4-3 and Figure 4-4).

For the PIC10F202/206, the register file is composed of eight Special Function registers and 24 General Purpose registers (see Figure 4-4).

4.3.1 GENERAL PURPOSE REGISTER FILE

The General Purpose Register file is accessed, either directly or indirectly, through the File Select Register (FSR). See Section 4.9 “Indirect Data Addressing: INDF and FSR Registers”.

Note 1: Address 0000h becomes the effective Reset vector. Location 01FFh contains the MOVJN XX internal oscillator calibration value.
### FIGURE 4-3: PIC10F200/204 REGISTER FILE MAP

<table>
<thead>
<tr>
<th>File Address</th>
<th>Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>INDF(^{(1)})</td>
</tr>
<tr>
<td>01h</td>
<td>TMR0</td>
</tr>
<tr>
<td>02h</td>
<td>PCL</td>
</tr>
<tr>
<td>03h</td>
<td>STATUS</td>
</tr>
<tr>
<td>04h</td>
<td>FSR</td>
</tr>
<tr>
<td>05h</td>
<td>OSCCAL</td>
</tr>
<tr>
<td>06h</td>
<td>GPIO</td>
</tr>
<tr>
<td>07h</td>
<td>CMCON0(^{(2)})</td>
</tr>
<tr>
<td>08h</td>
<td>Unimplemented(^{(3)})</td>
</tr>
<tr>
<td>0Fh</td>
<td></td>
</tr>
<tr>
<td>10h</td>
<td>General Purpose Registers</td>
</tr>
<tr>
<td>1Fh</td>
<td></td>
</tr>
</tbody>
</table>

**Note 1:** Not a physical register. See Section 4.9 “Indirect Data Addressing: INDF and FSR Registers”.

**Note 2:** PIC10F204 only. Unimplemented on the PIC10F200 and reads as 00h.

**Note 3:** Unimplemented, read as 00h.

### FIGURE 4-4: PIC10F202/206 REGISTER FILE MAP

<table>
<thead>
<tr>
<th>File Address</th>
<th>Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>INDF(^{(1)})</td>
</tr>
<tr>
<td>01h</td>
<td>TMR0</td>
</tr>
<tr>
<td>02h</td>
<td>PCL</td>
</tr>
<tr>
<td>03h</td>
<td>STATUS</td>
</tr>
<tr>
<td>04h</td>
<td>FSR</td>
</tr>
<tr>
<td>05h</td>
<td>OSCCAL</td>
</tr>
<tr>
<td>06h</td>
<td>GPIO</td>
</tr>
<tr>
<td>07h</td>
<td>CMCON0(^{(2)})</td>
</tr>
<tr>
<td>08h</td>
<td>General Purpose Registers</td>
</tr>
<tr>
<td>1Fh</td>
<td></td>
</tr>
</tbody>
</table>

**Note 1:** Not a physical register. See Section 4.9 “Indirect Data Addressing: INDF and FSR Registers”.

**Note 2:** PIC10F206 only. Unimplemented on the PIC10F202 and reads as 00h.
### 4.3.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral functions to control the operation of the device (Table 4-1).

The Special Function Registers can be classified into two sets. The Special Function Registers associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section for each peripheral feature.

#### TABLE 4-1: SPECIAL FUNCTION REGISTER (SFR) SUMMARY (PIC10F200/202/204/206)

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on Power-On Reset</th>
<th>Register on Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>00h</td>
<td>INDF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Uses Contents of FSR to Address Data Memory (not a physical register)</td>
<td>xxxx xxxx 19</td>
</tr>
<tr>
<td>01h</td>
<td>TMR0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8-bit Real-Time Clock/Counter</td>
<td>xxxx xxxx 23, 27</td>
</tr>
<tr>
<td>02h(1)</td>
<td>PCL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Low-order 8 bits of PC</td>
<td>1111 1111 18</td>
</tr>
<tr>
<td>03h</td>
<td>STATUS</td>
<td>GPWUF</td>
<td>CWUF(5)</td>
<td>—</td>
<td>TO</td>
<td>PD</td>
<td>Z</td>
<td>DC</td>
<td>C</td>
<td>00-1 lxxx(3)</td>
<td>15</td>
</tr>
<tr>
<td>04h</td>
<td>FSR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Indirect Data Memory Address Pointer</td>
<td>111x xxxx 19</td>
</tr>
<tr>
<td>05h</td>
<td>OSCCAL</td>
<td>CAL6</td>
<td>CAL5</td>
<td>CAL4</td>
<td>CAL3</td>
<td>CAL2</td>
<td>CAL1</td>
<td>CAL0</td>
<td>FOSC4</td>
<td>1111 1110</td>
<td>17</td>
</tr>
<tr>
<td>06h</td>
<td>GPIO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>GP3 GP2 GP1 GP0 wp —</td>
<td>—</td>
</tr>
<tr>
<td>07h(4)</td>
<td>CMCON0</td>
<td>CMP0</td>
<td>CMPOUT</td>
<td>COUTEN</td>
<td>POL</td>
<td>CMP10CS</td>
<td>CMP0</td>
<td>CMREF</td>
<td>CPREF</td>
<td>CMW</td>
<td>1111 1111</td>
</tr>
<tr>
<td>N/A</td>
<td>TRISGPIO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>I/O Control Register</td>
<td>—</td>
</tr>
<tr>
<td>N/A</td>
<td>OPTION</td>
<td>GPWU</td>
<td>GPP0</td>
<td>T0CS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</td>
<td>option</td>
<td>1111 1111</td>
</tr>
</tbody>
</table>

**Legend:**
- = unimplemented, read as ‘0’.
- x = unknown.
- u = unchanged.
- q = value depends on condition.

**Note 1:** The upper byte of the Program Counter is not directly accessible. See Section 4.7 "Program Counter" for an explanation of how to access these bits.

**Note 2:** Other (non Power-up) Resets include external Reset through MCLR, Watchdog Timer and wake-up on pin change Reset.

**Note 3:** See Table 9-1 for other Reset specific values.

**Note 4:** PIC10F204/206 only.

**Note 5:** PIC10F204/206 only. On all other devices, this bit is reserved and should not be used.
4.4 STATUS Register

This register contains the arithmetic status of the ALU, the Reset status and the page preselect bit.

The STATUS register can be the destination for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the TO and PD bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS`, will clear the upper three bits and set the Z bit. This leaves the STATUS register as `000u uluu` (where `u` = unchanged).

Therefore, it is recommended that only `BCF`, `BSF` and `MOVWF` instructions be used to alter the STATUS register. These instructions do not affect the Z, DC or C bits from the STATUS register. For other instructions which do affect Status bits, see Section 10.0 “Instruction Set Summary”.

REGISTER 4-1: STATUS REGISTER

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>U-1</th>
<th>R-1</th>
<th>R-1</th>
<th>R/W-x</th>
<th>R/W-x</th>
<th>R/W-x</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPWUF</td>
<td>CWUF(1)</td>
<td>—</td>
<td>TO</td>
<td>PD</td>
<td>Z</td>
<td>DC</td>
<td>C</td>
</tr>
</tbody>
</table>

bit 7  
GPWUF: GPIO Reset bit  
1 = Reset due to wake-up from Sleep on pin change  
0 = After power-up or other Reset

bit 6  
CWUF: Comparator Wake-up on Change Flag bit(1)  
1 = Reset due to wake-up from Sleep on comparator change  
0 = After power-up or other Reset conditions.

bit 5  
Reserved: Do not use. Use of this bit may affect upward compatibility with future products.

bit 4  
TO: Time-out bit  
1 = After power-up, `CLRWD` instruction or `SLEEP` instruction  
0 = A WDT time-out occurred

bit 3  
PD: Power-down bit  
1 = After power-up or by the `CLRWD` instruction  
0 = By execution of the `SLEEP` instruction

bit 2  
Z: Zero bit  
1 = The result of an arithmetic or logic operation is zero  
0 = The result of an arithmetic or logic operation is not zero

bit 1  
DC: Digit Carry/Borrow bit (for `ADDWF` and `SUBWF` instructions)  
`ADDWF`:  
1 = A carry from the 4th low-order bit of the result occurred  
0 = A carry from the 4th low-order bit of the result did not occur  
`SUBWF`:  
1 = A borrow from the 4th low-order bit of the result did not occur  
0 = A borrow from the 4th low-order bit of the result occurred

bit 0  
C: Carry/Borrow bit (for `ADDWF`, `SUBWF` and `RRF`, `RLF` instructions)  
`ADDWF`:  
1 = A carry occurred  
0 = A carry did not occur  
`SUBWF`:  
1 = A borrow did not occur  
0 = A borrow occurred  
`RRF` or `RLF`:  
1 = Load bit with LSb or MSb, respectively

Note 1: This bit is used on the PIC10F204/206. For code compatibility do not use this bit on the PIC10F200/202.
4.5 OPTION Register

The OPTION register is a 8-bit wide, write-only register, which contains various control bits to configure the Timer0/WDT prescaler and Timer0.

By executing the OPTION instruction, the contents of the W register will be transferred to the OPTION register. A Reset sets the OPTION<7:0> bits.

Note: If TRIS bit is set to ‘0’, the wake-up on change and pull-up functions are disabled for that pin (i.e., note that TRIS overrides Option control of GPPU and GPWU).

Note: If the T0CS bit is set to ‘1’, it will override the TRIS function on the T0CKI pin.

REGISTER 4-2: OPTION REGISTER

<table>
<thead>
<tr>
<th>GPWU</th>
<th>GPPU</th>
<th>T0CS</th>
<th>T0SE</th>
<th>PSA</th>
<th>PS2</th>
<th>PS1</th>
<th>PS0</th>
</tr>
</thead>
<tbody>
<tr>
<td>W-1</td>
<td>W-1</td>
<td>W-1</td>
<td>W-1</td>
<td>W-1</td>
<td>W-1</td>
<td>W-1</td>
<td>W-1</td>
</tr>
</tbody>
</table>

bit 7  GPWU: Enable Wake-up on Pin Change bit (GP0, GP1, GP3)

1 = Disabled
0 = Enabled

bit 6  GPPU: Enable Weak Pull-ups bit (GP0, GP1, GP3)

1 = Disabled
0 = Enabled

bit 5  T0CS: Timer0 Clock Source Select bit

1 = Transition on T0CKI pin (overrides TRIS on the T0CKI pin)
0 = Transition on internal instruction cycle clock, Fosc/4

bit 4  T0SE: Timer0 Source Edge Select bit

1 = Increment on high-to-low transition on the T0CKI pin
0 = Increment on low-to-high transition on the T0CKI pin

bit 3  PSA: Prescaler Assignment bit

1 = Prescaler assigned to the WDT
0 = Prescaler assigned to Timer0

bit 2-0  PS<2:0>: Prescaler Rate Select bits

<table>
<thead>
<tr>
<th>Bit Value</th>
<th>Timer0 Rate</th>
<th>WDT Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>1 : 2</td>
<td>1 : 1</td>
</tr>
<tr>
<td>001</td>
<td>1 : 4</td>
<td>1 : 2</td>
</tr>
<tr>
<td>010</td>
<td>1 : 8</td>
<td>1 : 4</td>
</tr>
<tr>
<td>011</td>
<td>1 : 16</td>
<td>1 : 8</td>
</tr>
<tr>
<td>100</td>
<td>1 : 32</td>
<td>1 : 16</td>
</tr>
<tr>
<td>101</td>
<td>1 : 64</td>
<td>1 : 32</td>
</tr>
<tr>
<td>110</td>
<td>1 : 128</td>
<td>1 : 64</td>
</tr>
<tr>
<td>111</td>
<td>1 : 256</td>
<td>1 : 128</td>
</tr>
</tbody>
</table>

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
4.6 OSCCAL Register

The Oscillator Calibration (OSCCAL) register is used to calibrate the internal precision 4 MHz oscillator. It contains seven bits for calibration.

**Note:** Erasing the device will also erase the pre-programmed internal calibration value for the internal oscillator. The calibration value must be read prior to erasing the part so it can be reprogrammed correctly later.

After you move in the calibration constant, do not change the value. See Section 9.2.2 “Internal 4 MHz Oscillator”.

### REGISTER 4-3: OSCCAL REGISTER

<table>
<thead>
<tr>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAL6</td>
<td>CAL5</td>
<td>CAL4</td>
<td>CAL3</td>
<td>CAL2</td>
<td>CAL1</td>
<td>CAL0</td>
<td>FOSC4</td>
</tr>
</tbody>
</table>

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

#### bit 7-1: CAL<6:0>: Oscillator Calibration bits

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0111111</td>
<td>Maximum frequency</td>
</tr>
<tr>
<td>00000001</td>
<td>Center frequency</td>
</tr>
<tr>
<td>111111</td>
<td>Minimum frequency</td>
</tr>
</tbody>
</table>

#### bit 0: FOSC4: INTOSC/4 Output Enable bit(1)

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTOSC/4 output onto GP2</td>
</tr>
<tr>
<td>0</td>
<td>GP2/T0CKI/COUT applied to GP2</td>
</tr>
</tbody>
</table>

**Note 1:** Overrides GP2/T0CKI/COUT control registers when enabled.
4.7 Program Counter

As a program instruction is executed, the Program Counter (PC) will contain the address of the next program instruction to be executed. The PC value is increased by one every instruction cycle, unless an instruction changes the PC.

For a GOTO instruction, bits 8-0 of the PC are provided by the GOTO instruction word. The Program Counter Low (PCL) is mapped to PC<7:0>.

For a CALL instruction, or any instruction where the PCL is the destination, bits 7:0 of the PC again are provided by the instruction word. However, PC<8> does not come from the instruction word, but is always cleared (Figure 4-5).

Instructions where the PCL is the destination, or modify PCL instructions, include MOVWF PC, ADDWF PC and BSF PC,5.

Note: Because PC<8> is cleared in the CALL instruction or any modify PCL instruction, all subroutine calls or computed jumps are limited to the first 256 locations of any program memory page (512 words long).

FIGURE 4-5: LOADING OF PC BRANCH INSTRUCTIONS

4.7.1 EFFECTS OF RESET

The PC is set upon a Reset, which means that the PC addresses the last location in program memory (i.e., the oscillator calibration instruction). After executing MOVWF XX, the PC will roll over to location 0000h and begin executing user code.

4.8 Stack

The PIC10F200/204 devices have a 2-deep, 8-bit wide hardware PUSH/POP stack.

The PIC10F202/206 devices have a 2-deep, 9-bit wide hardware PUSH/POP stack.

A CALL instruction will PUSH the current value of Stack 1 into Stack 2 and then PUSH the current PC value, incremented by one, into Stack Level 1. If more than two sequential CALLs are executed, only the most recent two return addresses are stored.

A RETLW instruction will POP the contents of Stack Level 1 into the PC and then copy Stack Level 2 contents into level 1. If more than two sequential RETLWs are executed, the stack will be filled with the address previously stored in Stack Level 2.

Note 1: The W register will be loaded with the literal value specified in the instruction. This is particularly useful for the implementation of the data look-up tables within the program memory.

2: There are no Status bits to indicate stack overflows or stack underflow conditions.

3: There are no instruction mnemonics called PUSH or POP. These are actions that occur from the execution of the CALL and RETLW instructions.
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.

4.10 Indirect Addressing

• Register file 09 contains the value 10h
• Register file 0A contains the value 0Ah
• Load the value 09 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 = 0A)
• 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
MOVF FSR ;to RAM
NEXT CLRF INDF ;clear INDF
;register
INC FSR,F ;inc pointer
BTSC 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.

Note: PIC10F200/202/204/206 – Do not use banking. FSR<7:5> are unimplemented and read as '1's.

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

FIGURE 4-6: DIRECT/INDIRECT ADDRESSING (PIC10F200/202/204/206)

Note 1: For register map detail, see Section 4.3 “Data Memory Organization”.
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 GPIO, 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.

5.1 GPIO

GPIO is an 8-bit I/O register. Only the low-order 4 bits are used (GP<3:0>). Bits 7 through 4 are unimplemented and read as '0's. Please note that GP3 is an input-only pin. Pins GP0, GP1 and GP3 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 GP3/MCLR is configured as MCLR, weak pull-up is always on and wake-up on change for this pin is not enabled.

5.2 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 GP3, which is input-only and the GP2/T0CKI/COUT/FOSC4 pin, which may be controlled by various registers. See Table 5-1.

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.

### TABLE 5-1: ORDER OF PRECEDENCE FOR PIN FUNCTIONS

<table>
<thead>
<tr>
<th>Priority</th>
<th>GP0</th>
<th>GP1</th>
<th>GP2</th>
<th>GP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CIN+</td>
<td>CIN-</td>
<td>FOSC4</td>
<td>I/MCLR</td>
</tr>
<tr>
<td>2</td>
<td>TRIS GPIO</td>
<td>TRIS GPIO</td>
<td>COUT</td>
<td>—</td>
</tr>
<tr>
<td>3</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>4</td>
<td>—</td>
<td>—</td>
<td>TRIS GPIO</td>
<td>—</td>
</tr>
</tbody>
</table>

5.3 I/O Interfacing

The equivalent circuit for an I/O port pin is shown in Figure 5-1. All port pins, except 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 GPIO, 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 GP3) can be programmed individually as input or output.

### FIGURE 5-1: PIC10F200/202/204/206 EQUIVALENT CIRCUIT FOR A SINGLE I/O PIN

Note 1: See Table 3-2 for buffer type.
5.4 I/O Programming Considerations

5.4.1 Bidirectional I/O Ports

Some instructions operate internally as read followed by write operations. The **BCF** and **BSF** instructions, for example, read the entire port into the CPU, execute the bit operation and rewrite the result. Caution must be used when these instructions are applied to a port where one or more pins are used as input/outputs. For example, a **BSF** operation on bit 2 of GPIO will cause all eight bits of GPIO to be read into the CPU, bit 2 to be set and the GPIO value to be written to the output latches. If another bit of GPIO is used as a bidirectional I/O pin (say bit 0), and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the Input mode, no problem occurs. However, if bit 0 is switched into Output mode later on, the content of the data latch may now be unknown.

**Example 5-1** shows the effect of two sequential Read-Modify-Write instructions (e.g., **BCF**, **BSF**, etc.) on an I/O port.

A pin actively outputting a high or a low should not be driven from external devices at the same time in order to change the level on this pin (“wired OR”, “wired AND”). The resulting high output currents may damage the chip.

### Table 5-2: Summary of Port Registers

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on Power-On Reset</th>
<th>Value on All Other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>TRISGPIO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—— 1111</td>
<td>—— 1111</td>
</tr>
<tr>
<td>N/A</td>
<td>OPTION</td>
<td>GPWU</td>
<td>GPPU</td>
<td>TOCS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</td>
<td>1111 1111</td>
<td>1111 1111</td>
</tr>
<tr>
<td>03h</td>
<td>STATUS</td>
<td>GPWUF</td>
<td>CWUF</td>
<td>—</td>
<td>—</td>
<td>TO</td>
<td>PD</td>
<td>Z</td>
<td>DC</td>
<td>00-1 1xxx qg-q quuu(1), (2)</td>
<td></td>
</tr>
<tr>
<td>06h</td>
<td>GPIO</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>GP3</td>
<td>GP2</td>
<td>GP1</td>
<td>—— xxxxxx</td>
<td>—— uuuu</td>
</tr>
</tbody>
</table>

Legend:  
Shaded cells are not used by PORT registers, read as ‘0’, – = unimplemented, read as ‘0’, x = unknown, u = unchanged, q = depends on condition.

Note 1:  
1. If Reset was due to wake-up on pin change, then bit 7 = 1. All other Resets will cause bit 7 = 0.
2. If Reset was due to wake-up on comparator change, then bit 6 = 1. All other Resets will cause bit 6 = 0.

### Example 5-1: Read-Modify-Write Instructions on an I/O Port

```plaintext
;Initial GPIO Settings
;GPIO<3:2> Inputs
;GPIO<1:0> Outputs
;
;              GPIO latch   GPIO pins
;-------------- ------------
;BCF GPIO,  1 ;---- pp01 ---- pp11
;BCF GPIO,  0 ;---- pp10 ---- pp11
;MOVLW 007h;
;TRIS GPIO ;---- pp10 ---- pp11
;
Note 1: The user may have expected the pin values to be ---- pp00. The 2nd BCF caused GP1 to be latched as the pin value (High).
```

5.4.2 Successive Operations on I/O Ports

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-2). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction causes that file to be read into the CPU. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a **NOP** or another instruction not accessing this I/O port.
FIGURE 5-2: SUCCESSIVE I/O OPERATION (PIC10F200/202/204/206)

<table>
<thead>
<tr>
<th>Instruction Fetched</th>
<th>Instruction Executed</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOVWF GPIO</td>
<td>MOVWF GPIO, W</td>
</tr>
<tr>
<td>MOVF GPIO, W</td>
<td>NOP</td>
</tr>
<tr>
<td>Port pin written</td>
<td>Port pin sampled</td>
</tr>
<tr>
<td>(Write to GPIO)</td>
<td>(Read GPIO)</td>
</tr>
</tbody>
</table>

This example shows a write to GPIO followed by a read from GPIO.

Data setup time = (0.25 TCY – Tpd)
where: TCY = instruction cycle
Tpd = propagation delay

Therefore, at higher clock frequencies, a write followed by a read may be problematic.
6.0 TIMER0 MODULE AND TMR0 REGISTER (PIC10F200/202)

The Timer0 module has the following features:
- 8-bit timer/counter register, TMR0
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select:
  - Edge select for external clock

Figure 6-1 is a simplified block diagram of the Timer0 module.

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 register is written, the increment is inhibited for the following two cycles (Figure 6-2 and Figure 6-3). The user can work around this by writing an adjusted value to the TMR0 register.

Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode, Timer0 will increment either on every rising or falling edge of pin T0CKI. The T0SE bit (OPTION<4>) determines the source edge. Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 6.1 “Using Timer0 with an External Clock (PIC10F200/202)

The prescaler may be used by either the Timer0 module or the Watchdog Timer, but not both. The prescaler assignment is controlled in software by the control bit, PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to Timer0. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, 1:256 are selectable. Section 6.2 “Prescaler” details the operation of the prescaler.

A summary of registers associated with the Timer0 module is found in Table 6-1.
6.1 Using Timer0 with an External Clock (PIC10F200/202)

When an external clock input is used for Timer0, it must meet certain requirements. The external clock requirement is due to internal phase clock (TOSC) synchronization. Also, there is a delay in the actual incrementing of Timer0 after synchronization.

### 6.1.1 EXTERNAL CLOCK SYNCHRONIZATION

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of TOCKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 6-4). Therefore, it is necessary for TOCKI to be high for at least 2 Tosc (and a small RC delay of 2 TIOH) and low for at least 2 Tosc (and a small RC delay of 2 TIOH). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple counter-type prescaler, so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for TOCKI to have a period of at least 4 Tosc (and a small RC delay of 4 TIOH) divided by the prescaler value. The only requirement on TOCKI high and low time is that they do not violate the minimum pulse width requirement of TIOH. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.
6.1.2 TIMER0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the Timer0 module is actually incremented. Figure 6-4 shows the delay from the external clock edge to the timer incrementing.

**FIGURE 6-4: TIMER0 TIMING WITH EXTERNAL CLOCK**

![Timer0 Timing Diagram]

**Note 1:** Delay from clock input change to Timer0 increment is 3 Tosc to 7 Tosc (Duration of Q = Tosc). Therefore, the error in measuring the interval between two edges on Timer0 input = ±4 Tosc max.

2: External clock if no prescaler selected; prescaler output otherwise.

3: The arrows indicate the points in time where sampling occurs.

6.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module or as a postscaler for the Watchdog Timer (WDT), respectively (see Section 9.6 “Watchdog Timer (WDT)”). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet.

**Note:** The prescaler may be used by either the Timer0 module or the WDT, but not both. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the WDT and vice versa.

The PSA and PS<2:0> bits (OPTION<3:0>) determine prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., CLRF 1, MOVWF 1, BSF 1,x, etc.) will clear the prescaler. When assigned to WDT, a CLRWDT instruction will clear the prescaler along with the WDT. The prescaler is neither readable nor writable. On a Reset, the prescaler contains all ‘0’s.

6.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution). To avoid an unintended device Reset, the following instruction sequence (Example 6-1) must be executed when changing the prescaler assignment from Timer0 to the WDT.

**EXAMPLE 6-1: CHANGING PRESCALER (TIMER0 → WDT)**

```
CLRWDT ;Clear WDT
CLRF TMR0 ;Clear TMRO & Prescaler
MOVLW '00xx1111'b ;These 3 lines (5, 6, 7) OPTION ;are required only if ;desired
MOVLW '00xx1xxx'b ;Set Postscaler to OPTION ;desired WDT rate
```

The PSA and PS<2:0> bits (OPTION<3:0>) determine prescaler assignment andprescale ratio.
To change the prescaler from the WDT to the Timer0 module, use the sequence shown in Example 6-2. This sequence must be used even if the WDT is disabled. A `CLRWDT` instruction should be executed before switching the prescaler.

**EXAMPLE 6-2: CHANGING PRESCALER (WDT→TIMER0)**

```
CLRWDT ;Clear WDT and prescaler
MOVLW 'xxxx0xxx' ;Select TMR0, new prescale value and clock source
OPTION
```

**FIGURE 6-5: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER**

**Note 1:** T0CS, T0SE, PSA, PS<2:0> are bits in the OPTION register.
**Note 2:** T0CKI is shared with pin GP2 on the PIC10F200/202/204/206.
The Timer0 module has the following features:

- 8-bit timer/counter register, TMR0
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select:
  - Edge select for external clock
  - External clock from either the T0CKI pin or from the output of the comparator

Figure 7-1 is a simplified block diagram of the Timer0 module.

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 register is written, the increment is inhibited for the following two cycles (Figure 7-2 and Figure 7-3). The user can work around this by writing an adjusted value to the TMR0 register.

There are two types of Counter mode. The first Counter mode uses the T0CKI pin to increment Timer0. It is selected by setting the T0CS bit (OPTION<5>), setting the CMPT0CS bit (CMCON0<4>) and setting the COUTEN bit (CMCON0<6>). In this mode, Timer0 will increment either on every rising or falling edge of pin T0CKI. The T0SE bit (OPTION<4>) determines the source edge. Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in Section 7.1 “Using Timer0 with an External Clock (PIC10F204/206)”.

The second Counter mode uses the output of the comparator to increment Timer0. It can be entered in two different ways. The first way is selected by setting the T0CS bit (OPTION<5>) and clearing the CMPT0CS bit (CMCON0<4>); (COUTEN [CMCON0<6>]) does not affect this mode of operation. This enables an internal connection between the comparator and the Timer0.

The second way is selected by setting the T0CS bit (OPTION<5>), setting the CMPT0CS bit (CMCON0<4>) and clearing the COUTEN bit (CMCON0<6>). This allows the output of the comparator onto the T0CKI pin, while keeping the T0CKI input active. Therefore, any comparator change on the COUT pin is fed back into the T0CKI input. The T0SE bit (OPTION<4>) determines the source edge. Clearing the T0SE bit selects the rising edge.

Restrictions on the external clock input as discussed in Section 7.1 “Using Timer0 with an External Clock (PIC10F204/206)”.

The prescaler may be used by either the Timer0 module or the Watchdog Timer, but not both. The prescaler assignment is controlled in software by the control bit, PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to Timer0. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4,..., 1:256 are selectable. Section 7.2 “Prescaler” details the operation of the prescaler.

A summary of registers associated with the Timer0 module is found in Table 7-1.
7.1 Using Timer0 with an External Clock (PIC10F204/206)

When an external clock input is used for Timer0, it must meet certain requirements. The external clock requirement is due to internal phase clock (TOSC) synchronization. Also, there is a delay in the actual incrementing of Timer0 after synchronization.

7.1.1 EXTERNAL CLOCK SYNCHRONIZATION

When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of an external clock with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 7-4). Therefore, it is necessary for T0CKI or the comparator output to be high for at least 2 Tosc and low for at least 2 T0H. Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple counter type prescaler, so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple counter must be taken into account. Therefore, it is necessary for T0CKI or the comparator output to have a period of at least 4 Tosc divided by the prescaler value. The only requirement on T0CKI or the comparator output high and low time is that they do not violate the minimum pulse width requirement of T0H. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.
7.1.2 TIMER0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the Timer0 module is actually incremented. Figure 7-4 shows the delay from the external clock edge to the timer incrementing.

FIGURE 7-4: TIMER0 TIMING WITH EXTERNAL CLOCK

![Timer0 Timing Diagram]

Note 1: Delay from clock input change to Timer0 increment is 3 Tosc to 7 Tosc (Duration of Q = Tosc). Therefore, the error in measuring the interval between two edges on Timer0 input = ±4 Tosc max.

2: External clock if no prescaler selected; prescaler output otherwise.

3: The arrows indicate the points in time where sampling occurs.

7.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module or as a postscaler for the Watchdog Timer (WDT), respectively (see Figure 9-6). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet.

Note: The prescaler may be used by either the Timer0 module or the WDT, but not both. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the WDT and vice versa.

The PSA and PS<2:0> bits (OPTION<3:0>) determine prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., CLRF 1, MOVWF 1, BSF 1,x, etc.) will clear the prescaler. When assigned to WDT, a CLRWDT instruction will clear the prescaler along with the WDT. The prescaler is neither readable nor writable. On a Reset, the prescaler contains all ‘0’s.

7.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution). To avoid an unintended device Reset, the following instruction sequence (Example 7-1) must be executed when changing the prescaler assignment from Timer0 to the WDT.

EXAMPLE 7-1: CHANGING PRESCALER (TIMER0 → WDT)

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLRWDT ;Clear WDT</td>
<td></td>
</tr>
<tr>
<td>CLRF TMR0 ;Clear TMRO &amp; Prescaler</td>
<td></td>
</tr>
<tr>
<td>MOVWL '00xx1111'b;These 3 lines (5, 6, 7) OPTION ;are required only if ;desired</td>
<td></td>
</tr>
<tr>
<td>CLRWDT ;PS&lt;2:0&gt; are 000 or 001</td>
<td></td>
</tr>
<tr>
<td>MOVWL '00xx1xxx'b;Set Postscaler to OPTION ;desired WDT rate</td>
<td></td>
</tr>
</tbody>
</table>

To change the prescaler from the WDT to the Timer0 module, use the sequence shown in Example 7.2. This sequence must be used even if the WDT is disabled. A CLRWDT instruction should be executed before switching the prescaler.
EXAMPLE 7-2: CHANGING PRESCALER
(WDT → TIMER0)

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLRWDT</td>
<td>Clear WDT and prescaler</td>
</tr>
<tr>
<td>MOVlw 'xxxx0xxx'</td>
<td>Select TMR0, new prescale value and clock source</td>
</tr>
</tbody>
</table>

OPTION

FIGURE 7-5: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER

- GP2/T0CKI
- Comparator Output
- Watchdog Timer
- WDT Enable bit
- T0CS, T0SE, PSA, PS<2:0> are bits in the OPTION register.
- T0CKI is shared with pin GP2.
- Bit CMPT0CS is located in the CMCON0 register.
### 8.0 COMPARATOR MODULE

The comparator module contains one Analog comparator. The inputs to the comparator are multiplexed with GP0 and GP1 pins. The output of the comparator can be placed on GP2.

The CMCON0 register, shown in Register 8-1, controls the comparator operation. A block diagram of the comparator is shown in Figure 8-1.

#### REGISTER 8-1: CMCON0 REGISTER

<table>
<thead>
<tr>
<th>R-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMPOUT</td>
<td>COUTEN</td>
<td>POL</td>
<td>CMPT0CS</td>
<td>CMPON</td>
<td>CNREF</td>
<td>CPREF</td>
<td>CWU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bit 7</th>
<th>bit 6</th>
<th>bit 5</th>
<th>bit 4</th>
<th>bit 3</th>
<th>bit 2</th>
<th>bit 1</th>
<th>bit 0</th>
</tr>
</thead>
</table>

Legend:

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

**bit 7**  
**CMPOUT**: Comparator Output bit  
1 = VIN+ > VIN-  
0 = VIN+ < VIN-  

**bit 6**  
**COUTEN**: Comparator Output Enable bit(1, 2)  
1 = Output of comparator is NOT placed on the COUT pin  
0 = Output of comparator is placed in the COUT pin  

**bit 5**  
**POL**: Comparator Output Polarity bit(2)  
1 = Output of comparator not inverted  
0 = Output of comparator inverted  

**bit 4**  
**CMPT0CS**: Comparator TMR0 Clock Source bit(2)  
1 = TMR0 clock source selected by T0CS control bit  
0 = Comparator output used as TMR0 clock source  

**bit 3**  
**CMPON**: Comparator Enable bit  
1 = Comparator is on  
0 = Comparator is off  

**bit 2**  
**CNREF**: Comparator Negative Reference Select bit(2)  
1 = CIN- pin(3)  
0 = Internal voltage reference  

**bit 1**  
**CPREF**: Comparator Positive Reference Select bit(2)  
1 = CIN+ pin(3)  
0 = CIN- pin(3)  

**bit 0**  
**CWU**: Comparator Wake-up on Change Enable bit(2)  
1 = Wake-up on comparator change is disabled  
0 = Wake-up on comparator change is enabled.

**Note 1:** Overrides T0CS bit for TRIS control of GP2.

2: When the comparator is turned on, these control bits assert themselves. When the comparator is off, these bits have no effect on the device operation and the other control registers have precedence.

3: PIC10F204/206 only.
8.1 Comparator Configuration

The on-board comparator inputs, (GP0/CIN+, GP1/CIN-), as well as the comparator output (GP2/COUT), are steerable. The CMCON0, OPTION and TRIS registers are used to steer these pins (see Figure 8-1). If the Comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in Table 12-1.

Note: The comparator can have an inverted output (see Figure 8-1).

FIGURE 8-1: BLOCK DIAGRAM OF THE COMPARATOR

TABLE 8-1: TMR0 CLOCK SOURCE FUNCTION MUXING

<table>
<thead>
<tr>
<th>T0CS</th>
<th>CMPT0CS</th>
<th>COUTEN</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>x</td>
<td>Internal Instruction Cycle</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>CMPOUT</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>CMPOUT</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>CMPOUT</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>T0CKI</td>
</tr>
</tbody>
</table>
8.2 Comparator Operation

A single comparator is shown in Figure 8-2 along with the relationship between the analog input levels and the digital output. When the analog input at \( \text{VIN}^+ \) is less than the analog input \( \text{VIN}^- \), the output of the comparator is a digital low level. When the analog input at \( \text{VIN}^+ \) is greater than the analog input \( \text{VIN}^- \), the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 8-2 represent the uncertainty due to input offsets and response time. See Table 12-1 for Common Mode Voltage.

8.5 Comparator Output

The comparator output is read through CMCON0 register. This bit is read-only. The comparator output may also be used internally, see Figure 8-1.

**Note:** Analog levels on any pin that is defined as a digital input may cause the input buffer to consume more current than is specified.

8.6 Comparator Wake-up Flag

The comparator wake-up flag is set whenever all of the following conditions are met:

- \( \text{CWU} = 0 \) (CMCON0<0>)
- CMCON0 has been read to latch the last known state of the CMPOUT bit (MOVEW CMCON0, W)
- Device is in Sleep
- The output of the comparator has changed state

The wake-up flag may be cleared in software or by another device Reset.

8.7 Comparator Operation During Sleep

When the comparator is active and the device is placed in Sleep mode, the comparator remains active. While the comparator is powered-up, higher Sleep currents than shown in the power-down current specification will occur. To minimize power consumption while in Sleep mode, turn off the comparator before entering Sleep.

8.8 Effects of a Reset

A Power-on Reset (POR) forces the CMCON0 register to its Reset state. This forces the comparator module to be in the comparator Reset mode. This ensures that all potential inputs are analog inputs. Device current is minimized when analog inputs are present at Reset time. The comparator will be powered-down during the Reset interval.

8.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 8-3. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur. A maximum source impedance of 10 k\( \Omega \) is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.
FIGURE 8-3: ANALOG INPUT MODE

Legend:
- CPIN = Input Capacitance
- VT = Threshold Voltage
- ILEAKAGE = Leakage Current at the Pin
- RIC = Interconnect Resistance
- RS = Source Impedance
- VA = Analog Voltage

TABLE 8-2: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on POR</th>
<th>Value on All Other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>03h</td>
<td>STATUS GPWUF CWUF — TO PD Z DC C</td>
<td>00-1 lxxx qqqq qqqq</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07h</td>
<td>CMCON0 CMPOUT COUTEN POL CMPT0CS CMPON CNREF CPREF CWU</td>
<td>1111 1111 &lt;u&gt;uuuu uuuu&lt;/u&gt;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>TRISGPIO -- -- -- -- --</td>
<td>I/O Control Register 1111 1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Legend: x = Unknown, u = Unchanged, – = Unimplemented, read as ‘0’, q = Depends on condition.
9.0 SPECIAL FEATURES OF THE CPU

What sets a microcontroller apart from other processors are special circuits that deal with the needs of real-time applications. The PIC10F200/202/204/206 microcontrollers have a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power-saving operating modes and offer code protection. These features are:

- Reset:
  - Power-on Reset (POR)
  - Device Reset Timer (DRT)
  - Watchdog Timer (WDT)
  - Wake-up from Sleep on pin change
  - Wake-up from Sleep on comparator change
- Sleep
- Code Protection
- ID Locations
- In-Circuit Serial Programming™
- Clock Out

The PIC10F200/202/204/206 devices have a Watchdog Timer, which can be shut off only through Configuration bit WDTE. It runs off of its own RC oscillator for added reliability. When using INTRC, there is an 18 ms delay only on VDD power-up. With this timer on-chip, most applications need no external Reset circuitry.

The Sleep mode is designed to offer a very low-current Power-Down mode. The user can wake-up from Sleep through a change on input pins, wake-up from comparator change, or through a Watchdog Timer time-out.

9.1 Configuration Bits

The PIC10F200/202/204/206 Configuration Words consist of 12 bits. Configuration bits can be programmed to select various device configurations. One bit is the Watchdog Timer enable bit, one bit is the MCLR enable bit and one bit is for code protection (see Register 9-1).

### REGISTER 9-1: CONFIGURATION WORD FOR PIC10F200/202/204/206(1,2)

<table>
<thead>
<tr>
<th>bit 11-0</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>R/P-1</th>
<th>MCLRE</th>
<th>CP</th>
<th>WDTE</th>
<th>—</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit 11</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>MCLRE</td>
<td>CP</td>
<td>WDTE</td>
<td>—</td>
</tr>
<tr>
<td>bit 0</td>
<td>R/P-1</td>
<td>R/P-1</td>
<td>R/P-1</td>
<td>R/P-1</td>
<td>R/P-1</td>
<td>R/P-1</td>
<td>R/P-1</td>
<td>R/P-1</td>
<td>R/P-1</td>
<td>R/P-1</td>
<td>R/P-1</td>
</tr>
</tbody>
</table>

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/MCLR Pin Function Select bit
  - 1 = GP3/MCLR pin function is MCLR
  - 0 = GP3/MCLR pin function is digital I/O, MCLR internally tied to VDD
- bit 3 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 Reserved: Read as ‘0’

Note 1: Refer to the “PIC10F200/202/204/206 Memory Programming Specifications” (DS41228) to determine how to access the Configuration Word. The Configuration Word is not user addressable during device operation.

2: INTRC is the only oscillator mode offered on the PIC10F200/202/204/206.
9.2 Oscillator Configurations

9.2.1 OSCILLATOR TYPES

The PIC10F200/202/204/206 devices are offered with Internal Oscillator mode only.

- INTOSC: Internal 4 MHz Oscillator

9.2.2 INTERNAL 4 MHz OSCILLATOR

The internal oscillator provides a 4 MHz (nominal) system clock (see Section 12.0 “Electrical Characteristics” for information on variation over voltage and temperature).

In addition, a calibration instruction is programmed into the last address of memory, which contains the calibration value for the internal oscillator. This location is always uncode protected, regardless of the code-protect settings. This value is programmed as a \texttt{MOVLW xx} instruction where \texttt{xx} is the calibration value and is placed at the Reset vector. This will load the \texttt{W} register with the calibration value upon Reset and the PC will then roll over to the users program at address 0x000. The user then has the option of writing the value to the OSCCAL Register (05h) or ignoring it.

OSCCAL, when written to with the calibration value, will “trim” the internal oscillator to remove process variation from the oscillator frequency.

Note: Erasing the device will also erase the pre-programmed internal calibration value for the internal oscillator. The calibration value must be read prior to erasing the part so it can be reprogrammed correctly later.

9.3 Reset

The device differentiates between various kinds of Reset:

- Power-on Reset (POR)
- MCLR Reset during normal operation
- MCLR Reset during Sleep
- WDT time-out Reset during normal operation
- WDT time-out Reset during Sleep
- Wake-up from Sleep on pin change
- Wake-up from Sleep on comparator change

Some registers are not reset in any way, they are unknown on POR and unchanged in any other Reset. Most other registers are reset to “Reset state” on Power-on Reset (POR), MCLR, WDT or Wake-up on pin change Reset during normal operation. They are not affected by a WDT Reset during Sleep or MCLR Reset during Sleep, since these Resets are viewed as resumption of normal operation. The exceptions to this are T0, PD, GPWUF and CWUF bits. They are set or cleared differently in different Reset situations. These bits are used in software to determine the nature of Reset. See Table 9-1 for a full description of Reset states of all registers.

### TABLE 9-1: RESET CONDITIONS FOR REGISTERS – PIC10F200/202/204/206

<table>
<thead>
<tr>
<th>Register</th>
<th>Address</th>
<th>Power-on Reset</th>
<th>MCLR Reset, WDT Time-out, Wake-up On Pin Change, Wake on Comparator Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>—</td>
<td>\texttt{qqqq qqqq} \textsuperscript{(1)}</td>
<td>\texttt{qqqq qqqq} \textsuperscript{(1)}</td>
</tr>
<tr>
<td>INDF</td>
<td>00h</td>
<td>\texttt{xxxx xxxx}</td>
<td>\texttt{uuuu uuuu}</td>
</tr>
<tr>
<td>TMR0</td>
<td>01h</td>
<td>\texttt{xxxx xxxx}</td>
<td>\texttt{uuuu uuuu}</td>
</tr>
<tr>
<td>PCL</td>
<td>02h</td>
<td>\texttt{1111 1111}</td>
<td>\texttt{1111 1111}</td>
</tr>
<tr>
<td>STATUS</td>
<td>03h</td>
<td>\texttt{00-1 1xxx}</td>
<td>\texttt{qq0q qqqq} \textsuperscript{(2)}</td>
</tr>
<tr>
<td>STATUS\textsuperscript{(3)}</td>
<td>03h</td>
<td>\texttt{00-1 1xxx}</td>
<td>\texttt{qqqq qqqq} \textsuperscript{(2)}</td>
</tr>
<tr>
<td>FSR</td>
<td>04h</td>
<td>\texttt{111x xxxx}</td>
<td>\texttt{111u uuuu}</td>
</tr>
<tr>
<td>OSCCAL</td>
<td>05h</td>
<td>\texttt{1111 1110}</td>
<td>\texttt{uuuu uuuu}</td>
</tr>
<tr>
<td>GPIO</td>
<td>06h</td>
<td>----- \texttt{xxxx}</td>
<td>----- \texttt{uuuu}</td>
</tr>
<tr>
<td>CMCON\textsuperscript{(3)}</td>
<td>07h</td>
<td>\texttt{1111 1111}</td>
<td>\texttt{uuuu uuuu}</td>
</tr>
<tr>
<td>OPTION</td>
<td>—</td>
<td>\texttt{1111 1111}</td>
<td>\texttt{1111 1111}</td>
</tr>
<tr>
<td>TRISGPIO</td>
<td>—</td>
<td>----- \texttt{1111}</td>
<td>----- \texttt{1111}</td>
</tr>
</tbody>
</table>

Legend: \texttt{u} = unchanged, \texttt{x} = unknown, \texttt{=} = unimplemented bit, read as '0', \texttt{q} = value depends on condition.

Note 1: Bits <7:2> of W register contain oscillator calibration values due to \texttt{MOVLW XX} instruction at top of memory.

Note 2: See Table 9-2 for Reset value for specific conditions.

Note 3: PIC10F204/206 only.
9.3.1 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 a I/O. See Figure 9-1.

FIGURE 9-1: MCLR SELECT

The Power-on Reset circuit and the Device Reset Timer (see Section 9.5 “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 9-3. VDD is allowed to rise and stabilize before bringing MCLR high. The chip will actually come out of Reset T_{DRT} msec after MCLR goes high.

In Figure 9-4, the on-chip Power-on Reset feature is being used (MCLR and VDD are tied together or the pin is programmed to be GP3). The VDD is stable before the Start-up Timer times out and there is no problem in getting a proper Reset. However, Figure 9-5 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 9-4).

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”, (DS00000607).

TABLE 9-2: RESET CONDITION FOR SPECIAL REGISTERS

<table>
<thead>
<tr>
<th></th>
<th>STATUS Address: 03h</th>
<th>PCL Address: 02h</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power-on Reset</td>
<td>00-1 1xxx</td>
<td>1111 1111</td>
</tr>
<tr>
<td>MCLR Reset during normal operation</td>
<td>000u uuuu</td>
<td>1111 1111</td>
</tr>
<tr>
<td>MCLR Reset during Sleep</td>
<td>0001 0uuu</td>
<td>1111 1111</td>
</tr>
<tr>
<td>WDT Reset during Sleep</td>
<td>0000 0uuu</td>
<td>1111 1111</td>
</tr>
<tr>
<td>WDT Reset normal operation</td>
<td>0000 0uuu</td>
<td>1111 1111</td>
</tr>
<tr>
<td>Wake-up from Sleep on pin change</td>
<td>1001 0uuu</td>
<td>1111 1111</td>
</tr>
<tr>
<td>Wake-up from Sleep on comparator change</td>
<td>0101 0uuu</td>
<td>1111 1111</td>
</tr>
</tbody>
</table>

Legend: u = unchanged, x = unknown, – = unimplemented bit, read as '0'.

9.4 Power-on Reset (POR)

The PIC10F200/202/204/206 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/MCLR/VPP pin as MCLR and tie through a resistor to VDD, or program the pin as GP3.

An internal weak pull-up resistor is implemented using a transistor (refer to Table 12-2 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 12.0 “Electrical Characteristics” for details.

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 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 9-2.
FIGURE 9-2: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

Vdd

Power-up Detect

POR (Power-on Reset)

GP3/MCLR/VPP

MCLRE

WDT Time-out

Pin Change

Sleep

Wake-up on pin change Reset

MCLR Reset

Start-up Timer (10 μs or 18 ms)

CHIP Reset

FIGURE 9-3: TIME-OUT SEQUENCE ON POWER-UP (MCLR PULLED LOW)

Vdd

MCLRE

Internal POR

DRT Time-out

Internal Reset

TDRT

FIGURE 9-4: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO Vdd): FAST Vdd RISE TIME

Vdd

MCLRE

Internal POR

DRT Time-out

Internal Reset
FIGURE 9-5: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD): SLOW VDD RISE TIME

Note: When VDD rises slowly, the TDRT time-out expires long before VDD has reached its final value. In this example, the chip will reset properly if, and only if, V1 ≥ VDD min.
9.5 Device Reset Timer (DRT)

On the PIC10F200/202/204/206 devices, the DRT runs any time the device is powered-up.

The DRT operates on an internal oscillator. The processor is kept in Reset as long as the DRT is active. The DRT delay allows VDD to rise above VDD min. and for the oscillator to stabilize.

The on-chip DRT keeps the devices in a Reset condition for approximately 18 ms after MCLR has reached a logic high (V_{IH MCLR}) level. Programming GP3/MCLR/VPP as MCLR and using an external RC network connected to the MCLR input is not required in most cases. This allows savings in cost-sensitive and/ or space restricted applications, as well as allowing the use of the GP3/MCLR/VPP pin as a general purpose input.

The Device Reset Time delays will vary from chip-to-chip due to VDD, temperature and process variation. See AC parameters for details.

Reset sources are POR, MCLR, WDT time-out and wake-up on pin change. See Section 9.9.2 “Wake-up from Sleep”, Notes 1, 2 and 3.

<table>
<thead>
<tr>
<th>Oscillator</th>
<th>POR Reset</th>
<th>Subsequent Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTOSC</td>
<td>18 ms (typical)</td>
<td>10 μs (typical)</td>
</tr>
</tbody>
</table>

9.6 Watchdog Timer (WDT)

The Watchdog Timer (WDT) is a free running on-chip RC oscillator, which does not require any external components. This RC oscillator is separate from the internal 4 MHz oscillator. This means that the WDT will run even if the main processor clock has been stopped, for example, by execution of a SLEEP instruction. During normal operation or Sleep, a WDT Reset or wake-up Reset, generates a device Reset.

The TO bit (STATUS<4>) will be cleared upon a Watchdog Timer Reset.

The WDT can be permanently disabled by programming the configuration WDTE as a ‘0’ (see Section 9.1 “Configuration Bits”). Refer to the PIC10F200/202/204/206 Programming Specifications to determine how to access the Configuration Word.

9.6.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). If a longer time-out period is desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT (under software control) by writing to the OPTION register. Thus, a time-out period of a nominal 2.3 seconds can be realized. These periods vary with temperature, VDD and part-to-part process variations (see DC specs).

Under worst-case conditions (VDD = Min., Temperature = Max., max. WDT prescaler), it may take several seconds before a WDT time-out occurs.

9.6.2 WDT PROGRAMMING CONSIDERATIONS

The CLRWD instruction clears the WDT and the postscaler, if assigned to the WDT, and prevents it from timing out and generating a device Reset.

The SLEEP instruction resets the WDT and the postscaler, if assigned to the WDT. This gives the maximum Sleep time before a WDT wake-up Reset.
FIGURE 9-6: WATCHDOG TIMER BLOCK DIAGRAM

TABLE 9-4: SUMMARY OF REGISTERS ASSOCIATED WITH THE WATCHDOG TIMER

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on Power-On Reset</th>
<th>Value on All Other Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>OPTION</td>
<td>GPWU</td>
<td>GPPU</td>
<td>T0CS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</td>
<td>1111 1111</td>
<td>1111 1111</td>
</tr>
</tbody>
</table>

Legend: Shaded boxes = Not used by Watchdog Timer, – = unimplemented, read as “0”, u = unchanged.
9.7 Time-out Sequence, Power-down and Wake-up from Sleep Status
Bits (TO, PD, GPWUF, CWUF)

The TO, PD, GPWUF and CWUF bits in the STATUS register can be tested to determine if a Reset condition has been caused by a power-up condition, a MCLR, Watchdog Timer (WDT) Reset, wake-up on comparator change or wake-up on pin change.

<table>
<thead>
<tr>
<th>CWUF</th>
<th>GPWUF</th>
<th>TO</th>
<th>PD</th>
<th>Reset Caused By</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>WDT wake-up from Sleep</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>WDT time-out (not from Sleep)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>MCLR wake-up from Sleep</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Power-up</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>x</td>
<td>x</td>
<td>MCLR not during Sleep</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Wake-up from Sleep on pin change</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Wake-up from Sleep on comparator change</td>
</tr>
</tbody>
</table>

Legend:  u = unchanged,  x = unknown,  – = unimplemented bit, read as ‘0’,  q = value depends on condition.

Note 1: The TO, PD, GPWUF and CWUF bits maintain their status (u) until a Reset occurs. A low-pulse on the MCLR input does not change the TO, PD, GPWUF or CWUF Status bits.

9.8 Reset on Brown-out

A Brown-out Reset is a condition where device power (Vdd) dips below its minimum value, but not to zero, and then recovers. The device should be reset in the event of a brown-out.

To reset PIC10F200/202/204/206 devices when a Brown-out Reset occurs, external brown-out protection circuits may be built, as shown in Figure 9-7 and Figure 9-8.

Note 1: This brown-out circuit is less expensive, although less accurate. Transistor Q1 turns off when Vdd is below a certain level such that:

\[
\frac{V_{dd} \cdot R_1}{R_1 + R_2} = 0.7V
\]

2: Pin must be confirmed as MCLR.
9.9 Power-down Mode (Sleep)

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

9.9.1 SLEEP

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

If enabled, the Watchdog Timer will be cleared but keeps running, the TO bit (STATUS<4>) is set, the 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 MCLR pin low.

For lowest current consumption while powered-down, the T0CKI input should be at VDD or VSS and the GP3/MCLR/VPP pin must be at a logic high level if MCLR is enabled.

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

9.9.2 WAKE-UP FROM SLEEP

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

1. An external Reset input on GP3/MCLR/VPP pin, when configured as MCLR.
2. A Watchdog Timer time-out Reset (if WDT was enabled).
3. A change on input pin GP0, GP1 or GP3 when wake-up on change is enabled.
4. A comparator output change has occurred when wake-up on comparator change is enabled.

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

**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.
9.10  **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 (Reset vector) can be read, regardless of the code protection bit setting.

9.11  **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 four bits of the ID locations and always program the upper eight bits as '0's.

9.12  **In-Circuit Serial Programming™**

The PIC10F200/202/204/206 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 and GP0 pins low while raising the MCLR (VPP) pin from VIL to VIHH (see programming specification). GP1 becomes the programming clock and GP0 becomes the programming data. Both GP1 and GP0 are Schmitt Trigger inputs in this mode.

After Reset, a 6-bit command is then supplied to the device. Depending on the command, 16 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 PIC10F200/202/204/206 Programming Specifications.

A typical In-Circuit Serial Programming connection is shown in **Figure 9-10**.
10.0 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 10-1, while the various opcode fields are summarized in Table 10-1.

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 10-1: OPCODE FIELD DESCRIPTIONS

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>Register file address (0x00 to 0x7F)</td>
</tr>
<tr>
<td>W</td>
<td>Working register (accumulator)</td>
</tr>
<tr>
<td>b</td>
<td>Bit address within an 8-bit file register</td>
</tr>
<tr>
<td>k</td>
<td>Literal field, constant data or label</td>
</tr>
<tr>
<td>x</td>
<td>Don’t care location (= 0 or 1)</td>
</tr>
</tbody>
</table>
| d     | Destination select: 
  d = 0 (store result in W) 
  d = 1 (store result in file register ‘f’) 
  Default is d = 1 |
| label | Label name |
| TOS   | Top-of-Stack |
| PC    | Program Counter |
| WDT   | Watchdog Timer counter |
| TO    | Time-out 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 |
| italic| User defined term (font is courier) |

FIGURE 10-1: GENERAL FORMAT FOR INSTRUCTIONS

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 10-1 shows the three general formats that the instructions can have. All examples in the figure use the following format to represent a hexadecimal number:

\[0x\text{hhh}\]

where ‘h’ signifies a hexadecimal digit.
### TABLE 10-2: INSTRUCTION SET SUMMARY

<table>
<thead>
<tr>
<th>Mnemonic, Operands</th>
<th>Description</th>
<th>Cycles</th>
<th>12-Bit Opcode</th>
<th>Status Affected</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>MSb</td>
<td>LSb</td>
<td></td>
</tr>
<tr>
<td>ADDWF f, d</td>
<td>Add W and f</td>
<td>1</td>
<td>0001 11df</td>
<td>ffff</td>
<td>C, DC, Z</td>
</tr>
<tr>
<td>ANDWF f, d</td>
<td>AND W with f</td>
<td>1</td>
<td>0001 01df</td>
<td>ffff</td>
<td>Z</td>
</tr>
<tr>
<td>CLRF f</td>
<td>Clear f</td>
<td>1</td>
<td>0000 01df</td>
<td>ffff</td>
<td>Z</td>
</tr>
<tr>
<td>CLRW —</td>
<td>Clear W</td>
<td>1</td>
<td>0000 0100 0000</td>
<td>0000</td>
<td>Z</td>
</tr>
<tr>
<td>COMF f, d</td>
<td>Complement f</td>
<td>1</td>
<td>0010 01df</td>
<td>ffff</td>
<td>Z</td>
</tr>
<tr>
<td>DECF f, d</td>
<td>Decrement f</td>
<td>1</td>
<td>0000 11df</td>
<td>ffff</td>
<td>Z</td>
</tr>
<tr>
<td>DECFSZ f, d</td>
<td>Decrement f, Skip if 0</td>
<td>(2)</td>
<td>0010 11df</td>
<td>ffff</td>
<td>None</td>
</tr>
<tr>
<td>INC f, d</td>
<td>Increment f</td>
<td>1</td>
<td>0010 10df</td>
<td>ffff</td>
<td>Z</td>
</tr>
<tr>
<td>INCFSZ f, d</td>
<td>Increment f, Skip if 0</td>
<td>(2)</td>
<td>0011 11df</td>
<td>ffff</td>
<td>None</td>
</tr>
<tr>
<td>IORWF f, d</td>
<td>Inclusive OR W with f</td>
<td>1</td>
<td>0001 00df</td>
<td>ffff</td>
<td>Z</td>
</tr>
<tr>
<td>MOVF f, d</td>
<td>Move f</td>
<td>1</td>
<td>0010 00df</td>
<td>ffff</td>
<td>Z</td>
</tr>
<tr>
<td>MOVWF f</td>
<td>Move W to f</td>
<td>1</td>
<td>0000 001f</td>
<td>ffff</td>
<td>None</td>
</tr>
<tr>
<td>NOP —</td>
<td>No Operation</td>
<td>1</td>
<td>0000 0000 0000</td>
<td>0000</td>
<td>None</td>
</tr>
<tr>
<td>RLF f, d</td>
<td>Rotate left f through Carry</td>
<td>1</td>
<td>0011 01df</td>
<td>ffff</td>
<td>C</td>
</tr>
<tr>
<td>RRF f, d</td>
<td>Rotate right f through Carry</td>
<td>1</td>
<td>0011 00df</td>
<td>ffff</td>
<td>C</td>
</tr>
<tr>
<td>SUBWF f, d</td>
<td>Subtract W from f</td>
<td>1</td>
<td>0000 10df</td>
<td>ffff</td>
<td>C, DC, Z</td>
</tr>
<tr>
<td>SWAPF f, d</td>
<td>Swap f</td>
<td>1</td>
<td>0011 10df</td>
<td>ffff</td>
<td>None</td>
</tr>
<tr>
<td>XORWF f, d</td>
<td>Exclusive OR W with f</td>
<td>1</td>
<td>0001 10df</td>
<td>ffff</td>
<td>Z</td>
</tr>
</tbody>
</table>

**BIT-ORIENTED FILE REGISTER OPERATIONS**

<table>
<thead>
<tr>
<th>Mnemonic, Operands</th>
<th>Description</th>
<th>Cycles</th>
<th>12-Bit Opcode</th>
<th>Status Affected</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>MSb</td>
<td>LSb</td>
<td></td>
</tr>
<tr>
<td>BCF f, b</td>
<td>Bit Clear f</td>
<td>1</td>
<td>0100 bbbf</td>
<td>ffff</td>
<td>None</td>
</tr>
<tr>
<td>BSF f, b</td>
<td>Bit Set f</td>
<td>1</td>
<td>0101 bbbf</td>
<td>ffff</td>
<td>None</td>
</tr>
<tr>
<td>BTFSC f, b</td>
<td>Bit Test f, Skip if Clear</td>
<td>(2)</td>
<td>0110 bbbf</td>
<td>ffff</td>
<td>None</td>
</tr>
<tr>
<td>BTFSS f, b</td>
<td>Bit Test f, Skip if Set</td>
<td>(2)</td>
<td>0111 bbbf</td>
<td>ffff</td>
<td>None</td>
</tr>
</tbody>
</table>

**LITERAL AND CONTROL OPERATIONS**

<table>
<thead>
<tr>
<th>Mnemonic, Operands</th>
<th>Description</th>
<th>Cycles</th>
<th>12-Bit Opcode</th>
<th>Status Affected</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>MSb</td>
<td>LSb</td>
<td></td>
</tr>
<tr>
<td>ANDLW k</td>
<td>AND literal with W</td>
<td>1</td>
<td>1110 kkkk kkkk</td>
<td>kkkk</td>
<td>Z</td>
</tr>
<tr>
<td>CALL k</td>
<td>Call Subroutine</td>
<td>2</td>
<td>1001 kkkk kkkk</td>
<td>None</td>
<td>1</td>
</tr>
<tr>
<td>CLRWDT</td>
<td>Clear Watchdog Timer</td>
<td>1</td>
<td>0000 0000 0100</td>
<td>TO, PD</td>
<td></td>
</tr>
<tr>
<td>GOTO k</td>
<td>Unconditional branch</td>
<td>2</td>
<td>101k kkkk kkkk</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>IORLW k</td>
<td>Inclusive OR literal with W</td>
<td>1</td>
<td>1101 kkkk kkkk</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>MOVLW k</td>
<td>Move literal to W</td>
<td>1</td>
<td>1100 kkkk kkkk</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>OPTION —</td>
<td>Load OPTION register</td>
<td>1</td>
<td>0000 0000 0010</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>RETLW k</td>
<td>Return, place Literal in W</td>
<td>2</td>
<td>1000 kkkk kkkk</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>SLEEP —</td>
<td>Go into Standby mode</td>
<td>1</td>
<td>0000 0000 0011</td>
<td>TO, PD</td>
<td></td>
</tr>
<tr>
<td>TRIS k</td>
<td>Load TRIS register</td>
<td>1</td>
<td>0000 0000 0fff</td>
<td>None</td>
<td>3</td>
</tr>
<tr>
<td>XORLW k</td>
<td>Exclusive OR literal to W</td>
<td>1</td>
<td>1111 kkkk kkkk</td>
<td>Z</td>
<td></td>
</tr>
</tbody>
</table>

**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”.

**Note 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'.

**Note 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.

**Note 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:**   
[\(\text{label}\)] ADDWF \(f,d\)

**Operands:**  
0 \(\leq f \leq 31\)  
d \(\in [0,1]\)

**Operation:**  
\((W) + (f) \rightarrow (\text{dest})\)

**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'.

### BCF

**Bit Clear f**

**Syntax:**   
[\(\text{label}\)] BCF \(f,b\)

**Operands:**  
0 \(\leq f \leq 31\)  
0 \(\leq b \leq 7\)

**Operation:**  
0 \(\rightarrow (f<b>)\)

**Status Affected:**  
None

**Description:**  
Bit 'b' in register 'f' is cleared.

### ANDLW

**AND literal with W**

**Syntax:**   
[\(\text{label}\)] ANDLW k

**Operands:**  
0 \(\leq k \leq 255\)

**Operation:**  
(W).AND. (k) \(\rightarrow (W)\)

**Status Affected:**  
Z

**Description:**  
The contents of the W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register.

### BSF

**Bit Set f**

**Syntax:**   
[\(\text{label}\)] BSF \(f,b\)

**Operands:**  
0 \(\leq f \leq 31\)  
0 \(\leq b \leq 7\)

**Operation:**  
1 \(\rightarrow (f<b>)\)

**Status Affected:**  
None

**Description:**  
Bit 'b' in register 'f' is set.

### ANDWF

**AND W with f**

**Syntax:**   
[\(\text{label}\)] ANDWF \(f,d\)

**Operands:**  
0 \(\leq f \leq 31\)  
d \(\in [0,1]\)

**Operation:**  
(W).AND. (f) \(\rightarrow (\text{dest})\)

**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'.

### BTFSC

**Bit Test f, Skip if Clear**

**Syntax:**   
[\(\text{label}\)] BTFSC \(f,b\)

**Operands:**  
0 \(\leq f \leq 31\)  
0 \(\leq b \leq 7\)

**Operation:**  
skip if (f\(<b>) = 0\)

**Status Affected:**  
None

**Description:**  
If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a 2-cycle instruction.
<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BTFSS</strong></td>
<td>Bit Test f, Skip if Set</td>
</tr>
<tr>
<td>Syntax:</td>
<td>[ label ] BTFSS f,b</td>
</tr>
<tr>
<td>Operands:</td>
<td>0 ≤ f ≤ 31 0 ≤ b &lt; 7</td>
</tr>
<tr>
<td>Operation:</td>
<td>skip if (f&lt;b) = 1</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>None</td>
</tr>
<tr>
<td>Description:</td>
<td>If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution, is discarded and a <strong>NOP</strong> is executed instead, making this a 2-cycle instruction.</td>
</tr>
</tbody>
</table>

| **CALL**   | Subroutine Call |
| Syntax:    | \[ label \] CALL k |
| Operands:  | 0 ≤ k ≤ 255 |
| Operation: | (PC) + 1 → Top-of-Stack; k → PC<7:0>; (STATUS<6:5>) → PC<10:9>; 0 → PC<8> |
| Status Affected: | None |
| Description: | Subroutine call. First, return address (PC + 1) is PUSHed onto the stack. The 8-bit immediate address is loaded into PC bits <7:0>. The upper bits PC<10:9> are loaded from STATUS<6:5>, PC<8> is cleared. **CALL** is a 2-cycle instruction. |

| **CLRW**   | Clear W |
| Syntax:    | \[ label \] CLRW |
| Operands:  | None |
| Operation: | 00h → (W); 1 → Z |
| Status Affected: | Z |
| Description: | The W register is cleared. Zero bit (Z) is set. |

| **CLRWDT** | Clear Watchdog Timer |
| Syntax:    | \[ label \] CLRWDT |
| Operands:  | None |
| Operation: | 00h → WDT; 0 → WDT prescaler (if assigned); 1 → TO; 1 → PD |
| Status Affected: | TO, PD |
| Description: | The **CLRWDT** instruction resets the WDT. It also resets the prescaler, if the prescaler is assigned to the WDT and not Timer0. Status bits TO and PD are set. |

| **CLRF**   | Clear f |
| Syntax:    | \[ label \] CLRF f |
| Operands:  | 0 ≤ f ≤ 31 |
| Operation: | 00h → (f); 1 → Z |
| Status Affected: | Z |
| Description: | The contents of register 'f' are cleared and the Z bit is set. |

<p>| <strong>COMF</strong>   | Complement f |
| Syntax:    | [ label ] COMF f,d |
| Operands:  | 0 ≤ f ≤ 31 d ∈ [0,1] |
| Operation: | (f) → (dest) |
| Status Affected: | Z |
| Description: | The contents of register 'f' are complemented. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'. |</p>
<table>
<thead>
<tr>
<th>DEC F</th>
<th>Decrement f</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax:</td>
<td>[ label ] DEC f,d</td>
</tr>
<tr>
<td>Operands:</td>
<td>0 ≤ f ≤ 31</td>
</tr>
<tr>
<td></td>
<td>d ∈ [0,1]</td>
</tr>
<tr>
<td>Operation:</td>
<td>(f) – 1 → (dest)</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>Z</td>
</tr>
<tr>
<td>Description:</td>
<td>Decrement 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'.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INC F</th>
<th>Increment f</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax:</td>
<td>[ label ] INC f,d</td>
</tr>
<tr>
<td>Operands:</td>
<td>0 ≤ f ≤ 31</td>
</tr>
<tr>
<td></td>
<td>d ∈ [0,1]</td>
</tr>
<tr>
<td>Operation:</td>
<td>(f) + 1 → (dest)</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>Z</td>
</tr>
<tr>
<td>Description:</td>
<td>The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DECFSZ</th>
<th>Decrement f, Skip if 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax:</td>
<td>[ label ] DECFSZ f,d</td>
</tr>
<tr>
<td>Operands:</td>
<td>0 ≤ f ≤ 31</td>
</tr>
<tr>
<td></td>
<td>d ∈ [0,1]</td>
</tr>
<tr>
<td>Operation:</td>
<td>(f) – 1 → d; skip if result = 0</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>None</td>
</tr>
<tr>
<td>Description:</td>
<td>The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '0', the next instruction, which is already fetched, is discarded and a NOP is executed instead making it a 2-cycle instruction.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INCFSZ</th>
<th>Increment f, Skip if 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax:</td>
<td>[ label ] INCFSZ f,d</td>
</tr>
<tr>
<td>Operands:</td>
<td>0 ≤ f ≤ 31</td>
</tr>
<tr>
<td></td>
<td>d ∈ [0,1]</td>
</tr>
<tr>
<td>Operation:</td>
<td>(f) + 1 → (dest), skip if result = 0</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>None</td>
</tr>
<tr>
<td>Description:</td>
<td>The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '0', then the next instruction, which is already fetched, is discarded and a NOP is executed instead making it a 2-cycle instruction.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GOTO</th>
<th>Unconditional Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax:</td>
<td>[ label ] GOTO k</td>
</tr>
<tr>
<td>Operands:</td>
<td>0 ≤ k ≤ 511</td>
</tr>
<tr>
<td>Operation:</td>
<td>k → PC&lt;8:0&gt;; STATUS&lt;6:5&gt; → PC&lt;10:9&gt;</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>None</td>
</tr>
<tr>
<td>Description:</td>
<td>GOTO is an unconditional branch. The 9-bit immediate value is loaded into PC bits &lt;8:0&gt;. The upper bits of PC are loaded from STATUS&lt;6:5&gt;. GOTO is a 2-cycle instruction.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IORLW</th>
<th>Inclusive OR literal with W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax:</td>
<td>[ label ] IORLW k</td>
</tr>
<tr>
<td>Operands:</td>
<td>0 ≤ k ≤ 255</td>
</tr>
<tr>
<td>Operation:</td>
<td>(W) .OR. (k) → (W)</td>
</tr>
<tr>
<td>Status Affected:</td>
<td>Z</td>
</tr>
<tr>
<td>Description:</td>
<td>The contents of the W register are OR'ed with the 8-bit literal 'k'. The result is placed in the W register.</td>
</tr>
</tbody>
</table>
### IORWF
**Inclusive OR W with f**

**Syntax:** `[ label ] IORWF f,d  
**Operands:** 0 ≤ f ≤ 31  
\( d \in [0,1] \)  
**Operation:** \((W).OR. (f) \rightarrow (dest)\)  
**Status Affected:** Z  
**Description:** Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

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

**Syntax:** `[ label ] MOVWF f  
**Operands:** 0 ≤ f ≤ 31  
**Operation:** \((W) \rightarrow (f)\)  
**Status Affected:** None  
**Description:** Move data from the W register to register 'f'.

### MOVF
**Move f**

**Syntax:** `[ label ] MOVF f,d  
**Operands:** 0 ≤ f ≤ 31  
\( d \in [0,1] \)  
**Operation:** \((f) \rightarrow (dest)\)  
**Status Affected:** Z  
**Description:** The contents of register 'f' are moved to destination 'd'. If 'd' is '0', destination is the W register. If 'd' is '1', the destination is file register 'f'. 'd' = 1 is useful as a test of a file register, since status flag Z is affected.

### MOVLW
**Move literal to W**

**Syntax:** `[ label ] MOVLW k  
**Operands:** 0 ≤ k ≤ 255  
**Operation:** \(k \rightarrow (W)\)  
**Status Affected:** None  
**Description:** The 8-bit literal 'k' is loaded into the W register. The "don't cares" will assembled as '0's.

### OPTION
**Load OPTION Register**

**Syntax:** `[ label ] OPTION  
**Operands:** None  
**Operation:** \((W) \rightarrow \text{Option}\)  
**Status Affected:** None  
**Description:** The content of the W register is loaded into the OPTION register.
**RETLW** Return with literal in W

**Syntax:** `[ label ] RETLW k

**Operands:** 0 ≤ k ≤ 255

**Operation:** k → (W);
TOS → PC

**Status Affected:** None

**Description:** The W register is loaded with the 8-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a 2-cycle instruction.

**SLEEP** Enter SLEEP Mode

**Syntax:** `[ label ] SLEEP

**Operands:** None

**Operation:** 00h → WDT;
0 → WDT prescaler;
1 → TO;
0 → PD

**Status Affected:** TO, PD, RBWUF

**Description:** Time-out Status bit (TO) is set. The Power-down Status bit (PD) is cleared.
RBWUF is unaffected.
The WDT and its prescaler are cleared.
The processor is put into Sleep mode with the oscillator stopped.
See Section 9.9 “Power-down Mode (Sleep)” for more details.

**RLF** Rotate Left f through Carry

**Syntax:** `[ label ] RLF f,d

**Operands:** 0 ≤ f ≤ 31
d ∈ [0,1]

**Operation:** See description below

**Status Affected:** C

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

**SUBWF** Subtract W from f

**Syntax:** `[ label ] SUBWF f,d

**Operands:** 0 ≤ f ≤ 31
d ∈ [0,1]

**Operation:** (f) – (W) → (dest)

**Status Affected:** C, DC, Z

**Description:** Subtract (2’s complement method) the 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'.

**RRF** Rotate Right f through Carry

**Syntax:** `[ label ] RRF f,d

**Operands:** 0 ≤ f ≤ 31
d ∈ [0,1]

**Operation:** See description below

**Status Affected:** C

**Description:** The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

**SWAPF** Swap Nibbles in f

**Syntax:** `[ label ] SWAPF f,d

**Operands:** 0 ≤ f ≤ 31
d ∈ [0,1]

**Operation:** (f<3:0>) → (dest<7:4>);
(f<7:4>) → (dest<3:0>)

**Status Affected:** None

**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'.
## TRIS

**Load TRIS Register**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>[ label ] TRIS f</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands</td>
<td>f = 6</td>
</tr>
<tr>
<td>Operation</td>
<td>(W) \rightarrow TRIS register f</td>
</tr>
<tr>
<td>Status Affected</td>
<td>None</td>
</tr>
<tr>
<td>Description</td>
<td>TRIS register 'f' (f = 6 or 7) is loaded with the contents of the W register</td>
</tr>
</tbody>
</table>

## XORWF

**Exclusive OR W with f**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>[ label ] XORWF f,d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands</td>
<td>0 \leq f \leq 31</td>
</tr>
<tr>
<td>d \in {0,1}</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>(W) \rightarrow (dest)</td>
</tr>
<tr>
<td>Status Affected</td>
<td>Z</td>
</tr>
<tr>
<td>Description</td>
<td>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'.</td>
</tr>
</tbody>
</table>

## XORLW

**Exclusive OR literal with W**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>[ label ] XORLW k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operands</td>
<td>0 \leq k \leq 255</td>
</tr>
<tr>
<td>Operation</td>
<td>(W) \rightarrow (W)</td>
</tr>
<tr>
<td>Status Affected</td>
<td>Z</td>
</tr>
<tr>
<td>Description</td>
<td>The contents of the W register are XOR'ed with the 8-bit literal 'k'. The result is placed in the W register.</td>
</tr>
</tbody>
</table>
11.0 DEVELOPMENT SUPPORT
The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:
- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/
    MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for
    Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards,
  Evaluation Kits and Starter Kits
- Third-party development tools

11.1 MPLAB X Integrated Development Environment Software
The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:
- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:
- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:
- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:
- Local file history feature
- Built-in support for Bugzilla issue tracker
11.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip’s 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

• Support for the entire device instruction set
• Support for fixed-point and floating-point data
• Command-line interface
• Rich directive set
• Flexible macro language
• MPLAB X IDE compatibility

11.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

• Integration into MPLAB X IDE projects
• User-defined macros to streamline assembly code
• Conditional assembly for multipurpose source files
• Directives that allow complete control over the assembly process

11.4 MPLINK Object Linker/
MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

• Efficient linking of single libraries instead of many smaller files
• Enhanced code maintainability by grouping related modules together
• Flexible creation of libraries with easy module listing, replacement, deletion and extraction

11.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

• Support for the entire device instruction set
• Support for fixed-point and floating-point data
• Command-line interface
• Rich directive set
• Flexible macro language
• MPLAB X IDE compatibility
11.6 **MPLAB X SIM Software Simulator**

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

11.7 **MPLAB REAL ICE In-Circuit Emulator System**

The MPLAB REAL ICE In-Circuit Emulator System is Microchip’s next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugging and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer’s PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

11.8 **MPLAB ICD 3 In-Circuit Debugger System**

The MPLAB ICD 3 In-Circuit Debugger System is Microchip’s most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugging and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer’s PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

11.9 **PICkit 3 In-Circuit Debugger/Programmer**

The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer’s PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

11.10 **MPLAB PM3 Device Programmer**

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.
11.11 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

11.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

• Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
• Software Tools from companies, such as Gimpel and Trace Systems
• Protocol Analyzers from companies, such as Saleae and Total Phase
• Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
• Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®
12.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings(†)

<table>
<thead>
<tr>
<th>Condition</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient temperature under bias</td>
<td>-40°C to +125°C</td>
</tr>
<tr>
<td>Storage temperature</td>
<td>-65°C to +150°C</td>
</tr>
<tr>
<td>Voltage on VDD with respect to VSS</td>
<td>0 to +6.5V</td>
</tr>
<tr>
<td>Voltage on MCLR with respect to VSS</td>
<td>0 to +13.5V</td>
</tr>
<tr>
<td>Voltage on all other pins with respect to VSS</td>
<td>-0.3V to (VDD + 0.3V)</td>
</tr>
<tr>
<td>Total power dissipation†</td>
<td>800 mW</td>
</tr>
<tr>
<td>Max. current out of VSS pin</td>
<td>80 mA</td>
</tr>
<tr>
<td>Max. current into VDD pin</td>
<td>80 mA</td>
</tr>
<tr>
<td>Input clamp current, I&lt;sub&gt;IK&lt;/sub&gt; (VI &lt; 0 or VI &gt; VDD)</td>
<td>±20 mA</td>
</tr>
<tr>
<td>Output clamp current, I&lt;sub&gt;OK&lt;/sub&gt; (VO &lt; 0 or VO &gt; VDD)</td>
<td>±20 mA</td>
</tr>
<tr>
<td>Max. output current sunk by any I/O pin</td>
<td>25 mA</td>
</tr>
<tr>
<td>Max. output current sourced by any I/O pin</td>
<td>25 mA</td>
</tr>
<tr>
<td>Max. output current sourced by I/O port</td>
<td>75 mA</td>
</tr>
<tr>
<td>Max. output current sunk by I/O port</td>
<td>75 mA</td>
</tr>
</tbody>
</table>

Note 1: Power dissipation is calculated as follows: 

$$P_{DIS} = VDD \times (I_{DD} - \sum IOH) + \sum (VDD - VOH) \times IOH + \sum (VOL \times IOL)$$

†NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
FIGURE 12-1: PIC10F200/202/204/206 VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C
### 12.1 DC Characteristics: PIC10F200/202/204/206 (Industrial)

<table>
<thead>
<tr>
<th>Param. No.</th>
<th>Sym.</th>
<th>Characteristic</th>
<th>Min.</th>
<th>Typ.(^{(1)})</th>
<th>Max.</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>D001</td>
<td>VDD</td>
<td>Supply Voltage</td>
<td>2.0</td>
<td>5.5</td>
<td>V</td>
<td>See Figure 12-1</td>
<td></td>
</tr>
<tr>
<td>D002</td>
<td>VDR</td>
<td>RAM Data Retention Voltage(^{(2)})</td>
<td>1.5*</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>Device in Sleep mode</td>
</tr>
<tr>
<td>D003</td>
<td>VPOR</td>
<td>VDD Start Voltage to ensure Power-on Reset</td>
<td>—</td>
<td>Vss</td>
<td>—</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>D004</td>
<td>SVDD</td>
<td>VDD Rise Rate to ensure Power-on Reset</td>
<td>0.05*</td>
<td>—</td>
<td>—</td>
<td>V/ms</td>
<td></td>
</tr>
</tbody>
</table>

| D010       | IDD  | Supply Current\(^{(3)}\) | —    | 175            | 275  | μA    | VDD = 2.0V |
|            |      |                             | —    | 0.63           | 1.1  | mA    | VDD = 5.0V |
| D020       | IPD  | Power-down Current\(^{(4)}\) | —    | 0.1            | 1.2  | μA    | VDD = 2.0V |
|            |      |                             | —    | 0.35           | 2.4  | μA    | VDD = 5.0V |
| D022       | IWDT | WDT Current\(^{(5)}\) | —    | 1.0            | 3    | μA    | VDD = 2.0V |
|            |      |                             | —    | 7              | 16   | μA    | VDD = 5.0V |
| D023       | ICMP | Comparator Current\(^{(6)}\) | —    | 12             | 23   | μA    | VDD = 2.0V |
|            |      |                             | —    | 44             | 80   | μA    | VDD = 5.0V |
| D024       | IVREF | Internal Reference Current\(^{(5,6)}\) | —    | 85             | 115  | μA    | VDD = 2.0V |
|            |      |                             | —    | 175            | 195  | μA    | VDD = 5.0V |

* These parameters are characterized but not tested.

**Note 1:** Data in the Typical (“Typ.”) column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

**Note 2:** This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

**Note 3:** The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, bus rate, internal code execution pattern and temperature also have an impact on the current consumption.

  a) The test conditions for all IDD measurements in active operation mode are:
  All I/O pins tri-stated, pulled to Vss, T0CKI = VDD, MCLR = VDD; WDT enabled/disabled as specified.

  b) For standby current measurements, the conditions are the same, except that the device is in Sleep mode.

**Note 4:** Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or Vss.

**Note 5:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled.

**Note 6:** Measured with the comparator enabled.
# 12.2 DC Characteristics: PIC10F200/202/204/206 (Extended)

## DC CHARACTERISTICS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D001</td>
<td>VDD</td>
<td>Supply Voltage</td>
<td>2.0</td>
<td></td>
<td>5.5</td>
<td>V</td>
<td>See Figure 12-1</td>
</tr>
<tr>
<td>D002</td>
<td>VDR</td>
<td>RAM Data Retention Voltage(2)</td>
<td>1.5*</td>
<td></td>
<td>—</td>
<td>V</td>
<td>Device in Sleep mode</td>
</tr>
<tr>
<td>D003</td>
<td>VPOR</td>
<td>VDD Start Voltage to ensure Power-on Reset</td>
<td>—</td>
<td>Vss</td>
<td>—</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>D004</td>
<td>SVDD</td>
<td>VDD Rise Rate to ensure Power-on Reset</td>
<td>0.05*</td>
<td></td>
<td>—</td>
<td>V/ms</td>
<td></td>
</tr>
<tr>
<td>D010</td>
<td>IDD</td>
<td>Supply Current(3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>175</td>
<td>275</td>
<td>μA</td>
<td>VDD = 2.0V</td>
</tr>
<tr>
<td>D020</td>
<td>IPD</td>
<td>Power-down Current(4)</td>
<td></td>
<td>0.1</td>
<td>9</td>
<td>μA</td>
<td>VDD = 2.0V</td>
</tr>
<tr>
<td>D022</td>
<td>IWDT</td>
<td>WDT Current(5)</td>
<td></td>
<td>1.0</td>
<td>18</td>
<td>μA</td>
<td>VDD = 2.0V</td>
</tr>
<tr>
<td>D023</td>
<td>ICMP</td>
<td>Comparator Current(5)</td>
<td></td>
<td>12</td>
<td>27</td>
<td>μA</td>
<td>VDD = 2.0V</td>
</tr>
<tr>
<td>D024</td>
<td>VREF</td>
<td>Internal Reference Current(5,6)</td>
<td></td>
<td>85</td>
<td>120</td>
<td>μA</td>
<td>VDD = 2.0V</td>
</tr>
</tbody>
</table>

* These parameters are characterized but not tested.

### Notes:

1. Data in the Typical ("Typ.") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.
2. This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.
3. The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, bus rate, internal code execution pattern and temperature also have an impact on the current consumption.
   a) The test conditions for all IDD measurements in active operation mode are:
      All I/O pins tri-stated, pulled to VSS, T0CKI = VDD, MCLR = VDD; WDT enabled/disabled as specified.
   b) For standby current measurements, the conditions are the same, except that the device is in Sleep mode.
4. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS.
5. The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled.
6. Measured with the Comparator enabled.
### 12.3 DC Characteristics: PIC10F200/202/204/206 (Industrial, Extended)

#### DC CHARACTERISTICS

<table>
<thead>
<tr>
<th>Param. No.</th>
<th>Sym.</th>
<th>Characteristic</th>
<th>Min.</th>
<th>Typ.†</th>
<th>Max.</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VIL</td>
<td>Input Low Voltage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D030</td>
<td></td>
<td>I/O ports:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>with TTL buffer</td>
<td>Vss</td>
<td>—</td>
<td>0.8</td>
<td>V</td>
<td>For all $4.5V \leq \text{VDD} \leq 5.5V$</td>
</tr>
<tr>
<td>D030A</td>
<td></td>
<td></td>
<td>Vss</td>
<td>—</td>
<td>0.15</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>D031</td>
<td></td>
<td>with Schmitt Trigger buffer</td>
<td>Vss</td>
<td>—</td>
<td>0.2</td>
<td>VDD</td>
<td></td>
</tr>
<tr>
<td>D032</td>
<td></td>
<td>MCLR, T0CKI</td>
<td>Vss</td>
<td>—</td>
<td>0.2</td>
<td>VDD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VIH</td>
<td>Input High Voltage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D040</td>
<td></td>
<td>I/O ports:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>with TTL buffer</td>
<td>2.0</td>
<td>—</td>
<td>—</td>
<td>VDD</td>
<td>$4.5V \leq \text{VDD} \leq 5.5V$</td>
</tr>
<tr>
<td>D040A</td>
<td></td>
<td></td>
<td>0.25</td>
<td>VDD + 0.8</td>
<td>—</td>
<td>VDD</td>
<td>Otherwise</td>
</tr>
<tr>
<td>D041</td>
<td></td>
<td>with Schmitt Trigger buffer</td>
<td>0.8</td>
<td>VDD</td>
<td>—</td>
<td>VDD</td>
<td>For entire VDD range</td>
</tr>
<tr>
<td>D042</td>
<td></td>
<td>MCLR, T0CKI</td>
<td>0.8</td>
<td>VDD</td>
<td>—</td>
<td>VDD</td>
<td></td>
</tr>
<tr>
<td>D070</td>
<td>IPUR</td>
<td>GPIO weak pull-up current(3)</td>
<td>50</td>
<td>250</td>
<td>400</td>
<td>μA</td>
<td>$\text{VDD} = 5V, \text{VPIN} = \text{VSS}$</td>
</tr>
<tr>
<td>D060</td>
<td></td>
<td>Input Leakage Current(1, 2)</td>
<td></td>
<td></td>
<td></td>
<td>μA</td>
<td></td>
</tr>
<tr>
<td>D061</td>
<td></td>
<td>I/O ports</td>
<td>—</td>
<td>±0.1</td>
<td>±1</td>
<td>±5</td>
<td>$\text{Vss} \leq \text{VPIN} \leq \text{VDD}$, Pin at high-impedance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GP3/MCLR(3)</td>
<td>—</td>
<td>±0.7</td>
<td>±5</td>
<td>μA</td>
<td>$\text{Vss} \leq \text{VPIN} \leq \text{VDD}$</td>
</tr>
<tr>
<td>D080</td>
<td></td>
<td>Output Low Voltage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D080A</td>
<td></td>
<td>I/O ports</td>
<td>—</td>
<td>—</td>
<td>0.6</td>
<td>V</td>
<td>$\text{IOL} = 8.5 \text{ mA, VDD} = 4.5V, -40°C to +85°C$</td>
</tr>
<tr>
<td>D090</td>
<td></td>
<td>Output High Voltage</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D090A</td>
<td></td>
<td>I/O ports(2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D101</td>
<td></td>
<td>Capacitive Loading Specs on Output Pins</td>
<td>—</td>
<td>—</td>
<td>50*</td>
<td>pF</td>
<td></td>
</tr>
</tbody>
</table>

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

* These parameters are for design guidance only and are not tested.

**Note 1:** The leakage current on the MCLR 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.

**Note 2:** Negative current is defined as coming out of the pin.

**Note 3:** This specification applies when GP3/MCLR is configured as an input with pull-up disabled. The leakage current of the MCLR circuit is higher than the standard I/O logic.
### TABLE 12-1: COMPARATOR SPECIFICATIONS

<table>
<thead>
<tr>
<th>Param. No.</th>
<th>Sym.</th>
<th>Characteristics</th>
<th>Min.</th>
<th>Typ.†</th>
<th>Max.</th>
<th>Units</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>D300</td>
<td>VOS</td>
<td>Input Offset Voltage</td>
<td>—</td>
<td>±5.0</td>
<td>±10</td>
<td>mV</td>
<td>*(VDD - 1.5)/2</td>
</tr>
<tr>
<td>D301</td>
<td>VCM</td>
<td>Input Common Mode Voltage</td>
<td>0</td>
<td>—</td>
<td>VDD–1.5*</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>D302</td>
<td>CMRR</td>
<td>Common Mode Rejection Ratio</td>
<td>55*</td>
<td>—</td>
<td>—</td>
<td>dB</td>
<td></td>
</tr>
<tr>
<td>D303*</td>
<td>TRT</td>
<td>Response Time</td>
<td>Falling</td>
<td>150</td>
<td>600</td>
<td>ns</td>
<td>*(Note 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rising</td>
<td>200</td>
<td>1000</td>
<td>ns</td>
<td></td>
</tr>
<tr>
<td>D304*</td>
<td>TMC2COV</td>
<td>Comparator Mode Change to Output Valid</td>
<td>—</td>
<td>—</td>
<td>10*</td>
<td>μs</td>
<td></td>
</tr>
<tr>
<td>D305</td>
<td>VIVRF</td>
<td>Internal Reference Voltage</td>
<td>0.55</td>
<td>0.6</td>
<td>0.65</td>
<td>V</td>
<td>2.0V ≤ VDD ≤ 5.5V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-40°C ≤ TA ≤ ±125°C (extended)</td>
</tr>
</tbody>
</table>

* 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:** Response time is measured with one comparator input at (VDD - 1.5)/2 - 100 mV to (VDD - 1.5)/2 + 20 mV.

### TABLE 12-2: PULL-UP RESISTOR RANGES

<table>
<thead>
<tr>
<th>VDD (Volts)</th>
<th>Temperature (°C)</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP0/GP1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>-40</td>
<td>73K</td>
<td>105K</td>
<td>186K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>73K</td>
<td>113K</td>
<td>187K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>85</td>
<td>82K</td>
<td>123K</td>
<td>190K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>125</td>
<td>86K</td>
<td>132K</td>
<td>190K</td>
<td>Ω</td>
</tr>
<tr>
<td>5.5</td>
<td>-40</td>
<td>15K</td>
<td>21K</td>
<td>33K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>15K</td>
<td>22K</td>
<td>34K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>85</td>
<td>19K</td>
<td>26K</td>
<td>35K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>125</td>
<td>23K</td>
<td>29K</td>
<td>35K</td>
<td>Ω</td>
</tr>
<tr>
<td>GP3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>-40</td>
<td>63K</td>
<td>81K</td>
<td>96K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>77K</td>
<td>93K</td>
<td>116K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>85</td>
<td>82K</td>
<td>96k</td>
<td>116K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>125</td>
<td>86K</td>
<td>100K</td>
<td>119K</td>
<td>Ω</td>
</tr>
<tr>
<td>5.5</td>
<td>-40</td>
<td>16K</td>
<td>20k</td>
<td>22K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>16K</td>
<td>21K</td>
<td>23K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>85</td>
<td>24K</td>
<td>25k</td>
<td>28K</td>
<td>Ω</td>
</tr>
<tr>
<td></td>
<td>125</td>
<td>26K</td>
<td>27K</td>
<td>29K</td>
<td>Ω</td>
</tr>
</tbody>
</table>
12.4 Timing Parameter Symbology and Load Conditions – PIC10F200/202/204/206

The timing parameter symbols have been created following one of the following formats:
1. TppS2ppS
2. TppS

<table>
<thead>
<tr>
<th>T</th>
<th>Frequency</th>
<th>T</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td></td>
<td>T</td>
<td></td>
</tr>
</tbody>
</table>

Lowercase subscripts (pp) and their meanings:

<table>
<thead>
<tr>
<th>pp</th>
<th>to</th>
<th>mc</th>
<th>MCLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>ck</td>
<td>CLKOUT</td>
<td>osc</td>
<td>Oscillator</td>
</tr>
<tr>
<td>cy</td>
<td>Cycle time</td>
<td>t0</td>
<td>T0CKI</td>
</tr>
<tr>
<td>drt</td>
<td>Device Reset Timer</td>
<td>wdt</td>
<td>Watchdog Timer</td>
</tr>
<tr>
<td>io</td>
<td>I/O port</td>
<td>wdt</td>
<td>Watchdog Timer</td>
</tr>
</tbody>
</table>

Uppercase letters and their meanings:

<table>
<thead>
<tr>
<th>S</th>
<th>F</th>
<th>Fall</th>
<th>P</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>H</td>
<td>High</td>
<td>R</td>
<td>Rise</td>
</tr>
<tr>
<td></td>
<td>I</td>
<td>Invalid (high-impedance)</td>
<td>V</td>
<td>Valid</td>
</tr>
<tr>
<td></td>
<td>L</td>
<td>Low</td>
<td>Z</td>
<td>High-impedance</td>
</tr>
</tbody>
</table>

FIGURE 12-2: LOAD CONDITIONS – PIC10F200/202/204/206

Legend:

\[ CL = 50 \text{ pF} \text{ for all pins} \]
TABLE 12-3: CALIBRATED INTERNAL RC FREQUENCIES – PIC10F200/202/204/206

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F10</td>
<td>Fosc</td>
<td>Internal Calibrated INTOSC Frequency(^{1,2})</td>
<td>± 1%</td>
<td>3.96</td>
<td>4.00</td>
<td>4.04</td>
<td>MHz</td>
<td>VDD=3.5V @ 25°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>± 2%</td>
<td>3.92</td>
<td>4.00</td>
<td>4.08</td>
<td>MHz</td>
<td>2.5V ≤ VDD ≤ 5.5V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>± 5%</td>
<td>3.80</td>
<td>4.00</td>
<td>4.20</td>
<td>MHz</td>
<td>0°C ≤ TA ≤ +85°C (industrial)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.0V ≤ VDD ≤ 5.5V</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-40°C ≤ TA ≤ +85°C (industrial)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-40°C ≤ TA ≤ +125°C (extended)</td>
</tr>
</tbody>
</table>

* These parameters are characterized but not tested.
† 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.

Note 1: To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 µF and 0.01 µF values in parallel are recommended.
2: Under stable VDD conditions.

FIGURE 12-3: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER TIMING – PIC10F200/202/204/206

Note 1: I/O pins must be taken out of High-Impedance mode by enabling the output drivers in software.
2: Runs on POR only.
TABLE 12-4: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER – PIC10F200/202/204/206

**AC CHARACTERISTICS**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>TMC₇</td>
<td>MCLR Pulse Width (low)</td>
<td>2*</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td>VDD = 5V, -40°C to +85°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5*</td>
<td>—</td>
<td>—</td>
<td>μs</td>
<td>VDD = 5.0V</td>
</tr>
<tr>
<td>31</td>
<td>TWD₇</td>
<td>Watchdog Timer Time-out</td>
<td>10</td>
<td>16</td>
<td>29</td>
<td>ms</td>
<td>VDD = 5.0V (industrial)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Period (no prescaler)</td>
<td>10</td>
<td>16</td>
<td>31</td>
<td>ms</td>
<td>VDD = 5.0V (extended)</td>
</tr>
<tr>
<td>32</td>
<td>TDR₇</td>
<td>Device Reset Timer Period (standard)</td>
<td>10</td>
<td>16</td>
<td>29</td>
<td>ms</td>
<td>VDD = 5.0V (industrial)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>16</td>
<td>31</td>
<td>ms</td>
<td>VDD = 5.0V (extended)</td>
</tr>
<tr>
<td>34</td>
<td>TIO₂</td>
<td>I/O High-impedance from MCLR low</td>
<td>—</td>
<td>—</td>
<td>2*</td>
<td>μs</td>
<td></td>
</tr>
</tbody>
</table>

* 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.

FIGURE 12-4: TIMER0 CLOCK TIMINGS – PIC10F200/202/204/206

![T0CKI clock timings diagram]

TABLE 12-5: TIMER0 CLOCK REQUIREMENTS – PIC10F200/202/204/206

**AC CHARACTERISTICS**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>T₁₀H</td>
<td>T₀CKI High Pulse Width</td>
<td>No Prescaler</td>
<td>0.5 ( T_{CY} + 20^* )</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>With Prescaler</td>
<td>10*</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td>41</td>
<td>T₁₀L</td>
<td>T₀CKI Low Pulse Width</td>
<td>No Prescaler</td>
<td>0.5 ( T_{CY} + 20^* )</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>With Prescaler</td>
<td>10*</td>
<td>—</td>
<td>—</td>
<td>ns</td>
</tr>
<tr>
<td>42</td>
<td>T₁₀P</td>
<td>T₀CKI Period</td>
<td>20 or ( \frac{T_{CY} + 40^*}{N} )</td>
<td>—</td>
<td>—</td>
<td>ns</td>
<td>Whichever is greater. ( N = ) Prescale Value (1, 2, 4, ..., 256)</td>
</tr>
</tbody>
</table>

* 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.
### TABLE 12-6: THERMAL CONSIDERATIONS

**Standard Operating Conditions (unless otherwise specified)**

<table>
<thead>
<tr>
<th>Param. No.</th>
<th>Sym.</th>
<th>Characteristic</th>
<th>Typ.</th>
<th>Units</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>TH01</td>
<td>( \theta_JA )</td>
<td>Thermal Resistance Junction to Ambient</td>
<td>60</td>
<td>°C/W</td>
<td>6-pin SOT-23 package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>80</td>
<td>°C/W</td>
<td>8-pin PDIP package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>90</td>
<td>°C/W</td>
<td>8-pin DFN package</td>
</tr>
<tr>
<td>TH02</td>
<td>( \theta_JC )</td>
<td>Thermal Resistance Junction to Case</td>
<td>31.4</td>
<td>°C/W</td>
<td>6-pin SOT-23 package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>24</td>
<td>°C/W</td>
<td>8-pin PDIP package</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>24</td>
<td>°C/W</td>
<td>8-pin DFN package</td>
</tr>
<tr>
<td>TH03</td>
<td>( T_{JMAX} )</td>
<td>Maximum Junction Temperature</td>
<td>150</td>
<td>°C</td>
<td></td>
</tr>
<tr>
<td>TH04</td>
<td>PD</td>
<td>Power Dissipation</td>
<td>—</td>
<td>W</td>
<td>( PD = P_{INTERNAL} + P_{I/O} )</td>
</tr>
<tr>
<td>TH05</td>
<td>P_{INTERNAL}</td>
<td>Internal Power Dissipation</td>
<td>—</td>
<td>W</td>
<td>( P_{INTERNAL} = IDD \times VDD^{(1)} )</td>
</tr>
<tr>
<td>TH06</td>
<td>P_{I/O}</td>
<td>I/O Power Dissipation</td>
<td>—</td>
<td>W</td>
<td>( P_{I/O} = \sum (IOL \times VOL) + \sum (IOH \times (VDD - VOH)) )</td>
</tr>
<tr>
<td>TH07</td>
<td>PDER</td>
<td>Derated Power</td>
<td>—</td>
<td>W</td>
<td>( PDER = PDMAX \times (TJ - TA)/\theta_JA^{(2)} )</td>
</tr>
</tbody>
</table>

**Note 1:** IDD is current to run the chip alone without driving any load on the output pins.

**2:** \( TA = \) Ambient Temperature; \( TJ = \) Junction Temperature.
13.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

The graphs and tables provided in this section are for design guidance and are not tested.

In some graphs or tables, the data presented are outside specified operating range (i.e., outside specified VDD range). This is for information only and devices are ensured to operate properly only within the specified range.

**Note:** The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

“Typical” represents the mean of the distribution at 25°C. “MAXIMUM”, “Max.”, “MINIMUM” or “Min.” represents (mean + 3σ) or (mean - 3σ) respectively, where σ is a standard deviation, over each temperature range.

**FIGURE 13-1: İDD VS. VDD OVER FOSC**
FIGURE 13-2: TYPICAL IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst-Case Temp) + 3σ
(-40°C to 125°C)

FIGURE 13-3: MAXIMUM IPD vs. VDD (SLEEP MODE, ALL PERIPHERALS DISABLED)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst-Case Temp) + 3σ
(-40°C to 125°C)
FIGURE 13-4: COMPARATOR IPD vs. VDD (COMPARATOR ENABLED)

Typical: Statistical Mean @25°C
      Maximum: Mean (Worst-Case Temp) + 3σ
              (-40°C to 125°C)

FIGURE 13-5: TYPICAL WDT IPD vs. VDD

Typical: Statistical Mean @25°C
      Maximum: Mean (Worst-Case Temp) + 3σ
              (-40°C to 125°C)
FIGURE 13-6: MAXIMUM WDT IPD vs. VDD OVER TEMPERATURE

Typical: Statistical Mean @25°C
Maximum: Mean (Worst-Case Temp) + 3σ
(-40°C to 125°C)

Max. 125°C

FIGURE 13-7: WDT TIME-OUT vs. VDD OVER TEMPERATURE (NO PRESCALER)

Typical: Statistical Mean @25°C
Maximum: Mean (Worst-Case Temp) + 3σ
(-40°C to 125°C)

Max. 125°C

Max. 85°C

Typical. 25°C

Min. -40°C
FIGURE 13-8: \(V_{OL} \) vs. \(I_{OL} \) OVER TEMPERATURE (\(V_{DD} = 3.0V\))

- Typical: Statistical Mean @25°C
- Maximum: Mean (Worst-Case Temp) + 3\(\sigma\) (-40°C to 125°C)

Max. 125°C
Max. 85°C
Typical 25°C
Min. -40°C

FIGURE 13-9: \(V_{OL} \) vs. \(I_{OL} \) OVER TEMPERATURE (\(V_{DD} = 5.0V\))

- Typical: Statistical Mean @25°C
- Maximum: Mean (Worst-Case Temp) + 3\(\sigma\) (-40°C to 125°C)

Max. 125°C
Max. 85°C
Typ. 25°C
Min. -40°C
**FIGURE 13-10:** $V_{OH}$ vs. $I_{OH}$ OVER TEMPERATURE ($V_{DD} = 3.0V$)

- **Typical:** Statistical Mean @25°C
- **Maximum:** Mean (Worst-Case Temp) + $3\sigma$ (-40°C to 125°C)

**FIGURE 13-11:** $V_{OH}$ vs. $I_{OH}$ OVER TEMPERATURE ($V_{DD} = 5.0V$)

- **Typical:** Statistical Mean @25°C
- **Maximum:** Mean (Worst-Case Temp) + $3\sigma$ (-40°C to 125°C)
FIGURE 13-12: TTL INPUT THRESHOLD $V_{IN}$ vs. $V_{DD}$

Typical: Statistical Mean @25°C
Maximum: Mean (Worst-Case Temp) + 3σ
(-40°C to 125°C)

FIGURE 13-13: SCHMITT TRIGGER INPUT THRESHOLD $V_{IN}$ vs. $V_{DD}$

Typical: Statistical Mean @25°C
Maximum: Mean (Worst-Case Temp) + 3σ
(-40°C to 125°C)
FIGURE 13-14: INTOSC (INTERNAL OSCILLATOR) POWER-UP TIMES vs. VDD

![Graph showing INTOSC power-up times vs. VDD for different temperatures: Max. 125°C, Max. 85°C, Typical 25°C, and Max. -40°C.](image-url)
14.0 PACKAGING INFORMATION

14.1 Package Marking Information

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.

*Standard PIC® device marking consists of Microchip part number, year code, week code, and traceability code. For PIC device 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.*
Package Marking Information (Continued)

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  
\( ^e_3 \) Pb-free JEDEC® designator for Matte Tin (Sn)  
* This package is Pb-free. The Pb-free JEDEC designator (\( ^e_3 \)) 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.

* Standard PIC® device marking consists of Microchip part number, year code, week code, and traceability code. For PIC device 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.
### TABLE 14-1: 8-LEAD 2x3 DFN (MC) PACKAGE TOP MARKING

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Marking</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC10F200-I/MC</td>
<td>BA0</td>
</tr>
<tr>
<td>PIC10F200-E/MC</td>
<td>BB0</td>
</tr>
<tr>
<td>PIC10F202-I/MC</td>
<td>BC0</td>
</tr>
<tr>
<td>PIC10F202-E/MC</td>
<td>BD0</td>
</tr>
<tr>
<td>PIC10F204-I/MC</td>
<td>BE0</td>
</tr>
<tr>
<td>PIC10F204-E/MC</td>
<td>BF0</td>
</tr>
<tr>
<td>PIC10F206-I/MC</td>
<td>BG0</td>
</tr>
<tr>
<td>PIC10F206-E/MC</td>
<td>BH0</td>
</tr>
</tbody>
</table>

### TABLE 14-2: 6-LEAD SOT-23 (OT) PACKAGE TOP MARKING

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Marking</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC10F200-I/OT</td>
<td>00NN</td>
</tr>
<tr>
<td>PIC10F200-E/OT</td>
<td>00NN</td>
</tr>
<tr>
<td>PIC10F202-I/OT</td>
<td>02NN</td>
</tr>
<tr>
<td>PIC10F202-E/OT</td>
<td>02NN</td>
</tr>
<tr>
<td>PIC10F204-I/OT</td>
<td>04NN</td>
</tr>
<tr>
<td>PIC10F204-E/OT</td>
<td>04NN</td>
</tr>
<tr>
<td>PIC10F206-I/OT</td>
<td>06NN</td>
</tr>
<tr>
<td>PIC10F206-E/OT</td>
<td>06NN</td>
</tr>
</tbody>
</table>

**Note:** NN represents the alphanumeric traceability code.
### 14.2 Package Details

The following sections give the technical details of the packages.

---

**6-Lead Plastic Small Outline Transistor (OT) [SOT-23]**

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

---

#### Dimensions

![Image of SOT-23 package]

**Dimensions in MILLIMETERS**

<table>
<thead>
<tr>
<th>Units</th>
<th>MILLIMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension Limits</td>
<td>MIN</td>
</tr>
<tr>
<td>Number of Pins</td>
<td>N</td>
</tr>
<tr>
<td>Pitch</td>
<td>e</td>
</tr>
<tr>
<td>Outside Lead Pitch</td>
<td>e1</td>
</tr>
<tr>
<td>Overall Height</td>
<td>A</td>
</tr>
<tr>
<td>Molded Package Thickness</td>
<td>A2</td>
</tr>
<tr>
<td>Standoff</td>
<td>A1</td>
</tr>
<tr>
<td>Overall Width</td>
<td>E</td>
</tr>
<tr>
<td>Molded Package Width</td>
<td>E1</td>
</tr>
<tr>
<td>Overall Length</td>
<td>D</td>
</tr>
<tr>
<td>Foot Length</td>
<td>L</td>
</tr>
<tr>
<td>Footprint</td>
<td>L1</td>
</tr>
<tr>
<td>Foot Angle</td>
<td>φ</td>
</tr>
<tr>
<td>Lead Thickness</td>
<td>c</td>
</tr>
<tr>
<td>Lead Width</td>
<td>b</td>
</tr>
</tbody>
</table>

**Notes:**

1. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.127 mm per side.
2. Dimensioning and tolerancing per ASME Y14.5M.
   - BSC: Basic Dimension. Theoretically exact value shown without tolerances.

---

Microchip Technology Drawing C04-028B
6-Lead Plastic Small Outline Transistor (OT) [SOT-23]

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

![Recommended Land Pattern Diagram]

<table>
<thead>
<tr>
<th>Units</th>
<th>MILLIMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dimension Limits</td>
</tr>
<tr>
<td>Contact Pitch</td>
<td>E</td>
</tr>
<tr>
<td>Contact Pad Spacing</td>
<td>C</td>
</tr>
<tr>
<td>Contact Pad Width (X6)</td>
<td>X</td>
</tr>
<tr>
<td>Contact Pad Length (X6)</td>
<td>Y</td>
</tr>
<tr>
<td>Distance Between Pads</td>
<td>G</td>
</tr>
<tr>
<td>Distance Between Pads</td>
<td>GX</td>
</tr>
<tr>
<td>Overall Width</td>
<td>Z</td>
</tr>
</tbody>
</table>

Notes:
1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2028A
8-Lead Plastic Dual In-Line (P) - 300 mil Body [PDIP]

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

<table>
<thead>
<tr>
<th>Dimension Limits</th>
<th>MIN</th>
<th>NOM</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Pins</td>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Pitch</td>
<td>e</td>
<td>.100 BSC</td>
<td></td>
</tr>
<tr>
<td>Top to Seating Plane</td>
<td>A</td>
<td>-</td>
<td>.210</td>
</tr>
<tr>
<td>Molded Package Thickness</td>
<td>A2</td>
<td>.115</td>
<td>.130</td>
</tr>
<tr>
<td>Base to Seating Plane</td>
<td>A1</td>
<td>.015</td>
<td>-</td>
</tr>
<tr>
<td>Shoulder to Shoulder Width</td>
<td>E</td>
<td>.290</td>
<td>.310</td>
</tr>
<tr>
<td>Molded Package Width</td>
<td>E1</td>
<td>.240</td>
<td>.250</td>
</tr>
<tr>
<td>Overall Length</td>
<td>D</td>
<td>.348</td>
<td>.365</td>
</tr>
<tr>
<td>Tip to Seating Plane</td>
<td>L</td>
<td>.115</td>
<td>.130</td>
</tr>
<tr>
<td>Lead Thickness</td>
<td>c</td>
<td>.008</td>
<td>.010</td>
</tr>
<tr>
<td>Upper Lead Width</td>
<td>b1</td>
<td>.040</td>
<td>.060</td>
</tr>
<tr>
<td>Lower Lead Width</td>
<td>b</td>
<td>.014</td>
<td>.018</td>
</tr>
<tr>
<td>Overall Row Spacing</td>
<td>§ eB</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within 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.
8-Lead Plastic Dual Flat, No Lead Package (MC) – 2x3x0.9 mm Body [DFN]

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

<table>
<thead>
<tr>
<th>Units</th>
<th>MIN</th>
<th>NOM</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Pins</td>
<td>N</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Pitch</td>
<td>e</td>
<td>0.50 BSC</td>
<td></td>
</tr>
<tr>
<td>Overall Height</td>
<td>A</td>
<td>0.80</td>
<td>0.90</td>
</tr>
<tr>
<td>Standoff</td>
<td>A1</td>
<td>0.00</td>
<td>0.02</td>
</tr>
<tr>
<td>Contact Thickness</td>
<td>A3</td>
<td>0.20 REF</td>
<td></td>
</tr>
<tr>
<td>Overall Length</td>
<td>D</td>
<td>2.00 BSC</td>
<td></td>
</tr>
<tr>
<td>Overall Width</td>
<td>E</td>
<td>3.00 BSC</td>
<td></td>
</tr>
<tr>
<td>Exposed Pad Length</td>
<td>D2</td>
<td>1.30</td>
<td>–</td>
</tr>
<tr>
<td>Exposed Pad Width</td>
<td>E2</td>
<td>1.50</td>
<td>0.25</td>
</tr>
<tr>
<td>Contact Width</td>
<td>b</td>
<td>0.20</td>
<td>0.25</td>
</tr>
<tr>
<td>Contact Length</td>
<td>L</td>
<td>0.30</td>
<td>0.40</td>
</tr>
<tr>
<td>Contact-to-Exposed Pad</td>
<td>K</td>
<td>0.20</td>
<td>–</td>
</tr>
</tbody>
</table>

**Notes:**
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package may have one or more exposed tie bars at ends.
3. Package is saw singulated.
4. Dimensioning and tolerancing per ASME Y14.5M.
   - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-123C
8-Lead Plastic Dual Flat, No Lead Package (MC) - 2x3x0.9mm Body [DFN]

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

---

**RECOMMENDED LAND PATTERN**

---

<table>
<thead>
<tr>
<th>Units</th>
<th>MILLIMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimension</td>
<td>Limits</td>
</tr>
<tr>
<td>Contact Pitch</td>
<td>E</td>
</tr>
<tr>
<td>Optional Center Pad Width</td>
<td>W2</td>
</tr>
<tr>
<td>Optional Center Pad Length</td>
<td>T2</td>
</tr>
<tr>
<td>Contact Pad Spacing</td>
<td>C1</td>
</tr>
<tr>
<td>Contact Pad Width (X8)</td>
<td>X1</td>
</tr>
<tr>
<td>Contact Pad Length (X8)</td>
<td>Y1</td>
</tr>
<tr>
<td>Distance Between Pads</td>
<td>G</td>
</tr>
</tbody>
</table>

Notes:
1. Dimensioning and tolerancing per ASME Y14.5M
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
APPENDIX A: REVISION HISTORY

Revision C (August 2006)
Added 8-Pin DFN Pin Diagram; Revised Table 1-1; Reformatted all Registers; Revised Section 4.8 and added note; Section 5.3 (changed Figure reference to Figure 5-1); Tables 6-1 and 7-1 (removed shading from TRISGPIO (I/O Control Register); Sections 8.1-8.4 (changed Table reference to Table 12-2); Section 14.1 Revised and replaced Package Marking Information and drawings, Added Tables 14-1 & 14-2, Added DFN Package drawing.

Revision D (April 2007)
Revised section 12.1, 12.2, 12.3, Table 1-1, 12-1, 12-3, 12-4. Added Section 13.0. Replaced Package Drawings (Rev. AP); Removed instances of PICmicro® and replaced it with PIC®.

Revision E (October 2013)
Revised Figure 8-1 (deleted OSCCAL); Revised Packaging Legend.

Revision F (September 2014)
Added Table 12-6 (Thermal Considerations); Updated Register 4-1, Register 9-1 and Chapter 14 (Packaging Information); Other minor corrections.
THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user’s guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip’s customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under “Support”, click on “Customer Change Notification” and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support
PRODUCT IDENTIFICATION SYSTEM

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

<table>
<thead>
<tr>
<th>PART NO.</th>
<th>Device</th>
<th>Tape and Reel Option</th>
<th>Temperature Range</th>
<th>Package</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PIC10F200</td>
<td>Blank = Standard packaging (tube or tray)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PIC10F202</td>
<td>T = Tape and Reel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PIC10F204</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PIC10F206</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PIC10F200T (Tape &amp; Reel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PIC10F202T (Tape &amp; Reel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PIC10F204T (Tape &amp; Reel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PIC10F206T (Tape &amp; Reel)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tape and Reel Option</th>
<th>Examples:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank = Standard packaging (tube or tray)</td>
<td>a) PIC10F200T - E/OT Tape and Reel Extended temperature SOT-23 package (Pb-free)</td>
</tr>
<tr>
<td>T = Tape and Reel</td>
<td>b) PIC10F200 - I/P Industrial temperature, PDIP package (Pb-free)</td>
</tr>
<tr>
<td></td>
<td>c) PIC10F204 - I/MC Industrial temperature DFN package (Pb-free)</td>
</tr>
</tbody>
</table>

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks
The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELoo, KEELoo logo, Kleer, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC™ logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.
The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.
Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KleerNet, KleerNet logo, MiWi, MPASM, MFP, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscent Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.
SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.
Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.
GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.
All other trademarks mentioned herein are property of their respective companies.
© 2004-2014, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.
ISBN: 978-1-63276-597-0

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KeelLoc® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.
Worldwide Sales and Service

AMERICAS
Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-3000
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC
Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-8733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9600

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC
India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or 82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE
Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim
Tel: 49-7231-424750

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

03/25/14