Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

PIC10F220/222
Data Sheet

High-Performance Microcontrollers with 8-Bit A/D

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

Printed on recycled paper.

ISBN: 9781620775912

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, 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 $\overline{MCLR}$ Input Pin
- Internal Weak Pull-Ups on I/O Pins
- Power-Saving Sleep mode
- Wake-up from Sleep on Pin Change

Low-Power Features/CMOS Technology:

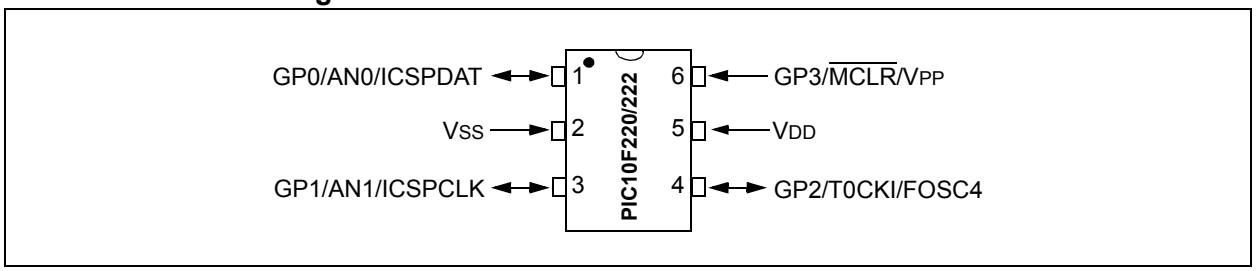
- Operating Current:
  - < 175 μA @ 2V, 4 MHz
- Standby Current:
  - 100 nA @ 2V, typical
- Low-Power, High-Speed Flash Technology:
  - 100,000 Flash endurance
  - > 40-year retention
- Fully Static Design
- Wide Operating Voltage Range: 2.0V to 5.5V
- Wide Temperature Range:
  - Industrial: -40°C to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

Peripheral Features:

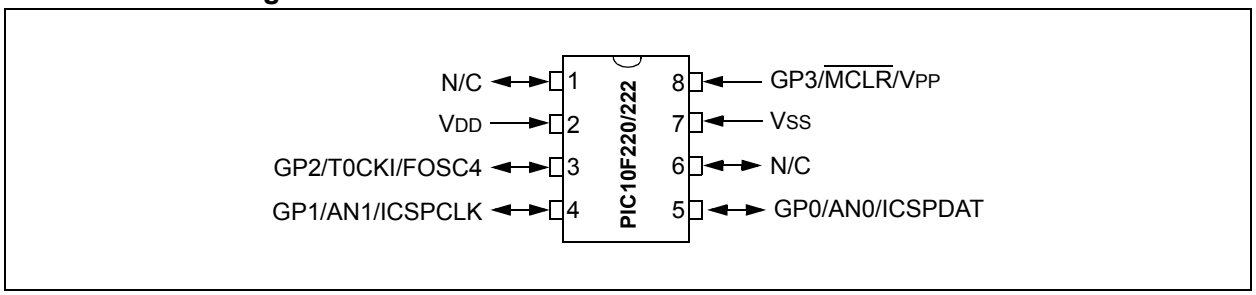
- 4 I/O Pins:
  - 3 I/O pins with individual direction control
  - 1 input only pin
  - High current sink/source for direct LED drive
  - Wake-on-change
  - Weak pull-ups
- 8-bit Real-Time Clock/Counter (TMR0) with 8-bit Programmable Prescaler
- Analog-to-Digital (A/D) Converter:
  - 8-bit resolution
  - 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

6-Lead SOT-23 Pin Diagram



8-Lead DIP Pin Diagram



8-Lead DFN Pin Diagram

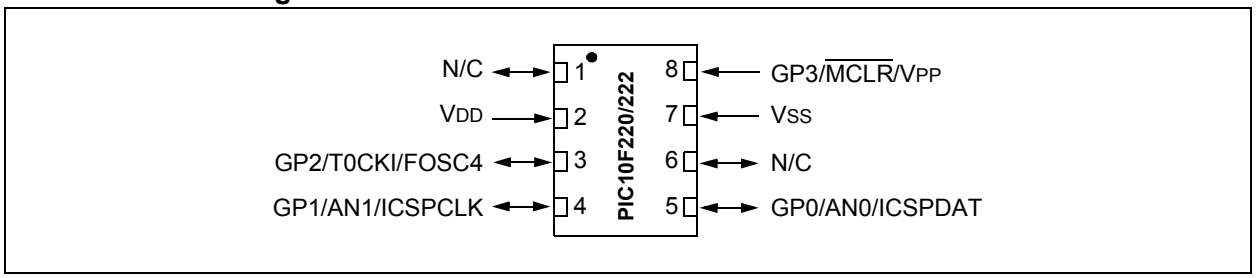


Table of Contents

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, DFN	6-pin SOT-23, 8-pin DIP, DFN

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

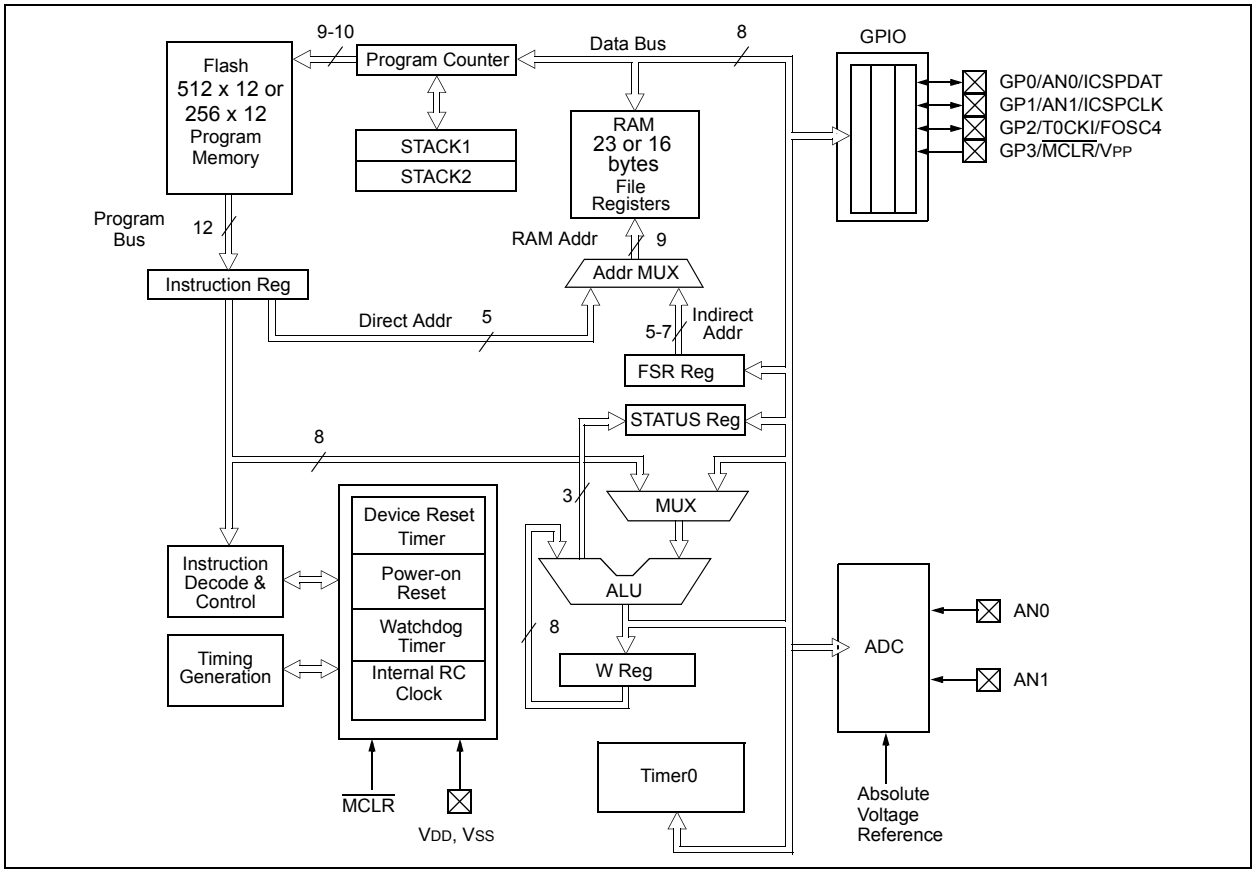


TABLE 3-1: PINOUT DESCRIPTION

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.
	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.

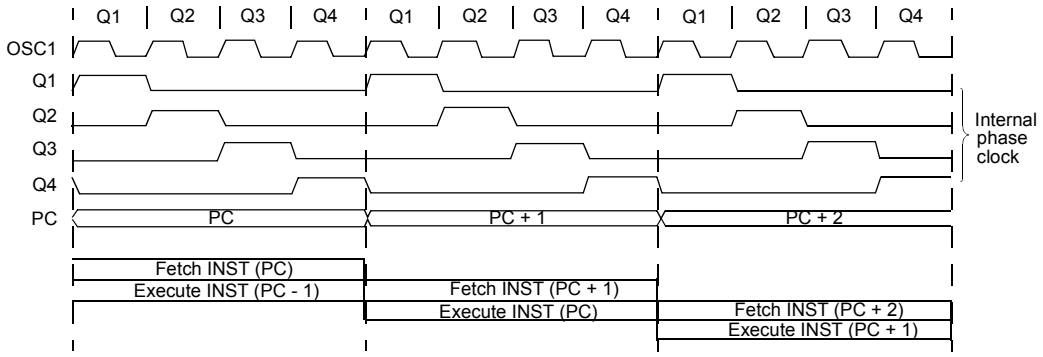
3.2 Instruction Flow/Pipelining

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).

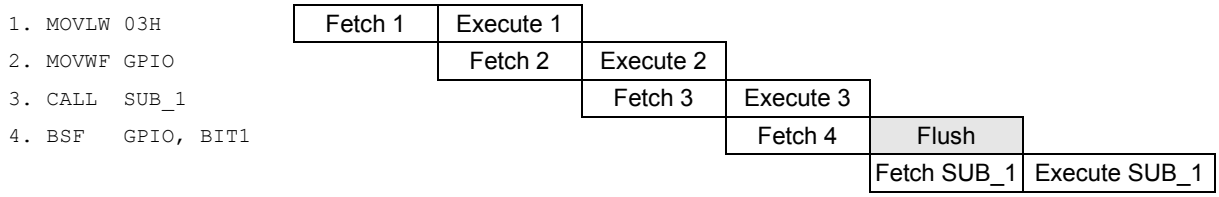
A fetch cycle begins with the PC incrementing in Q1.

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

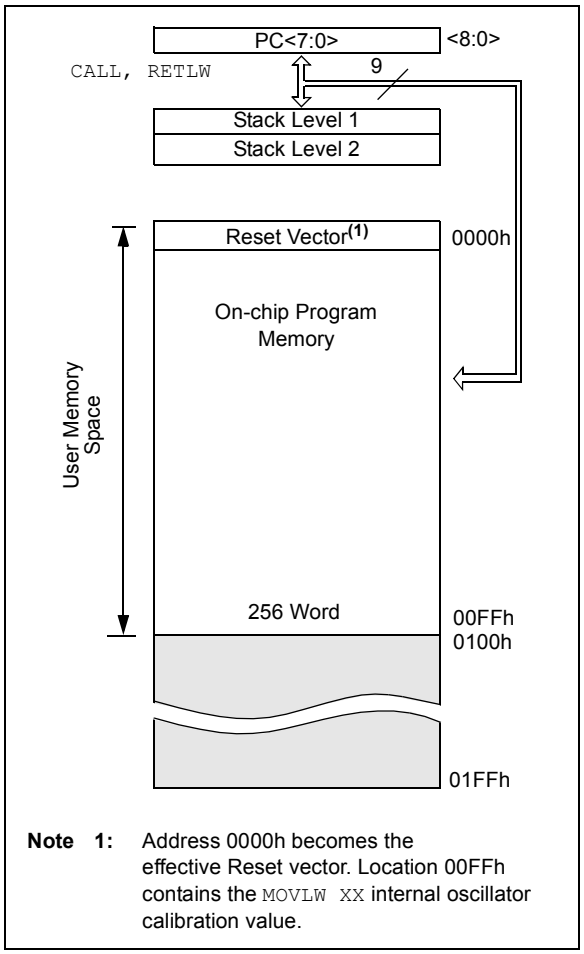
The PIC10F220/222 memories are organized into program memory and data memory. Data memory banks are accessed using the File Select Register (FSR).

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



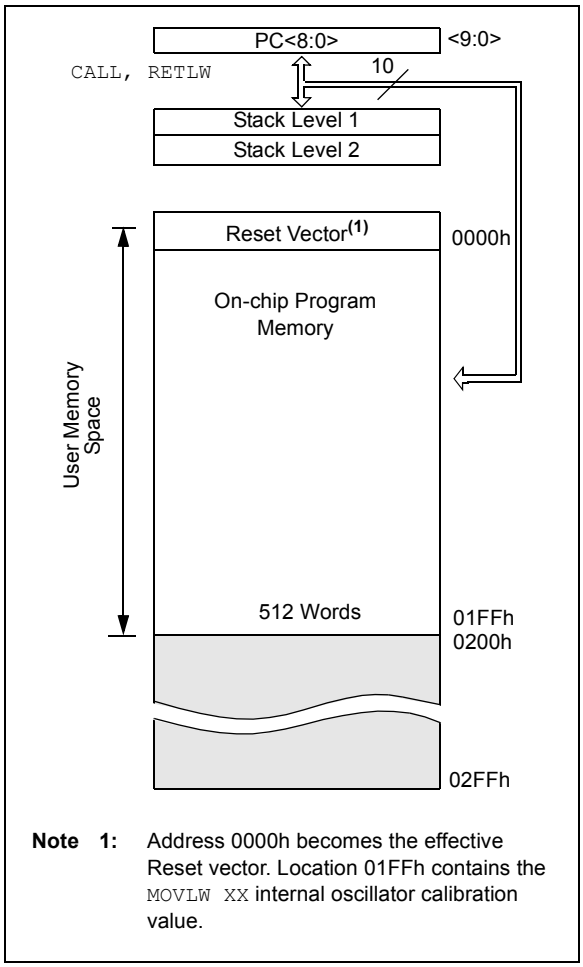
Note 1: Address 0000h becomes the effective Reset vector. Location 00FFh contains the `MOVLW XX` internal oscillator calibration value.

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.

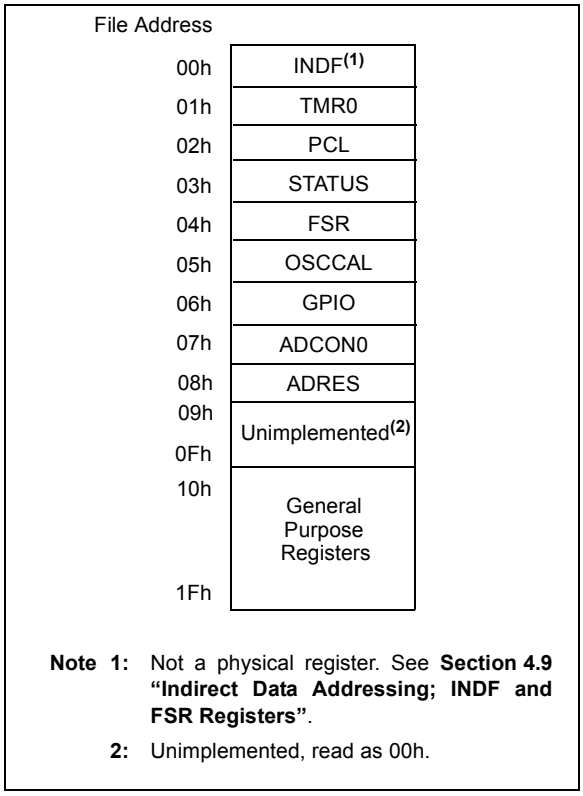
For the PIC10F220, the register file is composed of 9 Special Function Registers and 16 General Purpose Registers (Figure 4-3, Figure 4-4).

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"**.

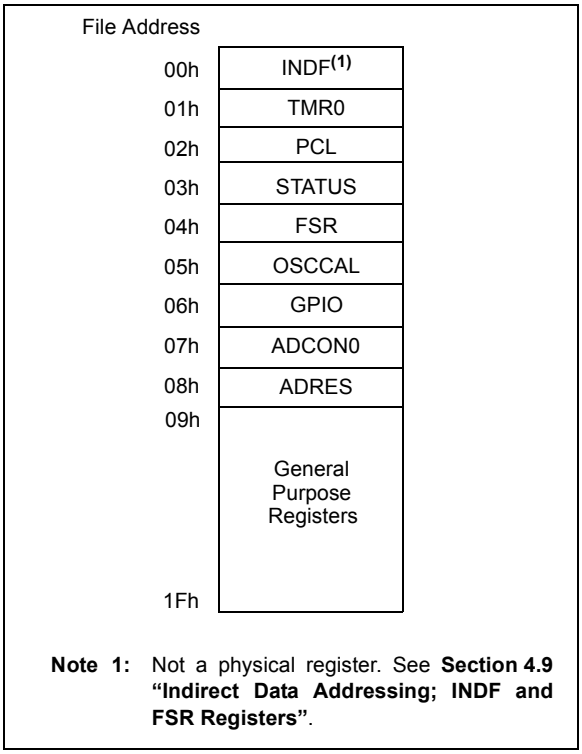
FIGURE 4-3: PIC10F220 REGISTER FILE MAP



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 GPPU and GPWU).

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

REGISTER 4-2: OPTION REGISTER

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

bit 6 **$\overline{\text{GPPU}}$:** 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 – OSCILLATOR CALIBRATION 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.

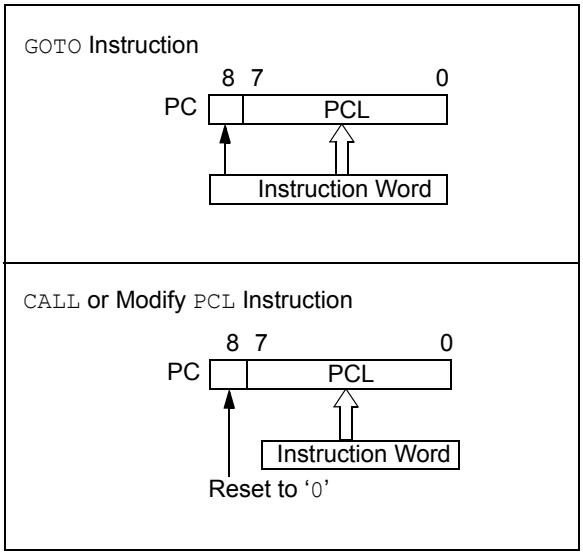
For a GOTO instruction, bits 8:0 of the PC are provided by the GOTO instruction word. The PC Latch (PCL) is mapped to PC<7:0>.

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

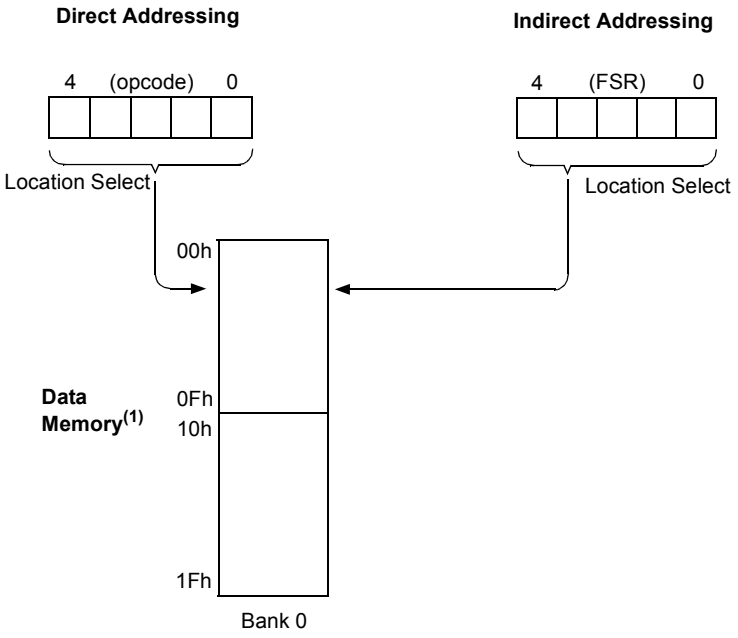
```
        MOVLW   0x10    ;initialize pointer
        MOVWF   FSR     ;to RAM
NEXT    CLRF    INDF    ;clear INDF
                        ;register
        INCF    FSR,F   ;inc pointer
        BTFSC   FSR,4   ;all done?
        GOTO    NEXT    ;NO, clear next
CONTINUE
            :           ;YES, 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/$\overline{\text{MCLR}}$ is configured as $\overline{\text{MCLR}}$, a weak pull-up can be enabled via the Configuration Word. Configuring GP3 as $\overline{\text{MCLR}}$ 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.

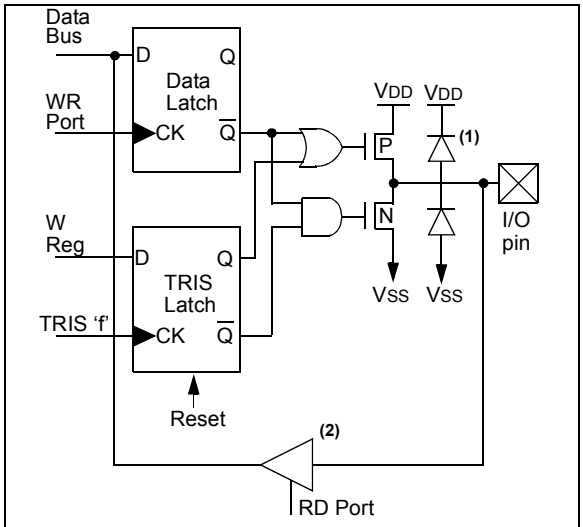
Note:	A read of the ports reads the pins, not the output data latches. That is, if an output driver on a pin is enabled and driven high, but the external system is holding it low, a read of the port will indicate that the pin is low.

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 5-1. All port pins, except GP3, which is input only, may be used for both input and output operations. For input operations, these ports are non-latching. Any input must be present until read by an input instruction (e.g., `MOVF GPIO, W`). The outputs are latched and remain unchanged until the output latch is rewritten. To use a port pin as output, the corresponding direction control bit in TRIS must be cleared (= 0). For use as an input, the corresponding TRIS bit must be set. Any I/O pin (except GP3) can be programmed individually as input or output.

FIGURE 5-1: EQUIVALENT CIRCUIT FOR A SINGLE I/O PIN



Note 1: I/O pins have protection diodes to V_{DD} and V_{SS}.

2: See Table 3-1 for buffer type.

TABLE 5-1: ORDER OF PRECEDENCE FOR PIN FUNCTIONS

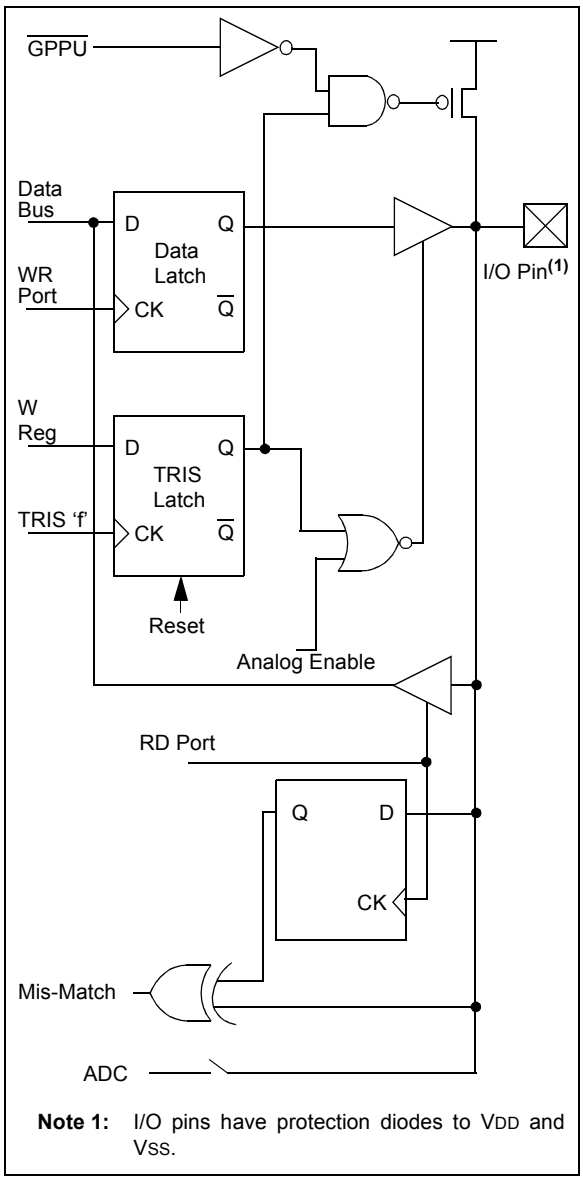
Priority	GP0	GP1	GP2	GP3
1	AN0	AN1	FOSC4	$\overline{\text{MCLR}}$
2	TRIS GPIO	TRIS GPIO	T0CKI	—
3	—	—	TRIS GPIO	—

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

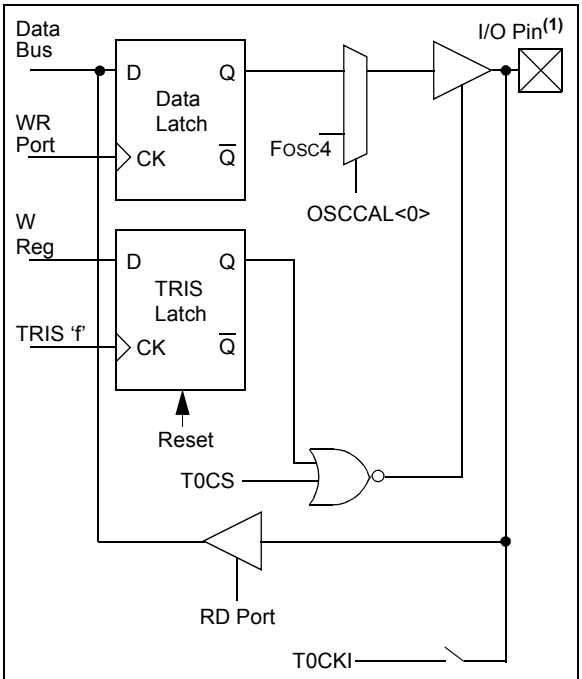
Legend: — = Condition of bit will have no effect on the setting of the pin to Digital mode.

FIGURE 5-2: BLOCK DIAGRAM OF GP0 AND GP1



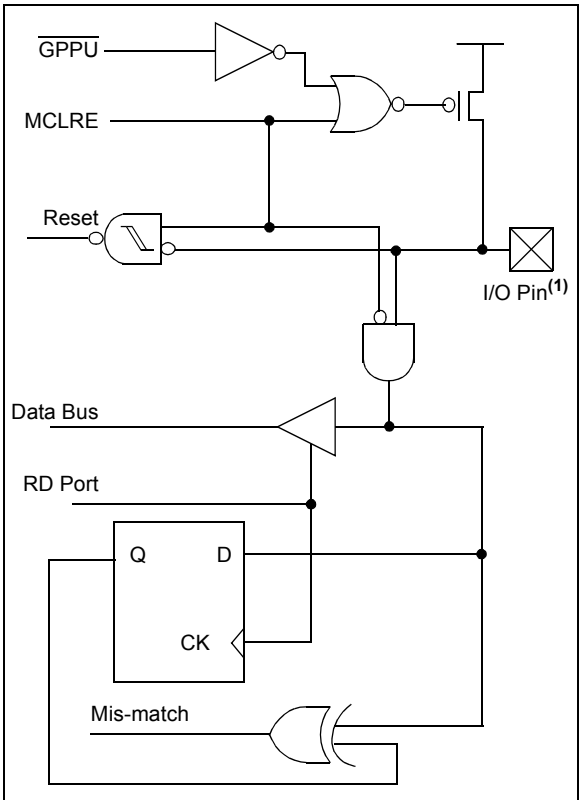
Note 1: I/O pins have protection diodes to VDD and VSS.

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/$\overline{MCLR}$ 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	$\overline{GPWU}$	$\overline{GPPU}$	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
03h	STATUS	GPWUF	—	—	$\overline{TO}$	$\overline{PD}$	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

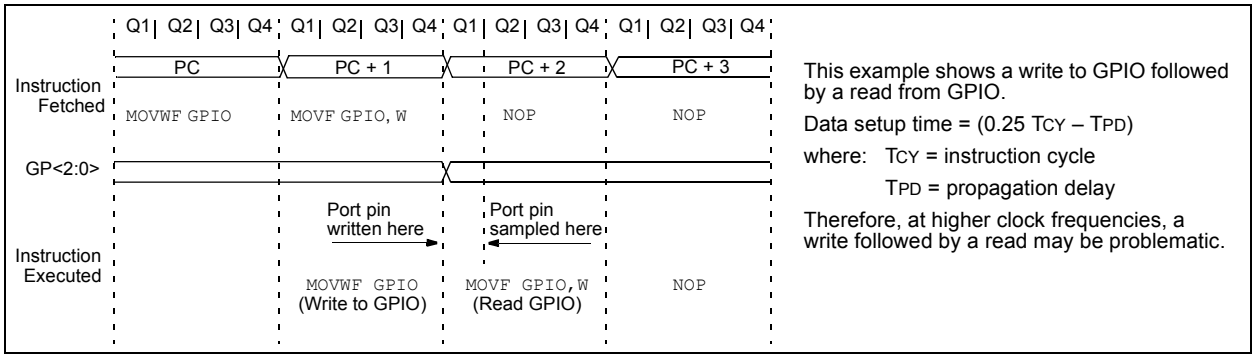
```
;Initial GPIO Settings
;GPIO<3:2> Inputs
;GPIO<1:0> Outputs
;
;                 GPIO latch   GPIO pins
;                 ----------   ----------
 BCF    GPIO,  1 ;---- pp01    ---- pp11
 BCF    GPIO,  0 ;---- pp10    ---- pp11
 MOVLW  007h;
 TRIS   GPIO     ;---- pp10    ---- pp11
;
```

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.