PIC16F5X
Data Sheet

Flash-Based, 8-Bit CMOS Microcontroller Series

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

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. 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.

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

PIC16F5X

Flash-Based, 8-Bit CMOS Microcontroller Series

High-Performance RISC CPU:

- Only 33 single-word instructions to learn
- All instructions are single cycle except for program branches which are two-cycle
- Two-level deep hardware stack
- Direct, Indirect and Relative Addressing modes for data and instructions
- Operating speed:
  - DC – 20 MHz clock speed
  - DC – 200 ns instruction cycle time
- On-chip Flash program memory:
  - 512 x 12 on PIC16F54
  - 2048 x 12 on PIC16F57
  - 2048 x 12 on PIC16F59
- General Purpose Registers (SRAM):
  - 25 x 8 on PIC16F54
  - 72 x 8 on PIC16F57
  - 134 x 8 on PIC16F59

Special Microcontroller Features:

- Power-on Reset (POR)
- Device Reset Timer (DRT)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable Code Protection
- Power-Saving Sleep mode
- In-Circuit Serial Programming™ (ICSP™)
- Selectable oscillator options:
  - RC: Low-cost RC oscillator
  - XT: Standard crystal/resonator
  - HS: High-speed crystal/resonator
  - LP: Power-saving, low-frequency crystal
- Packages:
  - 18-pin PDIP and SOIC for PIC16F54
  - 20-pin SSOP for PIC16F54
  - 28-pin PDIP, SOIC and SSOP for PIC16F57
  - 40-pin PDIP for PIC16F59
  - 44-pin TQFP for PIC16F59

Low-Power Features:

- Operating Current:
  - 170 µA @ 2V, 4 MHz, typical
  - 15 µA @ 2V, 32 kHz, typical
- Standby Current:
  - 500 nA @ 2V, typical

Peripheral Features:

- 12/20/32 I/O pins:
  - Individual direction control
  - High current source/sink
- 8-bit real-time clock/counter (TMR0) with 8-bit programmable prescaler

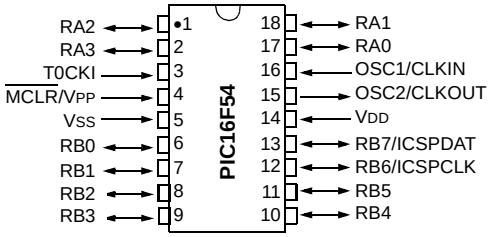
CMOS Technology:

- Wide operating voltage range:
  - Industrial: 2.0V to 5.5V
  - Extended: 2.0V to 5.5V
- Wide temperature range:
  - Industrial: -40°C to 85°C
  - Extended: -40°C to 125°C
- High-endurance Flash:
  - 100K write/erase cycles
  - > 40-year retention

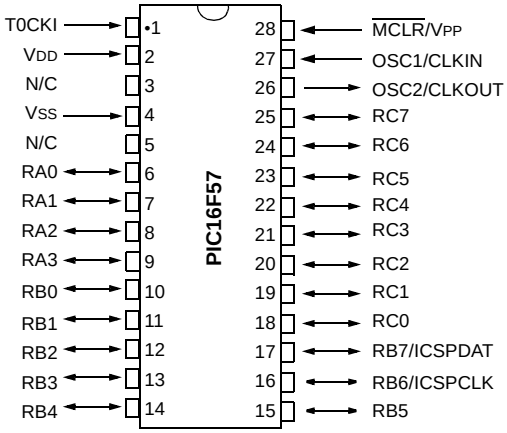
Device	Program Memory	Data Memory	I/O	Timers 8-bit
	Flash (words)	SRAM (bytes)		
PIC16F54	512	25	12	1
PIC16F57	2048	72	20	1
PIC16F59	2048	134	32	1

Pin Diagrams

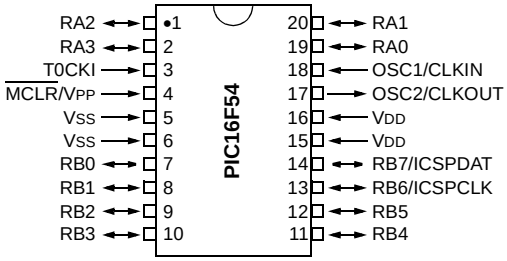
PDIP, SOIC



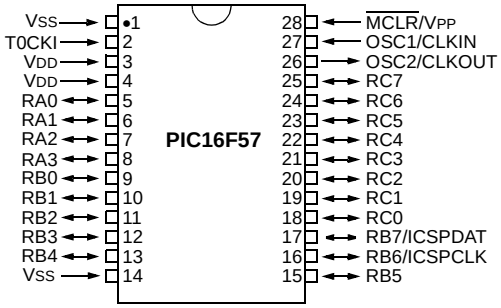
PDIP, SOIC



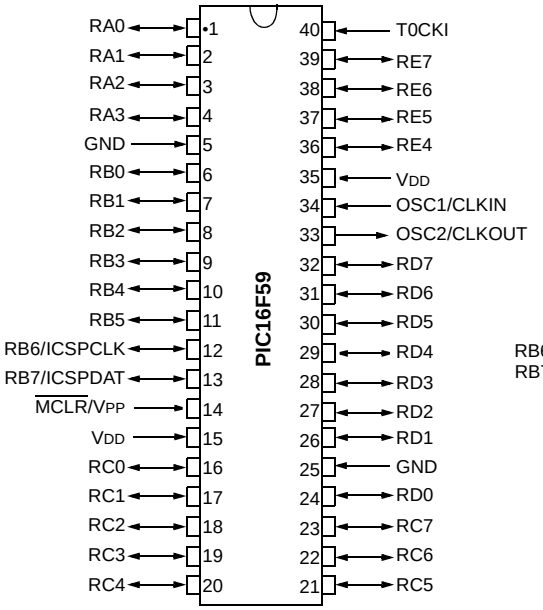
SSOP



SSOP



PDIP, 0.600"



TQFP

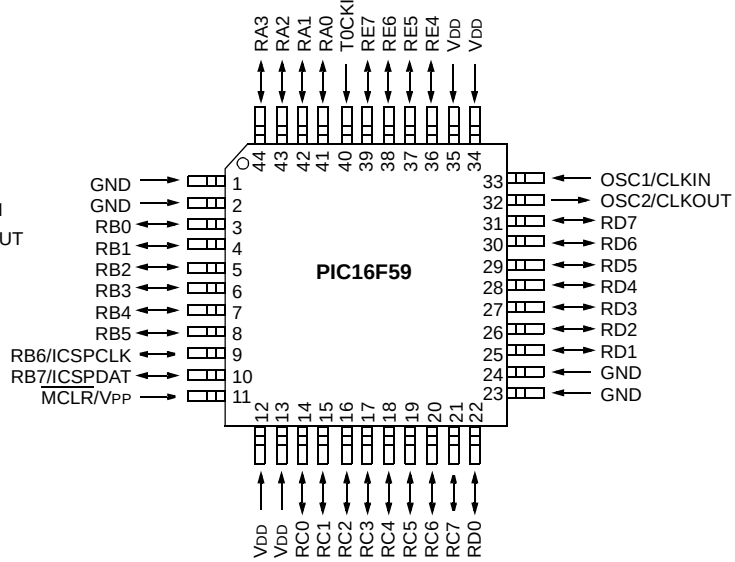


Table of Contents

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@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)

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

Customer Notification System

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

NOTES:

1.0 GENERAL DESCRIPTION

The PIC16F5X from Microchip Technology is a family of low-cost, high-performance, 8-bit, fully static, Flash-based CMOS microcontrollers. It employs a RISC architecture with only 33 single-word/single-cycle instructions. All instructions are single cycle except for program branches which take two cycles. The PIC16F5X delivers performance an order of magnitude higher than its competitors in the same price category. The 12-bit wide instructions are highly symmetrical resulting in 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 PIC16F5X products are equipped with special features that reduce system cost and power requirements. The Power-on Reset (POR) and Device Reset Timer (DRT) eliminate the need for external Reset circuitry. There are four oscillator configurations to choose from, including the power-saving LP (Low Power) oscillator and cost saving RC oscillator. Power-saving Sleep mode, Watchdog Timer and code protection features improve system cost, power and reliability.

The PIC16F5X products are supported by a full-featured macro assembler, a software simulator, 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 PIC16F5X series fits perfectly in applications ranging from high-speed automotive and appliance motor control to low-power remote transmitters/receivers, pointing devices and telecom processors. The Flash technology makes customizing application programs (transmitter codes, motor speeds, receiver frequencies, etc.) extremely fast and convenient. The small footprint packages, for through hole or surface mounting, make this microcontroller series perfect for applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC16F5X series very versatile, even in areas where no microcontroller use has been considered before (e.g., timer functions, replacement of “glue” logic in larger systems, co-processor applications).

TABLE 1-1: PIC16F5X FAMILY OF DEVICES

Features	PIC16F54	PIC16F57	PIC16F59
Maximum Operation Frequency	20 MHz	20 MHz	20 MHz
Flash Program Memory (x12 words)	512	2K	2K
RAM Data Memory (bytes)	25	72	134
Timer Module(s)	TMR0	TMR0	TMR0
I/O Pins	12	20	32
Number of Instructions	33	33	33
Packages	18-pin DIP, SOIC; 20-pin SSOP	28-pin DIP, SOIC; 28-pin SSOP	40-pin DIP, 44-pin TQFP

Note: All PIC® Family devices have Power-on Reset, selectable Watchdog Timer, selectable code-protect and high I/O current capability.

NOTES:

2.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16F5X family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16F5X uses a Harvard architecture in which program and data are accessed on separate buses. This improves bandwidth over traditional von Neumann architecture 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 except for program branches.

The PIC16F54 addresses 512 x 12 of program memory, the PIC16F57 and PIC16F59 addresses 2048 x 12 of program memory. All program memory is internal.

The PIC16F5X can directly or indirectly address its register files and data memory. All Special Function Registers (SFR), including the program counter, are mapped in the data memory. The PIC16F5X has 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 PIC16F5X simple, yet efficient. In addition, the learning curve is reduced significantly.

The PIC16F5X device contains 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, typically one operand is 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 2-1 with the corresponding device pins described in Table 2-1 (for PIC16F54), Table 2-2 (for PIC16F57) and Table 2-3 (for PIC16F59).

FIGURE 2-1: PIC16F5X SERIES BLOCK DIAGRAM

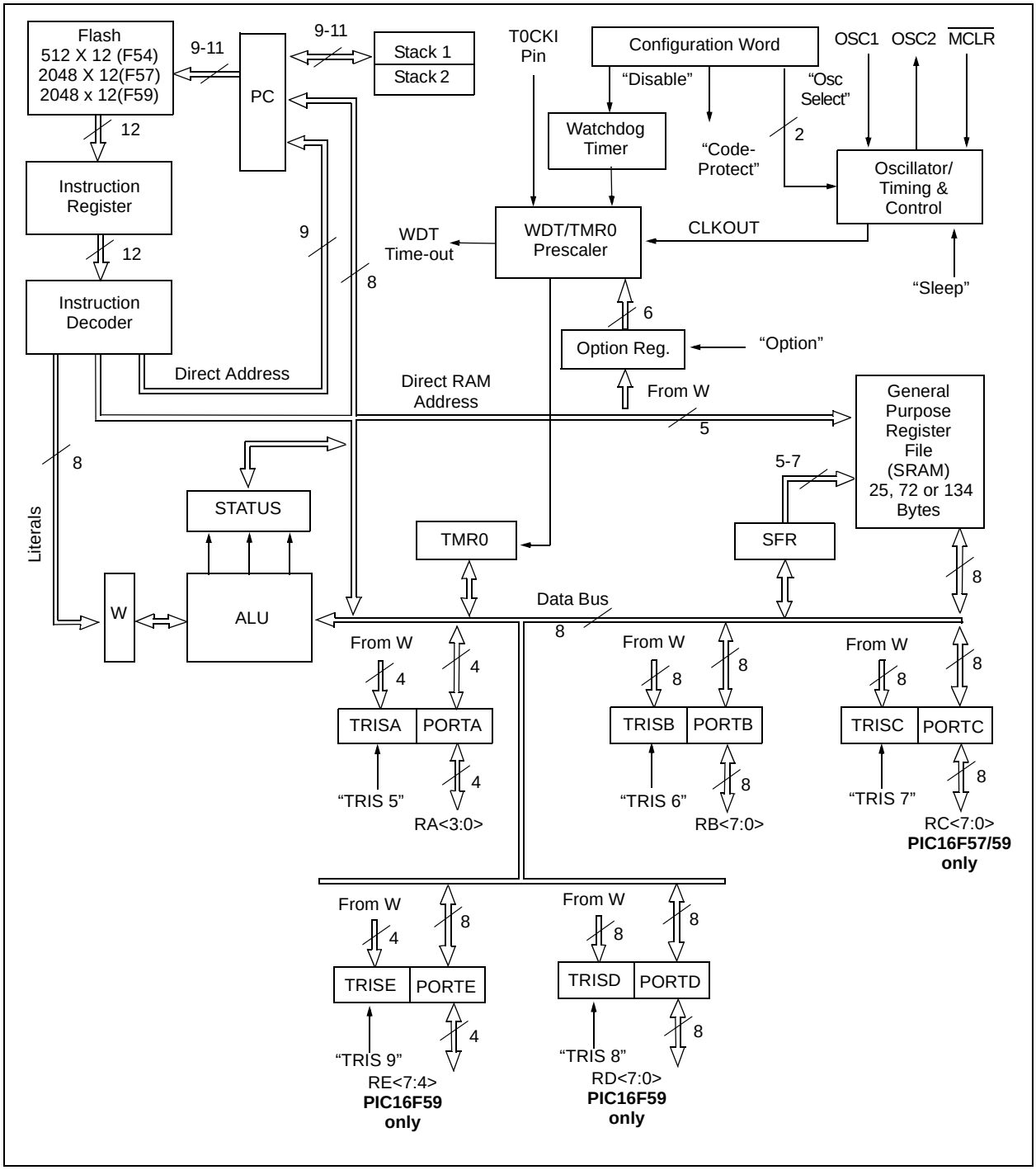


TABLE 2-1: PIC16F54 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0	RA0	TTL	CMOS	Bidirectional I/O pin
RA1	RA1	TTL	CMOS	Bidirectional I/O pin
RA2	RA2	TTL	CMOS	Bidirectional I/O pin
RA3	RA3	TTL	CMOS	Bidirectional I/O pin
RB0	RB0	TTL	CMOS	Bidirectional I/O pin
RB1	RB1	TTL	CMOS	Bidirectional I/O pin
RB2	RB2	TTL	CMOS	Bidirectional I/O pin
RB3	RB3	TTL	CMOS	Bidirectional I/O pin
RB4	RB4	TTL	CMOS	Bidirectional I/O pin
RB5	RB5	TTL	CMOS	Bidirectional I/O pin
RB6/ICSPCLK	RB6	TTL	CMOS	Bidirectional I/O pin
	ICSPCLK	ST	—	Serial Programming Clock
RB7/ICSPDAT	RB7	TTL	CMOS	Bidirectional I/O pin
	ICSPDAT	ST	CMOS	Serial Programming I/O
T0CKI	T0CKI	ST	—	Clock input to Timer0. Must be tied to VSS or VDD, if not in use, to reduce current consumption.
$\overline{MCLR}$/VPP	$\overline{MCLR}$	ST	—	Active-low Reset to device. Voltage on the $\overline{MCLR}$/VPP pin must not exceed VDD to avoid unintended entering of Programming mode.
	VPP	HV	—	Programming voltage input
OSC1/CLKIN	OSC1	XTAL	—	Oscillator crystal input
	CLKIN	ST	—	External clock source input
OSC2/CLKOUT	OSC2	—	XTAL	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
	CLKOUT	—	CMOS	In RC mode, OSC2 pin can output CLKOUT, which has 1/4 the frequency of OSC1.
VDD	VDD	Power	—	Positive supply for logic and I/O pins
VSS	VSS	Power	—	Ground reference for logic and I/O pins

Legend: I = input, O = output, ST = Schmitt Trigger input, I/O = input/output, — = Not Used, TTL = TTL input, CMOS = CMOS output, XTAL = Crystal input/output, HV = High Voltage

TABLE 2-2: PIC16F57 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0	RA0	TTL	CMOS	Bidirectional I/O pin
RA1	RA1	TTL	CMOS	Bidirectional I/O pin
RA2	RA2	TTL	CMOS	Bidirectional I/O pin
RA3	RA3	TTL	CMOS	Bidirectional I/O pin
RB0	RB0	TTL	CMOS	Bidirectional I/O pin
RB1	RB1	TTL	CMOS	Bidirectional I/O pin
RB2	RB2	TTL	CMOS	Bidirectional I/O pin
RB3	RB3	TTL	CMOS	Bidirectional I/O pin
RB4	RB4	TTL	CMOS	Bidirectional I/O pin
RB5	RB5	TTL	CMOS	Bidirectional I/O pin
RB6/ICSPCLK	RB6	TTL	CMOS	Bidirectional I/O pin
	ICSPCLK	ST	—	Serial programming clock
RB7/ICSPDAT	RB7	TTL	CMOS	Bidirectional I/O pin
	ICSPDAT	ST	CMOS	Serial programming I/O
RC0	RC0	TTL	CMOS	Bidirectional I/O pin
RC1	RC1	TTL	CMOS	Bidirectional I/O pin
RC2	RC2	TTL	CMOS	Bidirectional I/O pin
RC3	RC3	TTL	CMOS	Bidirectional I/O pin
RC4	RC4	TTL	CMOS	Bidirectional I/O pin
RC5	RC5	TTL	CMOS	Bidirectional I/O pin
RC6	RC6	TTL	CMOS	Bidirectional I/O pin
RC7	RC7	TTL	CMOS	Bidirectional I/O pin
T0CKI	T0CKI	ST	—	Clock input to Timer0. Must be tied to V_{SS} or V_{DD}, if not in use, to reduce current consumption.
$\overline{MCLR}/V_{PP}$	$\overline{MCLR}$	ST	—	Active-low Reset to device. Voltage on the $\overline{MCLR}/V_{PP}$ pin must not exceed V_{DD} to avoid unintended entering of Programming mode.
	V_{PP}	HV	—	Programming voltage input
OSC1/CLKIN	OSC1	XTAL	—	Oscillator crystal input
	CLKIN	ST	—	External clock source input
OSC2/CLKOUT	OSC2	—	XTAL	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
	CLKOUT	—	CMOS	In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1.
V_{DD}	V_{DD}	Power	—	Positive supply for logic and I/O pins
V_{SS}	V_{SS}	Power	—	Ground reference for logic and I/O pins
N/C	N/C	—	—	Unused, do not connect

Legend: I = input, O = output, ST = Schmitt Trigger input, I/O = input/output, — = Not Used, TTL = TTL input, CMOS = CMOS output, XTAL = Crystal input/output, HV = High Voltage

TABLE 2-3: PIC16F59 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0	RA0	TTL	CMOS	Bidirectional I/O pin
RA1	RA1	TTL	CMOS	Bidirectional I/O pin
RA2	RA2	TTL	CMOS	Bidirectional I/O pin
RA3	RA3	TTL	CMOS	Bidirectional I/O pin
RB0	RB0	TTL	CMOS	Bidirectional I/O pin
RB1	RB1	TTL	CMOS	Bidirectional I/O pin
RB2	RB2	TTL	CMOS	Bidirectional I/O pin
RB3	RB3	TTL	CMOS	Bidirectional I/O pin
RB4	RB4	TTL	CMOS	Bidirectional I/O pin
RB5	RB5	TTL	CMOS	Bidirectional I/O pin
RB6/ICSPCLK	RB6	TTL	CMOS	Bidirectional I/O pin
	ICSPCLK	ST	—	Serial programming clock
RB7/ICSPDAT	RB7	TTL	CMOS	Bidirectional I/O pin
	ICSPDAT	ST	CMOS	Serial programming I/O
RC0	RC0	TTL	CMOS	Bidirectional I/O pin
RC1	RC1	TTL	CMOS	Bidirectional I/O pin
RC2	RC2	TTL	CMOS	Bidirectional I/O pin
RC3	RC3	TTL	CMOS	Bidirectional I/O pin
RC4	RC4	TTL	CMOS	Bidirectional I/O pin
RC5	RC5	TTL	CMOS	Bidirectional I/O pin
RC6	RC6	TTL	CMOS	Bidirectional I/O pin
RC7	RC7	TTL	CMOS	Bidirectional I/O pin
RD0	RD0	TTL	CMOS	Bidirectional I/O pin
RD1	RD1	TTL	CMOS	Bidirectional I/O pin
RD2	RD2	TTL	CMOS	Bidirectional I/O pin
RD3	RD3	TTL	CMOS	Bidirectional I/O pin
RD4	RD4	TTL	CMOS	Bidirectional I/O pin
RD5	RD5	TTL	CMOS	Bidirectional I/O pin
RD6	RD6	TTL	CMOS	Bidirectional I/O pin
RD7	RD7	TTL	CMOS	Bidirectional I/O pin
RE4	RE4	TTL	CMOS	Bidirectional I/O pin
RE5	RE5	TTL	CMOS	Bidirectional I/O pin
RE6	RE6	TTL	CMOS	Bidirectional I/O pin
RE7	RE7	TTL	CMOS	Bidirectional I/O pin
T0CKI	T0CKI	ST	—	Clock input to Timer0. Must be tied to VSS or VDD, if not in use, to reduce current consumption.
$\overline{\text{MCLR}}$/VPP	$\overline{\text{MCLR}}$	ST	—	Active-low Reset to device. Voltage on the $\overline{\text{MCLR}}$/VPP pin must not exceed VDD to avoid unintended entering of Programming mode.
	VPP	HV	—	Programming voltage input
OSC1/CLKIN	OSC1	XTAL	—	Oscillator crystal input
	CLKIN	ST	—	External clock source input
OSC2/CLKOUT	OSC2	—	XTAL	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
	CLKOUT	—	CMOS	In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1.
VDD	VDD	Power	—	Positive supply for logic and I/O pins
VSS	VSS	Power	—	Ground reference for logic and I/O pins

Legend: I = input, O = output, ST = Schmitt Trigger input; I/O = input/output, — = Not Used, TTL = TTL input; CMOS = CMOS output, XTAL = Crystal input/output, HV = High Voltage

2.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN pin) is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. Internally, the Program Counter (PC) is incremented every Q1 and the instruction is fetched from program memory and latched into the instruction register in Q4. It is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 2-2 and Example 2-1.

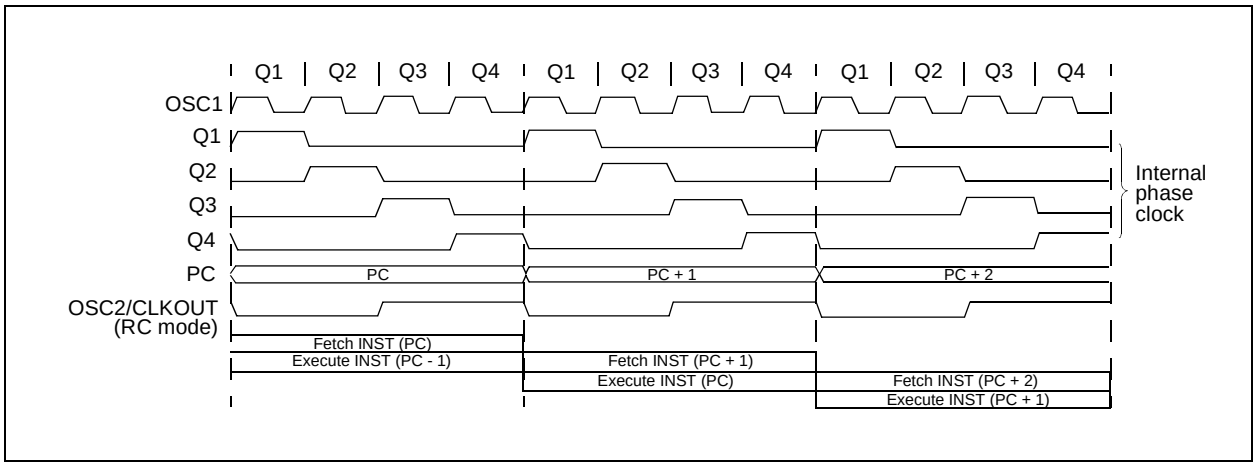
2.2 Instruction Flow/Pipelining

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 Program Counter to change (e.g., `GOTO`), then two cycles are required to complete the instruction (Example 2-1).

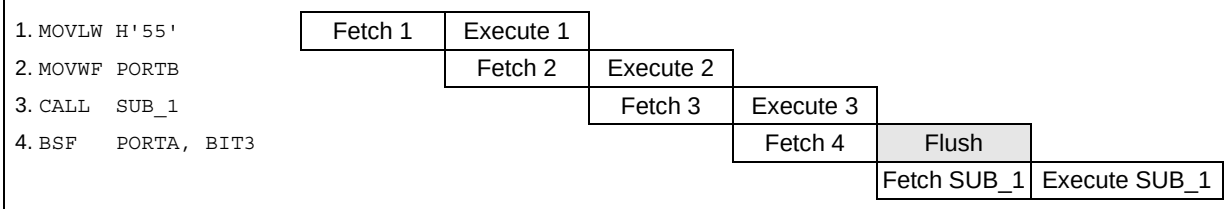
A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

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 2-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 2-1: 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.

3.0 MEMORY ORGANIZATION

PIC16F5X memory is organized into program memory and data memory. For the PIC16F57 and PIC16F59, which have more than 512 words of program memory, a paging scheme is used. Program memory pages are accessed using one or two STATUS register bits. For the PIC16F57 and PIC16F59, which have a data memory register file of more than 32 registers, a banking scheme is used. Data memory banks are accessed using the File Selection Register (FSR).

3.1 Program Memory Organization

The PIC16F54 has a 9-bit Program Counter (PC) capable of addressing a 512 x 12 program memory space (Figure 3-1). The PIC16F57 and PIC16F59 have an 11-bit Program Counter capable of addressing a 2K x 12 program memory space (Figure 3-2). Accessing a location above the physically implemented address will cause a wraparound.

A NOP at the Reset vector location will cause a restart at location 000h. The Reset vector for the PIC16F54 is at 1FFh. The Reset vector for the PIC16F57 and PIC16F59 is at 7FFh. See **Section 3.5 "Program Counter"** for additional information using CALL and GOTO instructions.

FIGURE 3-1: PIC16F54 PROGRAM MEMORY MAP AND STACK

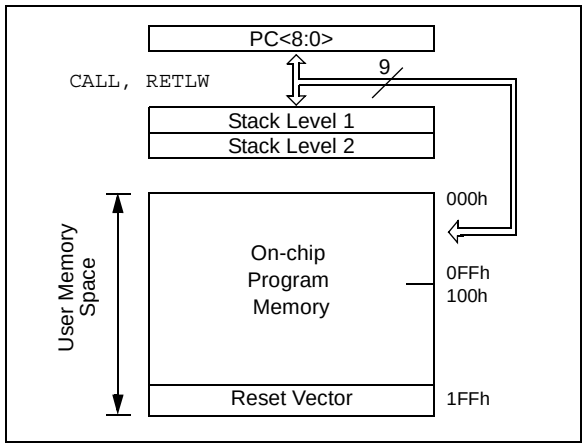
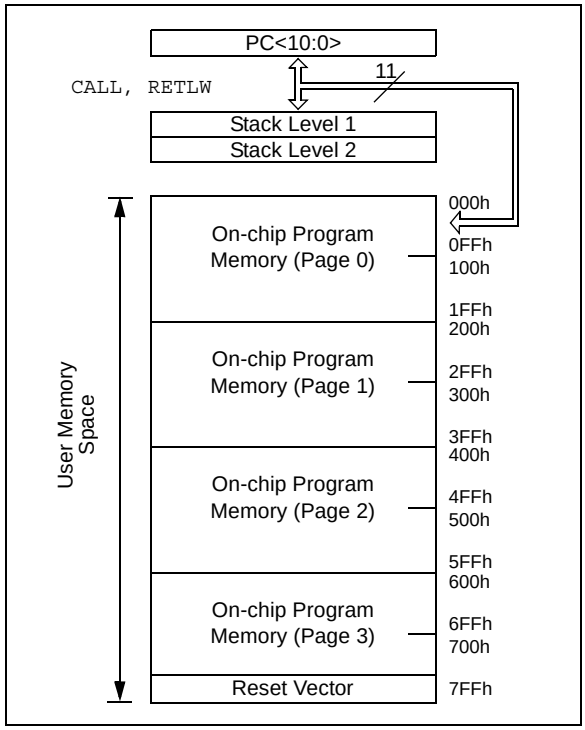


FIGURE 3-2: PIC16F57/PIC16F59 PROGRAM MEMORY MAP AND STACK



3.2 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 (PC), the STATUS register, the I/O registers (ports) and the File Select Register (FSR). In addition, Special Purpose 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 PIC16F54, the register file is composed of 7 Special Function Registers and 25 General Purpose Registers (Figure 3-3).

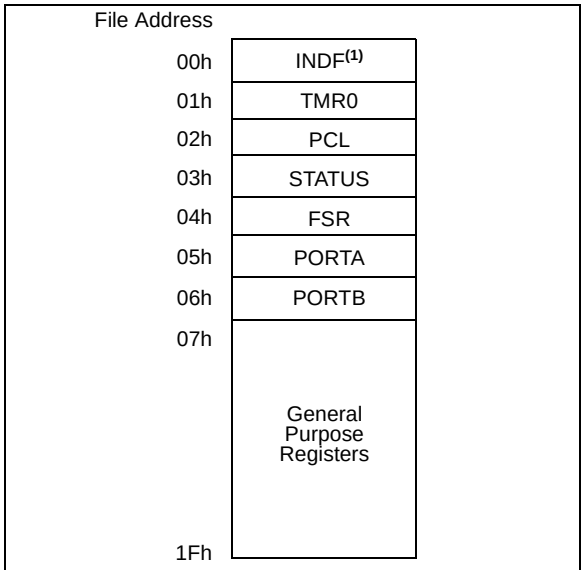
For the PIC16F57, the register file is composed of 8 Special Function Registers, 8 General Purpose Registers and 64 additional General Purpose Registers that may be addressed using a banking scheme (Figure 3-4).

For the PIC16F59, the register file is composed of 10 Special Function Registers, 6 General Purpose Registers and 128 additional General Purpose Registers that may be addressed using a banking scheme (Figure 3-5).

3.2.1 GENERAL PURPOSE REGISTER FILE

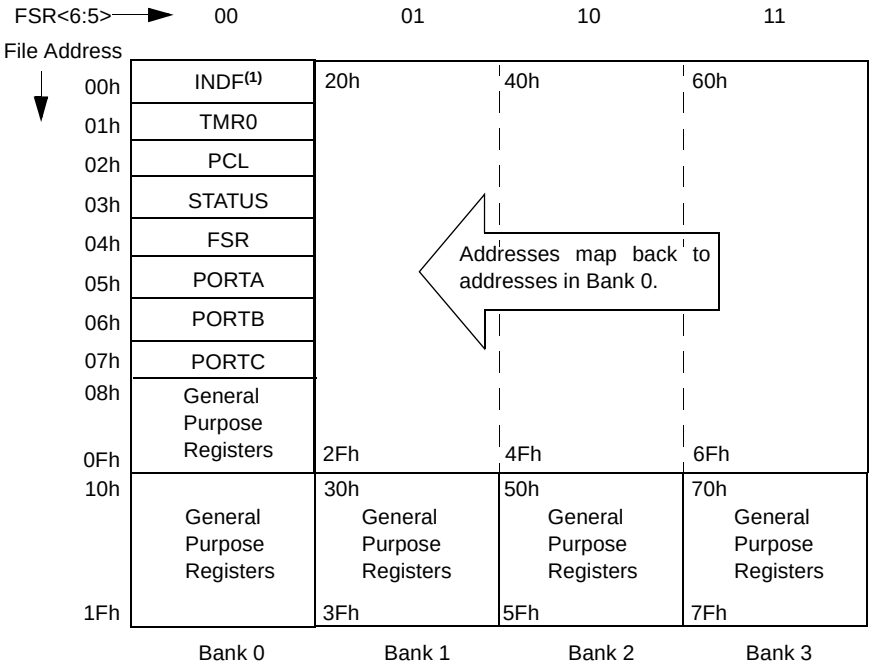
The register file is accessed either directly or indirectly through the File Select Register (FSR). The FSR register is described in **Section 3.7 "Indirect Data Addressing; INDF and FSR Registers"**.

FIGURE 3-3: PIC16F54 REGISTER FILE MAP



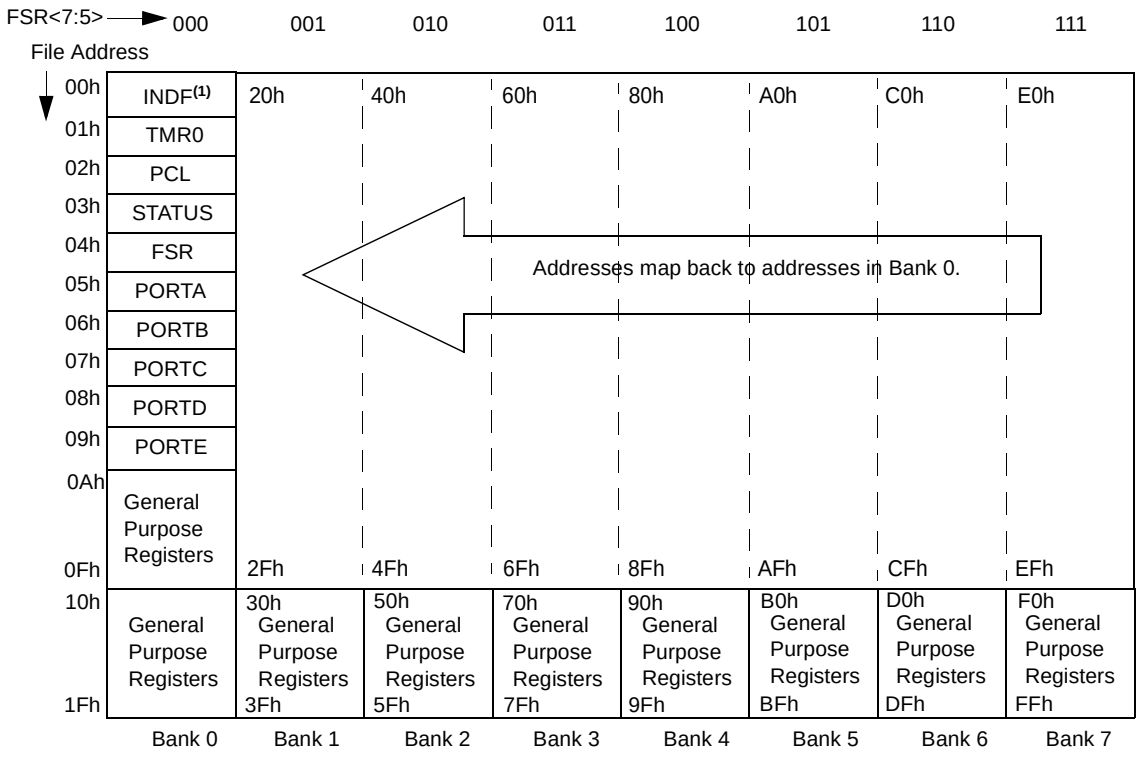
Note 1: Not a physical register. See **Section 3.7 "Indirect Data Addressing; INDF and FSR Registers"**.

FIGURE 3-4: PIC16F57 REGISTER FILE MAP



Note 1: Not a physical register. See **Section 3.7 "Indirect Data Addressing; INDF and FSR Registers"**.

FIGURE 3-5: PIC16F59 REGISTER FILE MAP



Note 1: Not a physical register.

3.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFR) are registers used by the CPU and peripheral functions to control the operation of the device (Table 3-1).

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 3-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Details on Page
N/A	TRIS	I/O Control Registers (TRISA, TRISB, TRISC, TRISD, TRISE)								1111 1111	29
N/A	OPTION	Contains control bits to configure Timer0 and Timer0/WDT prescaler								--11 1111	18
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								xxxx xxxx	20
01h	TMR0	Timer0 Module Register								xxxx xxxx	34
02h	PCL(1)	Low order 8 bits of PC								1111 1111	19
03h	STATUS	PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	17
04h	FSR(3)	Indirect data memory Address Pointer								111x xxxx	20
04h	FSR(4)	Indirect data memory Address Pointer								1xxx xxxx	20
04h	FSR(5)	Indirect data memory Address Pointer								xxxx xxxx	20
05h	PORTA(6)	—	—	—	—	RA3	RA2	RA1	RA0	---- xxxx	29
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	29
07h	PORTC(2)	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	29
08h	PORTD(7)	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	29
09h	PORTE(6), (7)	RE7	RE6	RE5	RE4	—	—	—	—	xxxx ----	29

Legend: Shaded cells = unimplemented or unused, – = unimplemented, read as ‘0’ (if applicable), x = unknown, u = unchanged

Note 1: The upper byte of the Program Counter is not directly accessible. See **Section 3.5 “Program Counter”** for an explanation of how to access these bits.

2: File address 07h is a General Purpose Register on the PIC16F54.

3: PIC16F54 only.

4: PIC16F57 only.

5: PIC16F59 only.

6: Unimplemented bits are read as ‘0’s.

7: File address 08h and 09h are General Purpose Registers on the PIC16F54 and PIC16F57.

3.3 STATUS Register

This register contains the arithmetic status of the ALU, the Reset status and the page preselect bits for program memories larger than 512 words.

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`, `MOVWF` and `SWAPF` instructions be used to alter the STATUS register because these instructions do not affect the Z, DC or C bits from the STATUS register. For other instructions which do affect Status bits, see **Section 9.0 "Instruction Set Summary"**.

REGISTER 3-1: 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
PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7 **PA2**: Reserved, do not use
Use of the PA2 bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

bit 6-5 **PA<1:0>**: Program Page Preselect bits (PIC16F57/PIC16F59)
`00` = Page 0 (000h-1FFh)
`01` = Page 1 (200h-3FFh)
`10` = Page 2 (400h-5FFh)
`11` = Page 3 (600h-7FFh)
Each page is 512 words. Using the PA<1:0> bits as general purpose read/write bits in devices which do not use them for program page preselect is not recommended. This 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 from the 4th low order bit of the result did not occur
SUBWF
`1` = A borrow to 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	Loaded with LSb or MSb, respectively
`0` = A carry did not occur	`0` = A borrow occurred	

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

3.4 Option Register

The Option register is a 6-bit wide, write-only register which contains various control bits to configure the Timer0/WDT prescaler and Timer0.

By executing the OPTION instruction, the contents of the W register will be transferred to the Option register. A Reset sets the Option<5:0> bits.

REGISTER 3-2: OPTION REGISTER

U-0	U-0	W-1	W-1	W-1	W-1	W-1	W-1
—	—	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7-6 **Unimplemented**: Read as '0'

bit 5 **T0CS**: Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE**: Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on 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

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

3.5 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> (Figure 3-6 and Figure 3-7).

For the PIC16F57 and PIC16F59, a page number must be supplied as well. Bit 5 and bit 6 of the STATUS register provide page information to bit 9 and bit 10 of the PC (Figure 3-6 and Figure 3-7).

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 3-6 and Figure 3-7).

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

For the PIC16F57 and PIC16F59, a page number again must be supplied. Bit 5 and bit 6 of the STATUS register provide page information to bit 9 and bit 10 of the PC (Figure 3-6 and Figure 3-7).

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

FIGURE 3-6: LOADING OF PC BRANCH INSTRUCTIONS – PIC16F54

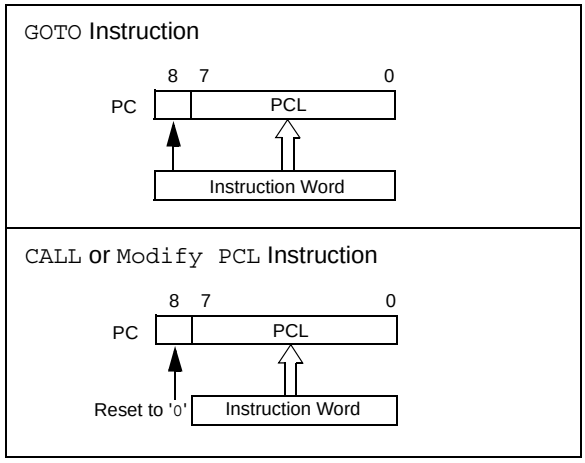
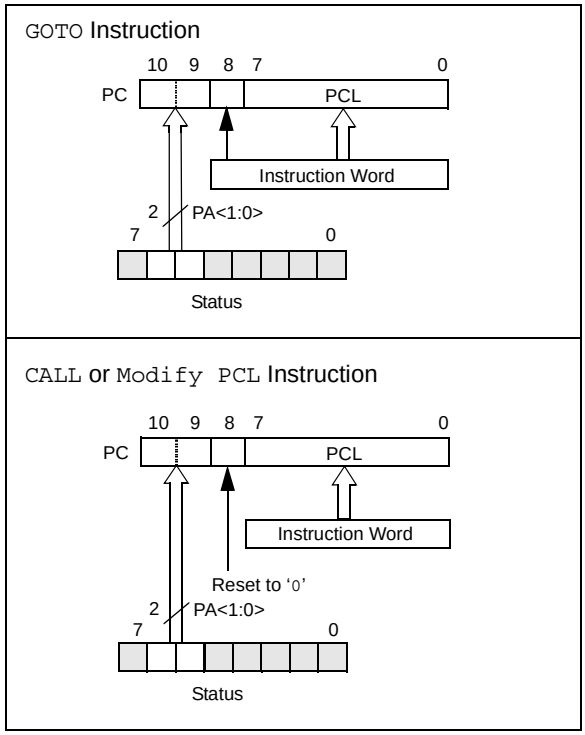


FIGURE 3-7: LOADING OF PC BRANCH INSTRUCTIONS – PIC16F57 AND PIC16F59



3.5.1 PAGING CONSIDERATIONS PIC16F57 AND PIC16F59

If the PC is pointing to the last address of a selected memory page, when it increments, it will cause the program to continue in the next higher page. However, the page preselect bits in the STATUS register will not be updated. Therefore, the next GOTO, CALL or MODIFY PCL instruction will send the program to the page specified by the page preselect bits (PA0 or PA<1:0>).

For example, a NOP at location 1FFh (page 0) increments the PC to 200h (page 1). A GOTO xxx at 200h will return the program to address xxh on page 0 (assuming that PA<1:0> are clear).

To prevent this, the page preselect bits must be updated under program control.

3.5.2 EFFECTS OF RESET

The PC is set upon a Reset, which means that the PC addresses the last location in the last page (i.e., the Reset vector).

The STATUS register page preselect bits are cleared upon a Reset, which means that page 0 is preselected.

Therefore, upon a Reset, a GOTO instruction at the Reset vector location will automatically cause the program to jump to page 0.

3.6 Stack

The PIC16F54 device has a 9-bit wide, two-level hardware PUSH/POP stack. The PIC16F57 and PIC16F59 devices have an 11-bit wide, two-level hardware PUSH/POP stack.

A CALL instruction will PUSH the current value of stack 1 into stack 2 and then PUSH the current program counter 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 program counter 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:	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.

For the RETLW instruction, the PC is loaded with the Top-of-Stack (TOS) contents. All of the devices covered in this data sheet have a two-level stack. The stack has the same bit width as the device PC, therefore, paging is not an issue when returning from a subroutine.

3.7 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.

EXAMPLE 3-1: INDIRECT ADDRESSING

- Register file 08 contains the value 10h
- Register file 09 contains the value 0Ah
- Load the value 08 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 = 09h)
- A read of the INDF 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 3-2.

EXAMPLE 3-2: HOW TO CLEAR RAM USING INDIRECT ADDRESSING

```
          MOVLW  H'10'  ;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 either a 5-bit (PIC16F54), 7-bit (PIC16F57) or 8-bit (PIC16F59) 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.

PIC16F54: This does not use banking. FSR<7:5> bits are unimplemented and read as '1's.

PIC16F57: FSR<7> bit is unimplemented and read as '1'. FSR<6:5> are the bank select bits and are used to select the bank to be addressed (00 = Bank 0, 01 = Bank 1, 10 = Bank 2, 11 = Bank 3).

PIC16F59: FSR<7:5> are the bank select bits and are used to select the bank to be addressed (000 = Bank 0, 001 = Bank 1, 010 = Bank 2, 011 = Bank 3, 100 = Bank 4, 101 = Bank 5, 110 = Bank 6, 111 = Bank 7).

Note:	A CLRF FSR instruction may not result in an FSR value of 00h if there are unimplemented bits present in the FSR.

4.0 OSCILLATOR CONFIGURATIONS

4.1 Oscillator Types

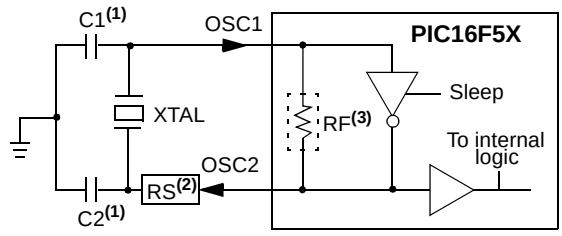
The PIC16F5X devices can be operated in four different oscillator modes. The user can program two Configuration bits (FOSC1:FOSC0) to select one of these four modes:

- LP: Low-power Crystal
- XT: Crystal/Resonator
- HS: High-speed Crystal/Resonator
- RC: Resistor/Capacitor

4.2 Crystal Oscillator/Ceramic Resonators

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 4-1). The PIC16F5X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency outside of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source drive the OSC1/CLKIN pin (Figure 4-2).

FIGURE 4-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



Note 1: See Capacitor Selection tables for recommended values of C1 and C2.
2: A series resistor (RS) may be required.
3: RF varies with the Oscillator mode chosen (approx. value = 10 MΩ).

FIGURE 4-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

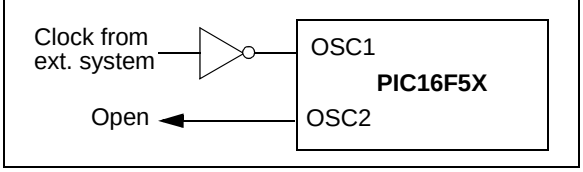


TABLE 4-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Osc Type	Resonator Freq.	Cap. Range C1	Cap. Range C2
XT	455 kHz 2.0 MHz 4.0 MHz	68-100 pF 15-33 pF 10-22 pF	68-100 pF 15-33 pF 10-22 pF
HS	8.0 MHz 16.0 MHz	10-22 pF 10 pF	10-22 pF 10 pF

These values are for design guidance only. Since each resonator has its own characteristics, the user should consult the resonator manufacturer for appropriate values of external components.

TABLE 4-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq.	Cap.Range C1	Cap. Range C2
LP	32 kHz(1)	15 pF	15 pF
XT	100 kHz 200 kHz 455 kHz 1 MHz 2 MHz 4 MHz	15-30 pF 15-30 pF 15-30 pF 15-30 pF 15 pF 15 pF	200-300 pF 100-200 pF 15-100 pF 15-30 pF 15 pF 15 pF
HS	4 MHz 8 MHz 20 MHz	15 pF 15 pF 15 pF	15 pF 15 pF 15 pF

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

These values are for design guidance only. Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specifications. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

Note 1: This device has been designed to perform to the parameters of its data sheet. It has been tested to an electrical specification designed to determine its conformance with these parameters. Due to process differences in the manufacture of this device, this device may have different performance characteristics than its earlier version. These differences may cause this device to perform differently in your application than the earlier version of this device.

2: The user should verify that the device oscillator starts and performs as expected. Adjusting the loading capacitor values and/or the Oscillator mode may be required.

4.3 External Crystal Oscillator Circuit

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

Figure 4-3 shows an implementation example of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180° phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This circuit could be used for external oscillator designs.

FIGURE 4-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT (USING XT, HS OR LP OSCILLATOR MODE)

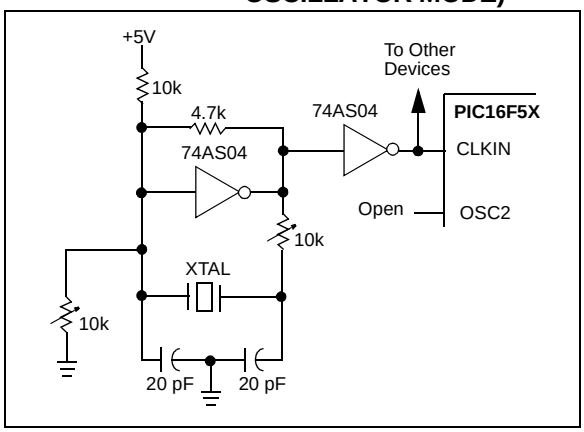
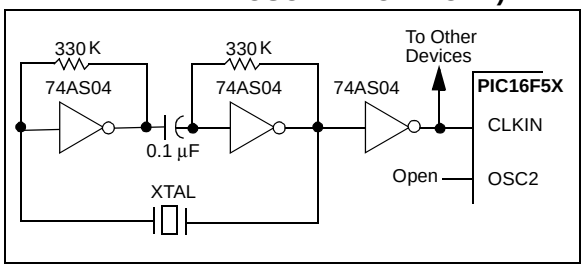


Figure 4-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverters perform a 360° phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 4-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT (USING XT, HS OR LP OSCILLATOR MODE)



4.4 RC Oscillator

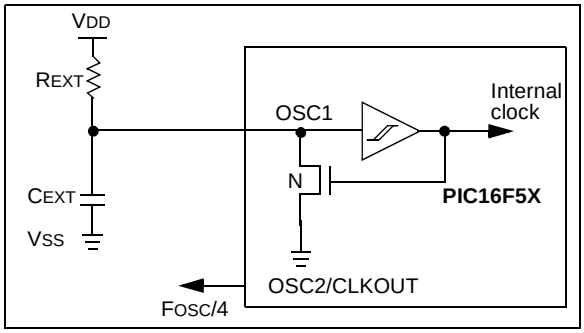
For applications where precise timing is not a requirement, the RC oscillator option is available. The operation and functionality of the RC oscillator is dependent upon a number of variables. The RC oscillator frequency is a function of:

- Supply voltage
- Resistor (REXT) and capacitor (CEXT) values
- Operating temperature.

The oscillator frequency will vary from unit to unit due to normal process parameter variation. The difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low CEXT values. The user also needs to account for the tolerance of the external R and C components. Figure 4-5 shows how the R/C combination is connected.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin and can be used for test purposes or to synchronize other logic.

FIGURE 4-5: RC OSCILLATOR MODE



5.0 RESET

The PIC16F5X devices may be reset in one of the following ways:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ Reset (normal operation)
- $\overline{\text{MCLR}}$ Wake-up Reset (from Sleep)
- WDT Reset (normal operation)
- WDT Wake-up Reset (from Sleep)

Table 5-1 shows these Reset conditions for the PCL and STATUS registers.

Some registers are not affected in any Reset condition. Their status is unknown on POR and unchanged in any other Reset. Most other registers are reset to a "Reset state" on Power-on Reset (POR), $\overline{\text{MCLR}}$ or WDT Reset. A $\overline{\text{MCLR}}$ or WDT wake-up from Sleep also results in a device Reset and not a continuation of operation before Sleep.

The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits (STATUS <4:3>) are set or cleared depending on the different Reset conditions (Table 5-1). These bits may be used to determine the nature of the Reset.

Table 5-3 lists a full description of Reset states of all registers. Figure 5-1 shows a simplified block diagram of the on-chip Reset circuit.

TABLE 5-1: STATUS BITS AND THEIR SIGNIFICANCE

Condition	$\overline{\text{TO}}$	$\overline{\text{PD}}$
Power-on Reset	1	1
$\overline{\text{MCLR}}$ Reset (normal operation)	u	u
$\overline{\text{MCLR}}$ Wake-up (from Sleep)	1	0
WDT Reset (normal operation)	0	1
WDT Wake-up (from Sleep)	0	0

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

TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH RESET

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on $\overline{\text{MCLR}}$ and WDT Reset
03h	STATUS	PA2	PA1	PA0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	000q quuu

Legend: u = unchanged, x = unknown, q = see Table 5-1 for possible values.

TABLE 5-3: RESET CONDITIONS FOR ALL REGISTERS

Register	Address	Power-on Reset	$\overline{\text{MCLR}}$ or WDT Reset
W	N/A	xxxx xxxx	uuuu uuuu
TRIS	N/A	1111 1111	1111 1111
OPTION	N/A	--11 1111	--11 1111
INDF	00h	xxxx xxxx	uuuu uuuu
TMR0	01h	xxxx xxxx	uuuu uuuu
PCL	02h	1111 1111	1111 1111
STATUS	03h	0001 1xxx	000q quuu
FSR(1)	04h	111x xxxx	111u uuuu
FSR(2)	04h	1xxx xxxx	1uuu uuuu
FSR(3)	04h	xxxx xxxx	uuuu uuuu
PORTA	05h	---- xxxx	---- uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu
PORTC(4)	07h	xxxx xxxx	uuuu uuuu
PORTD(5)	08h	xxxx xxxx	uuuu uuuu
PORTE(5)	09h	xxxx ----	uuuu ----

Legend: u = unchanged, x = unknown, – = unimplemented, read as '0', q = see tables in Table 5-1 for possible values.

Note 1: PIC16F54 only.

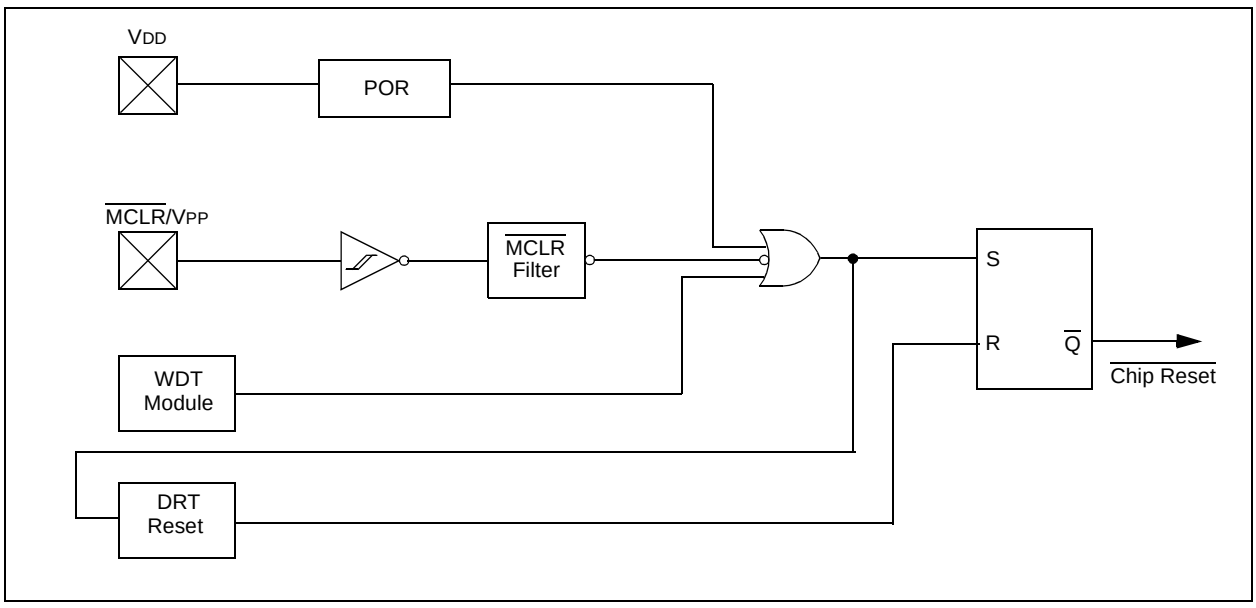
2: PIC16F57 only.

3: PIC16F59 only.

4: General purpose register file on PIC16F54.

5: General purpose register file on PIC16F54 and PIC16F57.

FIGURE 5-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



5.1 Power-on Reset (POR)

The PIC16F5X family of devices incorporate on-chip Power-on Reset (POR) circuitry which provides an internal chip Reset for most power-up situations. To use this feature, the user merely ties the MCLR/VPP pin to VDD. A simplified block diagram of the on-chip Power-on Reset circuit is shown in Figure 5-1.

The Power-on Reset circuit and the Device Reset Timer (Section 5.2) circuit are closely related. On power-up, the Reset latch is set and the DRT is reset. The DRT timer begins counting once it detects MCLR to be high. After the time-out period, which is typically 18 ms, it will reset the Reset latch and thus end the on-chip Reset signal.

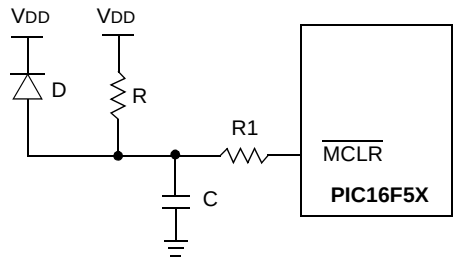
A power-up example where MCLR is not tied to VDD is shown in Figure 5-3. VDD is allowed to rise and stabilize before bringing MCLR high. The chip will actually come out of Reset TDRT msec after MCLR goes high.

In Figure 5-4, the on-chip Power-on Reset feature is being used (MCLR and VDD are tied together). The VDD is stable before the start-up timer times out and there is no problem in getting a proper Reset. However, Figure 5-5 depicts a problem situation where VDD rises too slowly. The time between when the DRT senses a high on the MCLR/VPP pin and the MCLR/VPP pin (and VDD) actually reach their full value is too long. In this situation, when the start-up timer times out, VDD has not reached the VDD (min) value and the chip is, therefore, not ensured to function correctly. For such situations, we recommend that external RC circuits be used to achieve longer POR delay times (Figure 5-2).

Note 1:	When the device starts normal operation (exits 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.
2:	The POR is disabled when the device is in Sleep.

For more information on the PIC16F5X POR, see Application Note AN522, "*Power-Up Considerations*" at www.microchip.com.

FIGURE 5-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



- External Power-on Reset circuit is required only if VDD power-up is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.
- R < 40 kΩ is recommended to make sure that voltage drop across R does not violate the device electrical specification.
- R1 = 100Ω to 1 kΩ will limit any current flowing into MCLR from external capacitor C in the event of MCLR pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

FIGURE 5-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO V_{DD})

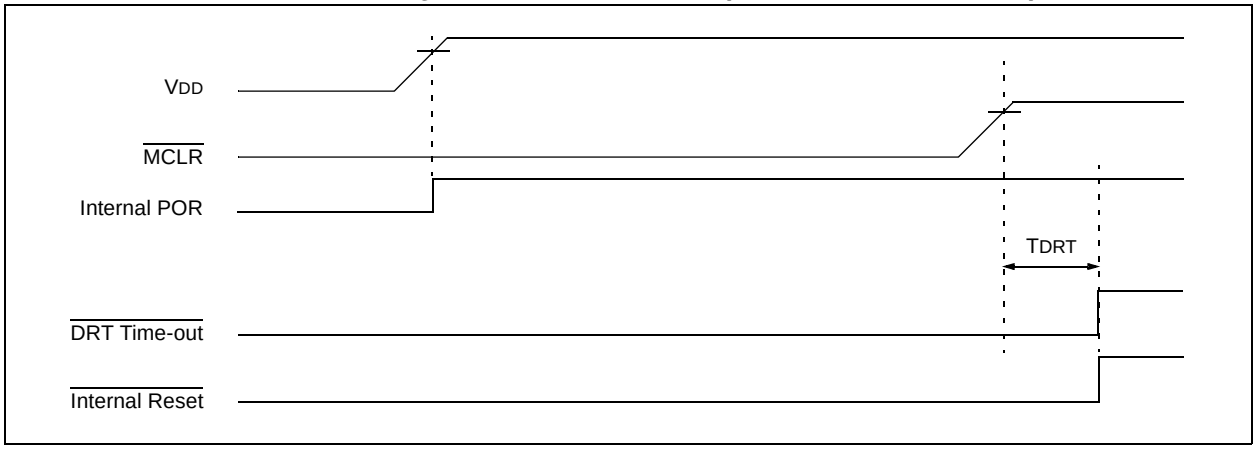


FIGURE 5-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD}): FAST V_{DD} RISE TIME

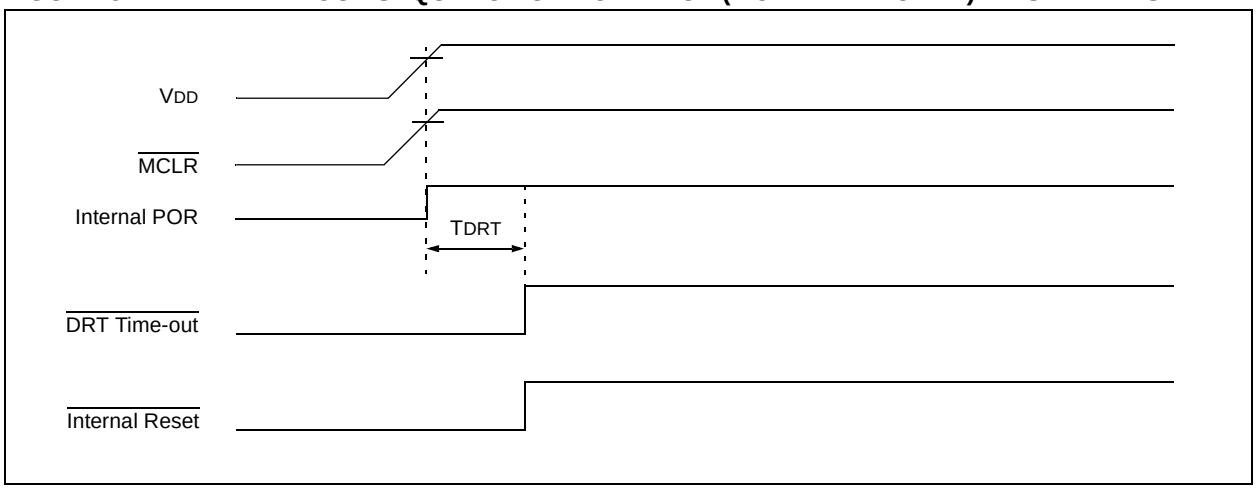
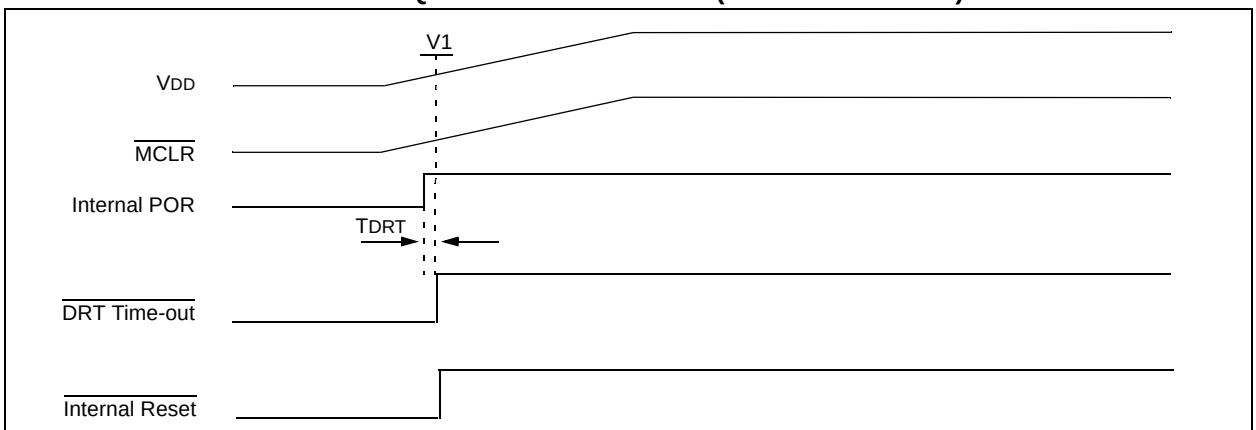


FIGURE 5-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO V_{DD}): SLOW V_{DD} RISE TIME



Note : When V_{DD} rises slowly, the T_{DRT} time-out expires long before V_{DD} has reached its final value. In this example, the chip will reset properly if, and only if, V1 ≥ V_{DD} min.

5.2 Device Reset Timer (DRT)

The Device Reset Timer (DRT) provides an 18 ms nominal time-out on Reset regardless of the oscillator mode used. The DRT operates on an internal RC 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 chosen oscillator to stabilize.

Oscillator circuits, based on crystals or ceramic resonators, require a certain time after power-up to establish a stable oscillation. The on-chip DRT keeps the device in a Reset condition for approximately 18 ms after the voltage on the $\overline{MCLR}/V_{PP}$ pin has reached a logic high (V_{IH}) level. Thus, external RC networks connected to the $\overline{MCLR}$ input are not required in most cases, allowing for savings in cost-sensitive and/or space restricted applications.

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

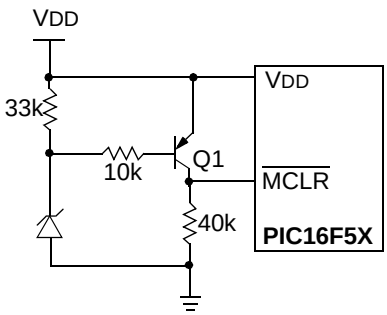
The DRT will also be triggered upon a Watchdog Timer time-out. This is particularly important for applications using the WDT to wake the PIC16F5X from Sleep mode automatically.

5.3 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.

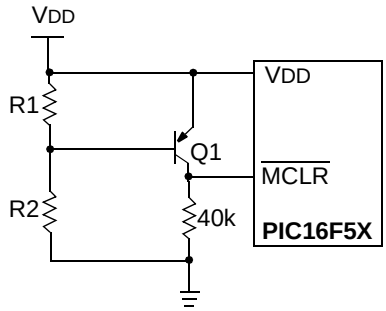
To reset PIC16F5X devices when a Brown-out occurs, external Brown-out protection circuits may be built, as shown in Figure 5-6, Figure 5-7 and Figure 5-8.

FIGURE 5-6: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 1



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

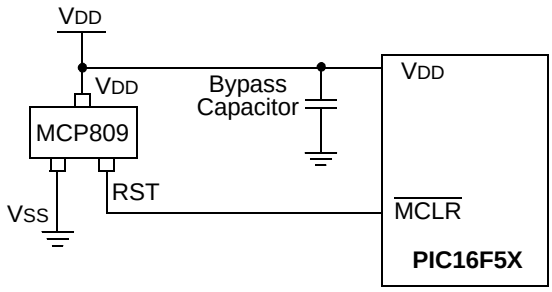
FIGURE 5-7: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2



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

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

FIGURE 5-8: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 3



This brown-out protection circuit employs Microchip Technology's MCP809 microcontroller supervisor. The MCP8XX and MCP1XX families of supervisors provide push-pull and open collector outputs with both "active-high and active-low" Reset pins. There are 7 different trip point selections to accommodate 5V and 3V systems.

NOTES:

6.0 I/O PORTS

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

6.1 PORTA

PORTA is a 4-bit I/O register. Only the low order 4 bits are used (PORTA<3:0>). The high order 4 bits (PORTA<7:4>) are unimplemented and read as '0's.

6.2 PORTB

PORTB is an 8-bit I/O register (PORTB<7:0>).

6.3 PORTC

PORTC is an 8-bit I/O register (PORTC<7:0>) for the PIC16F57 and PIC16F59.

PORTC is a General Purpose Register for the PIC16F54.

6.4 PORTD

PORTD is an 8-bit I/O register (PORTD<7:0>) for the PIC16F59.

PORTD is a General Purpose Register for the PIC16F54 and PIC16F57.

6.5 PORTE

PORTE is an 4-bit I/O register for the PIC16F59. Only the high order 4 bits are used (PORTE<7:4>). The low order 4 bits (PORTE<3:0>) are unimplemented and read as '0's.

PORTE is a General Purpose Register for the PIC16F54 and PIC16F57.

6.6 TRIS Registers

The output driver control registers are 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 (Input) mode. A '0' puts the contents of the output data latch on the selected pins, enabling the output buffer.

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.

6.7 I/O Interfacing

The equivalent circuit for an I/O port pin is shown in Figure 6-1. All ports may be used for both input and output operation. For input operations, these ports are non-latching. Any input must be present until read by an input instruction (e.g., `MOVF PORTB, W`). The outputs are latched and remain unchanged until the output latch is rewritten. To use a port pin as output, the corresponding direction control bit (in TRISA, TRISB, TRISC, TRISD and TRISE) must be cleared (= 0). For use as an input, the corresponding TRIS bit must be set. Any I/O pin can be programmed individually as input or output.

FIGURE 6-1: EQUIVALENT CIRCUIT FOR A SINGLE I/O PIN

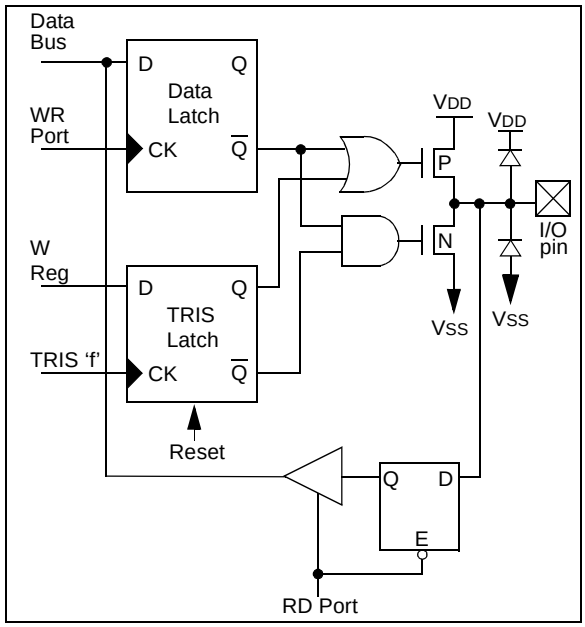


TABLE 6-1: SUMMARY OF PORT REGISTERS

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 MCLR and WDT Reset
N/A	TRIS	I/O Control Registers (TRISA, TRISB, TRISC, TRISD and TRISE)								1111 1111	1111 1111
05h	PORTA	—	—	—	—	RA3	RA2	RA1	RA0	---- xxxx	---- uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
07h	PORTC(1)	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
08h	PORTD(2)	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
09h	PORTE(2)	RE7	RE6	RE5	RE4	—	—	—	—	xxxx ----	uuuu ----

Legend: Shaded cells = unimplemented, read as '0', – = unimplemented, read as '0', x = unknown, u = unchanged

Note 1: File address 07h is a General Purpose Register on the PIC16F54.

2: File address 08h and 09h are General Purpose Registers on the PIC16F54 and PIC16F57.

6.8 I/O Programming Considerations

6.8.1 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 5 of PORTB will cause all eight bits of PORTB to be read into the CPU, bit 5 to be set and the PORTB value to be written to the output latches. If another bit of PORTB 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 6-1 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.

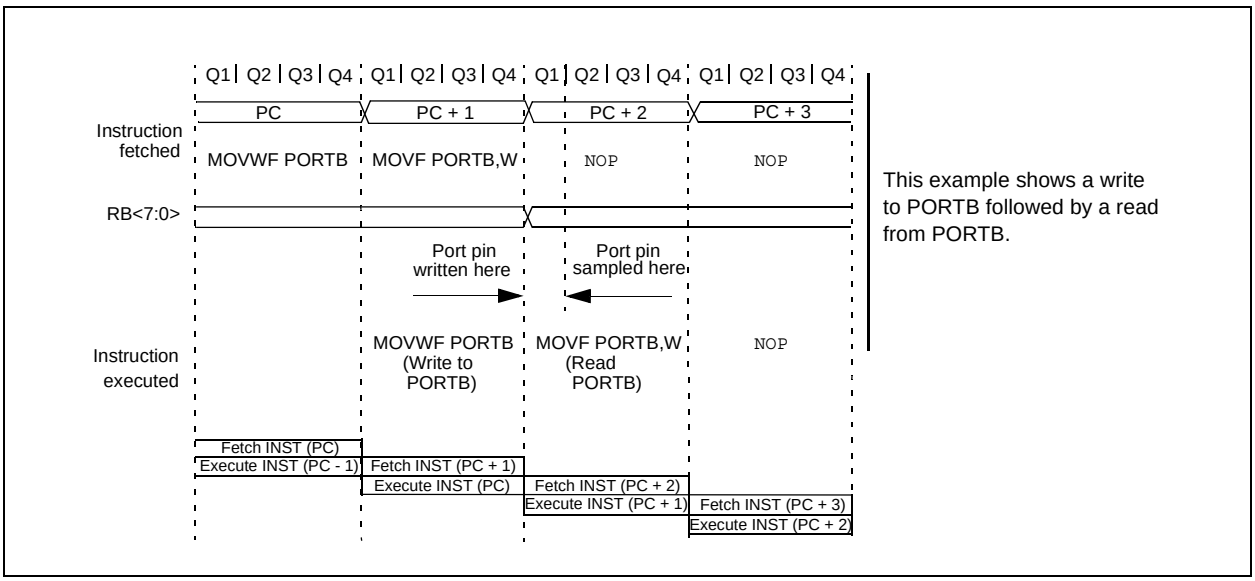
EXAMPLE 6-1: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
;Initial PORT Settings
;PORTB<7:4> Inputs
;PORTB<3:0> Outputs
;PORTB<7:6> have external pull-ups and are
;not connected to other circuitry
;
;                 PORT latch PORT pins
;                 ---------------------
  BCF   PORTB, 7  ;01pp pppp   11pp pppp
  BCF   PORTB, 6  ;10pp pppp   11pp pppp
  MOVLW H'3F'     ;
  TRIS  PORTB     ;10pp pppp   10pp pppp
;
;Note that the user may have expected the
 pin
;values to be 00pp pppp. The 2nd BCF caused
;RB7 to be latched as the pin value (High).
```

6.8.2 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 (see Figure 6-2). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction, which causes that file to be read into the CPU, is executed. 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 6-2: SUCCESSIVE I/O OPERATION



NOTES:

7.0 TIMER0 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 7-1 is a simplified block diagram of the Timer0 module.

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

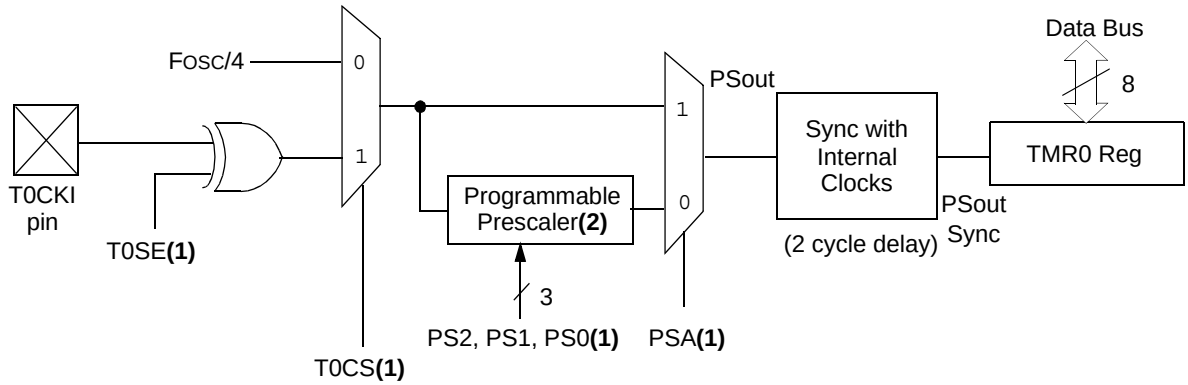
Counter mode is selected by setting the T0CS bit (OPTION<5>). In this mode, Timer0 will increment either on every rising or falling edge of pin T0CKI. The incrementing edge is determined by the source edge select bit T0SE (OPTION<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in **Section 7.1 "Using Timer0 with an External Clock"**.

Note:	The prescaler may be used by either the Timer0 module or the Watchdog Timer, but not both.

The prescaler assignment is controlled in software by the control bit PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to Timer0. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4,..., 1:256 are selectable. **Section 7.2 "Prescaler"** details the operation of the prescaler.

A summary of registers associated with the Timer0 module is found in Table 7-1.

FIGURE 7-1: TIMER0 BLOCK DIAGRAM



Note 1: Bits T0CS, T0SE, PSA, PS2, PS1 and PS0 are located in **Section 3.4 "Option Register"**.

2: The prescaler is shared with the Watchdog Timer (Figure 7-5).

FIGURE 7-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALER

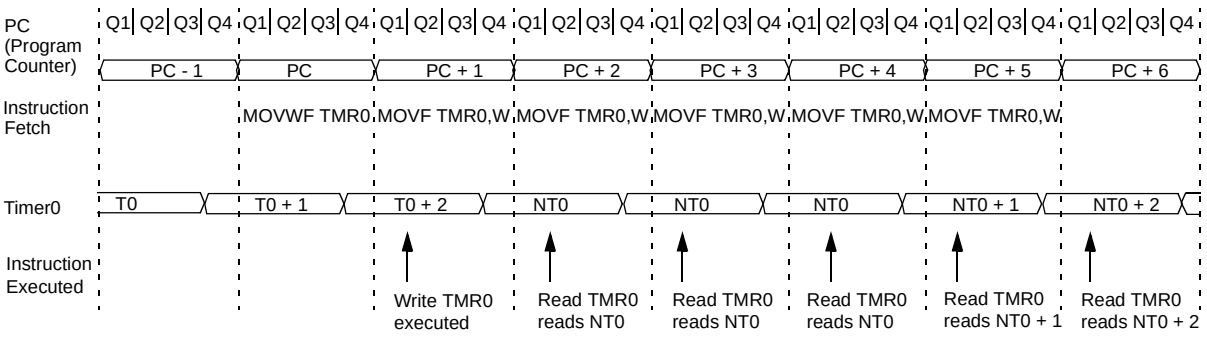


FIGURE 7-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALER 1:2

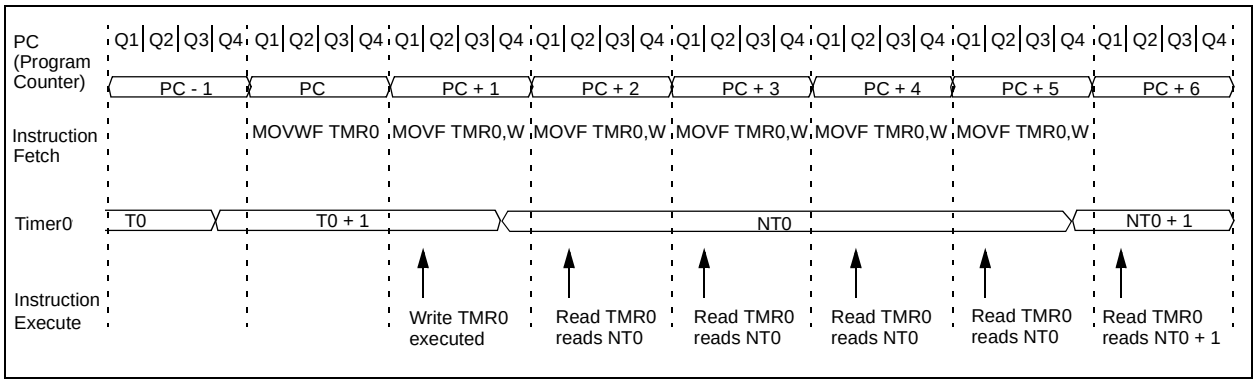


TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER0

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 MCLR and WDT Reset
01h	TMR0	Timer0 - 8-bit real-time clock/counter								xxxx xxxx	uuuu uuuu
N/A	OPTION	—	—	T0CS	T0SE	PSA	PS2	PS1	PS0	--11 1111	--11 1111

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

7.1 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.

7.1.1 EXTERNAL CLOCK SYNCHRONIZATION

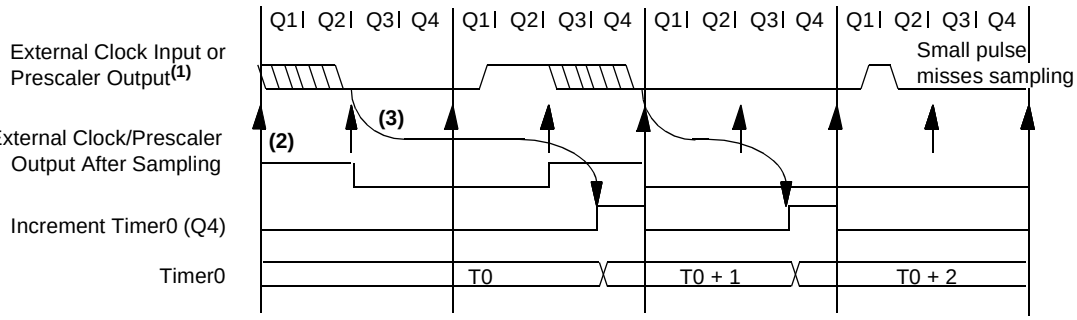
When no prescaler is used, the external clock is the Timer0 input. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 7-4). Therefore, it is necessary for T0CKI to be high for at least 2TOSC (and a small RC delay of 20 ns) and low for at least 2TOSC (and a small RC delay of 20 ns). 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 40 ns) 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 10 ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

7.1.2 TIMER0 INCREMENT DELAY

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

FIGURE 7-4: TIMER0 TIMING WITH EXTERNAL CLOCK



Note 1: External clock if no prescaler selected; prescaler output otherwise.
2: The arrows indicate the points in time where sampling occurs.
3: 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.

7.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer (WDT), respectively (**Section 8.2.1 "WDT Period"**). For simplicity, this counter is being referred to as "prescaler" throughout this data sheet. Note that the prescaler may be used by either the Timer0 module or the WDT, but not both. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the WDT, and vice-versa.

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

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

7.2.1 SWITCHING PRESCALER ASSIGNMENT

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

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

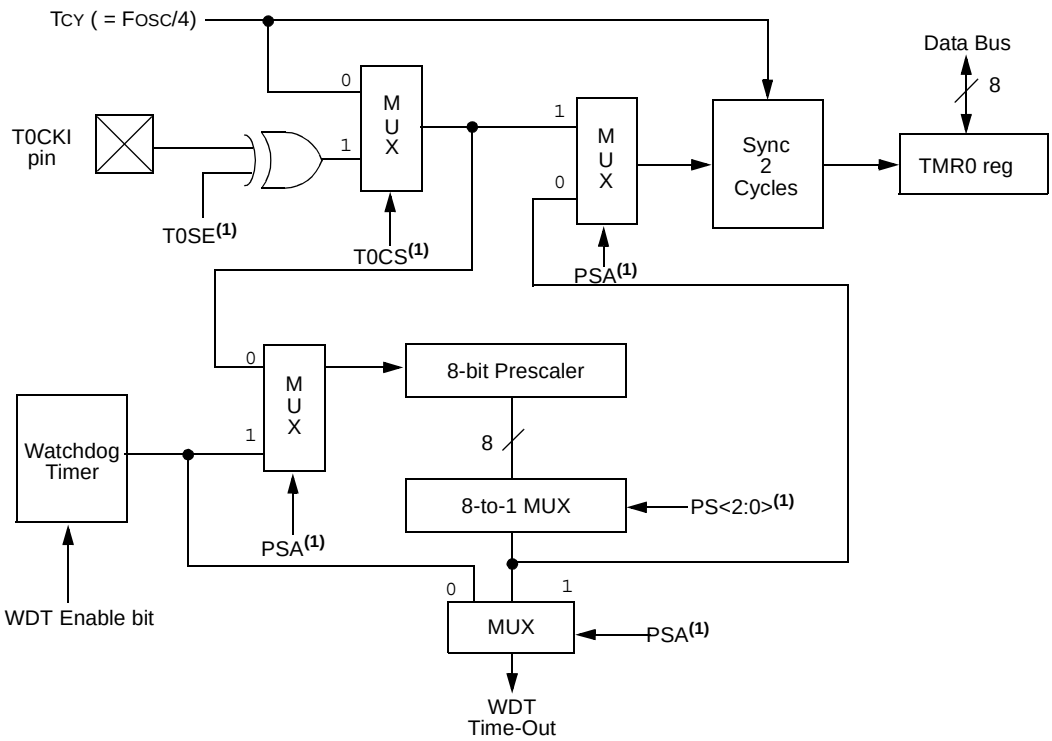
```
CLRWDT                ;Clear WDT
CLRF    TMR0          ;Clear TMR0 &  ;Prescaler
MOVLW   B'00xx1111'   ;Last 3 instructions
                      ;in this example
OPTION                ;are required only if
                      ;desired
CLRWDT                ;PS<2:0> are 000 or 001
MOVLW   B'00xx1xxx'   ;Set Prescaler to
OPTION                ;desired WDT rate
```

To change prescaler from the WDT to the Timer0 module, use the sequence shown in Example 7-2. This sequence must be used even if the WDT is disabled. A CLRWDT instruction should be executed before switching the prescaler.

EXAMPLE 7-2: CHANGING PRESCALER (WDT→TIMER0)

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

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



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

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 PIC16F5X family of 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:

- Oscillator Selection
- Reset
- Power-on Reset
- Device Reset Timer
- Watchdog Timer (WDT)
- Sleep
- Code protection
- User ID locations
- In-Circuit Serial Programming™ (ICSP™)

The PIC16F5X family has a Watchdog Timer which can be shut off only through Configuration bit WDTE. It runs off of its own RC oscillator for added reliability. There is an 18 ms delay provided by the Device Reset Timer (DRT), intended to keep the chip in Reset until the crystal oscillator is stable. 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 external Reset or through a Watchdog Timer time-out. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost, while the LP crystal option saves power. A set of Configuration bits are used to select various options.

8.1 Configuration Bits

Configuration bits can be programmed to select various device configurations. Two bits are for the selection of the oscillator type; one bit is the Watchdog Timer enable bit; one bit is for code protection for the PIC16F5X devices (Register 8-1).

REGISTER 8-1: CONFIGURATION WORD FOR PIC16F5X

—	—	—	—	—	—	—	—	$\overline{CP}$	WDTE	FOSC1	FOSC0

bit 11 bit 0

bit 11-4: **Unimplemented**: Read as '1'

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

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

bit 1-0: **FOSC1:FOSC0:** Oscillator Selection bits
00 = LP oscillator
01 = XT oscillator
10 = HS oscillator
11 = RC oscillator

Note 1: Refer to the PIC16F54, PIC16F57 and PIC16F59 Programming Specifications to determine how to access the Configuration Word. These documents can be found on the Microchip web site at www.microchip.com.

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

8.2 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 RC oscillator of the OSC1/CLKIN pin. That means that the WDT will run even if the clock on the OSC1/CLKIN and OSC2/CLKOUT pins have 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 (**Section 3.3 "STATUS Register"**).

The WDT can be permanently disabled by programming the Configuration bit WDTE as a '0' (**Section 8.1 "Configuration Bits"**). Refer to the PIC16F54 and PIC16F57 Programming Specifications to determine how to access the Configuration Word. These documents can be found on the Microchip web site at www.microchip.com.

8.2.1 WDT PERIOD

An 8-bit counter is available as a prescaler for the Timer0 module (**Section 7.2 "Prescaler"**), or as a postscaler for the Watchdog Timer (WDT), respectively. 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 (**Section 3.4 "Option Register"**).

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 time-out, a period of a nominal 2.3 seconds, can be realized. These periods vary with temperature, V_{DD} and part-to-part process variations (see Device Characterization).

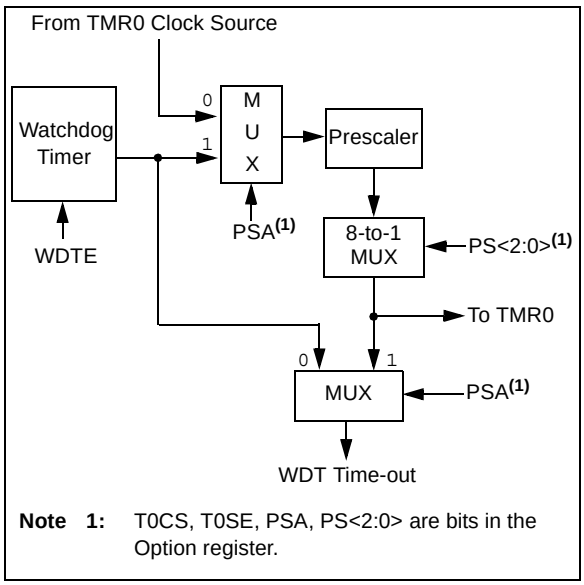
Under worst case conditions (V_{DD} = Min., Temperature = Max., WDT prescaler = 1:128), it may take several seconds before a WDT time-out occurs.

8.2.2 WDT PROGRAMMING CONSIDERATIONS

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

The SLEEP instruction resets the WDT and the prescaler, if assigned to the WDT. This gives the maximum Sleep time before a WDT Wake-up Reset.

FIGURE 8-1: WATCHDOG TIMER BLOCK DIAGRAM



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

TABLE 8-1: 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 MCLR and WDT Reset
N/A	OPTION	—	—	T0CS	T0SE	PSA	PS2	PS1	PS0	--11 1111	--11 1111

Legend: Shaded cells not used by Watchdog Timer, - = unimplemented, read as '0', u = unchanged

8.3 Power-Down Mode (Sleep)

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

8.3.1 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).

It should be noted that a Reset generated by a WDT time-out does not drive the $\overline{MCLR}/V_{PP}$ pin low.

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

8.3.2 WAKE-UP FROM SLEEP

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

1. An external Reset input on $\overline{MCLR}/V_{PP}$ pin.
2. A Watchdog Timer time-out Reset (if WDT was enabled).

Both of these events cause a device Reset. The $\overline{TO}$ and $\overline{PD}$ bits can be used to determine the cause of device Reset. The $\overline{TO}$ bit is cleared if a WDT time-out occurred (and caused wake-up). The $\overline{PD}$ bit, which is set on power-up, is cleared when SLEEP is invoked.

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

8.4 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.

Once code protection is enabled, all program memory locations above 0x3F read all '0's. Program memory locations 0x00-0x3F are always unprotected. The user ID locations and the Configuration Word read out in an unprotected fashion. It is possible to program the user ID locations and the Configuration Word after code protect is enabled.

8.5 User ID Locations

Four memory locations are designated as user 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 user ID locations and always program the upper 8 bits as '1's.

Note:	Microchip will assign a unique pattern number for QTP and SQTP requests. This pattern number will be unique and traceable to the submitted code.

8.6 In-Circuit Serial Programming™ (ICSP™)

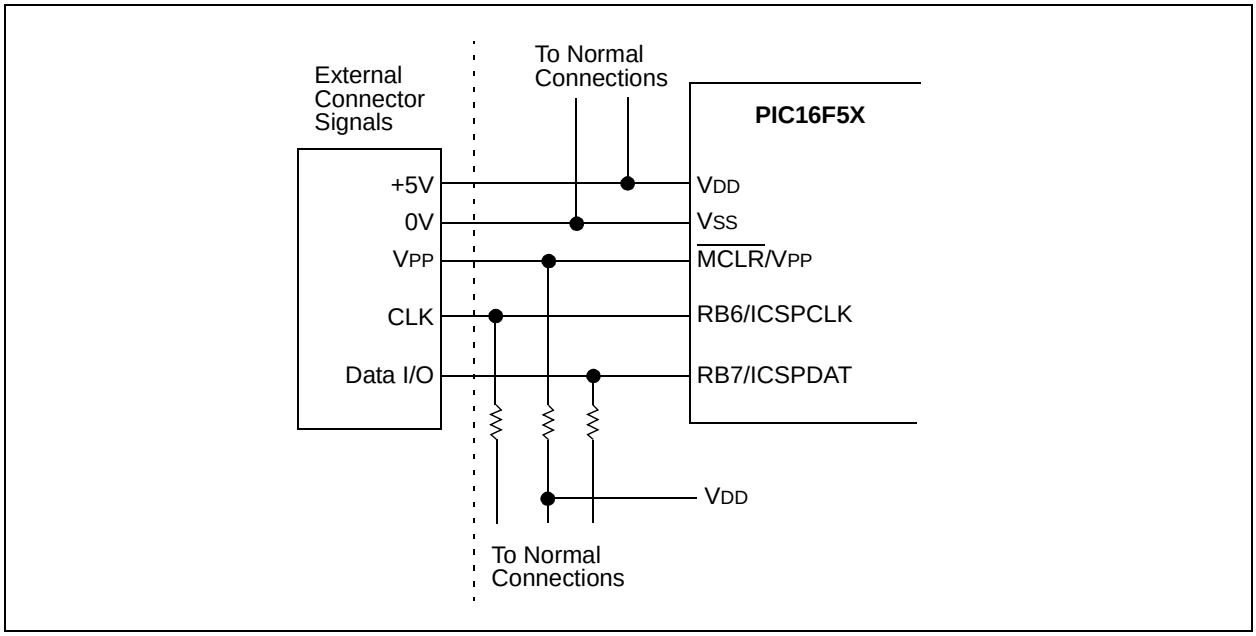
The PIC16F5X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. Thus, the most recent firmware or custom firmware can be programmed.

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

A 6-bit command is then supplied to the device. Depending on the command, 14 bits of program data are then supplied to or from the device, depending if the command was a Load or a Read. For complete details of serial programming, please refer to the respective Programming Specifications: *"PIC16F54 Memory Programming Specification"* (DS41207), *"PIC16F57 Memory Programming Specification*" (DS41208), and "*PIC16F59 Memory Programming Specification"* (DS41243).

A typical In-Circuit Serial Programming connection is shown in Figure 8-1.

FIGURE 8-1: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING™ CONNECTION



9.0 INSTRUCTION SET SUMMARY

Each PIC16F5X 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 PIC16F5X instruction set summary in Table 9-2 groups the instructions into byte-oriented, bit-oriented, and literal and control operations. Table 9-1 shows the opcode field descriptions.

For **byte-oriented** instructions, 'f' represents a file register designator and 'd' represents a destination designator. The file register designator is used to specify which one of the 32 file registers in that bank 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 0x1F)
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

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

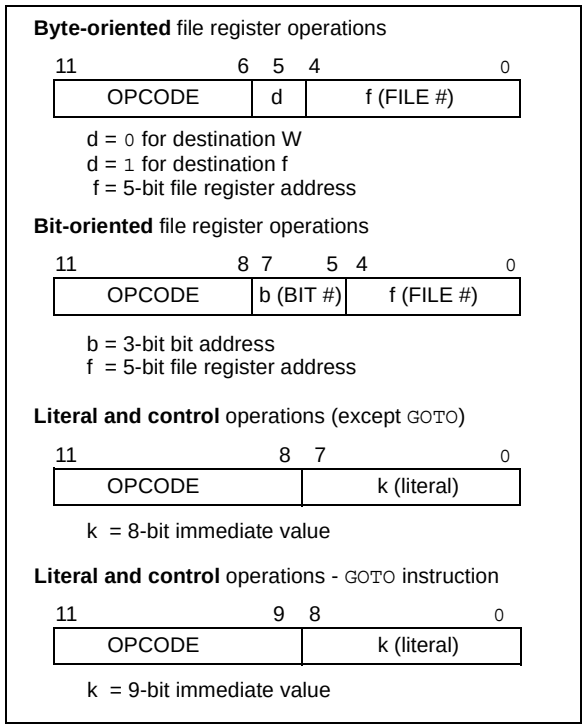


TABLE 9-2: INSTRUCTION SET SUMMARY

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	Subroutine Call	2	1001	kkkk	kkkk	None	1
CLRWDT	—	Clear Watchdog Timer	1	0000	0000	0100	$\overline{TO}$, $\overline{PD}$	
GOTO	k	Unconditional branch	2	101k	kkkk	kkkk	None	
IORLW	k	Inclusive OR Literal with W	1	1101	kkkk	kkkk	Z	
MOVLW	k	Move Literal to W	1	1100	kkkk	kkkk	None	
OPTION	—	Load OPTION register	1	0000	0000	0010	None	
RETLW	k	Return, place Literal in W	2	1000	kkkk	kkkk	None	
SLEEP	—	Go into Standby mode	1	0000	0000	0011	$\overline{TO}$, $\overline{PD}$	
TRIS	f	Load TRIS register	1	0000	0000	0fff	None	3
XORLW	k	Exclusive OR Literal to W	1	1111	kkkk	kkkk	Z	

Note 1: The 9th bit of the program counter will be forced to a '0' by any instruction that writes to the PC except for GOTO (see **Section 3.5 "Program Counter"** for more on 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 = 5, 6 or 7 causes the contents of the W register to be written to the tri-state latches of PORTA, B or C, respectively. A '1' forces the pin to a high-impedance state and disables the output buffers.

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

ADDWF **Add W and f**

Syntax: [*label*] ADDWF f, d

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

Operation: (W) + (f) → (dest)

Status Affected: C, DC, Z

Encoding:

0001	11df	ffff

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'.

Words: 1

Cycles: 1

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

ANDWF **AND W with f**

Syntax: [*label*] ANDWF f, d

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

Operation: (W) .AND. (f) → (dest)

Status Affected: Z

Encoding:

0001	01df	ffff

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'.

Words: 1

Cycles: 1

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

ANDLW **AND literal with W**

Syntax: [*label*] ANDLW k

Operands: $0 \le k \le 255$

Operation: (W).AND. (k) → (W)

Status Affected: Z

Encoding:

1110	kkkk	kkkk

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

Words: 1

Cycles: 1

Example:
```
ANDLW  H'5F'
```
Before Instruction
W = 0xA3
After Instruction
W = 0x03

BCF **Bit Clear f**

Syntax: [*label*] BCF f, b

Operands: $0 \le f \le 31$
$0 \le b \le 7$

Operation: 0 → (f<b>)

Status Affected: None

Encoding:

0100	bbbf	ffff

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

Words: 1

Cycles: 1

Example:
```
BCF    FLAG_REG,  7
```
Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BSF **Bit Set f**

Syntax: [*label*] BSF f, b

Operands: 0 ≤ f ≤ 31
0 ≤ b ≤ 7

Operation: 1 → (f<b>)

Status Affected: None

Encoding:

0101	bbbf	ffff

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

Words: 1

Cycles: 1

Example:
```
BSF     FLAG_REG,    7
```
Before Instruction
FLAG_REG = 0x0A
After Instruction
FLAG_REG = 0x8A

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

Encoding:

0110	bbbf	ffff

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 two-cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE   BTFSC  FLAG,1
FALSE  GOTO   PROCESS_CODE
TRUE   •
       •
       •
```
Before Instruction
PC = address (HERE)
After Instruction
if FLAG<1> = 0,
PC = address (TRUE);
if FLAG<1> = 1,
PC = address (FALSE)

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

Encoding:

0111	bbbf	ffff

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 two-cycle instruction.

Words: 1

Cycles: 1(2)

Example:
```
HERE   BTFSS  FLAG,1
FALSE  GOTO   PROCESS_CODE
TRUE   •
       •
       •
```
Before Instruction
PC = address (HERE)
After Instruction
If FLAG<1> = 0,
PC = address (FALSE);
if FLAG<1> = 1,
PC = address (TRUE)

CALL **Subroutine Call**

Syntax: [*label*] CALL k

Operands: $0 \le k \le 255$

Operation: (PC) + 1→ TOS;
k → PC<7:0>;
(Status<6:5>) → PC<10:9>;
0 → PC<8>

Status Affected: None

Encoding:

1001	kkkk	kkkk

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.

Words: 1

Cycles: 2

Example:
```
HERE    CALL    THERE
```
Before Instruction
PC = address (HERE)
After Instruction
PC = address (THERE)
TOS = address (HERE + 1)

CLRF **Clear f**

Syntax: [*label*] CLRF f

Operands: $0 \le f \le 31$

Operation: 00h → (f);
1 → Z

Status Affected: Z

Encoding:

0000	011f	ffff

Description: The contents of register ‘f’ are cleared and the Z bit is set.

Words: 1

Cycles: 1

Example:
```
CLRF    FLAG_REG
```
Before Instruction
FLAG_REG = 0x5A
After Instruction
FLAG_REG = 0x00
Z = 1

CLRW **Clear W**

Syntax: [*label*] CLRW

Operands: None

Operation: 00h → (W);
1 → Z

Status Affected: Z

Encoding:

0000	0100	0000

Description: The W register is cleared. Zero bit (Z) is set.

Words: 1

Cycles: 1

Example:
```
CLRW
```
Before Instruction
W = 0x5A
After Instruction
W = 0x00
Z = 1

CLRWDT **Clear Watchdog Timer**

Syntax: [*label*] CLRWDT

Operands: None

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

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

Encoding:

0000	0000	0100

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.

Words: 1

Cycles: 1

Example:
```
CLRWDT
```
Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescaler = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

COMF **Complement f**

Syntax: [*label*] COMF f, d

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

Operation: $(\bar{f}) \rightarrow (dest)$

Status Affected: Z

Encoding:

0010	01df	ffff

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'.

Words: 1

Cycles: 1

Example:
```
COMF   REG1,0
```
Before Instruction
REG1 = 0x13
After Instruction
REG1 = 0x13
W = 0xEC

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

Encoding:

0000	11df	ffff

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'.

Words: 1

Cycles: 1

Example:
```
DECF    CNT, 1
```
Before Instruction
CNT = 0x01
Z = 0
After Instruction
CNT = 0x00
Z = 1

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

Encoding:

0010	11df	ffff

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.

Words: 1

Cycles: 1(2)

Example:
```
HERE     DECFSZ   CNT, 1
         GOTO     LOOP
CONTINUE •
         •
         •
```
Before Instruction
PC = address(HERE)
After Instruction
CNT = CNT - 1;
if CNT = 0,
PC = address (CONTINUE);
if CNT ≠ 0,
PC = address (HERE+1)

GOTO **Unconditional Branch**

Syntax: [*label*] GOTO k

Operands: 0 ≤ k ≤ 511

Operation: k → PC<8:0>;
STATUS<6:5> → PC<10:9>

Status Affected: None

Encoding:

101k	kkkk	kkkk

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.

Words: 1

Cycles: 2

Example:
```
GOTO THERE
```
After Instruction
PC = address (THERE)

INCF **Increment f**

Syntax: [*label*] INCF f, d

Operands: 0 ≤ f ≤ 31
d ∈ [0,1]

Operation: (f) + 1 → (dest)

Status Affected: Z

Encoding:

0010	10df	ffff

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'.

Words: 1

Cycles: 1

Example:
```
INCF   CNT,   1
```
Before Instruction
CNT = 0xFF
Z = 0
After Instruction
CNT = 0x00
Z = 1

INCFSZ **Increment f, Skip if 0**

Syntax: [*label*] INCFSZ f, d

Operands: 0 ≤ f ≤ 31
d ∈ [0,1]

Operation: (f) + 1 → (dest), skip if result = 0

Status Affected: None

Encoding:

0011	11df	ffff

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.

Words: 1

Cycles: 1(2)

Example:
```
HERE      INCFSZ     CNT, 1
          GOTO       LOOP
CONTINUE  •
          •
          •
```
Before Instruction
PC = address (HERE)
After Instruction
CNT = CNT + 1;
if CNT = 0,
PC = address (CONTINUE);
if CNT ≠ 0,
PC = address (HERE +1)

IORLW **Inclusive OR literal with W**

Syntax: [*label*] IORLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) .OR. (k) → (W)

Status Affected: Z

Encoding:

1101	kkkk	kkkk

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

Words: 1

Cycles: 1

Example:
```
IORLW  0x35
```
Before Instruction
W = 0x9A
After Instruction
W = 0xBF
Z = 0

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

Encoding:

0001	00df	ffff

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'.

Words: 1

Cycles: 1

Example:
```
IORWF   RESULT, 0
```
Before Instruction
RESULT = 0x13
W = 0x91
After Instruction
RESULT = 0x13
W = 0x93
Z = 0

MOVF **Move f**

Syntax: [*label*] MOVF f, d

Operands: 0 ≤ f ≤ 31
d ∈ [0,1]

Operation: (f) → (dest)

Status Affected: Z

Encoding:

0010	00df	ffff

Description: The contents of register 'f' is moved to destination 'd'. If 'd' is '0', destination is the W register. If 'd' is '1', the destination is file register 'f'. 'd' is '1' is useful to test a file register since Status flag Z is affected.

Words: 1

Cycles: 1

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

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W)

Status Affected: None

Encoding:

1100	kkkk	kkkk

Description: The eight-bit literal 'k' is loaded into the W register.

Words: 1

Cycles: 1

Example:
```
MOVLW  0x5A
```
After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f

Operands: $0 \le f \le 31$

Operation: (W) → (f)

Status Affected: None

Encoding:

0000	001f	ffff

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

Words: 1

Cycles: 1

Example:

```
MOVWF  TEMP_REG
```

Before Instruction
TEMP_REG = 0xFF
W = 0x4F
After Instruction
TEMP_REG = 0x4F
W = 0x4F

NOP **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000	0000	0000

Description: No operation.

Words: 1

Cycles: 1

Example:

```
NOP
```

OPTION **Load OPTION Register**

Syntax: [*label*] OPTION

Operands: None

Operation: (W) → OPTION

Status Affected: None

Encoding:

0000	0000	0010

Description: The content of the W register is loaded into the Option register.

Words: 1

Cycles: 1

Example:

```
OPTION
```

Before Instruction
W = 0x07
After Instruction
OPTION = 0x07

RETLW **Return with Literal in W**

Syntax: [*label*] RETLW k

Operands: $0 \le k \le 255$

Operation: k → (W);
TOS → PC

Status Affected: None

Encoding:

1000	kkkk	kkkk

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

Words: 1

Cycles: 2

Example:

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

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

RLF **Rotate Left f through Carry**

Syntax: [*label*] RLF f, d

Operands: 0 ≤ f ≤ 31
d ∈ [0,1]

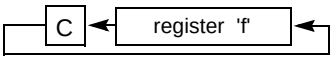
Operation: See description below

Status Affected: C

Encoding:

0011	01df	ffff

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



Words: 1

Cycles: 1

Example:
```
RLF    REG1,0
```
Before Instruction
REG1 = 1110 0110
C = 0
After Instruction
REG1 = 1110 0110
W = 1100 1100
C = 1

RRF **Rotate Right f through Carry**

Syntax: [*label*] RRF f, d

Operands: 0 ≤ f ≤ 31
d ∈ [0,1]

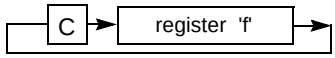
Operation: See description below

Status Affected: C

Encoding:

0011	00df	ffff

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag (STATUS<0>). If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example:
```
RRF    REG1,0
```
Before Instruction
REG1 = 1110 0110
C = 0
After Instruction
REG1 = 1110 0110
W = 0111 0011
C = 0

Sleep **Go into Standby Mode**

Syntax: [*label*] Sleep

Operands: None

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

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

Encoding:

0000	0000	0011

Description: Time-out Status bit ($\overline{TO}$) is set. The power-down Status bit ($\overline{PD}$) is cleared. 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.

Words: 1

Cycles: 1

Example:
```
SLEEP
```

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

Encoding:

0000	10df	ffff

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'.

Words: 1

Cycles: 1

Example 1: `SUBWF REG1, 1`

Before Instruction
REG1 = 3
W = 2
C = ?
After Instruction
REG1 = 1
W = 2
C = 1 ; result is positive

Example 2:
Before Instruction
REG1 = 2
W = 2
C = ?
After Instruction
REG1 = 0
W = 2
C = 1 ; result is zero

Example 3:
Before Instruction
REG1 = 1
W = 2
C = ?
After Instruction
REG1 = 0xFF
W = 2
C = 0 ; result is negative

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

Encoding:

0011	10df	ffff

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'.

Words: 1

Cycles: 1

Example: `SWAPF REG1, 0`

Before Instruction
REG1 = 0xA5
After Instruction
REG1 = 0xA5
W = 0x5A

TRIS **Load TRIS Register**

Syntax: [*label*] TRIS f

Operands: f = 5, 6, 7, 8 or 9

Operation: (W) → TRIS register f

Status Affected: None

Encoding:

0000	0000	0fff

Description: TRIS register 'f' (f = 5, 6 or 7) is loaded with the contents of the W register.

Words: 1

Cycles: 1

Example: `TRIS PORTB`

Before Instruction
W = 0xA5
After Instruction
TRISB = 0xA5

XORLW **Exclusive OR literal with W**

Syntax: [*label*] XORLW k

Operands: $0 \le k \le 255$

Operation: (W) .XOR. k → (W)

Status Affected: Z

Encoding:

1111	kkkk	kkkk

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

Words: 1

Cycles: 1

Example:

```
XORLW  0xAF
```

Before Instruction
W = 0xB5
After Instruction
W = 0x1A

XORWF **Exclusive OR W with f**

Syntax: [*label*] XORWF f, d

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

Operation: (W) .XOR. (f) → (dest)

Status Affected: Z

Encoding:

0001	10df	ffff

Description: Exclusive OR the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

Words: 1

Cycles: 1

Example:

```
XORWF  REG,1
```

Before Instruction
REG = 0xAF
W = 0xB5
After Instruction
REG = 0x1A
W = 0xB5

10.0 DEVELOPMENT SUPPORT

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

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

10.1 MPLAB Integrated Development Environment Software

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

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

The MPLAB IDE allows you to:

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

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

10.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC 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

10.3 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 and PIC24 families of microcontrollers and the dsPIC30 and dsPIC33 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.

10.4 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

10.5 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

10.6 MPLAB SIM Software Simulator

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

The MPLAB 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 hardware laboratory environment, making it an excellent, economical software development tool.

10.7 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 PIC 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 PIC 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.

10.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC® and MCU devices. It debugs and programs PIC® and dsPIC® Flash microcontrollers with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The MPLAB REAL ICE probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with the popular MPLAB ICD 2 system (RJ11) or with the new high speed, noise tolerant, low-voltage differential signal (LVDS) interconnection (CAT5).

MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added, such as software breakpoints and assembly code trace. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, real-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

10.9 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 PIC MCUs and can be used to develop for these and other PIC 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 PIC devices.

10.10 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 PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an SD/MMC card for file storage and secure data applications.

10.11 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 PIC 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.

10.12 PICkit 2 Development Programmer

The PICkit™ 2 Development Programmer is a low-cost programmer and selected Flash device debugger with an easy-to-use interface for programming many of Microchip's baseline, mid-range and PIC18F families of Flash memory microcontrollers. The PICkit 2 Starter Kit includes a prototyping development board, twelve sequential lessons, software and HI-TECH's PICC™ Lite C compiler, and is designed to help get up to speed quickly using PIC® microcontrollers. The kit provides everything needed to program, evaluate and develop applications using Microchip's powerful, mid-range Flash memory family of microcontrollers.

10.13 Demonstration, Development and Evaluation Boards

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

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

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

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

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.

11.0 ELECTRICAL SPECIFICATIONS FOR PIC16F54/57

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} ..... 0V to +6.5V

Voltage on $\overline{MCLR}$ with respect to V_{SS}(1) ..... 0V to +13.5V

Voltage on all other pins with respect to V_{SS} ..... -0.6V to (V_{DD} + 0.6V)

Total power dissipation(2) ..... 800 mW

Max. current out of V_{SS} pin ..... 150 mA

Max. current into V_{DD} pin ..... 100 mA

Max. current into an input pin (T0CKI only) ..... ±500 µA

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 a single I/O port (PORTA, B or C) ..... 50 mA

Max. output current sunk by a single I/O port (PORTA, B or C) ..... 50 mA

Note 1: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50 to 100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to V_{SS}.

2: 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.

11.0 ELECTRICAL SPECIFICATIONS FOR PIC16F59 (continued)

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} ..... 0V to +6.5V

Voltage on $\overline{MCLR}$ with respect to V_{SS}(1) ..... 0V to +13.5V

Voltage on all other pins with respect to V_{SS} ..... -0.6V to (V_{DD} + 0.6V)

Total power dissipation(2) ..... 900 mW

Max. current out of V_{SS} pins ..... 250 mA

Max. current into V_{DD} pins ..... 200 mA

Max. current into an input pin (T0CKI only) ..... ±500 µA

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 a single I/O port (PORTA, B, C, D or E) ..... 100 mA

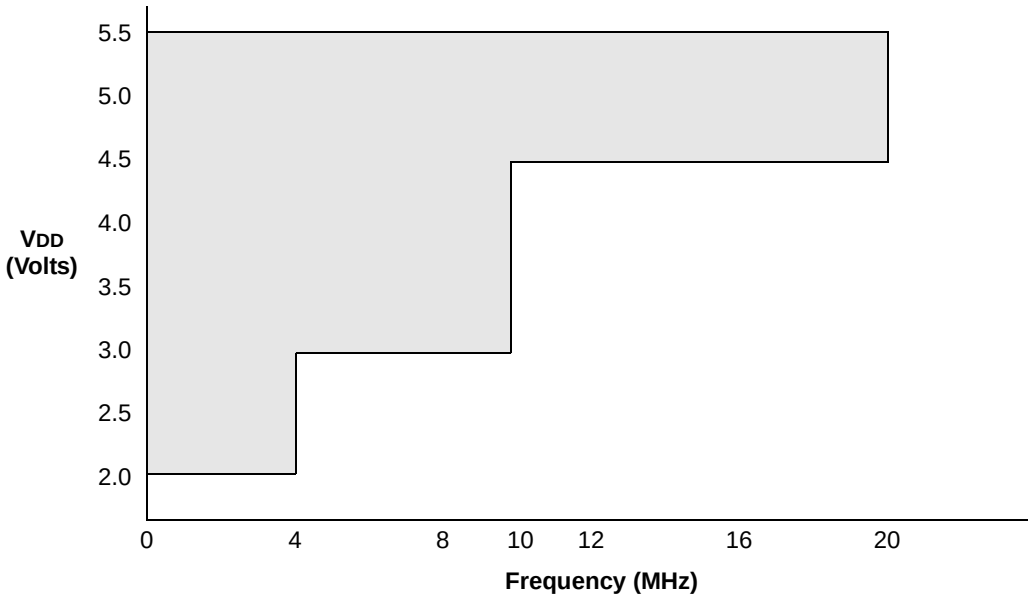
Max. output current sunk by a single I/O port (PORTA, B, C, D or E) ..... 100 mA

Note 1: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50 to 100Ω should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to V_{SS}.

2: 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 11-1: PIC16F5X VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA ≤ +125°C



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

11.1 DC Characteristics: PIC16F5X (Industrial)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C for industrial				
Param No.	**Sym.**	**Characteristic/Device**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**	2.0	—	5.5	V	
D002	VDR	**RAM Data Retention Voltage(1)**	—	1.5*	—	V	Device in Sleep mode
D003	VPOR	**VDD Start Voltage** to ensure Power-on Reset	—	Vss	—	V	See **Section 5.1 "Power-on Reset (POR)"** for details on Power-on Reset
D004	SVDD	**VDD Rise Rate** to ensure Power-on Reset	0.05*	—	—	V/ms	See **Section 5.1 "Power-on Reset (POR)"** for details on Power-on Reset
D010	IDD	**Supply Current(2)**					
			—	170	350	µA	FOSC = 4 MHz, VDD = 2.0V, XT or RC mode(3)
			—	0.4	1.0	mA	FOSC = 10 MHz, VDD = 3.0V, HS mode
			—	1.7	5.0	mA	FOSC = 20 MHz, VDD = 5.0V, HS mode
			—	15	22.5	µA	FOSC = 32 kHz, VDD = 2.0V, LP mode, WDT disabled
D020	IPD	**Power-down Current(2)**					
			—	1.0	6.0	µA	VDD = 2.0V, WDT enabled
			—	0.5	2.5	µA	VDD = 2.0V, WDT disabled

* These parameters are characterized but not tested.

† Data in "Typ" column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

Note 1: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, 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: OSC1 = external square wave, from rail-to-rail; 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. The Power-down Current in Sleep mode does not depend on the oscillator type.

3: Does not include current through REXT. The current through the resistor can be estimated by the formula: IR = VDD/2REXT (mA) with REXT in kΩ.

11.2 DC Characteristics: PIC16F5X (Extended)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym.**	**Characteristic/Device**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
D001	VDD	**Supply Voltage**	2.0	—	5.5	V	
D002	VDR	**RAM Data Retention Voltage(1)**	—	1.5*	—	V	Device in Sleep mode
D003	VPOR	**VDD Start Voltage** to ensure Power-on Reset	—	VSS	—	V	See **Section 5.1 "Power-on Reset (POR)"** for details on Power-on Reset
D004	SVDD	**VDD Rise Rate** to ensure Power-on Reset	0.05*	—	—	V/ms	See **Section 5.1 "Power-on Reset (POR)"** for details on Power-on Reset
D010	IDD	**Supply Current(2)**					
			— — — —	170 0.4 1.7 15	450 2.0 7.0 40	µA mA mA µA	FOSC = 4 MHz, VDD = 2.0V, XT or RC mode(3) FOSC = 10 MHz, VDD = 3.0V, HS mode FOSC = 20 MHz, VDD = 5.0V, HS mode FOSC = 32 kHz, VDD = 2.0V, LP mode, WDT disabled
D020	IPD	**Power-down Current(2)**					
			— —	1.0 0.5	15.0 8.0	µA µA	VDD = 2.0V, WDT enabled VDD = 2.0V, WDT disabled

* These parameters are characterized but not tested.

† Data in "Typ" column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

Note 1: This is the limit to which VDD can be lowered in Sleep mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as bus loading, oscillator type, 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: OSC1 = external square wave, from rail-to-rail; 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. The Power-down Current in Sleep mode does not depend on the oscillator type.

3: Does not include current through REXT. The current through the resistor can be estimated by the formula: IR = VDD/2REXT (mA) with REXT in kΩ.

11.3 DC Characteristics PIC16F5X

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
D030		I/O Ports	VSS	—	0.8V	V	4.5V <VDD ≤ 5.5V
		I/O Ports	VSS	—	0.15 VDD	V	VDD ≤ 4.5V
		$\overline{MCLR}$ (Schmitt Trigger)	VSS	—	0.15 VDD	V	
		T0CKI (Schmitt Trigger)	VSS	—	0.15 VDD	V	
		OSC1 (Schmitt Trigger)	VSS	—	0.15 VDD	V	RC mode(3)
		OSC1	VSS	—	0.3 VDD	V	HS mode
			VSS	—	0.3	V	XT mode
			VSS	—	0.3	V	LP mode
	VIH	**Input High Voltage**					
D040		I/O ports	2.0	—	VDD	V	4.5V < VDD ≤ 5.5V
		I/O ports	0.25 VDD + 0.8	—	VDD	V	VDD ≤ 4.5V
		$\overline{MCLR}$ (Schmitt Trigger)	0.85 VDD	—	VDD	V	
		T0CKI (Schmitt Trigger)	0.85 VDD	—	VDD	V	
		OSC1 (Schmitt Trigger)	0.85 VDD	—	VDD	V	RC mode(3)
		OSC1	0.7 VDD	—	VDD	V	HS mode
			1.6	—	VDD	V	XT mode
			1.6	—	VDD	V	LP mode
	IIL	**Input Leakage Current(1, 2)**					
D060		I/O ports	—	—	±1.0	µA	VSS ≤ VPIN ≤ VDD, pin at high-impedance
		$\overline{MCLR}$	—	—	±5.0	µA	VSS ≤ VPIN ≤ VDD
		T0CKI	—	—	±5.0	µA	VSS ≤ VPIN ≤ VDD
		OSC1	—	—	±5.0	µA	VSS ≤ VPIN ≤ VDD, XT, HS and LP modes
	VOL	**Output Low Voltage**					
D080		I/O ports	—	—	0.6	V	IOL = 8.5 mA, VDD = 4.5V
D083		OSC2/CLKOUT (RC mode)	—	—	0.6	V	IOL = 1.6 mA, VDD = 4.5V
	VOH	**Output High Voltage(2)**					
D090		I/O ports(2)	VDD – 0.7	—	—	V	IOH = -3.0 mA, VDD = 4.5V
D092		OSC2/CLKOUT (RC mode)	VDD – 0.7	—	—	V	IOH = -1.3 mA, VDD = 4.5V

* These parameters are characterized but not tested.

† Data in the Typical ("Typ") column is based on characterization results at 25°C. This data is for design guidance only and is not tested.

Note 1: The leakage current on the $\overline{MCLR}$/VPP 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 voltage.

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

3: For the RC mode, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16F5X be driven with external clock in RC mode.

11.4 Timing Parameter Symbology and Load Conditions

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

1. TppS2ppS
2. TppS

T	
F Frequency	T Time

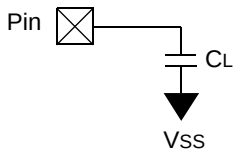
Lowercase letters (pp) and their meanings:

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 11-2: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS – PIC16F5X



Legend:

C_L =	50 pF	for all pins and OSC2 for RC mode
	0-15 pF	for OSC2 in XT, HS or LP modes when external clock is used to drive OSC1

11.5 Timing Diagrams and Specifications

FIGURE 11-3: EXTERNAL CLOCK TIMING

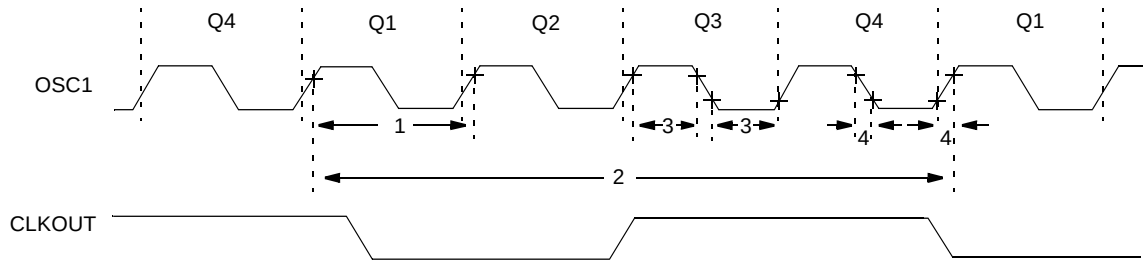


TABLE 11-1: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS		Standard Operating Conditions (unless otherwise specified) Operating Temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Parameter No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	FOSC	External CLKIN Frequency(1)	DC	—	4.0	MHz	XT Osc mode
			DC	—	20	MHz	HS Osc mode
			DC	—	200	kHz	LP Osc mode
		Oscillator Frequency(1)	DC	—	4.0	MHz	RC Osc mode
			0.1	—	4.0	MHz	XT Osc mode
			4.0	—	20	MHz	HS Osc mode
			5.0	—	200	kHz	LP Osc mode
1	TOSC	External CLKIN Period(1)	250	—	—	ns	XT Osc mode
			50	—	—	ns	HS Osc mode
			5.0	—	—	µs	LP Osc mode
		Oscillator Period(1)	250	—	—	ns	RC Osc mode
			250	—	10,000	ns	XT Osc mode
			50	—	250	ns	HS Osc mode
			5.0	—	—	µs	LP Osc mode
2	TCY	Instruction Cycle Time(2)	—	4/FOSC	—	—	
3	TosL, TosH	Clock in (OSC1) Low or High Time	50*	—	—	ns	XT oscillator
			20*	—	—	ns	HS oscillator
			2.0*	—	—	µs	LP oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	25*	ns	XT oscillator
			—	—	5*	ns	HS oscillator
			—	—	50*	ns	LP oscillator

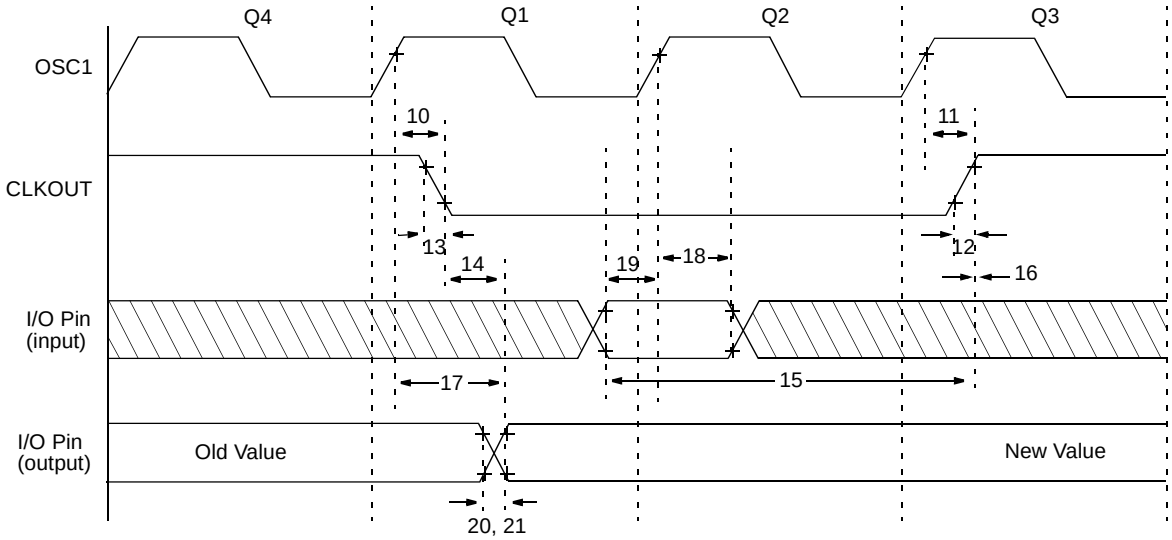
* These parameters are characterized but not tested.

† 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: All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption.
When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.

2: Instruction cycle period (TCY) equals four times the input oscillator time base period.

FIGURE 11-4: CLKOUT AND I/O TIMING – PIC16F5X



Note: Please refer to Figure 11-2 for load conditions.

TABLE 11-2: CLKOUT AND I/O TIMING REQUIREMENTS – PIC16F5X

Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units
10	TosH2ckL	OSC1↑ to CLKOUT↓ (1)	—	15	30**	ns
11	TosH2ckH	OSC1↑ to CLKOUT↑ (1)	—	15	30**	ns
12	TckR	CLKOUT rise time (1)	—	5.0	15**	ns
13	TckF	CLKOUT fall time (1)	—	5.0	15**	ns
14	TckL2ioV	CLKOUT↓ to Port out valid (1)	—	—	40**	ns
15	TioV2ckH	Port in valid before CLKOUT↑ (1)	0.25 TCY+30*	—	—	ns
16	TckH2ioI	Port in hold after CLKOUT↑ (1)	0*	—	—	ns
17	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid (2)	—	—	100*	ns
18	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	TBD	—	—	ns
19	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	TBD	—	—	ns
20	TioR	Port output rise time (2, 3)	—	10	25**	ns
20	TioR	Port output rise time (2, 4)	—	10	50**	ns
21	TioF	Port output fall time (2, 3)	—	10	25**	ns
21	TioF	Port output fall time (2, 4)	—	10	50**	ns

Legend: TBD = To Be Determined.

* These parameters are characterized but not tested.

** These parameters are design targets and are not tested. No characterization data available at this time.

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

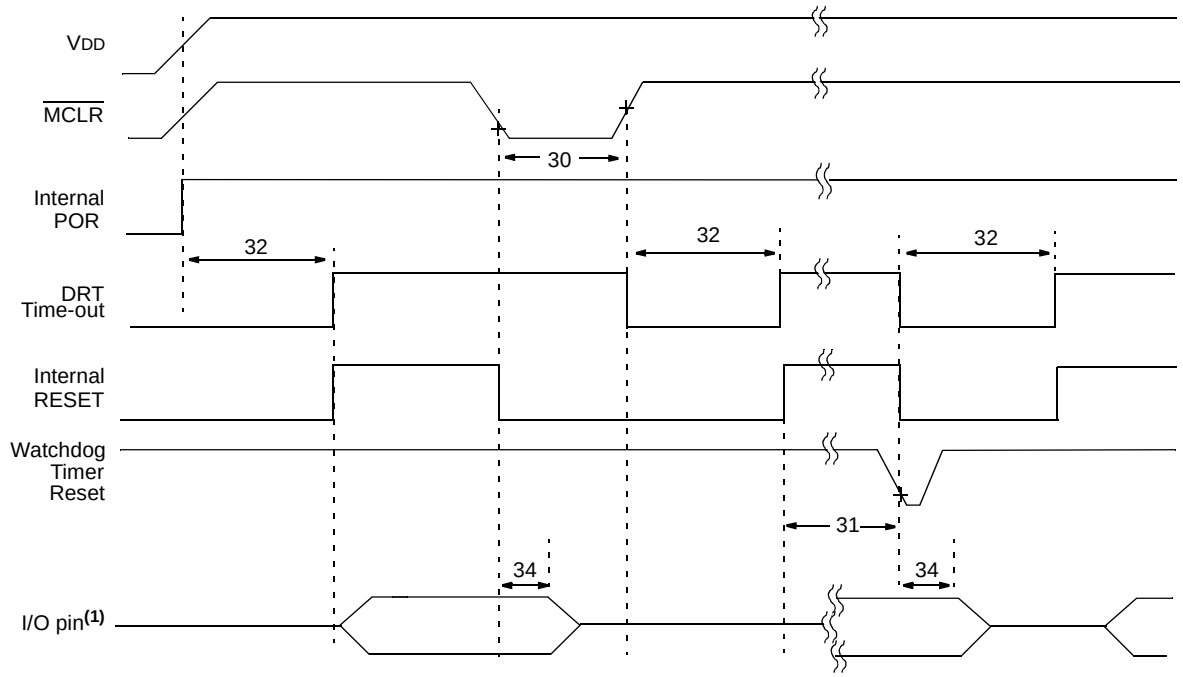
Note 1: Measurements are taken in RC mode where CLKOUT output is 4 x TOSC.

2: Please refer to Figure 11-2 for load conditions.

3: PIC16F54/57 only.

4: PIC16F59 only.

FIGURE 11-5: RESET, WATCHDOG TIMER, AND DEVICE RESET TIMER TIMING -– PIC16F5X



Note 1: Please refer to Figure 11-2 for load conditions.

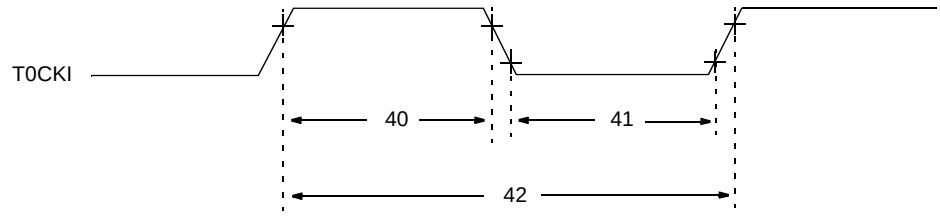
TABLE 11-3: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER – PIC16F5X

AC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise specified)** Operating Temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**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)	9.0* 9.0*	18* 18*	30* 40*	ms	VDD = 5.0V (industrial) VDD = 5.0V (extended)
32	TDRT	Device Reset Timer Period	9.0* 9.0*	18* 18*	30* 40*	ms	VDD = 5.0V (industrial) VDD = 5.0V (extended)
34	TIOZ	I/O high-impedance from $\overline{\text{MCLR}}$ Low	100*	300*	2000*	ns	

* These parameters are characterized but not tested.

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

FIGURE 11-6: TIMER0 CLOCK TIMINGS – PIC16F5X



Note: Please refer to Figure 11-2 for load conditions.

TABLE 11-4: TIMER0 CLOCK REQUIREMENTS – PIC16F5X

AC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise specified)** Operating Temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**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 $\frac{T_{CY} + 40}{N}$*	—	—	ns	Whichever is greater. N = Prescale Value (1, 2, 4,..., 256)

* These parameters are characterized but not tested.

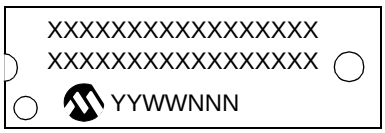
† Data in the Typical (“Typ”) column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

NOTES:

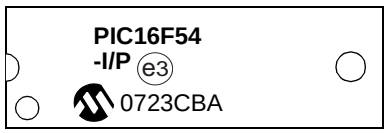
12.0 PACKAGING INFORMATION

12.1 Package Marketing Information

18-Lead PDIP



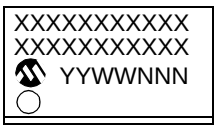
Example



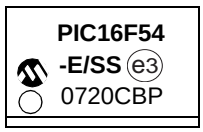
18-Lead SOIC

Example

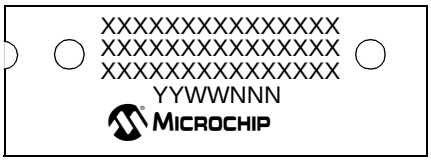
20-Lead SSOP



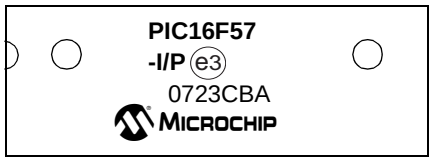
Example



28-Lead PDIP



Example



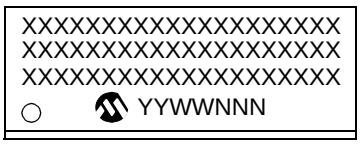
Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free 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 PIC device marking consists of Microchip part number, year code, week code, and traceability code. For PIC device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

Package Marking Information (Continued)

28-Lead SOIC

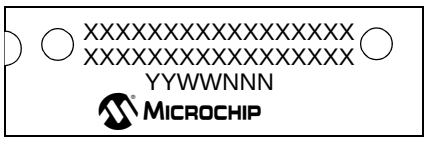


Example

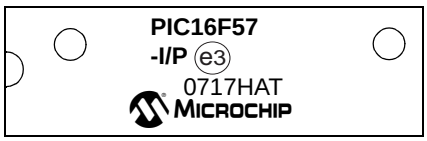
28-Lead SSOP

Example

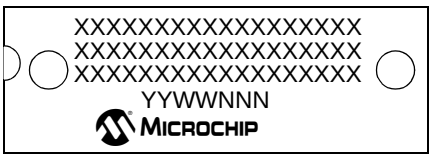
28-Lead SPDIP (.300")

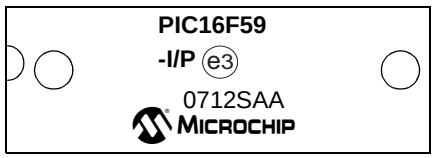


Example

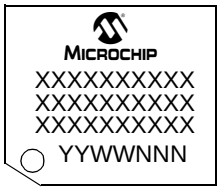


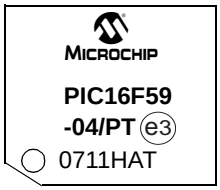
40-Lead PDIP (.600")





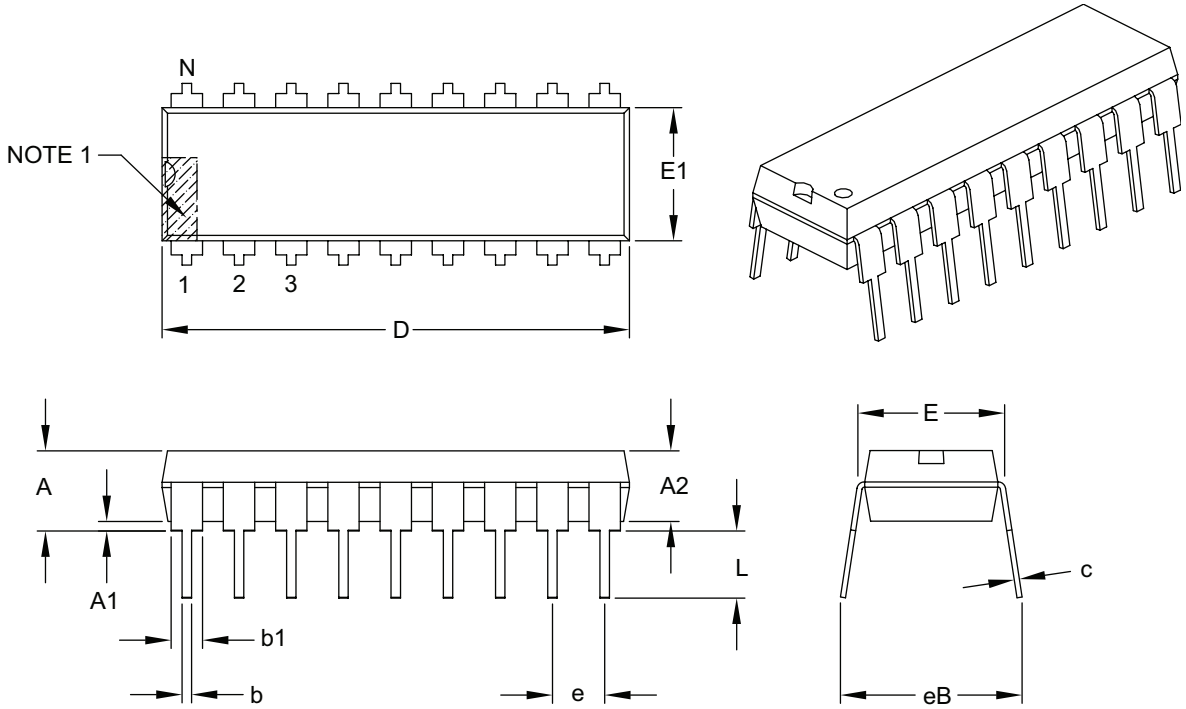
44-Lead TQFP





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

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		18	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.880	.900	.920
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.014
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

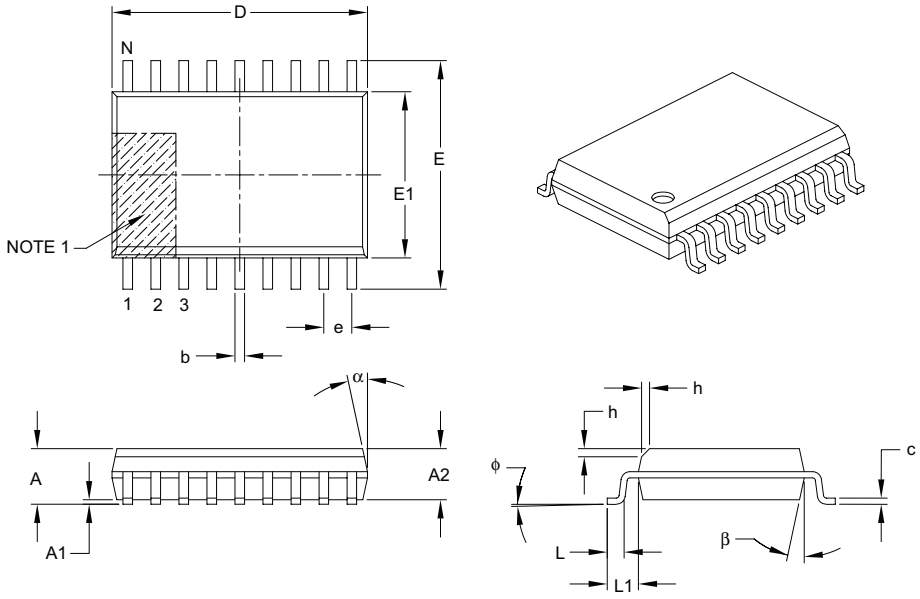
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-007B

18-Lead Plastic Small Outline (SO) – Wide, 7.50 mm Body [SOIC]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		18	
Pitch	e		1.27 BSC	
Overall Height	A	–	–	2.65
Molded Package Thickness	A2	2.05	–	–
Standoff §	A1	0.10	–	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		11.55 BSC	
Chamfer (optional)	h	0.25	–	0.75
Foot Length	L	0.40	–	1.27
Footprint	L1		1.40 REF	
Foot Angle	ϕ	0°	–	8°
Lead Thickness	c	0.20	–	0.33
Lead Width	b	0.31	–	0.51
Mold Draft Angle Top	α	5°	–	15°
Mold Draft Angle Bottom	β	5°	–	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

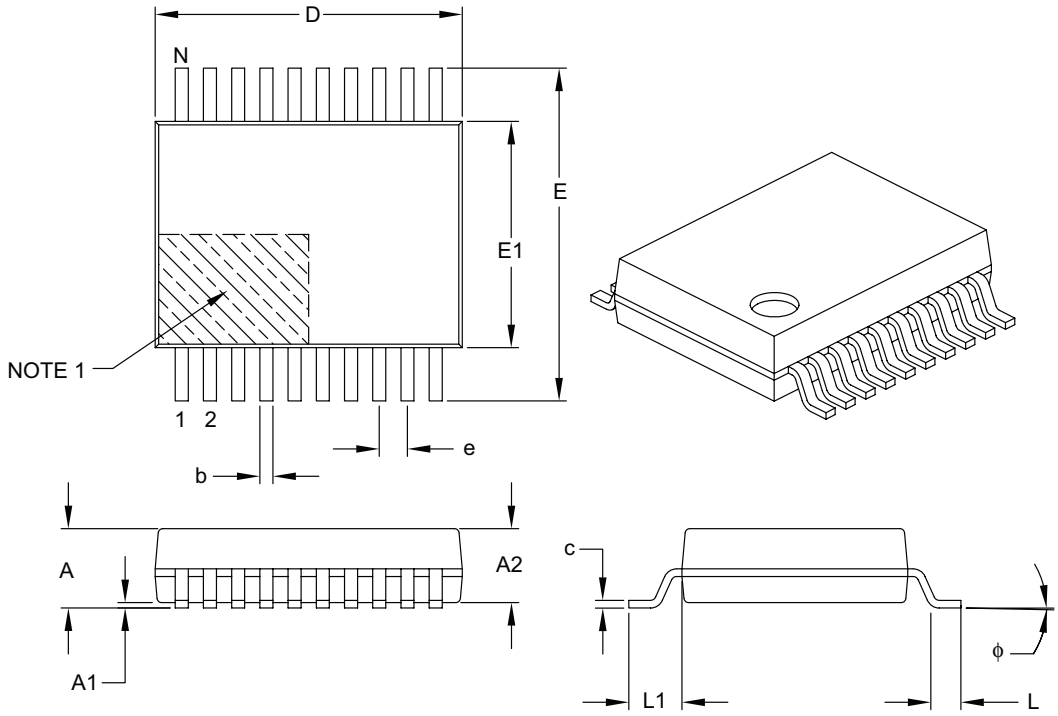
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-051B

20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

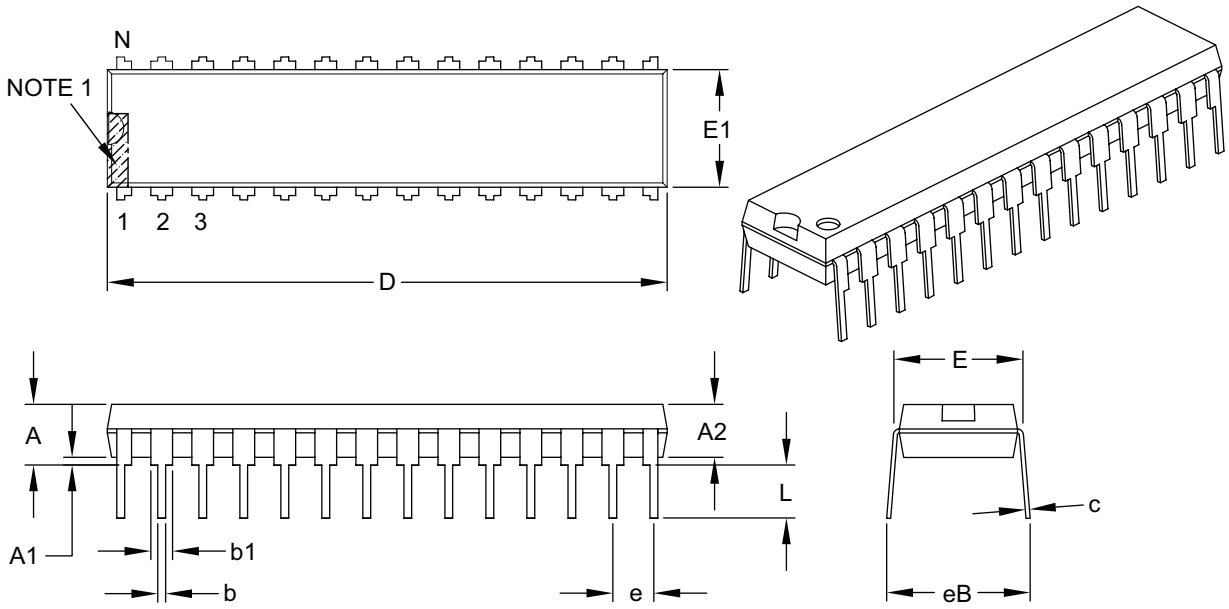
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

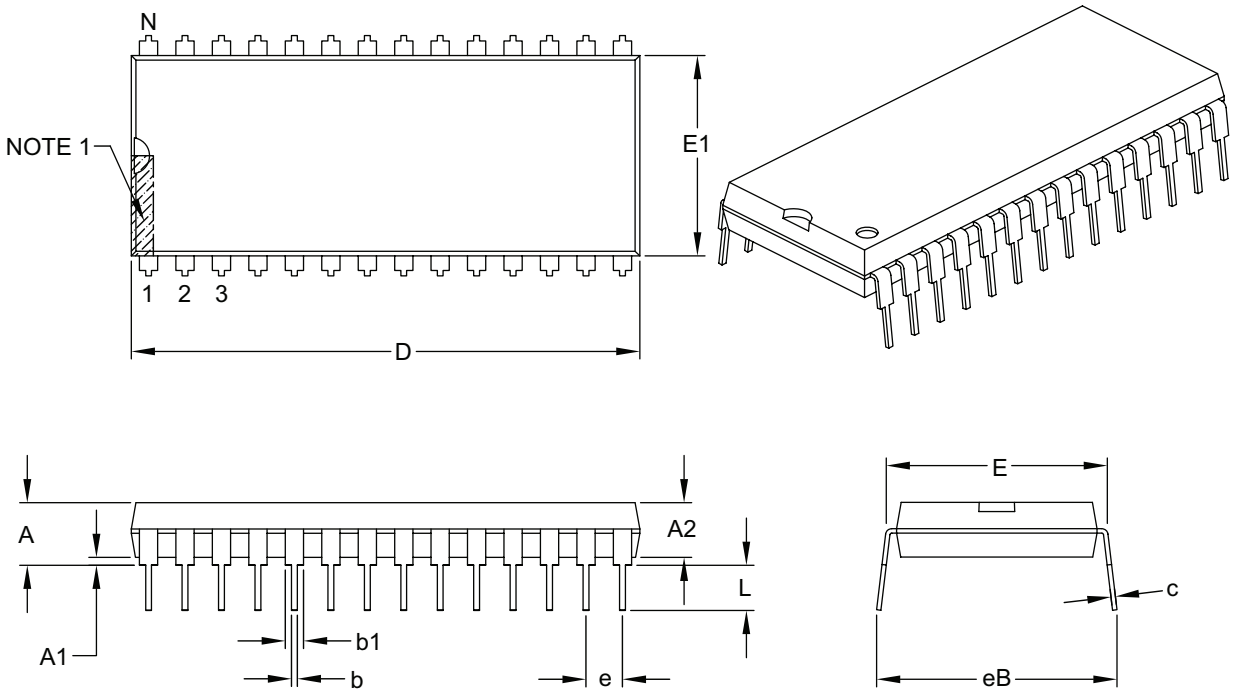
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-070B

28-Lead Plastic Dual In-Line (P) – 600 mil Body [PDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.250
Molded Package Thickness	A2	.125	–	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.590	–	.625
Molded Package Width	E1	.485	–	.580
Overall Length	D	1.380	–	1.565
Tip to Seating Plane	L	.115	–	.200
Lead Thickness	c	.008	–	.015
Upper Lead Width	b1	.030	–	.070
Lower Lead Width	b	.014	–	.022
Overall Row Spacing §	eB	–	–	.700

Notes:

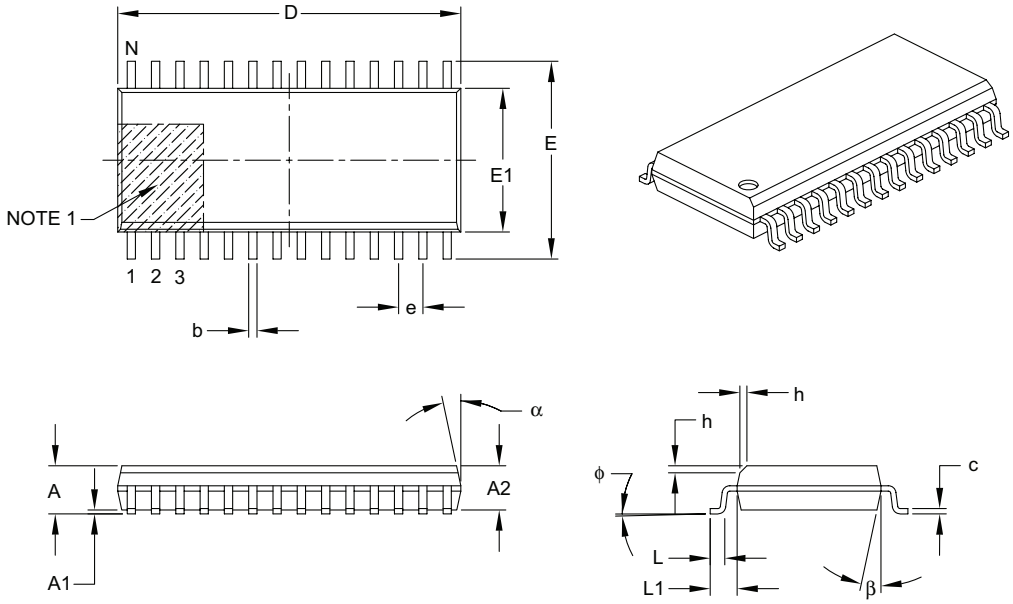
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-079B

28-Lead Plastic Small Outline (SO) – Wide, 7.50 mm Body [SOIC]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		1.27 BSC	
Overall Height	A	–	–	2.65
Molded Package Thickness	A2	2.05	–	–
Standoff §	A1	0.10	–	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		17.90 BSC	
Chamfer (optional)	h	0.25	–	0.75
Foot Length	L	0.40	–	1.27
Footprint	L1		1.40 REF	
Foot Angle Top	ϕ	0°	–	8°
Lead Thickness	c	0.18	–	0.33
Lead Width	b	0.31	–	0.51
Mold Draft Angle Top	α	5°	–	15°
Mold Draft Angle Bottom	β	5°	–	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

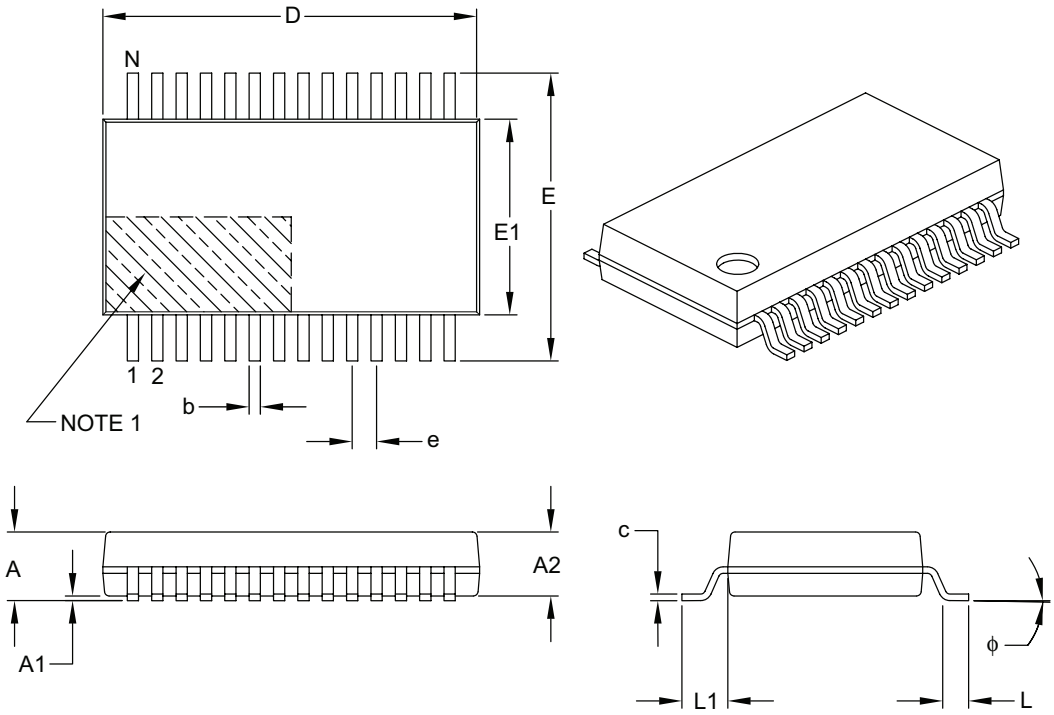
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-052B

28-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	ϕ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

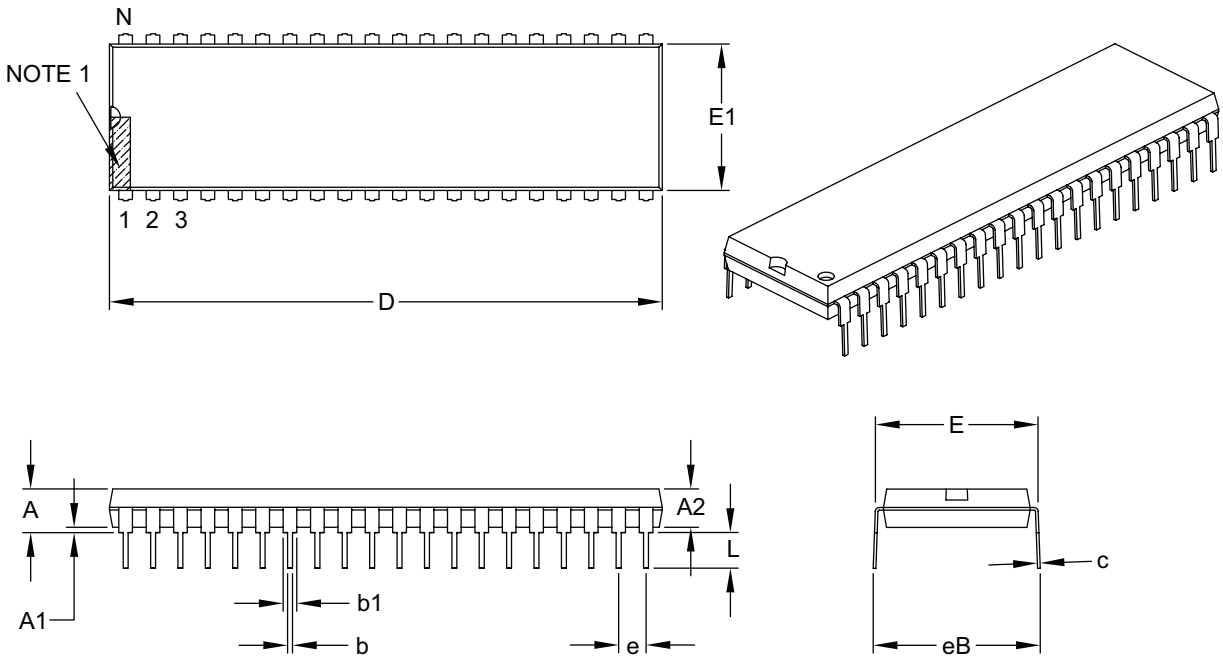
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

40-Lead Plastic Dual In-Line (P) – 600 mil Body [PDIP]

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



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		40	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.250
Molded Package Thickness	A2	.125	–	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.590	–	.625
Molded Package Width	E1	.485	–	.580
Overall Length	D	1.980	–	2.095
Tip to Seating Plane	L	.115	–	.200
Lead Thickness	c	.008	–	.015
Upper Lead Width	b1	.030	–	.070
Lower Lead Width	b	.014	–	.023
Overall Row Spacing §	eB	–	–	.700

Notes:

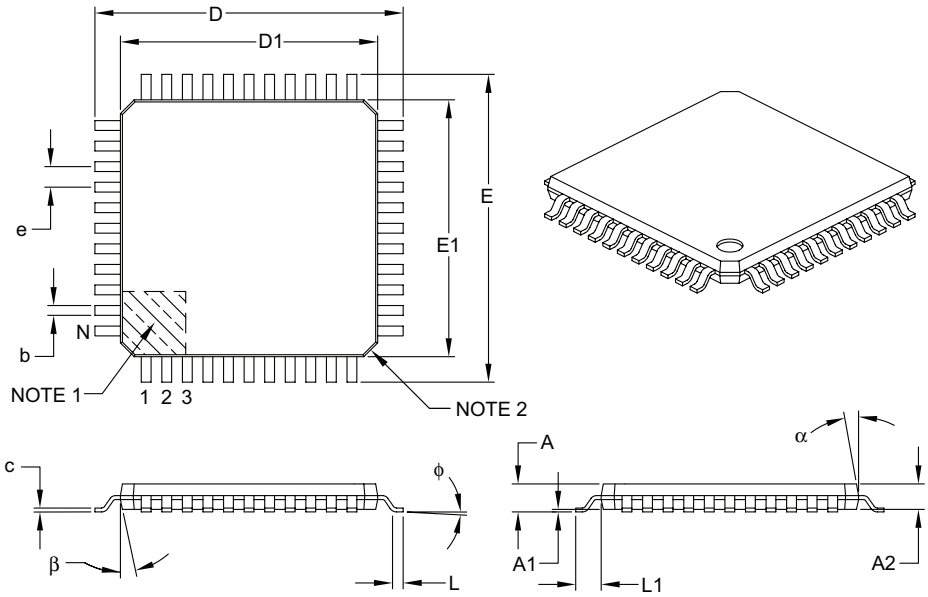
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-016B

44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm Footprint [TQFP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N		44	
Lead Pitch	e		0.80 BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	φ	0°	3.5°	7°
Overall Width	E		12.00 BSC	
Overall Length	D		12.00 BSC	
Molded Package Width	E1		10.00 BSC	
Molded Package Length	D1		10.00 BSC	
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.30	0.37	0.45
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076B

APPENDIX A: DATA SHEET REVISION HISTORY

Revision D (04/2007)

Changed PICmicro to PIC; Replaced Dev. Tool Section; Updated Package Marking Information and replaced Package Drawings (Rev. AP)

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: PIC16F5X Literature Number: DS41213D

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	PIC16F54	– VDD range 2.0V to 5.5V
	PIC16F54T(1)	– VDD range 2.0V to 5.5V
	PIC16F57	– VDD range 2.0V to 5.5V
	PIC16F57T(1)	– VDD range 2.0V to 5.5V
Temperature Range	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Extended)
Package	SO	= SOIC
	SS	= SSOP
	P	= PDIP
	SP	= Skinny Plastic DIP (SPDIP)(2)
	SOG	= SOIC (Pb-free)
	SSG	= SOIC (Pb-free)
	PG	= SOIC (Pb-free)
	SPG	= SOIC (Pb-free)
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

a) PIC16F54–I/P = Industrial temp, PDIP package
b) PIC16F54T–I/SSG = Industrial temp, SSOP package (Pb -free), tape and reel
c) PIC16F57–E/SP6 = Extended temp, Skinny Plastic DIP package (Pb-free)
d) PIC16F57T–E/SS = Extended temp, SSOP package, tape and reel
e) PIC16F54–I/SOG = Industrial temp, SOIC package (Pb-free)

Note 1: T = in tape and reel SOIC and SSOP packages only.
2: PIC16F57 only

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

Device	PIC16F59	– VDD range 2.0V to 5.5V
	PIC16F59T(1)	– VDD range 2.0V to 5.5V
Temperature Range	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Extended)
Package	P	= PDIP
	PT	= TQFP
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

a) PIC16F59–I/P = Industrial temp, PDIP package (Pb-free).
b) PIC16F59T–I/PT = Industrial temp, TQFP package (Pb-free), tape and reel.

Note 1: T = in tape and reel TQFP packages only.

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

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

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

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

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Habour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

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: 86-28-8665-5511
Fax: 86-28-8665-7889

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: 91-11-4160-8631
Fax: 91-11-4160-8632

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: 43-7242-2244-39
Fax: 43-7242-2244-393

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

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

Germany - 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

12/08/06

单击下面可查看定价，库存，交付和生命周期等信息

>>Microchip Technology