PIC10F220/222
Data Sheet

6-Pin, 8-Bit Flash Microcontrollers

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.

Trademarks

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® 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.

PIC10F220/222

6-Pin, 8-Bit Flash Microcontrollers

Device Included In This Data Sheet:

- PIC10F220
- PIC10F222

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
- 8 special function hardware registers
- Operating speed:
  - 500 ns instruction cycle with 8 MHz internal clock
  - 1 µs instruction cycle with 4 MHz internal clock

Special Microcontroller Features:

- 4 or 8 MHz precision internal oscillator:
  - Factory calibrated to ±1%
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit Debugging (ICD) support
- Power-on Reset (POR)
- Short Device Reset Timer, DRT (1.125 ms typical)
- Watchdog Timer (WDT) with dedicated on-chip RC oscillator for reliable operation
- Programmable code protection
- Multiplexed $\overline{\text{MCLR}}$ 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:
  - < 170 µA @ 2V, 4 MHz
- 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

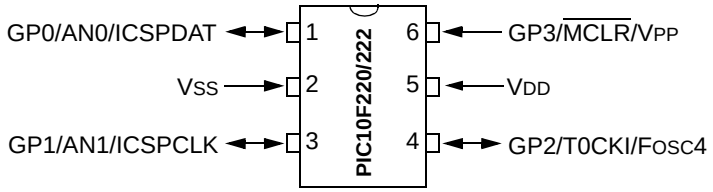
Peripheral Features:

- 4 I/O pins:
  - 3 I/O pins with individual direction control
  - 1 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
- Analog-to-Digital (A/D) Converter:
  - 8-bit resolution
  - 2 external input channels
  - 1 internal input channel dedicated

Device	Program Memory	Data Memory	I/O	Timers 8-bit	8-Bit A/D (ch)
	Flash (words)	SRAM (bytes)			
PIC10F220	256	16	4	1	2
PIC10F222	512	23	4	1	2

Pin Diagrams

6-Lead SOT-23



8-Lead DIP

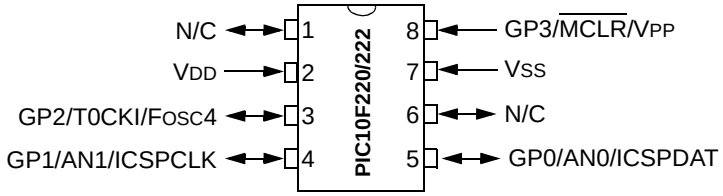


Table of Contents

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

http://www.microchip.com

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

Errata

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

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

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

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

Customer Notification System

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

NOTES:

1.0 GENERAL DESCRIPTION

The PIC10F220/222 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 PIC10F220/222 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 PIC10F220/222 products are equipped with special features that reduce system cost and power requirements. The Power-on Reset (POR) and Device Reset Timer (DRT) eliminates the need for the external Reset circuitry. INTOSC 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 PIC10F220/222 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 PIC10F220/222 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.

1.1 Applications

The PIC10F220/222 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 PIC10F220/222 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).

TABLE 1-1: PIC10F220/222 DEVICES(1), (2)

		PIC10F220	PIC10F222
Clock	Maximum Frequency of Operation (MHz)	8	8
Memory	Flash Program Memory	256	512
	Data Memory (bytes)	16	23
Peripherals	Timer Module(s)	TMR0	TMR0
	Wake-up from Sleep on pin change	Yes	Yes
	Analog inputs	2	2
Features	I/O Pins	3	3
	Input Only Pins	1	1
	Internal Pull-ups	Yes	Yes
	In-Circuit Serial Programming™	Yes	Yes
	Number of instructions	33	33
	Packages	6-pin SOT-23, 8-pin DIP	6-pin SOT-23, 8-pin DIP

Note 1: The PIC10F220/222 devices have Power-on Reset, selectable Watchdog Timer, selectable code-protect, high I/O current capability and precision internal oscillator.

2: The PIC10F220/222 devices use serial programming with data pin GP0 and clock pin GP1.

NOTES:

2.0 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 PIC10F220/222 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^{SM} (SQTP^{SM}) 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.

NOTES:

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC10F220/222 devices can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC10F220/222 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 µs @ 4 MHz or 500 ns @ 8 MHz) except for program branches.

The table below lists program memory (Flash) and data memory (RAM) for the PIC10F220/222 devices.

Device	Memory	
	Program	Data
PIC10F220	256 x 12	16 x 8
PIC10F222	512 x 12	23 x 8

The PIC10F220/222 devices can directly or indirectly address its register files and data memory. All Special Function Registers (SFR), including the PC, are mapped in the data memory. The PIC10F220/222 devices have a highly orthogonal (symmetrical) instruction set that makes it possible to carry out any operation, on any register, using any addressing mode. This symmetrical nature and lack of “special optimal situations” make programming with the PIC10F220/222 devices simple, yet efficient. In addition, the learning curve is reduced significantly.

The PIC10F220/222 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 with the corresponding device pins described in Table 3-1.

FIGURE 3-1: BLOCK DIAGRAM

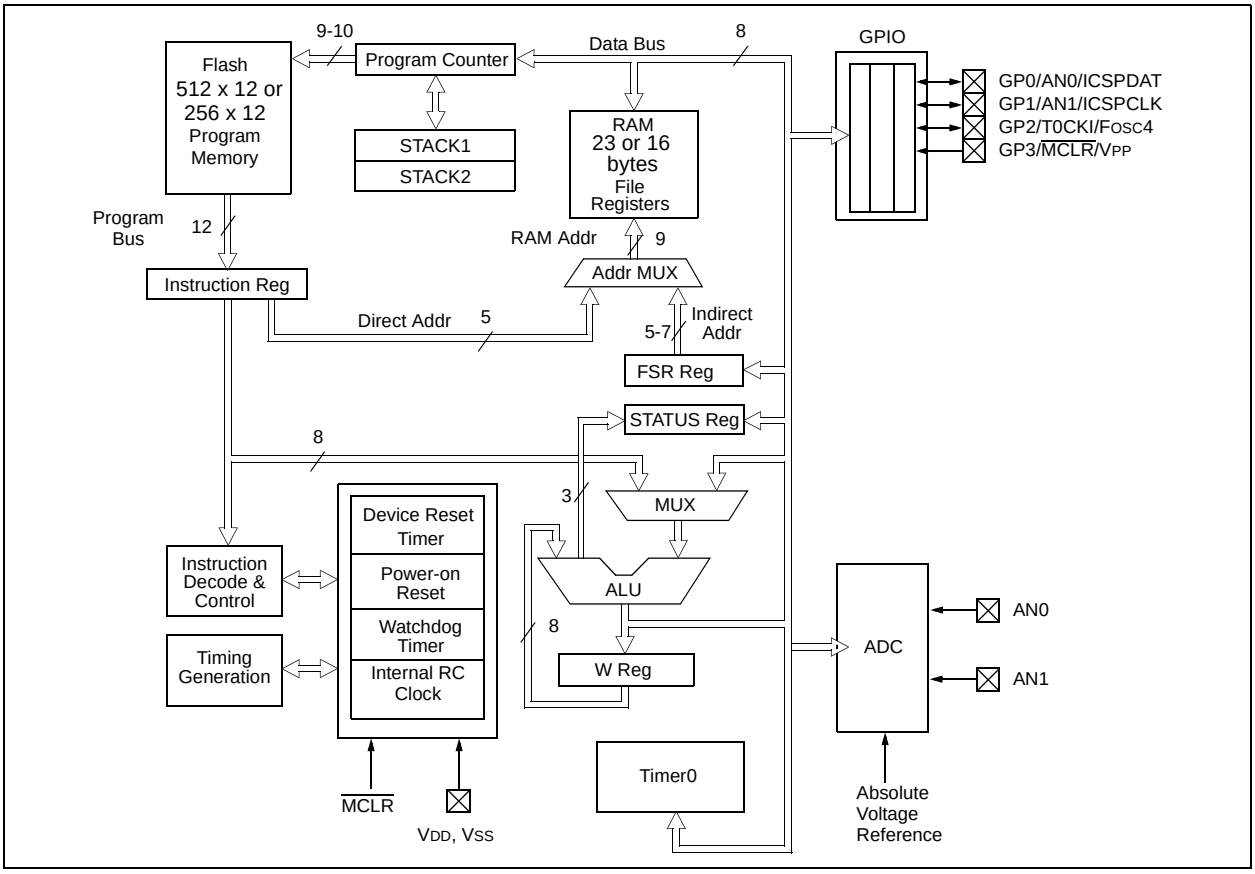


TABLE 3-1: PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
GP0/AN0/ICSPDAT	GP0	TTL	CMOS	Bidirectional I/O pin. Can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.
	AN0	AN	—	Analog Input
	ICSPDAT	ST	CMOS	In-Circuit programming data
GP1/AN1/ICSPCLK	GP1	TTL	CMOS	Bidirectional I/O pin. Can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.
	AN1	AN	—	Analog Input
	ICSPCLK	ST	—	In-Circuit programming clock
GP2/T0CKI/FOSC4	GP2	TTL	CMOS	Bidirectional I/O pin
	T0CKI	ST	—	Clock input to TMR0
	FOSC4	—	CMOS	Oscillator/4 output
GP3/$\overline{\text{MCLR}}$/VPP	GP3	TTL	—	Input pin. Can be software programmed for internal weak pull-up and wake-up from Sleep on pin change.
	$\overline{\text{MCLR}}$	ST	—	Master Clear (Reset). When configured as $\overline{\text{MCLR}}$, this pin is an active-low Reset to the device. Voltage on $\overline{\text{MCLR}}$/VPP must not exceed VDD during normal device operation or the device will enter Programming mode. Weak pull-up always on if configured as $\overline{\text{MCLR}}$.
	VPP	HV	—	Programming voltage input
VDD	VDD	P	—	Positive supply for logic and I/O pins
VSS	VSS	P	—	Ground reference for logic and I/O pins

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 (IR) in Q4. It is decoded and executed during Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2 and Example 3-1.

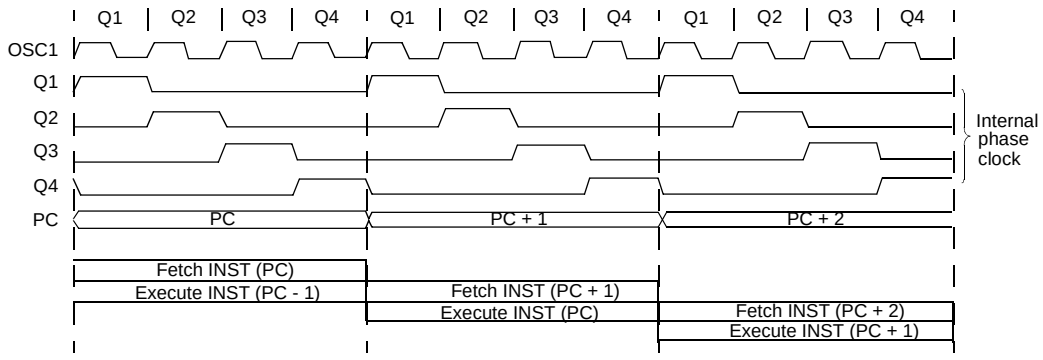
3.2 Instruction Flow/Pipelining

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 takes 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).

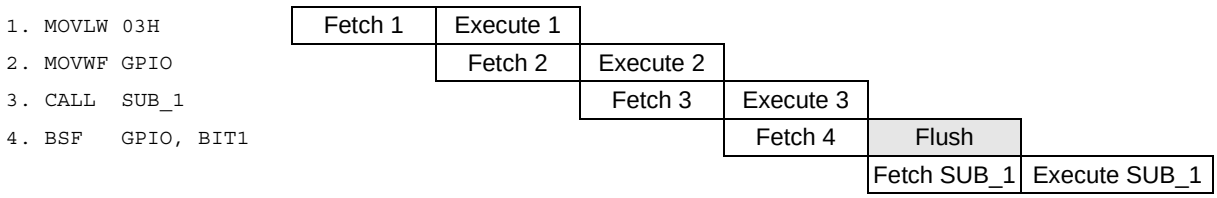
A fetch cycle begins with the PC incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the Instruction Register 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 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW



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.

NOTES:

4.0 MEMORY ORGANIZATION

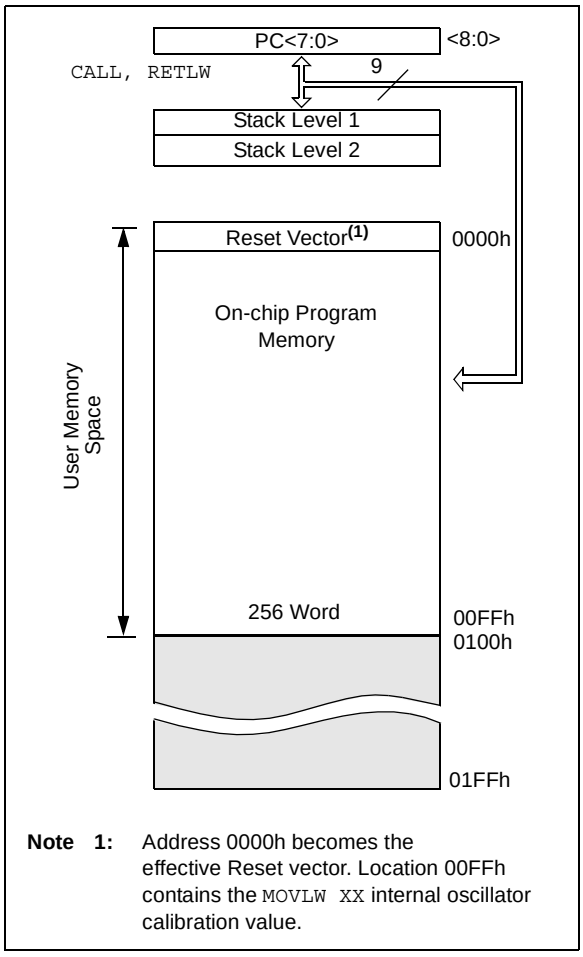
The PIC10F220/222 memories are organized into program memory and data memory. Data memory banks are accessed using the File Select Register (FSR).

4.1 Program Memory Organization for the PIC10F220

The PIC10F220 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 PIC10F220 are physically implemented (see Figure 4-1). Accessing a location above these boundaries will cause a wrap-around within the first 256 x 12 space (PIC10F220). The effective Reset vector is at 0000h, (see Figure 4-1). Location 00FFh (PIC10F220) contains the internal clock oscillator calibration value. This value should never be overwritten.

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC10F220



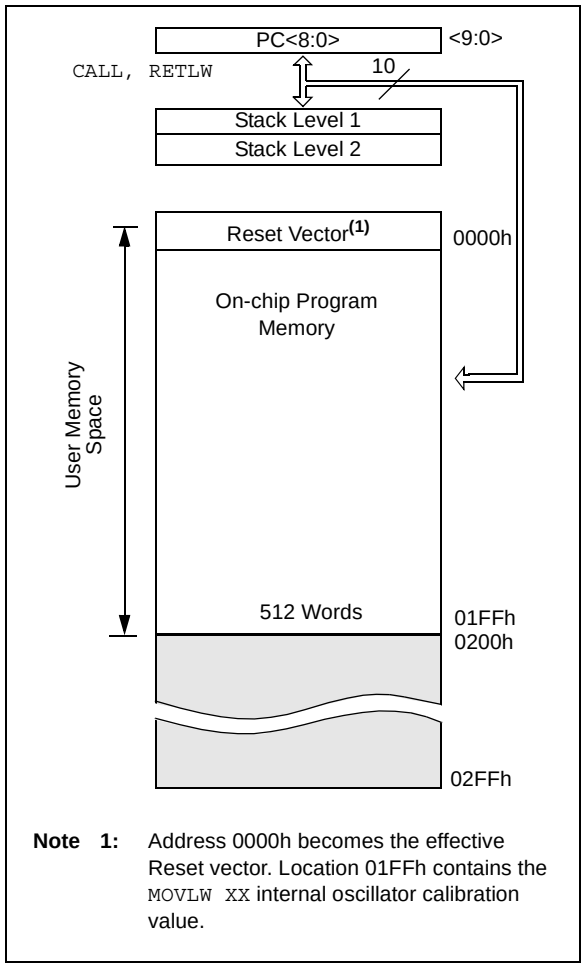
Note 1: Address 0000h becomes the effective Reset vector. Location 00FFh contains the `MOVLW XX` internal oscillator calibration value.

4.2 Program Memory Organization for the PIC10F222

The PIC10F222 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 Mem-High are physically implemented (see Figure 4-2). Accessing a location above these boundaries will cause a wrap-around within the first 512 x 12 space (PIC10F222). The effective Reset vector is at 0000h, (see Figure 4-2). Location 01FFh (PIC10F222) contains the internal clock oscillator calibration value. This value should never be overwritten.

FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC10F222



Note 1: Address 0000h becomes the effective Reset vector. Location 01FFh contains the `MOVLW XX` internal oscillator calibration value.

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.

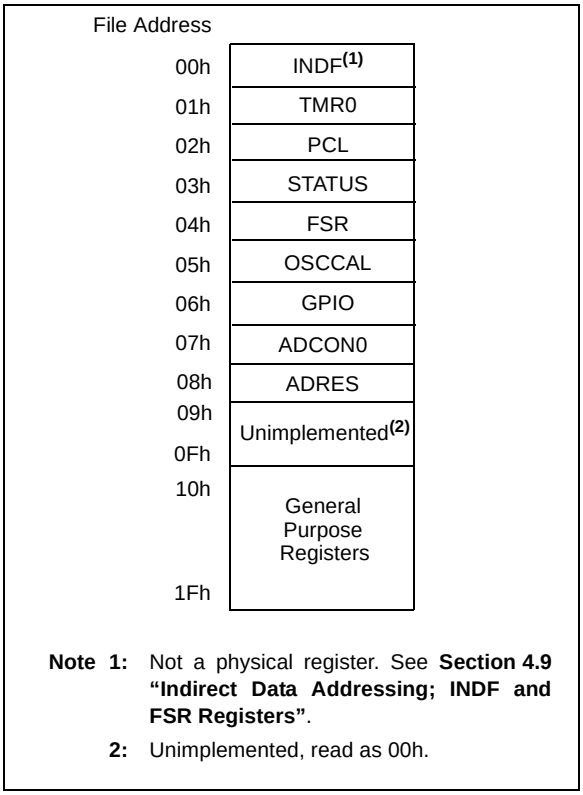
For the PIC10F220, the register file is composed of 9 Special Function Registers and 16 General Purpose Registers (Figure 4-3, Figure 4-4).

For the PIC10F222, the register file is composed of 9 Special Function Registers and 23 General Purpose Registers (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"**.

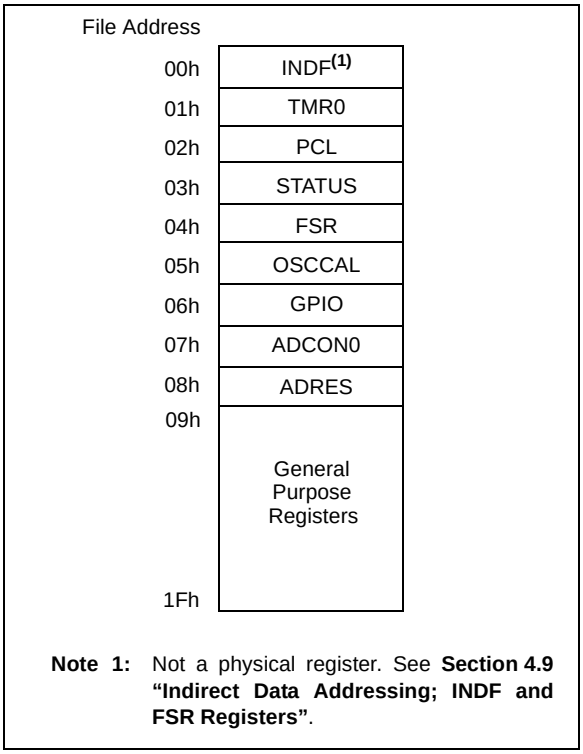
FIGURE 4-3: PIC10F220 REGISTER FILE MAP



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

2: Unimplemented, read as 00h.

FIGURE 4-4: PIC10F222 REGISTER FILE MAP



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

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

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset(2)	Page #
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	20
01h	TMR0	8-Bit Real-Time Clock/Counter								xxxx xxxx	25
02h	PCL(1)	Low Order 8 Bits of PC								1111 1111	19
03h	STATUS	GPWUF	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0--1 1xxx(3)	15
04h	FSR	Indirect Data Memory Address Pointer								111x xxxx	20
05h	OSCCAL	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	FOSC4	1111 1110	18
06h	GPIO	—	—	—	—	GP3	GP2	GP1	GP0	---- xxxx	21
07h	ADCON0	ANS1	ANS0	—	—	CHS1	CHS0	GO/$\overline{DONE}$	ADON	11-- 1100	30
08h	ADRES	Result of Analog-to-Digital Conversion								xxxx xxxx	31
N/A	TRISGPIO	—	—	—	—	I/O Control Register				---- 1111	23
N/A	OPTION	$\overline{GPWU}$	$\overline{GPPU}$	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	17

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.

2: Other (non Power-up) Resets include external Reset through $\overline{MCLR}$, Watchdog Timer and wake-up on pin change Reset.

3: See Table 8-1 for other Reset specific values.

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 $\overline{TO}$ and $\overline{PD}$ 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 u1uu (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 Instruction Set Summary.

REGISTER 4-1: STATUS REGISTER: (ADDRESS: 03h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
GPWUF	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

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

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

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

bit 4 **$\overline{TO}$:** Time-out bit
1 = After power-up, CLRWDT instruction or SLEEP instruction
0 = A WDT time-out occurred

bit 3 **$\overline{PD}$:** Power-down bit
1 = After power-up or by the CLRWDT 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/$\overline{borrow}$ bit (for ADDWF and SUBWF instructions)
ADDWF:
1 = A carry to the 4th low-order bit of the result occurred
0 = A carry to 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/$\overline{borrow}$ bit (for ADDWF, SUBWF and RRF, RLF instructions)

ADDWF:	SUBWF:	RRF or RLF:
1 = A carry occurred	1 = A borrow did not occur	Load bit with LSb or MSb, respectively
0 = A carry did not occur	0 = A borrow occurred	

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.

The OPTION register is not memory mapped and is therefore only addressable 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 $\overline{GPPU}$ and $\overline{GPWU}$).

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

REGISTER 4-2: OPTION REGISTER: (PIC10F22X)

W-1	W-1	W-1	W-1	W-1	W-1	W-1	W-1
$\overline{GPWU}$	$\overline{GPPU}$	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

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

bit 7 **$\overline{GPWU}$:** Enable Wake-up On Pin Change bit (GP0, GP1, GP3)
1 = Disabled
0 = Enabled

bit 6 **$\overline{GPPU}$:** 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

Bit Value	Timer0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

4.6 OSCCAL Register

The Oscillator Calibration (OSCCAL) register is used to calibrate the internal precision 4/8 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 8.2.2 "Internal 4/ 8 MHz Oscillator"**.

REGISTER 4-3: OSCCAL REGISTER: (ADDRESS: 05h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0
CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	FOSC4
bit 7							bit 0

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

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

0111111 = Maximum frequency
•
•
•
0000001
0000000 = Center frequency
1111111
•
•
•
1000000 = Minimum frequency

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

1 = INTOSC/4 output onto GP2
0 = GP2/T0CKI applied to GP2

Note 1: Overrides GP2/T0CKI 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.

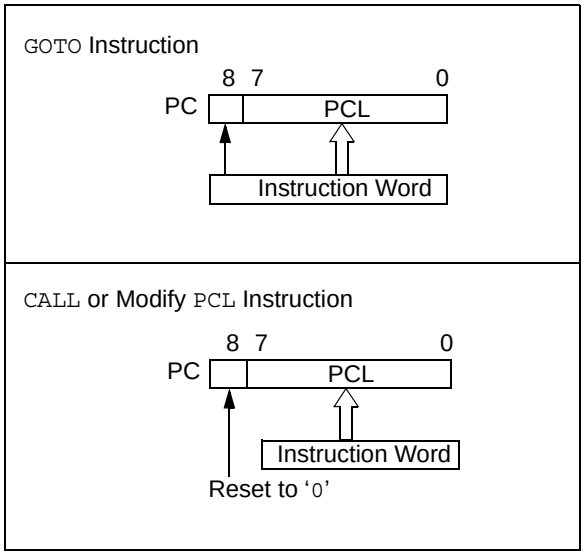
For a GOTO instruction, bits 8:0 of the PC are provided by the GOTO instruction word. The PC Latch (PCL) is mapped to PC<7:0>.

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 MOVLW XX, the PC will roll over to location 0000h and begin executing user code.

4.8 Stack

The PIC10F220 device has a 2-deep, 8-bit wide hardware PUSH/POP stack.

The PIC10F222 device has 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 CALL's 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 RETLW's are executed, the stack will be filled with the address previously stored in 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 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 instructions 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.

4.9.1 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

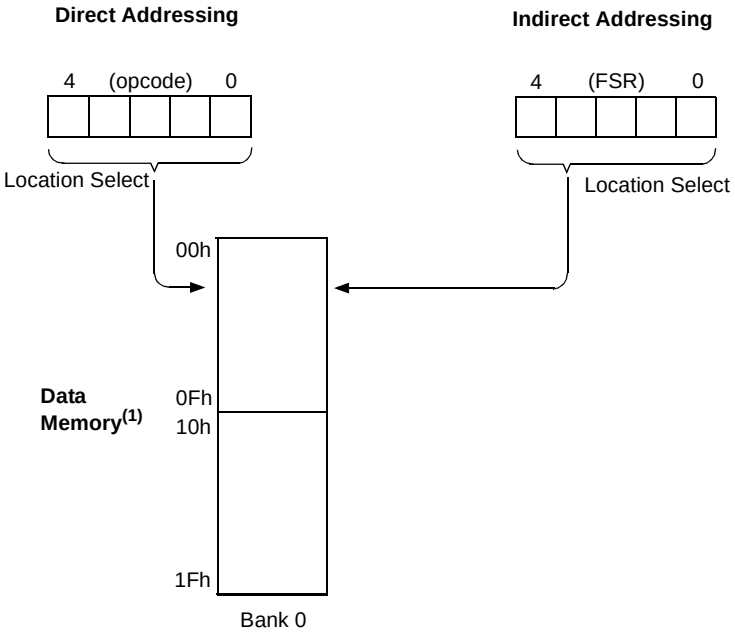
```
        MOVLW   0x10    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF
                        ;register
        INCF    FSR,F   ;inc pointer
        BTFSC   FSR,4   ;all done?
        GOTO    NEXT    ;NO, clear next
CONTINUE
            :           ;YES,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

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

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

Note:	Do not use banking. FSR <7:5> are unimplemented and read as '1's.

FIGURE 4-6: DIRECT/INDIRECT ADDRESSING



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 individually pin selectable. If GP3/$\overline{MCLR}$ is configured as $\overline{MCLR}$, a weak pull-up can be enabled via the Configuration Word. Configuring GP3 as $\overline{MCLR}$ disables the wake-up on change function for this pin.

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/FOSC4 pin, which may be controlled by various registers. See Table 5-1.

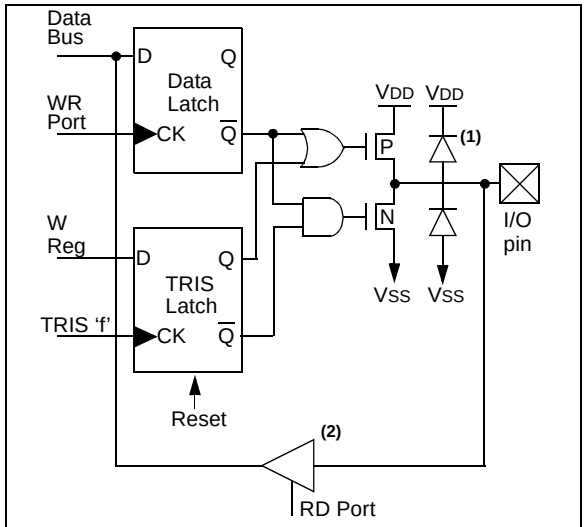
Note:	A read of the ports reads the pins, not the output data latches. That is, if an output driver on a pin is enabled and driven high, but the external system is holding it low, a read of the port will indicate that the pin is low.

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

5.3 I/O Interfacing

The equivalent circuit for an I/O port pin is shown in Figure 5-5. 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: EQUIVALENT CIRCUIT FOR A SINGLE I/O PIN



Note 1: I/O pins have protection diodes to VDD and VSS.

2: See Table 3-1 for buffer type.

TABLE 5-1: ORDER OF PRECEDENCE FOR PIN FUNCTIONS

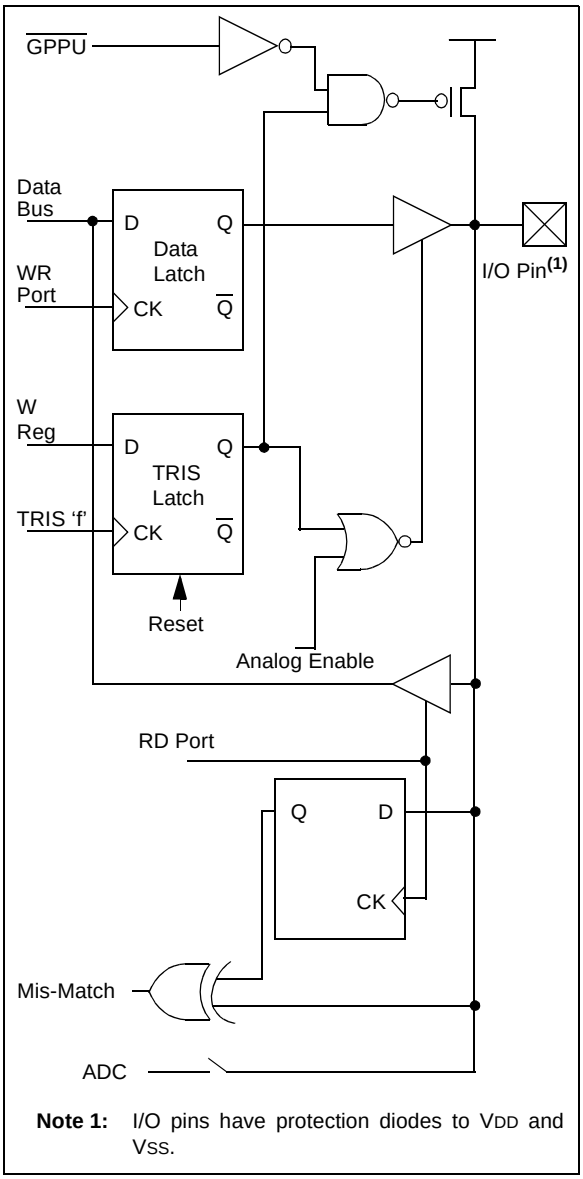
Priority	GP0	GP1	GP2	GP3
1	AN0	AN1	FOSC4	$\overline{MCLR}$
2	TRIS GPIO	TRIS GPIO	T0CKI	—
3	—	—	TRIS GPIO	—

TABLE 5-2: REQUIREMENTS TO MAKE PINS AVAILABLE IN DIGITAL MODE

Bit	GP0	GP1	GP2	GP3
FOSC4	—	—	0	—
T0CS	—	—	0	—
ANS1	—	0	—	—
ANS0	0	—	—	—
MCLRE	—	—	—	0

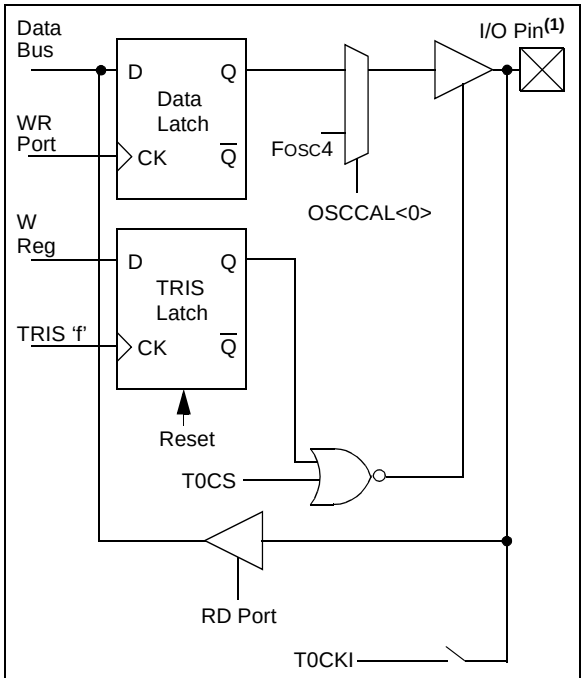
Legend: — = Condition of bit will have no effect on the setting of the pin to Digital mode.

FIGURE 5-2: BLOCK DIAGRAM OF GP0 AND GP1



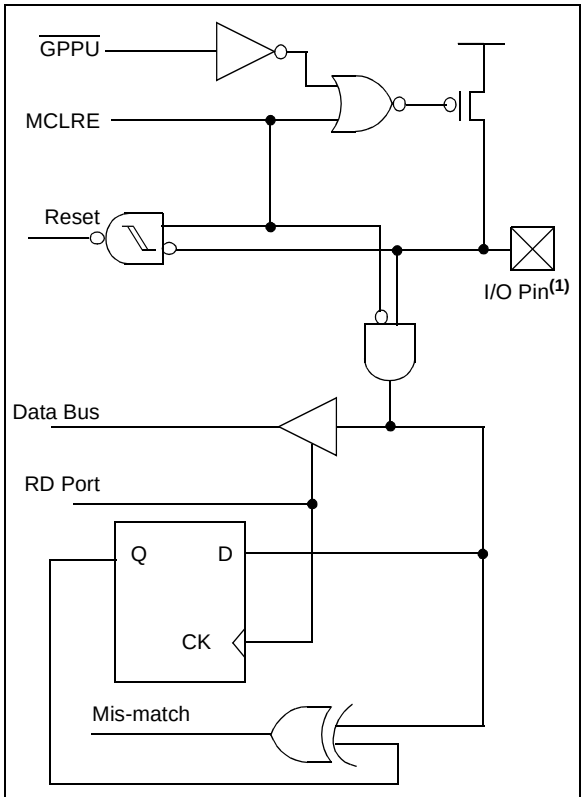
Note 1: I/O pins have protection diodes to VDD and VSS.

FIGURE 5-3: BLOCK DIAGRAM OF GP2



Note 1: I/O pins have protection diodes to VDD and VSS.

FIGURE 5-4: BLOCK DIAGRAM OF GP3



Note 1: GP3/MCLR pin has a protection diode to VSS only.

TABLE 5-3: SUMMARY OF PORT REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on All Other Resets
N/A	TRISGPIO	—	—	—	—	I/O Control Registers				---- 1111	---- 1111
N/A	OPTION	$\overline{\text{GPWU}}$	$\overline{\text{GPPU}}$	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
03h	STATUS	GPWUF	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	q00q quuu(1)
06h	GPIO	—	—	—	—	GP3	GP2	GP1	GP0	---- xxxx	---- uuuu

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

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

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 re-write 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.

EXAMPLE 5-1: I/O PORT READ-MODIFY-WRITE INSTRUCTIONS

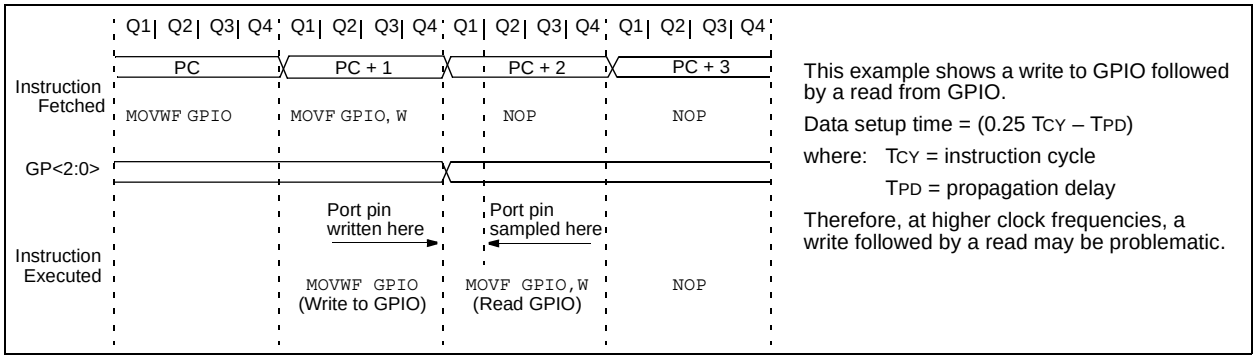
```
;Initial GPIO Settings
;GPIO<3:2> Inputs
;GPIO<1:0> Outputs
;
;                GPIO latch   GPIO pins
;                ----------   ----------
 BCF    GPIO,  1 ;---- pp01    ---- pp11
 BCF    GPIO,  0 ;---- pp10    ---- pp11
 MOVLW  007h;
 TRIS   GPIO     ;---- pp10    ---- pp11
;
```

Note: The user may have expected the pin values to be ---- pp00. The second 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 5-5). 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 5-5: SUCCESSIVE I/O OPERATION



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.

NOTES:

6.0 TMR0 MODULE AND TMR0 REGISTER

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.

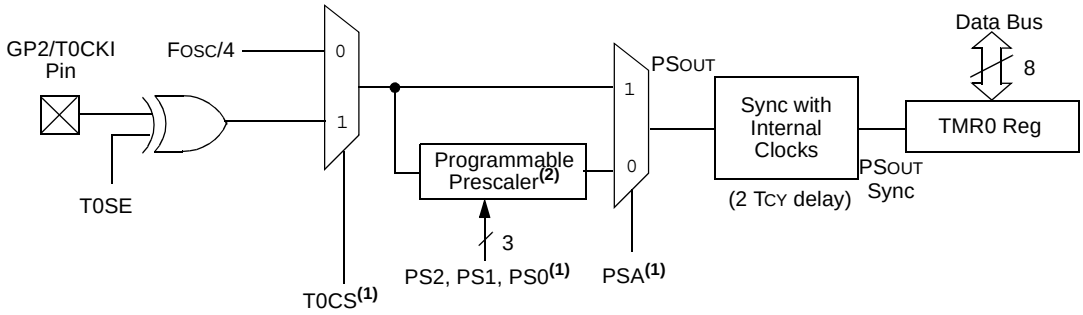
Timer mode is selected by clearing the T0CS bit (OPTION<5>). In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If TMR0 register is written, the increment is inhibited for the following two cycles (Figure 6-2 and Figure 6-3). The user can work around this by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

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.

FIGURE 6-1: TIMER0 BLOCK DIAGRAM



Note 1: Bits T0CS, T0SE, PSA, PS2, PS1 and PS0 are located in the OPTION register.

2: The prescaler is shared with the Watchdog Timer (Figure 6-5).

FIGURE 6-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALE

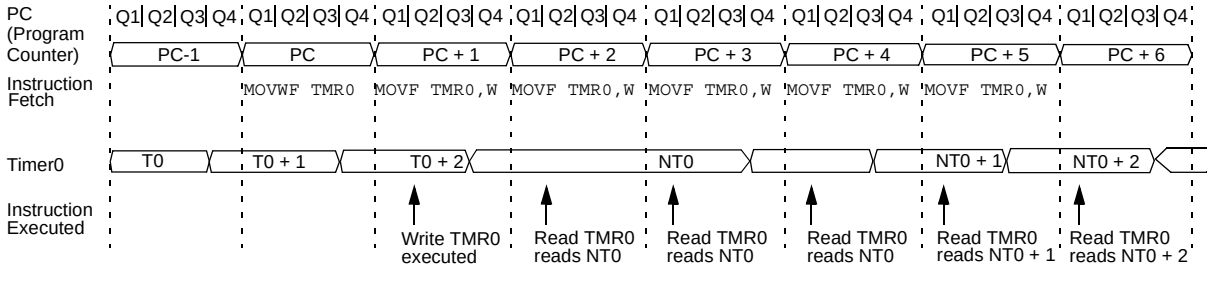


FIGURE 6-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALE 1:2

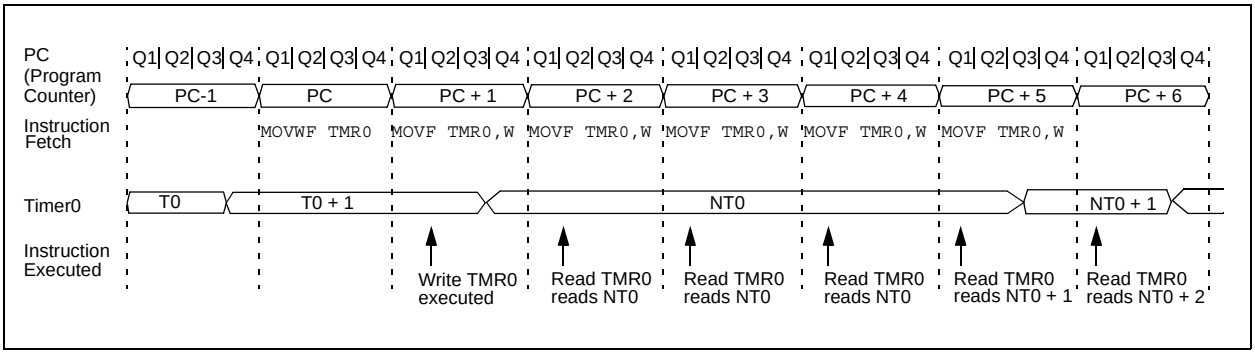


TABLE 6-1: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on All Other Resets
01h	TMR0	Timer0 – 8-Bit Real-Time Clock/Counter								xxxx xxxx	uuuu uuuu
N/A	OPTION	$\overline{\text{GPWU}}$	$\overline{\text{GPPU}}$	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
N/A	TRISGPIO(1)	—	—	—	—	I/O Control Register				---- 1111	---- 1111

Legend: Shaded cells not used by Timer0, – = unimplemented, x = unknown, u = unchanged.

Note 1: The TRIS of the T0CKI pin is overridden when T0CS = 1

6.1 Using Timer0 With An External Clock

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

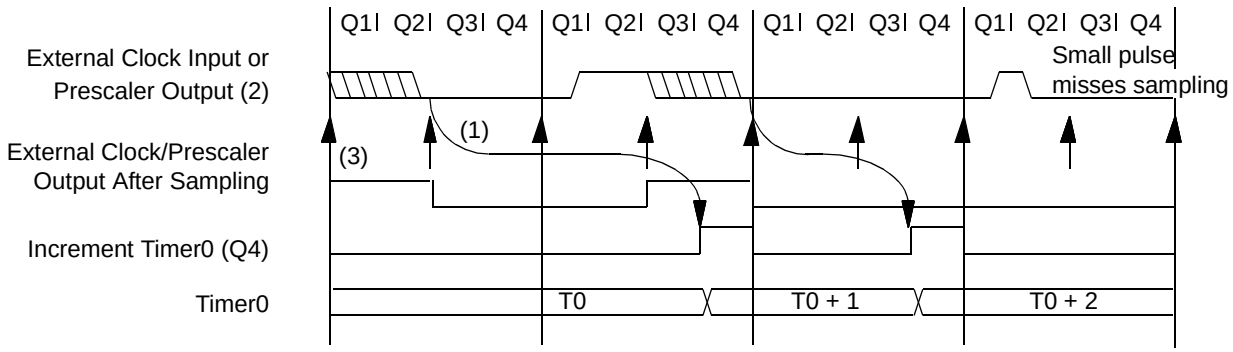
When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 6-4). Therefore, it is necessary for T0CKI to be high for at least 2TOSC (and a small RC delay of 2Tt0H) and low for at least 2TOSC (and a small RC delay of 2Tt0H). 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 to have a period of at least 4TOSC (and a small RC delay of 4Tt0H) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of Tt0H. 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



Note 1: Delay from clock input change to Timer0 increment is 3TOSC to 7TOSC. (Duration of Q = TOSC). Therefore, the error in measuring the interval between two edges on Timer0 input = ±4TOSC 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 8.6 "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)

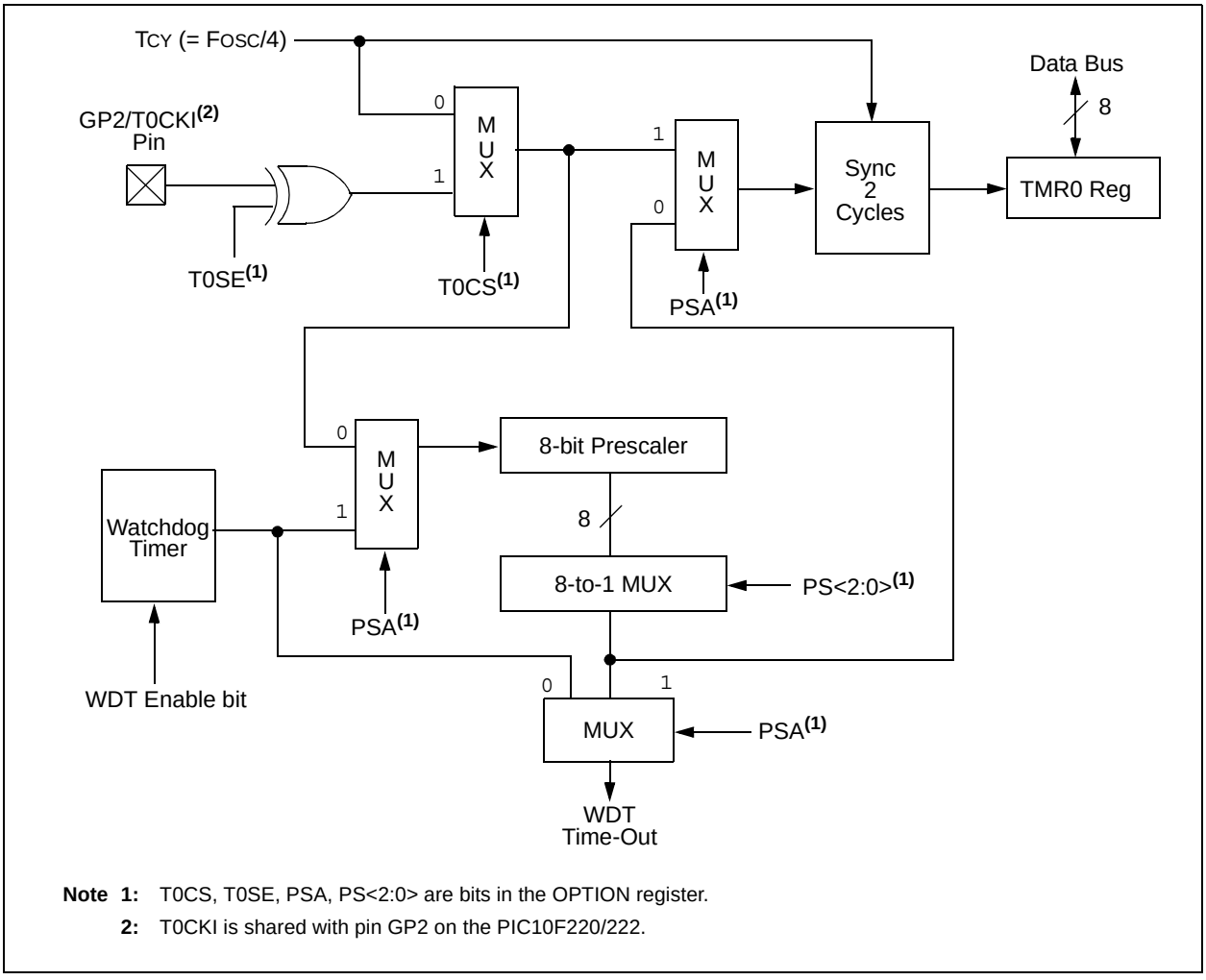
```
CLRWDT              ;Clear WDT
CLRF   TMR0         ;Clear TMR0 & Prescaler
MOVLW  '00xx1111'b  ;These 3 lines (5, 6, 7)
OPTION              ;are required only if
                    ;desired
CLRWDT              ;PS<2:0> are 000 or 001
MOVLW  '00xx1xxx'b  ;Set Postscaler to
OPTION              ;desired WDT rate
```

To change 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.

2: T0CKI is shared with pin GP2 on the PIC10F220/222.

7.0 ANALOG-TO-DIGITAL (A/D) CONVERTER

The A/D converter allows conversion of an analog signal into an 8-bit digital signal.

7.1 Clock Divisors

The A/D Converter has a single clock source setting, INTOSC/4. The A/D Converter requires 13 TAD periods to complete a conversion. The divisor values do not affect the number of TAD periods required to perform a conversion. The divisor values determine the length of the TAD period.

Note:	Due to the fixed clock divisor, a conversion will complete in 13 CPU instruction cycles.

7.2 Voltage Reference

Due to the nature of the design, there is no external voltage reference allowed for the A/D Converter. The A/D Converter reference voltage will always be VDD.

7.3 Analog Mode Selection

The ANS<1:0> bits are used to configure pins for analog input. Upon any Reset ANS<1:0> defaults to 11. This configures pins AN0 and AN1 as analog inputs. Pins configured as analog inputs are not available for digital output. Users should not change the ANS bits while a conversion is in process. ANS bits are active regardless of the condition of ADON.

7.4 A/D Converter Channel Selection

The CHS bits are used to select the analog channel to be sampled by the A/D Converter. The CHS bits should not be changed during a conversion. To acquire an analog signal, the CHS selection must match one of the pin(s) selected by the ANS bits. The Internal Absolute Voltage Reference can be selected regardless of the condition of the ANS bits. All channel selection information will be lost when the device enters Sleep.

Note:	The A/D Converter module consumes power when the ADON bit is set even when no channels are selected as analog inputs. For low-power applications, it is recommended that the ADON bit be cleared when the A/D Converter is not in use.

7.5 The GO/$\overline{\text{DONE}}$ bit

The GO/$\overline{\text{DONE}}$ bit is used to determine the status of a conversion, to start a conversion and to manually halt a conversion in process. Setting the GO/$\overline{\text{DONE}}$ bit starts a conversion. When the conversion is complete, the A/D Converter module clears the GO/$\overline{\text{DONE}}$ bit. A conversion can be terminated by manually clearing the GO/$\overline{\text{DONE}}$ bit while a conversion is in process. Manual termination of a conversion may result in a partially converted result in ADRES.

The GO/$\overline{\text{DONE}}$ bit is cleared when the device enters Sleep, stopping the current conversion. The A/D Converter does not have a dedicated oscillator, it runs off of the system clock.

The GO/$\overline{\text{DONE}}$ bit cannot be set when ADON is clear.

7.6 Sleep

This A/D Converter does not have a dedicated A/D Converter clock and therefore no conversion in Sleep is possible. If a conversion is underway and a Sleep command is executed, the GO/$\overline{\text{DONE}}$ and ADON bit will be cleared. This will stop any conversion in process and power-down the A/D Converter module to conserve power. Due to the nature of the conversion process, the ADRES may contain a partial conversion. At least 1 bit must have been converted prior to Sleep to have partial conversion data in ADRES. The CHS bits are reset to their default condition and CHS<1:0> = 11.

For accurate conversions, TAD must meet the following:

- 500 ns < TAD < 50 µs
- TAD = 1/(FOSC/divisor)

TABLE 7-1: EFFECTS OF SLEEP AND WAKE ON ADCON0

	ANS1	ANS0	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
Prior to Sleep	x	x	x	x	0	0
Prior to Sleep	x	x	x	x	1	1
Entering Sleep	Unchanged	Unchanged	1	1	0	0
Wake	1	1	1	1	0	0

7.7 Analog Conversion Result Register

The ADRES register contains the results of the last conversion. These results are present during the sampling period of the next analog conversion process. After the sampling period is over, ADRES is cleared (= 0). A 'leading one' is then right shifted into the ADRES to serve as an internal conversion complete bit. As each bit weight, starting with the MSb, is converted, the leading one is shifted right and the converted bit is stuffed into ADRES. After a total of 9 right shifts of the 'leading one' have taken place, the conversion is complete; the 'leading one' has been shifted out and the GO/DONE bit is cleared.

If the GO/DONE bit is cleared in software during a conversion, the conversion stops. The data in ADRES is the partial conversion result. This data is valid for the bit weights that have been converted. The position of the 'leading one' determines the number of bits that have been converted. The bits that were not converted before the GO/DONE was cleared are unrecoverable.

7.8 Internal Absolute Voltage Reference

The function of the Internal Absolute Voltage Reference is to provide a constant voltage for conversion across the devices VDD supply range. The A/D Converter is ratiometric with the conversion reference voltage being VDD. Converting a constant voltage of 0.6V (typical) will result in a result based on the voltage applied to VDD of the device. The result of conversion of this reference across the VDD range can be approximated by: Conversion Result = 0.6V/(VDD/256)

Note:	The actual value of the Absolute Voltage Reference varies with temperature and part-to-part variation. The conversion is also susceptible to analog noise on the VDD pin and noise generated by the sinking or sourcing of current on the I/O pins.

REGISTER 7-1: ADCON0 REGISTER (ADDRESS 07h)

R/W-1	R/W-1	U-0	U-0	R/W-1	R/W-1	R/W-0	R/W-0
ANS1	ANS0	—	—	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

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

bit 7 **ANS1:** ADC Analog Input Pin Select bit
1 = GP1/AN1 configured for analog input
0 = GP1/AN1 configured as digital I/O

bit 6 **ANS0:** ADC Analog Input Pin Select bit(1), (2)
1 = GP0/AN0 configured as an analog input
0 = GP0/AN0 configured as digital I/O

bit 5-4 **Unimplemented:** Read as '0'

bit 3-2 **CHS<1:0>:** ADC Channel Select bits(3)
00 = Channel 00 (GP0/AN0)
01 = Channel 01 (GP1/AN1)
1X = 0.6V absolute Voltage reference

bit 1 **GO/DONE:** ADC Conversion Status bit(4)
1 = ADC conversion in progress. Setting this bit starts an ADC conversion cycle. This bit is automatically cleared by hardware when the ADC is done converting.
0 = ADC conversion completed/not in progress. Manually clearing this bit while a conversion is in process terminates the current conversion.

Note 1: When the ANS bits are set, the channel(s) selected are automatically forced into analog mode regardless of the pin function previously defined.
2: The ANS<1:0> bits are active regardless of the condition of ADON
3: CHS<1:0> bits default to 11 after any Reset.
4: If the ADON bit is clear, the GO/DONE bit cannot be set.

REGISTER 7-1: ADCON0 REGISTER (ADDRESS 07h) (CONTINUED)

bit 0 **ADON:** ADC Enable bit
1 = ADC module is operating
0 = ADC module is shut-off and consumes no power

Note 1: When the ANS bits are set, the channel(s) selected are automatically forced into analog mode regardless of the pin function previously defined.
2: The ANS<1:0> bits are active regardless of the condition of ADON
3: CHS<1:0> bits default to 11 after any Reset.
4: If the ADON bit is clear, the GO/$\overline{\text{DONE}}$ bit cannot be set.

REGISTER 7-2: ADRES REGISTER (ADDRESS 08h)

R-X	R-X	R-X	R-X	R-X	R-X	R-X	R-X
ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
bit 7							bit 0

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

bit 7-0 **ADRES<7:0>**

7.9 A/D Acquisition Requirements

For the ADC to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure 7-1. The source impedance (RS) and the internal sampling switch (RSS) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (RSS) impedance varies over the device voltage (VDD), see Figure 7-1. **The maximum recommended impedance for analog sources is 10 kΩ.** As the source impedance is decreased, the acquisition time may be decreased. After the analog input channel is selected (or changed), an A/D acquisition must be done before the conversion can be started. To calculate the minimum acquisition time, Equation 7-1 may be used. This equation assumes that 1/2 LSb error is used (256 steps for the ADC). The 1/2 LSb error is the maximum error allowed for the ADC to meet its specified resolution.

EQUATION 7-1: ACQUISITION TIME EXAMPLE

Assumptions: $Temperature = 50°C$ *and external impedance of* $10k\Omega$ $5.0V$ V_{DD}

$$T_{ACQ} = \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient}$$
$$= T_{AMP} + T_C + T_{COFF}$$
$$= 2\mu s + T_C + [(\text{Temperature} - 25°C)(0.05\mu s/°C)]$$

Solving for T_C*:*

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ ln(1/512)$$
$$= -25pF(1k\Omega + 7k\Omega + 10k\Omega)\ ln(0.00196)$$
$$= 2.81\mu s$$

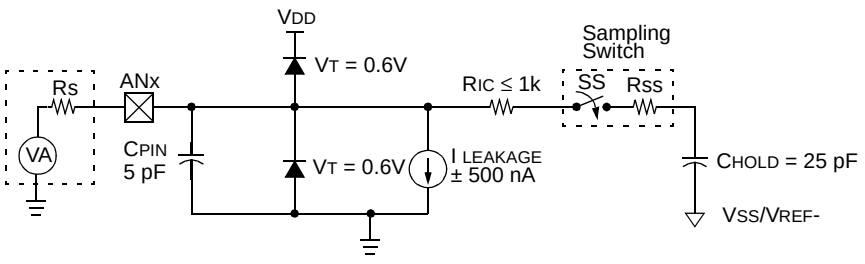
Therefore:

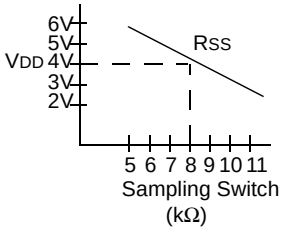
$$T_{ACQ} = 2\mu s + 2.81\mu s + [(50°C - 25°C)(0.05\mu s/°C)]$$
$$= 6.06us$$

Note 1: The charge holding capacitor (CHOLD) is not discharged after each conversion.

2: The maximum recommended impedance for analog sources is 10 kΩ. This is required to meet the pin leakage specification.

FIGURE 7-1: ANALOG INPUT MODEL





Legend:
CPIN = Input Capacitance
VT = Threshold Voltage
I LEAKAGE = Leakage current at the pin due to various junctions
RIC = Interconnect Resistance
SS = Sampling Switch
CHOLD = Sample/Hold Capacitance

8.0 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 PIC10F220/222 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
- Sleep
- Code Protection
- ID Locations
- In-Circuit Serial Programming™
- Clock Out

The PIC10F220/222 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 DRT, there is an 1.125 ms (typical) delay only on V_{DD} 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 or through a Watchdog Timer time-out.

8.1 Configuration Bits

The PIC10F220/222 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 $\overline{MCLR}$ enable bit and one bit is for code protection (see Register 8-1).

REGISTER 8-1: CONFIGURATION WORD FOR PIC10F220/222(1)

—	—	—	—	—	—	—	MCLRE	$\overline{CP}$	WDTE	$\overline{MCPU}$	IOSCFS
bit 11											bit 0

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

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

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

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

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

bit 1 **$\overline{MCPU}$:** Master Clear Pull-up Enable(2)
1 = Pull-up disabled
0 = Pull-up enabled

bit 0 **IOSCFS:** Internal Oscillator Frequency Select
1 = 8 MHz
0 = 4 MHz

Note 1: Refer to the "*PIC10F220/222 Memory Programming Specification*" (DS41266), to determine how to access the Configuration Word. The Configuration Word is not user addressable during device operation.
2: MCLRE must be a '1' to enable this selection.

8.2 Oscillator Configurations

8.2.1 OSCILLATOR TYPES

The PIC10F220/222 devices are offered with internal oscillator mode only.

- INTOSC: Internal 4/8 MHz Oscillator

8.2.2 INTERNAL 4/8 MHz OSCILLATOR

The internal oscillator provides a 4/8 MHz (nominal) system clock (see **Section 10.0 "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 `MOVLW XX` instruction where `XX` is the calibration value and is placed at the Reset vector. This will load the 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.

8.3 Reset

The device differentiates between various kinds of Reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset during normal operation
- $\overline{\text{MCLR}}$ Reset during Sleep
- WDT Time-out Reset during normal operation
- WDT Time-out Reset during Sleep
- Wake-up from Sleep on pin 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), $\overline{\text{MCLR}}$, WDT or Wake-up on pin change Reset during normal operation. They are not affected by a WDT Reset during Sleep or $\overline{\text{MCLR}}$ Reset during Sleep, since these Resets are viewed as resumption of normal operation. The exceptions to this are $\overline{\text{TO}}$, $\overline{\text{PD}}$ and GPWUF 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 8-1 for a full description of Reset states of all registers.

TABLE 8-1: RESET CONDITIONS FOR REGISTERS – PIC10F220/222

Register	Address	Power-on Reset	$\overline{\text{MCLR}}$ Reset, WDT Time-out, Wake-up On Pin Change,
W	—	qqqq qqqu(1)	qqqq qqqu(1)
INDF	00h	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu
PC	02h	1111 1111	1111 1111
STATUS	03h	0--1 1xxx	q00q quuu
FSR	04h	111x xxxx	111u uuuu
OSCCAL	05h	1111 1110	uuuu uuuu
GPIO	06h	---- xxxx	---- uuuu
ADCON0	07h	11-- 1100	11-- 1100
ADRES	08h	xxxx xxxx	uuuu uuuu
OPTION	—	1111 1111	1111 1111
TRIS	—	---- 1111	---- 1111

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

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

TABLE 8-2: RESET CONDITION FOR SPECIAL REGISTERS

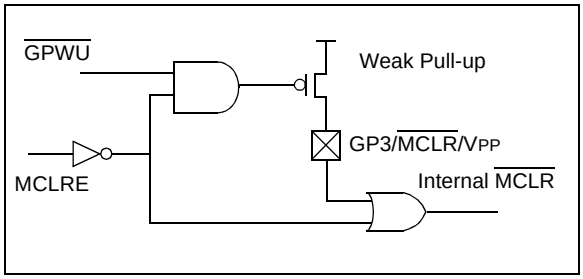
	STATUS Addr: 03h	PCL Addr: 02h
Power-on Reset	0--1 1xxx	1111 1111
$\overline{\text{MCLR}}$ Reset during normal operation	0--u uuuu	1111 1111
$\overline{\text{MCLR}}$ Reset during Sleep	0--1 0uuu	1111 1111
WDT Reset during Sleep	0--0 0uuu	1111 1111
WDT Reset normal operation	0--0 uuuu	1111 1111
Wake-up from Sleep on pin change	1--1 0uuu	1111 1111

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

8.3.1 $\overline{\text{MCLR}}$ ENABLE

This Configuration bit, when unprogrammed (left in the '1' state), enables the external $\overline{\text{MCLR}}$ function. When programmed, the $\overline{\text{MCLR}}$ function is tied to the internal V_{DD} and the pin is assigned to be a I/O. See Figure 8-1.

FIGURE 8-1: $\overline{\text{MCLR}}$ SELECT



8.4 Power-on Reset (POR)

The PIC10F220/222 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 V_{DD} has reached a high enough level for proper operation. To take advantage of the internal POR, program the GP3/$\overline{\text{MCLR}}$/V_{PP} pin as $\overline{\text{MCLR}}$ and tie through a resistor to V_{DD}, or program the pin as GP3. An internal weak pull-up resistor is implemented using a transistor (refer to Table 10-2 for the pull-up resistor ranges). This will eliminate external RC components usually needed to create a Power-on Reset. A maximum rise time for V_{DD} is specified. See **Section 10.0 "Electrical Characteristics"** for details.

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

A simplified block diagram of the on-chip Power-on Reset circuit is shown in Figure 8-2.

The Power-on Reset circuit and the Device Reset Timer (see **Section 8.5 "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 $\overline{\text{MCLR}}$ to be high. After the time-out period, which is typically 1.125 ms, it will reset the Reset latch and thus end the on-chip Reset signal.

A power-up example where $\overline{\text{MCLR}}$ is held low is shown in Figure 8-3. V_{DD} is allowed to rise and stabilize before bringing $\overline{\text{MCLR}}$ high. The chip will actually come out of Reset T_{DRT} msec after $\overline{\text{MCLR}}$ goes high.

In Figure 8-4, the on-chip Power-on Reset feature is being used ($\overline{\text{MCLR}}$ and V_{DD} are tied together or the pin is programmed to be GP3). The V_{DD} is stable before the Start-up timer times out and there is no problem in getting a proper Reset. However, Figure 8-5 depicts a problem situation where V_{DD} rises too slowly. The time between when the DRT senses that $\overline{\text{MCLR}}$ is high and when $\overline{\text{MCLR}}$ and V_{DD} actually reach their full value, is too long. In this situation, when the start-up timer times out, V_{DD} has not reached the V_{DD} (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 8-4).

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"* (DS00607).

FIGURE 8-2: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT

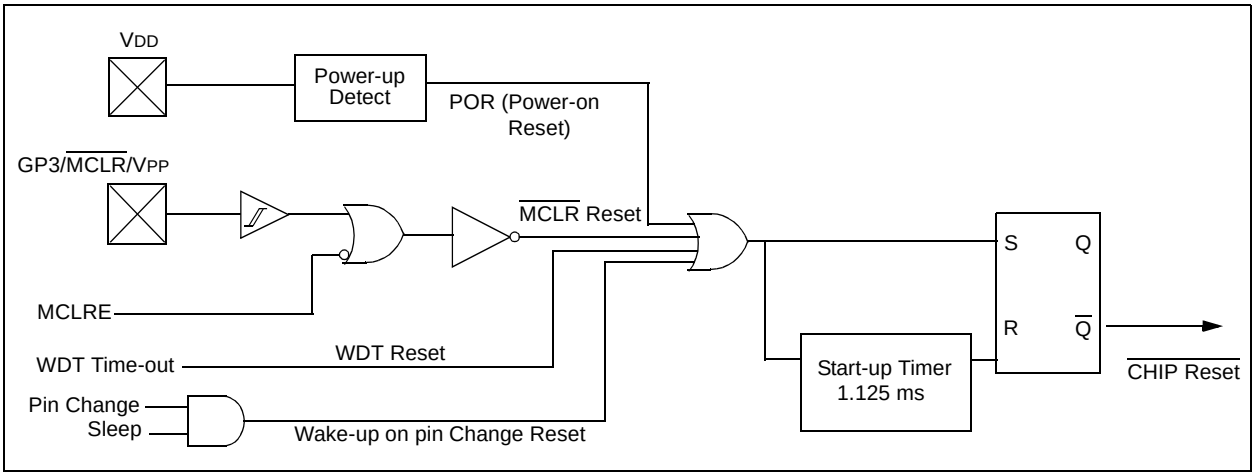


FIGURE 8-3: TIME-OUT SEQUENCE ON POWER-UP (MCLR PULLED LOW)

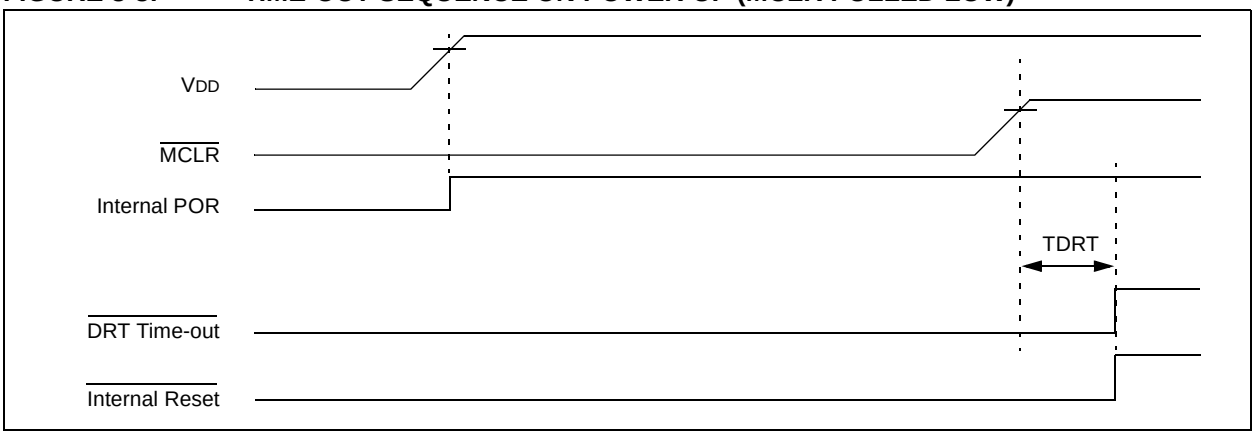


FIGURE 8-4: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD): FAST VDD RISE TIME

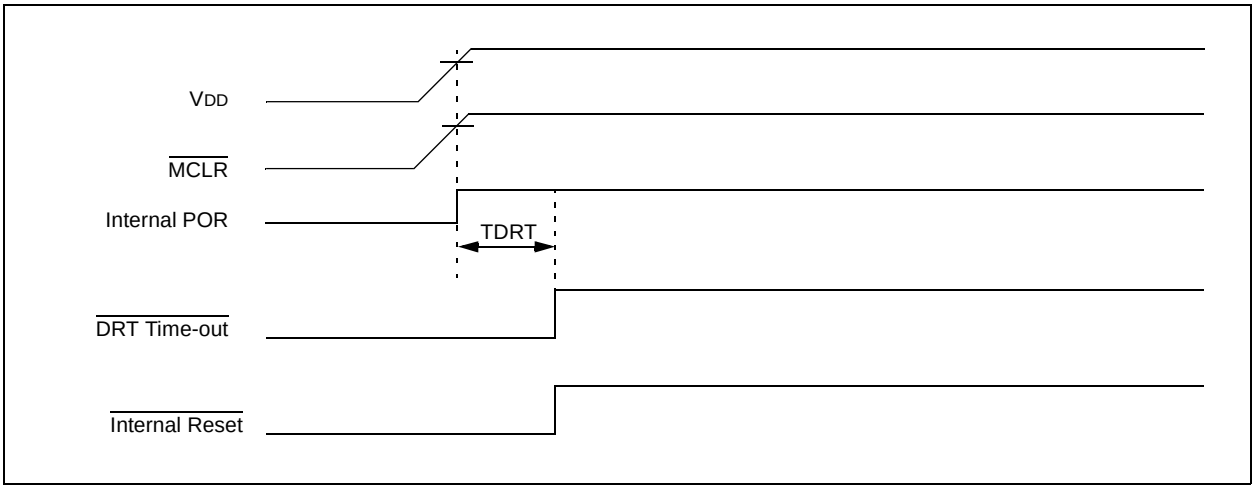
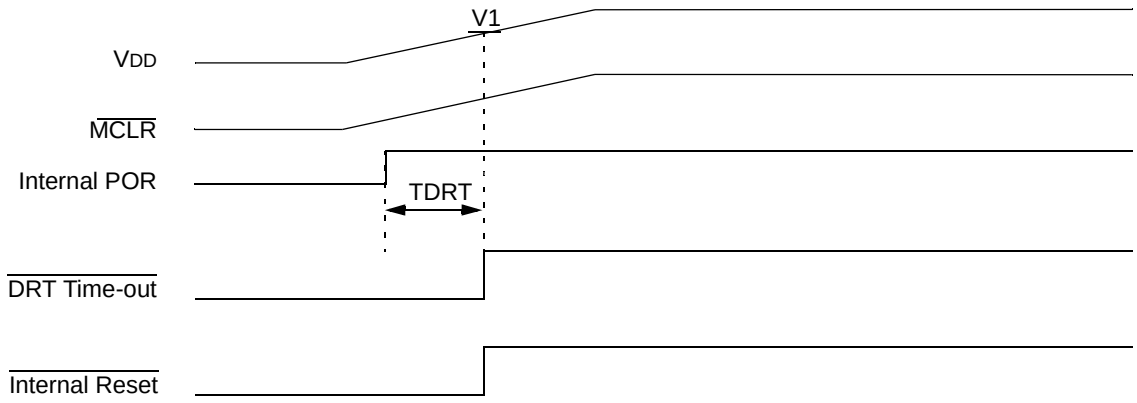


FIGURE 8-5: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD): SLOW VDD RISE TIME



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.

8.5 Device Reset Timer (DRT)

On the PIC10F220/222 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 V_{DD} to rise above V_{DD} min. and for the oscillator to stabilize.

The on-chip DRT keeps the devices in a Reset condition for approximately 1.125 ms after $\overline{MCLR}$ has reached a logic high (V_{IH} $\overline{MCLR}$) level. Programming GP3/$\overline{MCLR}$/V_{PP} as $\overline{MCLR}$ and using an external RC network connected to the $\overline{MCLR}$ 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/$\overline{MCLR}$/V_{PP} pin as a general purpose input.

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

Reset sources are POR, $\overline{MCLR}$, WDT time-out and wake-up on pin change. See **Section 8.9.2 "Wake-up from Sleep", Notes 1, 2 and 3**.

TABLE 8-3: DRT (DEVICE RESET TIMER PERIOD)

Oscillator	POR Reset	Subsequent Resets
INTOSC	1.125 ms (typical)	10 µs (typical)

8.6 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 4/8 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 $\overline{TO}$ 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 8.1 "Configuration Bits"**). Refer to the PIC10F220/222 Programming Specification to determine how to access the Configuration Word.

8.6.1 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, V_{DD} and part-to-part process variations (see DC specs).

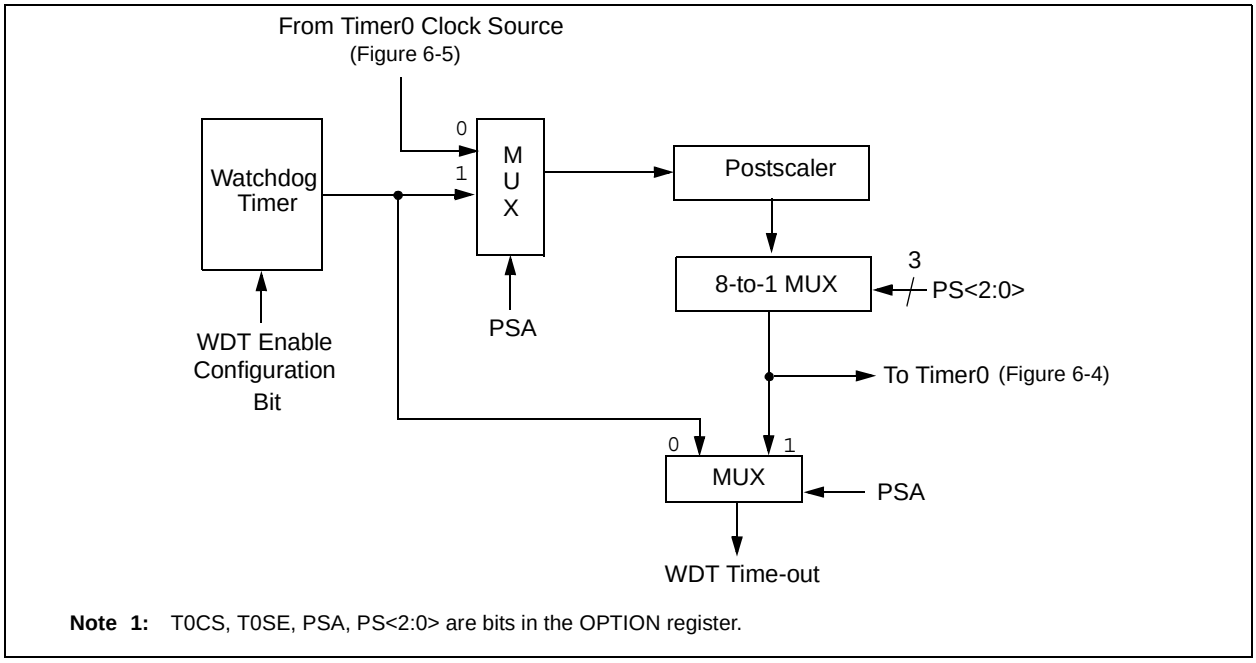
Under worst-case conditions (V_{DD} = Min., Temperature = Max., max. WDT prescaler), it may take several seconds before a WDT time-out occurs.

8.6.2 WDT PROGRAMMING CONSIDERATIONS

The `CLRWDT` 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 8-6: WATCHDOG TIMER BLOCK DIAGRAM



Note 1: T0CS, T0SE, PSA, PS<2:0> are bits in the OPTION register.

TABLE 8-4: SUMMARY OF REGISTERS ASSOCIATED WITH THE WATCHDOG TIMER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-On Reset	Value on All Other Resets
N/A	OPTION	$\overline{\text{GPWU}}$	$\overline{\text{GPPU}}$	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: Shaded boxes = Not used by Watchdog Timer, – = unimplemented, read as '0', u = unchanged.

8.7 Time-out Sequence, Power-down and Wake-up from Sleep Status Bits (TO/PD/GPWUF/CWUF)

The $\overline{\text{TO}}$, $\overline{\text{PD}}$ and GPWUF bits in the STATUS register can be tested to determine if a Reset condition has been caused by a Power-up condition, a $\overline{\text{MCLR}}$, Watchdog Timer (WDT) Reset or wake-up on pin change.

TABLE 8-5: $\overline{\text{TO}}$/$\overline{\text{PD}}$/GPWUF STATUS AFTER RESET

GPWUF	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Reset Caused By
0	0	0	WDT wake-up from Sleep
0	0	u	WDT time-out (not from Sleep)
0	1	0	$\overline{\text{MCLR}}$ wake-up from Sleep
0	1	1	Power-up
0	u	u	$\overline{\text{MCLR}}$ not during Sleep
1	1	0	Wake-up from Sleep on pin change

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

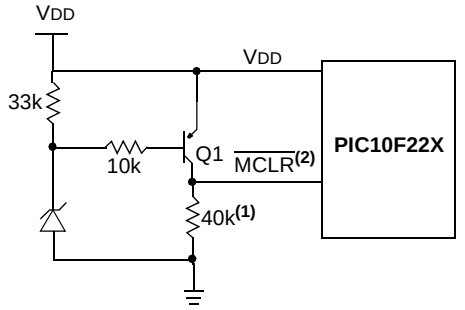
Note 1: The $\overline{\text{TO}}$, $\overline{\text{PD}}$ and GPWUF bits maintain their status (u) until a Reset occurs. A low-pulse on the $\overline{\text{MCLR}}$ input does not change the $\overline{\text{TO}}$, $\overline{\text{PD}}$ or GPWUF Status bits.

8.8 Reset on Brown-out

A Brown-out is a condition where device power (V_{DD}) 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 PIC10F220/222 devices when a Brown-out occurs, external Brown-out protection circuits may be built, as shown in Figure 8-7 and Figure 8-8.

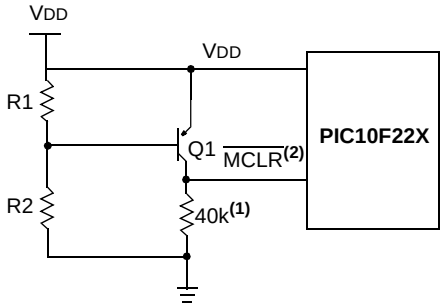
FIGURE 8-7: BROWN-OUT PROTECTION CIRCUIT 1



Note 1: This circuit will activate Reset when V_{DD} goes below Vz + 0.7V (where Vz = Zener voltage).

2: Pin must be configured as $\overline{MCLR}$.

FIGURE 8-8: BROWN-OUT PROTECTION CIRCUIT 2

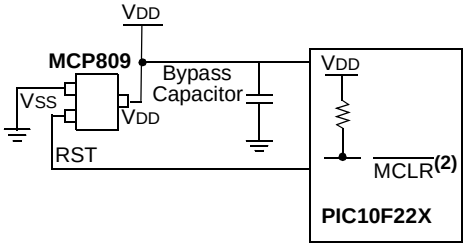


Note 1: This brown-out circuit is less expensive, although less accurate. Transistor Q1 turns off when V_{DD} is below a certain level such that:

$$V_{DD} \bullet \frac{R1}{R1 + R2} = 0.7V$$

2: Pin must be configured as $\overline{MCLR}$.

FIGURE 8-9: BROWN-OUT PROTECTION CIRCUIT 3



Note 1: This Brown-out Protection circuit employs Microchip Technology's MCP809 micro-controller supervisor. There are 7 different trip point selections to accommodate 5V to 3V systems.

2: Pin must be configured as $\overline{MCLR}$.

8.9 Power-down Mode (Sleep)

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

8.9.1 SLEEP

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

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

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

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

8.9.2 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.

These events cause a device Reset. The TO, PD GPWUF bits can be used to determine the cause of a 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).

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.

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

8.10 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.

8.11 ID Locations

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

Use only the lower 4 bits of the ID locations and always program the upper 8 bits as '1's.

8.12 In-Circuit Serial Programming™

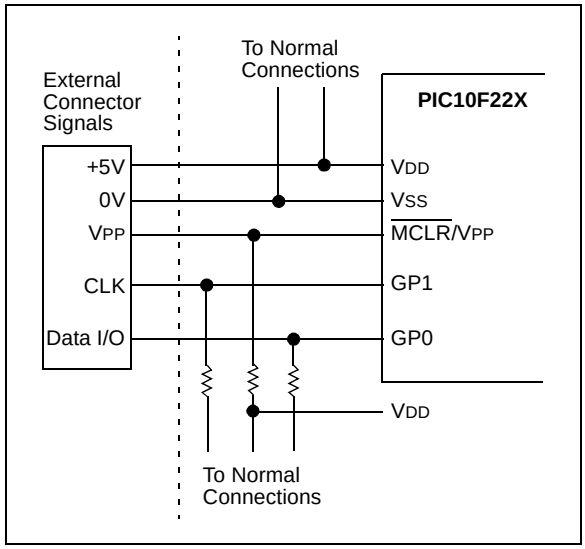
The PIC10F220/222 microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then 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 PIC10F220/222 Programming Specifications.

A typical In-Circuit Serial Programming connection is shown in Figure 8-10.

FIGURE 8-10: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



NOTES:

9.0 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 9-1, while the various opcode fields are summarized in Table 9-1.

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 9-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (0x00 to 0x7F)
W	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select; d = 0 (store result in W) 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
$\overline{TO}$	Time-out bit
$\overline{PD}$	Power-down bit
dest	Destination, either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

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

Figure 9-1 shows the three general formats that the instructions can have. All examples in the figure use the following format to represent a hexadecimal number:

'0xhhh'

where 'h' signifies a hexadecimal digit.

FIGURE 9-1: GENERAL FORMAT FOR INSTRUCTIONS

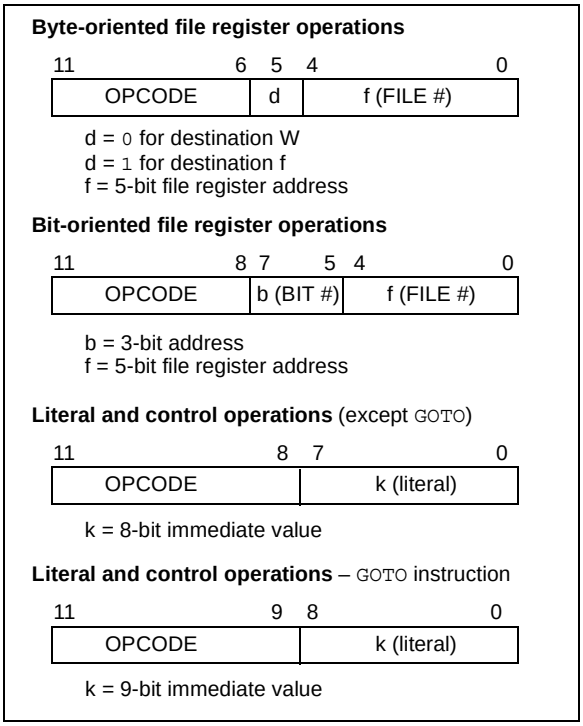


TABLE 9-2: INSTRUCTION SET SUMMARY

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

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

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

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

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

9.1 Instruction Description

ADDWF	Add W and f
Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register and register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

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

ANDWF	AND W with f
Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(W) AND (f) → (destination)
Status Affected:	Z
Description:	The contents of the W register are AND'ed with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

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

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

BTFSC	Bit Test f, Skip if Clear
Syntax:	[*label*] BTFSC f,b
Operands:	0 ≤ f ≤ 31 0 ≤ b ≤ 7
Operation:	skip if (f<b>) = 0
Status Affected:	None
Description:	If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a 2-cycle instruction.

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 31 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Description:	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 NOP is executed instead, making this a 2-cycle instruction.

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 eight-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 two-cycle instruction.

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.

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 k
Operands:	None
Operation:	00h → WDT; 0 → WDT prescaler (if assigned); 1 → $\overline{TO}$; 1 → $\overline{PD}$
Status Affected:	$\overline{TO}$, $\overline{PD}$
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 $\overline{TO}$ and $\overline{PD}$ are set.

COMF	Complement f
Syntax:	[*label*] COMF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	($\bar{f}$) → (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'.

DECF	Decrement f
Syntax:	[*label*] DECF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f) – 1 → (dest)
Status Affected:	Z
Description:	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’.

DECFSZ	Decrement f, Skip if 0
Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f) – 1 → d; skip if result = 0
Status Affected:	None
Description:	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 two-cycle instruction.

GOTO	Unconditional Branch
Syntax:	[*label*] GOTO k
Operands:	$0 \le k \le 511$
Operation:	k → PC<8:0>; STATUS<6:5> → PC<10:9>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 9-bit immediate value is loaded into PC bits <8:0>. The upper bits of PC are loaded from STATUS<6:5>. GOTO is a two-cycle instruction.

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f) + 1 → (dest)
Status Affected:	Z
Description:	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’.

INCFSZ	Increment f, Skip if 0
Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \le f \le 31$ $d \in [0,1]$
Operation:	(f) + 1 → (dest), skip if result = 0
Status Affected:	None
Description:	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 two-cycle instruction.

IORLW	Inclusive OR literal with W
Syntax:	[*label*] IORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .OR. (k) → (W)
Status Affected:	Z
Description:	The contents of the W register are OR’ed with the eight-bit literal ‘k’. The result is placed in the W register.

IORWF	Inclusive OR W with f
Syntax:	[*label*] IORWF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(W).OR. (f) → (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'.

MOVF	Move f
Syntax:	[*label*] MOVF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(f) → (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 → (W)
Status Affected:	None
Description:	The eight-bit literal 'k' is loaded into the W register. The "don't cares" will assembled as '0's.

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

NOP	No Operation
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Description:	No operation.

OPTION	Load OPTION Register
Syntax:	[*label*] OPTION
Operands:	None
Operation:	(W) → 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: $0 \le k \le 255$

Operation: $k \rightarrow (W)$;
$TOS \rightarrow PC$

Status Affected: None

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

RLF **Rotate Left f through Carry**

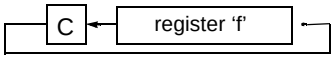
Syntax: [*label*] RLF f,d

Operands: $0 \le f \le 31$
$d \in [0,1]$

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'.



RRF **Rotate Right f through Carry**

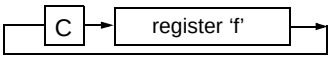
Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 31$
$d \in [0,1]$

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'.



SLEEP **Enter SLEEP Mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT;
0 → WDT prescaler;
1 → $\overline{TO}$;
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$, RBWUF

Description: Time-out Status bit ($\overline{TO}$) is set. The Power-down Status bit ($\overline{PD}$) 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 on Sleep for more details.

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 31$
$d \in [0,1]$

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'.

SWAPF **Swap Nibbles in f**

Syntax: [*label*] SWAPF f,d

Operands: $0 \le f \le 31$
$d \in [0,1]$

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
Syntax:	[*label*] TRIS f
Operands:	f = 6
Operation:	(W) → TRIS register f
Status Affected:	None
Description:	TRIS register 'f' (f = 6 or 7) is loaded with the contents of the W register

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

XORWF	Exclusive OR W with f
Syntax:	[*label*] XORWF f,d
Operands:	0 ≤ f ≤ 31 d ∈ [0,1]
Operation:	(W) .XOR. (f) → (dest)
Status Affected:	Z
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

10.0 ELECTRICAL CHARACTERISTICS

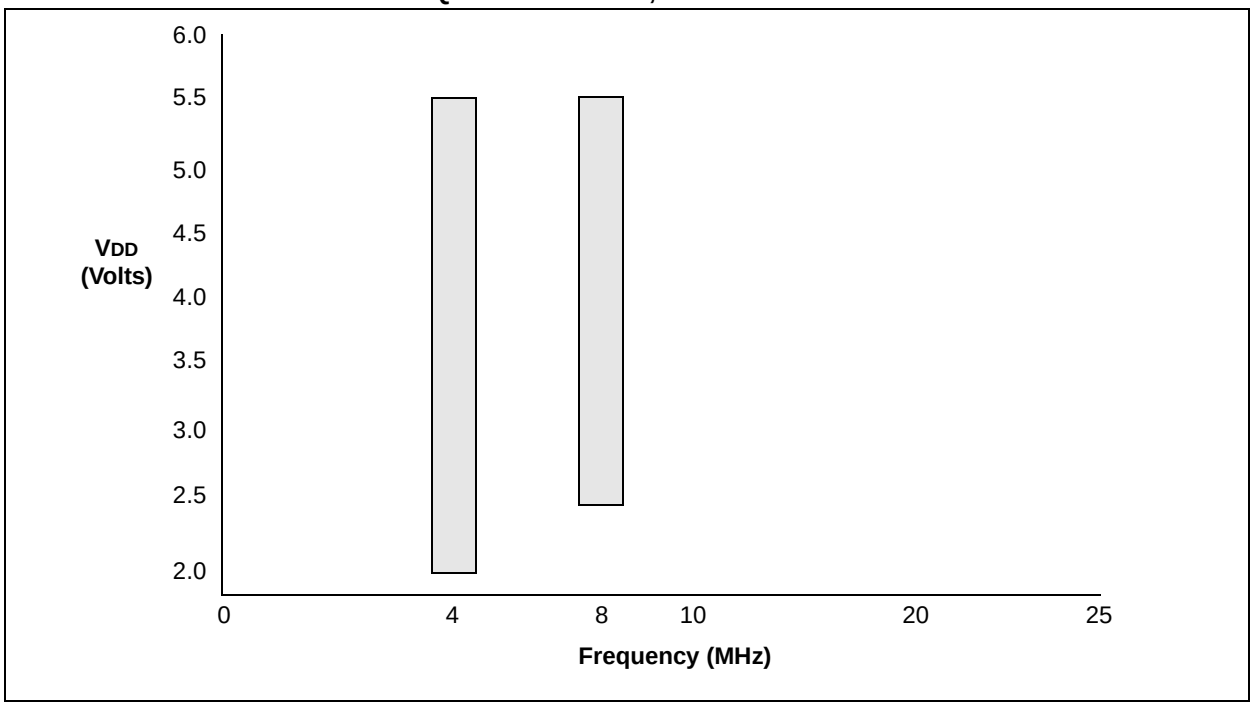
Absolute Maximum Ratings(†)

Ambient temperature under bias ..... -40°C to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... 0 to +6.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} ..... 0 to +13.5V
Voltage on all other pins with respect to V_{SS} ..... -0.3V to (V_{DD} + 0.3V)
Total power dissipation(1) ..... 200 mW
Max. current out of V_{SS} pin ..... 80 mA
Max. current into V_{DD} pin ..... 80 mA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) ..... ±20 mA
Output clamp current, I_{OK} (V_O < 0 or V_O > V_{DD}) ..... ±20 mA
Max. output current sunk by any I/O pin ..... 25 mA
Max. output current sourced by any I/O pin ..... 25 mA
Max. output current sourced by I/O port ..... 75 mA
Max. output current sunk by I/O port ..... 75 mA

Note 1: Power dissipation is calculated as follows: $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} .. + \sum(V_{OL} \times I_{OL})$

†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 10-1: VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C



10.1 DC Characteristics: PIC10F220/222 (Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial)				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**	2.0	—	5.5	V	See **Figure 10-1**
D002	VDR	**RAM Data Retention Voltage(2)**	—	1.5*	—	V	Device in Sleep mode
D003	VPOR	**VDD Start Voltage** to ensure Power-on Reset	—	Vss	—	V	See **Section 8.4 "Power-on Reset (POR)"** for details
D004	SVDD	**VDD Rise Rate** to ensure Power-on Reset	0.05*	—	—	V/ms	See **Section 8.4 "Power-on Reset (POR)"** for details
D010	IDD	**Supply Current(3)**	— — — —	170 350 250 450	TBD TBD TBD TBD	µA µA µA µA	FOSC = 4 MHz, VDD = 2.0V FOSC = 4 MHz, VDD = 5.0V FOSC = 8 MHz, VDD = 2.0V FOSC = 8 MHz, VDD = 5.0V
D020	IPD	**Power-down Current(4)**	—	0.1	TBD	µA	VDD = 2.0V
D022	ΔIWDT	**WDT Current(4)**	—	1.0	TBD	µA	VDD = 2.0V
D024	ΔIADC	**A/D Current**	—	80	TBD	µA	VDD = 2.0V

Legend: TBD = To Be Determined.

* 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.

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, $\overline{MCLR}$ = 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.

10.2 DC Characteristics: PIC10F220/222 (Extended)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +125°C (extended)				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**	2.0		5.5	V	See **Figure 10-1**
D002	VDR	**RAM Data Retention Voltage(2)**	—	1.5*	—	V	Device in Sleep mode
D003	VPOR	**VDD Start Voltage** to ensure Power-on Reset	—	Vss	—	V	See **Section 8.4 "Power-on Reset (POR)"** for details
D004	SVDD	**VDD Rise Rate** to ensure Power-on Reset	0.05*	—	—	V/ms	See **Section 8.4 "Power-on Reset (POR)"** for details
D010	IDD	**Supply Current(3)**	— — — —	170 350 250 450	TBD TBD TBD TBD	µA µA µA µA	FOSC = 4 MHz, VDD = 2.0V FOSC = 4 MHz, VDD = 5.0V FOSC = 8 MHz, VDD = 2.0V FOSC = 8 MHz, VDD = 5.0V
D020	IPD	**Power-down Current(4)**	—	0.1	TBD	µA	VDD = 2.0V
D022	ΔIWDT	**WDT Current(4)**	—	1.0	TBD	µA	VDD = 2.0V
D024	ΔIADC	**A/D Current**	—	80	TBD	µA	VDD = 2.0V

Legend: TBD = To Be Determined.

* 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.

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, $\overline{MCLR}$ = 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.

TABLE 10-1: DC CHARACTERISTICS: PIC10F220/222 (Industrial, Extended)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating temperature -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended) Operating voltage VDD range as described in DC specification				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O ports:					
D030		with TTL buffer	Vss	—	0.8V	V	For all 4.5 ≤ VDD ≤ 5.5V
D030A			Vss	—	0.15 VDD	V	Otherwise
D031		with Schmitt Trigger buffer	Vss	—	0.15 VDD	V	
D032		$\overline{MCLR}$, T0CKI	Vss	—	0.15 VDD	V	
	VIH	**Input High Voltage**					
		I/O ports:		—			
D040		with TTL buffer	2.0	—	VDD	V	4.5 ≤ VDD ≤ 5.5V
D040A			0.25 VDD + 0.8V	—	VDD	V	Otherwise
D041		with Schmitt Trigger buffer	0.85 VDD	—	VDD	V	For entire VDD range
D042		$\overline{MCLR}$, T0CKI	0.85 VDD	—	VDD	V	
D070	IPUR	**GPIO weak pull-up current**	TBD	250	TBD	µA	VDD = 5V, VPIN = VSS
	IIL	**Input Leakage Current(1), (2)**					
D060		I/O ports	—	—	± 1	µA	Vss ≤ VPIN ≤ VDD, Pin at high-impedance
D061		GP3/$\overline{MCLR}$(3)	—	—	± 5	µA	Vss ≤ VPIN ≤ VDD
		Output Low Voltage					
D080		I/O ports	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V, -40°C to +85°C
D080A			—	—	0.6	V	IOL = 7.0 mA, VDD = 4.5V, +85°C to +125°C
		Output High Voltage					
D090		I/O ports(2)	VDD–0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V, -40°C to +85°C
D090A			VDD–0.7	—	—	V	IOH = -2.5 mA, VDD = 4.5V, +85°C to +125°C
		Capacitive Loading Specs on Output Pins					
D101		All I/O pins	—	—	50*	pF	

Legend: TBD = To Be Determined.

† 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 $\overline{MCLR}$ 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.

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

3: This specification applies when GP3/$\overline{MCLR}$ is configured as an input with pull-up disabled. The leakage current of the $\overline{MCLR}$ circuit is higher than the standard I/O logic.

TABLE 10-2: PULL-UP RESISTOR RANGES

VDD (Volts)	Temperature (°C)	Min	Typ	Max	Units
GP0/GP1					
2.0	-40	TBD	91K	TBD	Ω
	25	TBD	105K	TBD	Ω
	85	TBD	118K	TBD	Ω
	125	TBD	125K	TBD	Ω
5.5	-40	TBD	18K	TBD	Ω
	25	TBD	23K	TBD	Ω
	85	TBD	26K	TBD	Ω
	125	TBD	28K	TBD	Ω
GP3					
2.0	-40	TBD	63K	TBD	Ω
	25	TBD	74K	TBD	Ω
	85	TBD	83K	TBD	Ω
	125	TBD	87K	TBD	Ω
5.5	-40	TBD	16K	TBD	Ω
	25	TBD	21K	TBD	Ω
	85	TBD	25K	TBD	Ω
	125	TBD	27K	TBD	Ω

Legend: TBD = To Be determined.

* These parameters are characterized but not tested.

10.3 Timing Parameter Symbology and Load Conditions

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS

T	
F Frequency	T Time

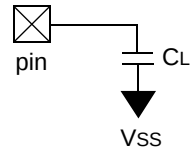
Lowercase subscripts (pp) and their meanings:

pp			
2	to	mc	$\overline{\text{MCLR}}$
ck	CLKOUT	osc	Oscillator
cy	Cycle time	os	OSC1
drt	Device Reset Timer	t0	T0CKI
io	I/O port	wdt	Watchdog Timer

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (high-impedance)	V	Valid
L	Low	Z	High-impedance

FIGURE 10-2: LOAD CONDITIONS



Legend:
C_L = 50 pF for all pins

TABLE 10-3: CALIBRATED INTERNAL RC FREQUENCIES

AC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise specified)** Operating Temperature -40°C ≤ TA ≤ +85°C (industrial), -40°C ≤ TA ≤ +125°C (extended)					
Param No.	**Sym**	**Characteristic**	**Freq. Tolerance**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
F10	FOSC	Internal Calibrated INTOSC Frequency(1), (2)	± 1%	7.92	8	8.08	MHz	3.5V @ TA = 25°C
			± 2%	7.84	8	8.16	MHz	2.5V ≤ VDD ≤ 5.5V Temperature 0-85°C
			± 5%	7.60	8	8.4	MHz	2.0V ≤ VDD ≤ 5.5V -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended)

Legend: TBD = To Be Determined.

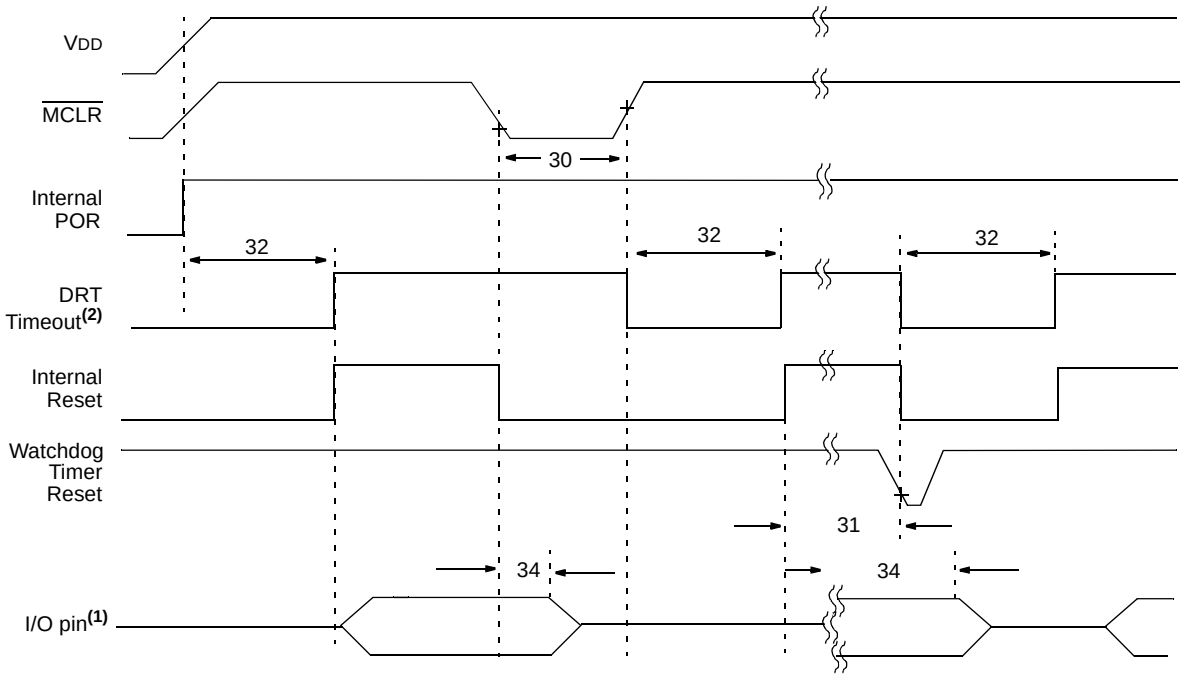
* These parameters are characterized but not tested.

† 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: The 4 MHz clock is derived from the 8 MHz oscillator. To obtain 4 MHz tolerance values, divide the appropriate 8 MHz value by 2.

FIGURE 10-3: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER TIMING



Note 1: I/O pins must be taken out of High-impedance mode by enabling the output drivers in software.
2: Runs on POR Reset only.

TABLE 10-4: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER

AC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended)				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
30	TMCL	$\overline{\text{MCLR}}$ Pulse Width (low)	2000*	—	—	ns	VDD = 5.0V
31	TWDT	Watchdog Timer Time-out Period (no prescaler) Watchdog Timer Time-out Period (no prescaler)	9* 9*	18* 18*	30* 40*	ms ms	VDD = 5.0V (Industrial) VDD = 5.0V (Extended)
32	TDRT	Device Reset Timer Period	0.5* 0.5*	1.125* 1.125*	2* 2.5*	ms ms	VDD = 5.0V (Industrial) VDD = 5.0V (Extended)
34	TIOZ	I/O High-impedance from $\overline{\text{MCLR}}$ low	—	—	2000*	ns	

* 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 10-4: TIMER0 CLOCK TIMINGS

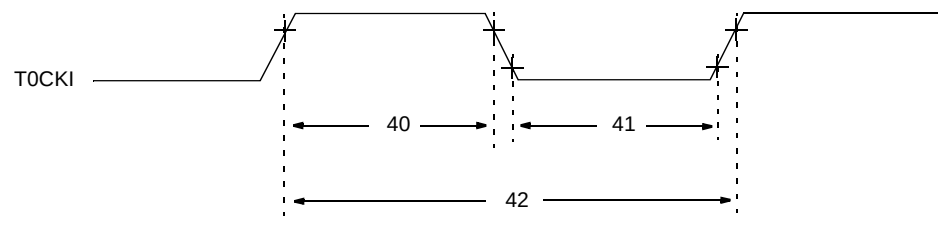


TABLE 10-5: TIMER0 CLOCK REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C (industrial) -40°C ≤ TA ≤ +125°C (extended)				
Param No.	**Sym**	**Characteristic**		**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		20 or TCY + 40* N	—	—	ns	Whichever is greater. N = Prescale Value (1, 2, 4,..., 256)

* 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 10-6: A/D CONVERTER CHARACTERISTICS (PIC10F220)

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
A01	NR	Resolution	—	—	8 bits	bit	
A02	EABS	Total Absolute Error*(1)	—	—	TBD	LSb	VDD = 5.0V
A03	EIL	Integral Error	—	—	±1	LSb	VDD = 5.0V
A04	EDL	Differential Error	—	—	-1 < EDL ≤ + 1.0	LSb	No missing codes to 8 bits VDD = 5.0V
A05	EFS	Full-scale Range	2.0*	—	5.5*	V	VDD
A06	EOFF	Offset Error	—	—	±1	LSb	VREF = 5.0V
A07	EGN	Gain Error	—	—	±1	LSb	VREF = 5.0V
A10	—	Monotonicity	—	guaranteed(2)	—	—	VSS ≤ VAIN ≤ VDD
A25	VAIN	Analog Input Voltage	VSS	—	VDD	V	
A30	ZAIN	Recommended Impedence of Analog Voltage Source	—	—	10	kΩ	

* These parameters are characterized but not tested.

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

Note 1: Total Absolute Error includes integral, differential, offset and gain errors.

2: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

3: VREF current is from external VREF or VDD pin, whichever is selected as reference input.

4: When A/D is off, it will not consume any current other than leakage current. The power-down current specification includes any such leakage from the A/D module.

TABLE 10-7: A/D CONVERTER CHARACTERISTICS (PIC10F222)

Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
A01	NR	Resolution	—	—	8 bits	bit	
A03	EIL	Integral Error	—	—	±1	LSb	VDD = 5.0V
A04	EDL	Differential Error	—	—	-1 < EDL ≤ + 1.0	LSb	No missing codes to 8 bits VDD = 5.0V
A05	EFS	Full-scale Range	2.0*	—	5.5*	V	VDD
A06	EOFF	Offset Error	—	—	±1	LSb	VREF = 5.0V
A07	EGN	Gain Error	—	—	±1	LSb	VREF = 5.0V
A10	—	Monotonicity	—	guaranteed(1)	—	—	VSS ≤ VAIN ≤ VDD
A25	VAIN	Analog Input Voltage	VSS	—	VDD	V	
A30	ZAIN	Recommended Impedence of Analog Voltage Source	—	—	10	kΩ	

* These parameters are characterized but not tested.

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

Note 1: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

TABLE 10-8: PIC10F220/222 A/D CONVERSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C							
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
AD131	TCNV	Conversion Time (not including Acquisition Time)(1)	—	13	—	TCY	Set GO/$\overline{\text{DONE}}$ bit to new data in A/D Result register
AD132*	TACQ	Acquisition Time	—	3.5 5	—	µs µs	VDD = 5V VDD = 2.5V

* These parameters are characterized but not tested.

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

11.0 DEVELOPMENT SUPPORT

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

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C18 and MPLAB C30 C Compilers
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

11.1 MPLAB Integrated Development Environment Software

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

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

The MPLAB IDE allows you to:

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

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

11.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PICmicro 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 IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

11.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 family of microcontrollers and dsPIC30F family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

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

11.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. 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 ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C 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 dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

11.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PICmicro 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, as well as internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

11.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PICmicro microcontrollers.

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft® Windows® 32-bit operating system were chosen to best make these features available in a simple, unified application.

11.8 MPLAB ICE 4000 High-Performance In-Circuit Emulator

The MPLAB ICE 4000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for high-end PICmicro MCUs and dsPIC DSCs. Software control of the MPLAB ICE 4000 In-Circuit Emulator is provided by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 4000 is a premium emulator system, providing the features of MPLAB ICE 2000, but with increased emulation memory and high-speed performance for dsPIC30F and PIC18XXXX devices. Its advanced emulator features include complex triggering and timing, and up to 2 Mb of emulation memory.

The MPLAB ICE 4000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft Windows 32-bit operating system were chosen to best make these features available in a simple, unified application.

11.9 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PICmicro MCUs and can be used to develop for these and other PICmicro MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming™ (ICSP™) protocol, offers cost-effective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PICmicro devices.

11.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} 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 PICmicro 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 SD/MMC card for file storage and secure data applications.

11.11 PICSTART Plus Development Programmer

The PICSTART Plus Development Programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus Development Programmer supports most PICmicro devices in DIP packages up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus Development Programmer is CE compliant.

11.12 Demonstration, Development and Evaluation Boards

A wide variety of demonstration, development and evaluation boards for various PICmicro 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.

Check the Microchip web page (www.microchip.com) and the latest *"Product Selector Guide"* (DS00148) for the complete list of demonstration, development and evaluation kits.

12.0 DC AND AC CHARACTERISTICS GRAPHS AND CHARTS

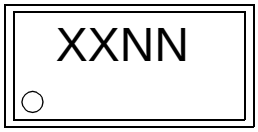
Graphs and charts are not available at this time.

NOTES:

13.0 PACKAGING INFORMATION

13.1 Package Marking Information

6-Lead SOT-23



Example

8-Lead PDIP

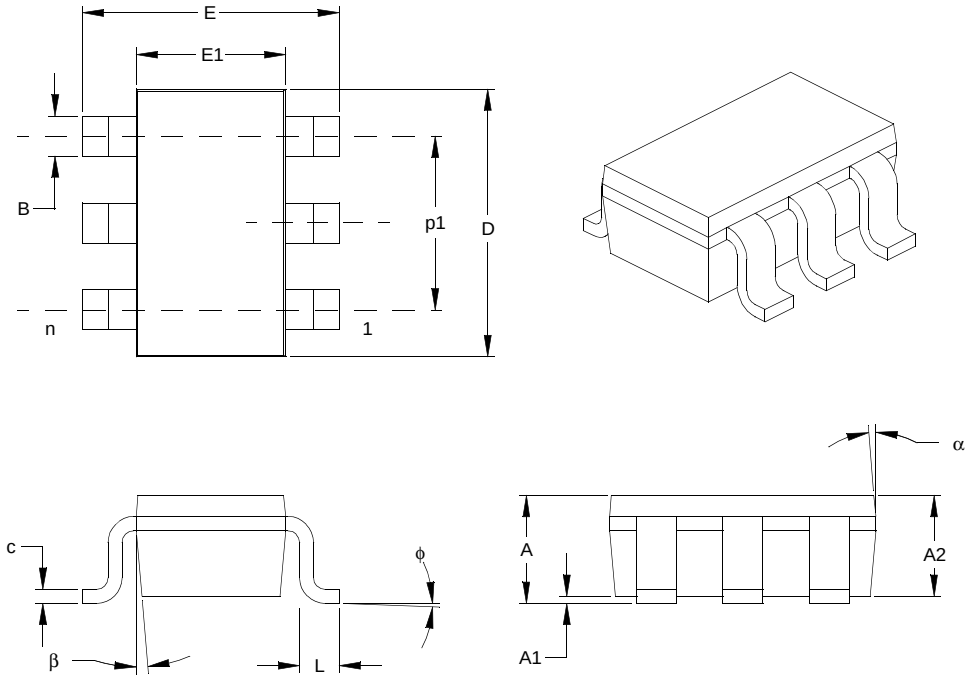
Example

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

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

***** Standard PICmicro device marking consists of Microchip part number, year code, week code, and traceability code. For PICmicro 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.

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



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		6			6	
Pitch	p		.038 BSC			0.95 BSC	
Outside lead pitch	p1		.075 BSC			1.90 BSC	
Overall Height	A	.035	.046	.057	0.90	1.18	1.45
Molded Package Thickness	A2	.035	.043	.051	0.90	1.10	1.30
Standoff	A1	.000	.003	.006	0.00	0.08	0.15
Overall Width	E	.102	.110	.118	2.60	2.80	3.00
Molded Package Width	E1	.059	.064	.069	1.50	1.63	1.75
Overall Length	D	.110	.116	.122	2.80	2.95	3.10
Foot Length	L	.014	.018	.022	0.35	0.45	0.55
Foot Angle	ϕ	0	5	10	0	5	10
Lead Thickness	c	.004	.006	.008	0.09	0.15	0.20
Lead Width	B	.014	.017	.020	0.35	0.43	0.50
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

***** Controlling Parameter

Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .005" (0.127mm) per side.

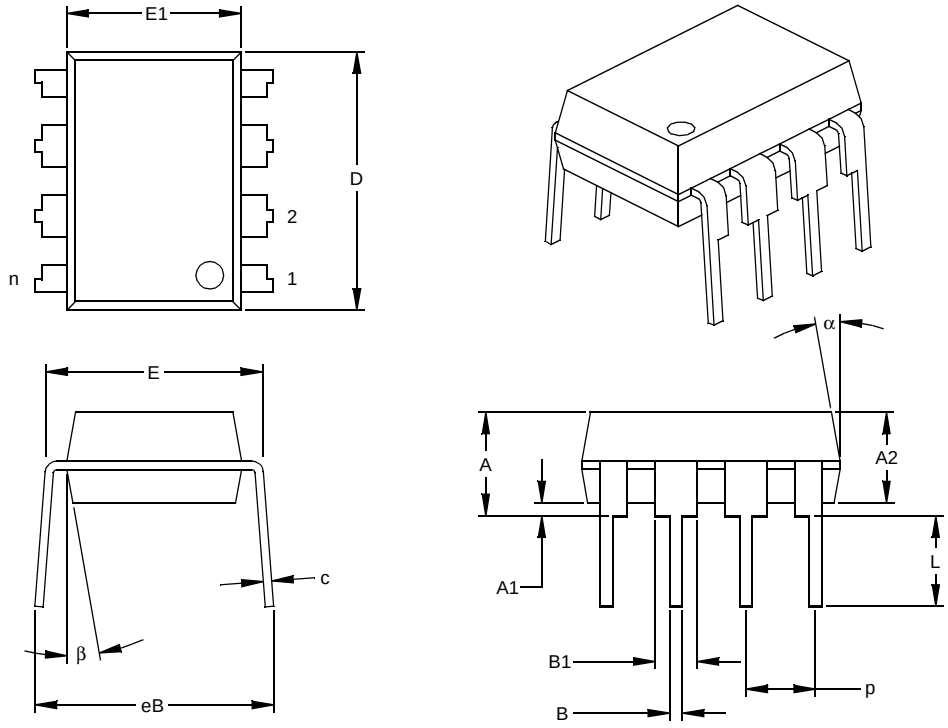
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
See ASME Y14.5M

JEITA (formerly EIAJ) equivalent: SC-74A

Drawing No. C04-120

Revised 09-12-05

8-Lead Plastic Dual In-line (P) – 300 mil Body (PDIP)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		8			8	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.360	.373	.385	9.14	9.46	9.78
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

***** Controlling Parameter
§ Significant Characteristic
Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-018

APPENDIX A: REVISION HISTORY

Revision A

Original release of document.

Revision B (03/2006)

Table 3-1, GP1; Section 4.7, Program Counter; Table 5-2; Figure 8-5; Section 9.1, ANDWF, SLEEP, SUBWF, SWAPF, XORLW.

INDEX

Z

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, 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
- Development Systems Information Line

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://support.microchip.com

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC10F220/222 Literature Number: DS41270B

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

PRODUCT IDENTIFICATION SYSTEM

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

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

Device:	PIC10F220[(1)], PIC10F222[(1)]; V_{DD} range 2.0V to 5.5V	
Temperature Range:	I = -40°C to +85°C	(Industrial)
	E = -40°C to +125°C	(Extended)
Package:	OT =	SOT, 6-LD (Pb-free)
	P =	300 mil PDIP, 8-LD (Pb-free)
Pattern:	Special Requirements	

Examples:

a) PIC10F220 – I/P = Industrial temp., PDIP package (Pb-free)

b) PIC10F222 – T-I/OT = Industrial temp., SOT package (Pb-free)

Note 1: SOT packages are only available in tape and reel.

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://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Alpharetta, GA
Tel: 770-640-0034
Fax: 770-640-0307

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

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

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

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

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

San Jose
Mountain View, CA
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8676-6200
Fax: 86-28-8676-6599

China - Fuzhou
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

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: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Shunde
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

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

China - Xian
Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi
Tel: 91-11-5160-8631
Fax: 91-11-5160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Gumi
Tel: 82-54-473-4301
Fax: 82-54-473-4302

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

Malaysia - Penang
Tel: 60-4-646-8870
Fax: 60-4-646-5086

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: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

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

EUROPE

Austria - Wels
Tel: 43-7242-2244-399
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 - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

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

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

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

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

02/16/06