PIC16F5X
Data Sheet

Flash-Based, 8-Bit CMOS Microcontrollers

Note the following details of the code protection feature on Microchip devices:

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

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

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, MXDEV, MXLAB, PICMASTER, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2004, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

♻ Printed on recycled paper.

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

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
- General Purpose Registers (SRAM):
  - 25 x 8 on PIC16F54
  - 72 x 8 on PIC16F57

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

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

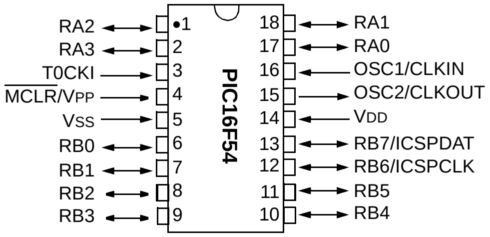
CMOS Technology:

- 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

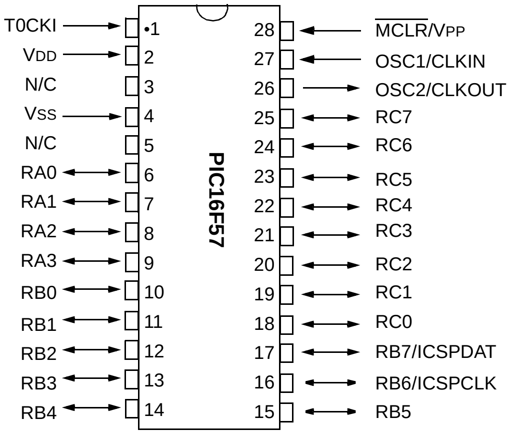
Device	Program Memory	Data Memory	I/O	Timers 8-bit
	Flash (words)	SRAM (bytes)		
PIC16F54	512	25	12	1
PIC16F57	2048	72	20	1

Pin Diagrams

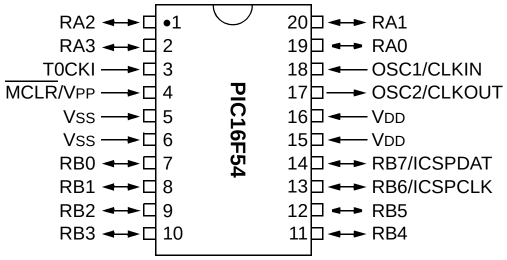
PDIP, SOIC



PDIP, SOIC



SSOP



SSOP

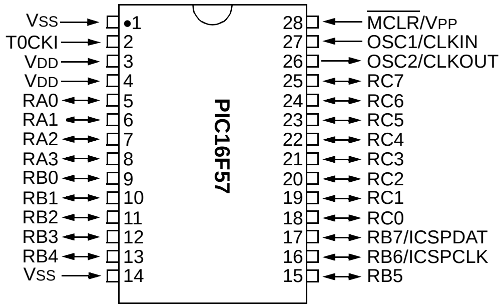


Table of Contents

1.0 General Description 5
2.0 Architectural Overview 7
3.0 Memory Organization 13
4.0 Oscillator Configurations 21
5.0 Reset 23
6.0 I/O Ports 29
7.0 Timer0 Module and TMR0 Register 31
8.0 Special Features of the CPU 35
9.0 Instruction Set Summary 39
10.0 Development Support 51
11.0 Electrical Specifications 57
12.0 DC and AC Characteristics Graphs and Tables 67
13.0 Packaging Information 69
Index 79
On-Line Support 81
Systems Information and Upgrade Hot Line 81
Reader Response 82
Product Identification System 83

TO OUR VALUED CUSTOMERS

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

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

Most Current Data Sheet

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

http://www.microchip.com

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

Errata

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

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

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

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

Customer Notification System

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

NOTES:

1.0 GENERAL DESCRIPTION

The 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 in 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, an in-circuit emulator, 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
Maximum Operation Frequency	20 MHz	20 MHz
Flash Program Memory (x12 words)	512	2K
RAM Data Memory (bytes)	25	72
Timer Module(s)	TMR0	TMR0
I/O Pins	12	20
Number of Instructions	33	33
Packages	18-pin PDIP, SOIC; 20-pin SSOP	28-pin PDIP, SOIC; 28-pin SSOP

All PICmicro® 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 and the PIC16F57 addresses 2 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 (PC), 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 $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ 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) and Table 2-2 (for PIC16F57).

FIGURE 2-1: PIC16F5X SERIES BLOCK DIAGRAM

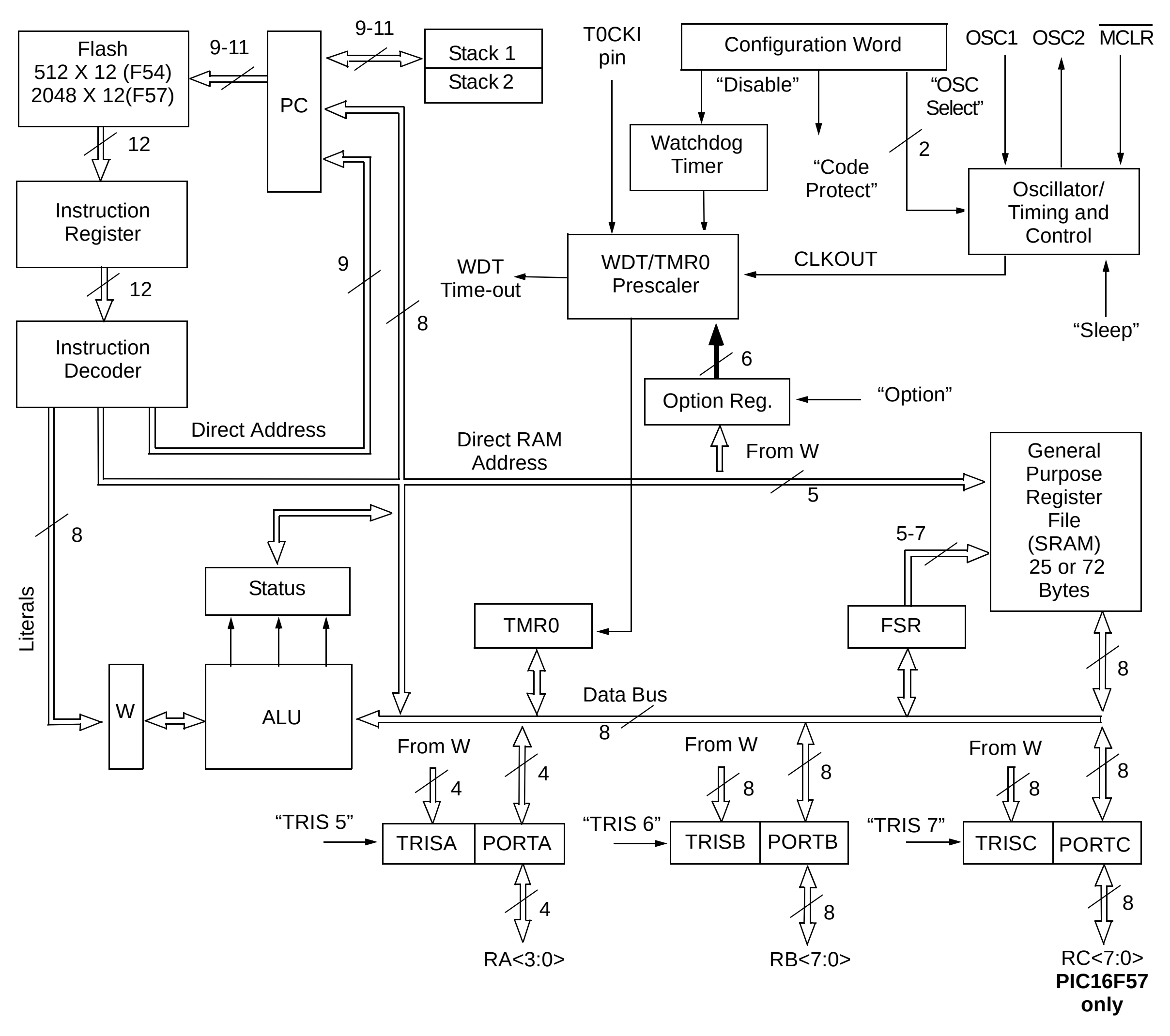


TABLE 2-1: PIC16F54 PINOUT DESCRIPTION

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.
$\overline{\text{MCLR}}$/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 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 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.
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

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 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.

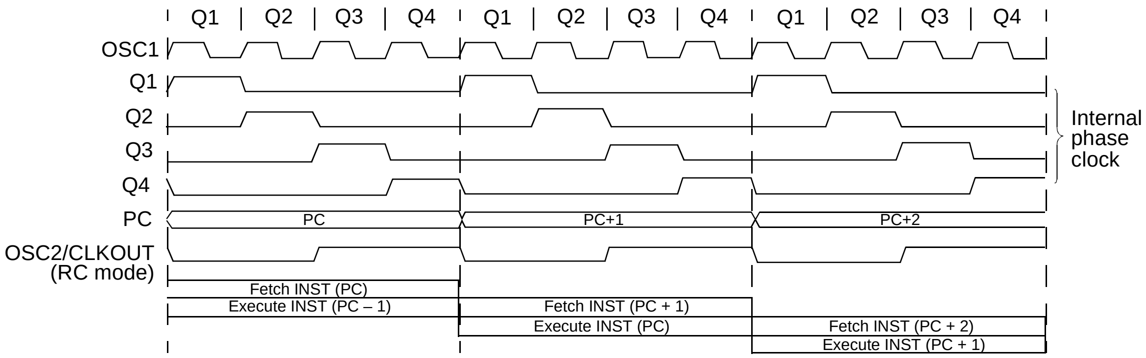
2.2 Instruction Flow/Pipelining

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

1. `MOVLW H'55'`	Fetch 1	Execute 1				
2. `MOVWF PORTB`		Fetch 2	Execute 2			
3. `CALL SUB_1`			Fetch 3	Execute 3		
4. `BSF PORTA, BIT3`				Fetch 4	Flush	
					Fetch SUB_1	Execute SUB_1

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

NOTES:

3.0 MEMORY ORGANIZATION

PIC16F5X memory is organized into program memory and data memory. For the PIC16F57, which has 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, which has 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 has an 11-bit program counter capable of addressing a 2 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 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

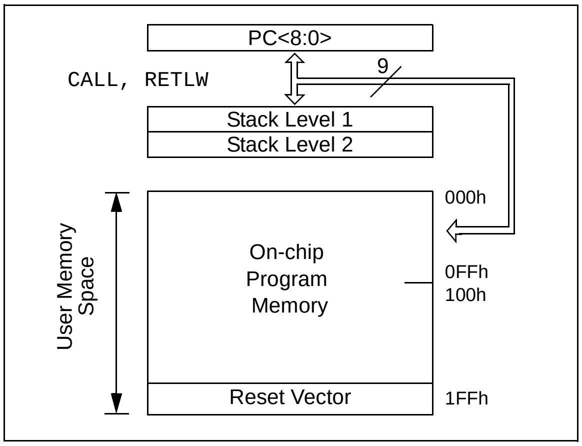
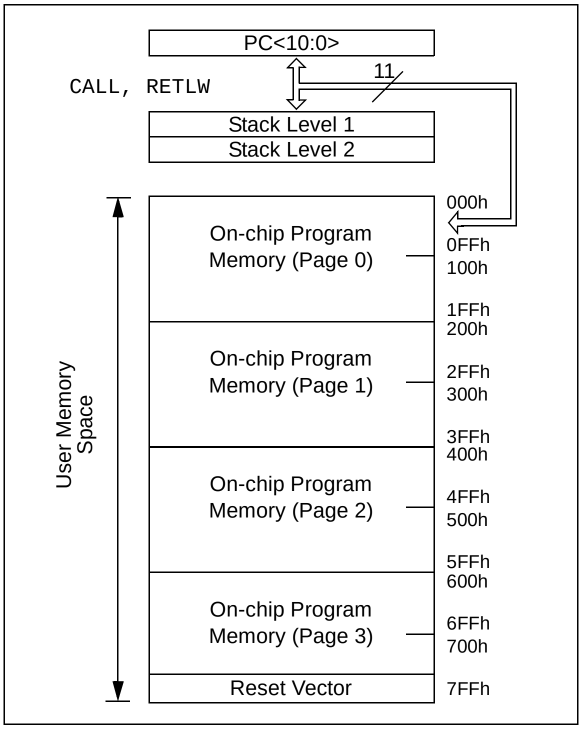


FIGURE 3-2: PIC16F57 PROGRAM MEMORY MAP AND STACK



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.

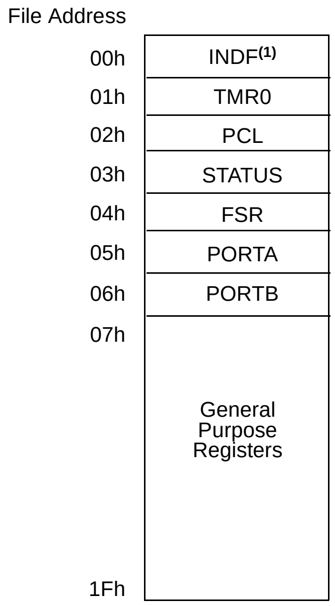
For the PIC16F54, the register file is composed of 7 Special Function Registers and 25 General Purpose Registers (Figure 3-3).

For the PIC16F57, the register file is composed of 8 Special Function Registers, 18 General Purpose Registers and 64 additional General Purpose Registers that may be addressed using a banking scheme (Figure 3-4).

3.2.1 GENERAL PURPOSE REGISTER FILE

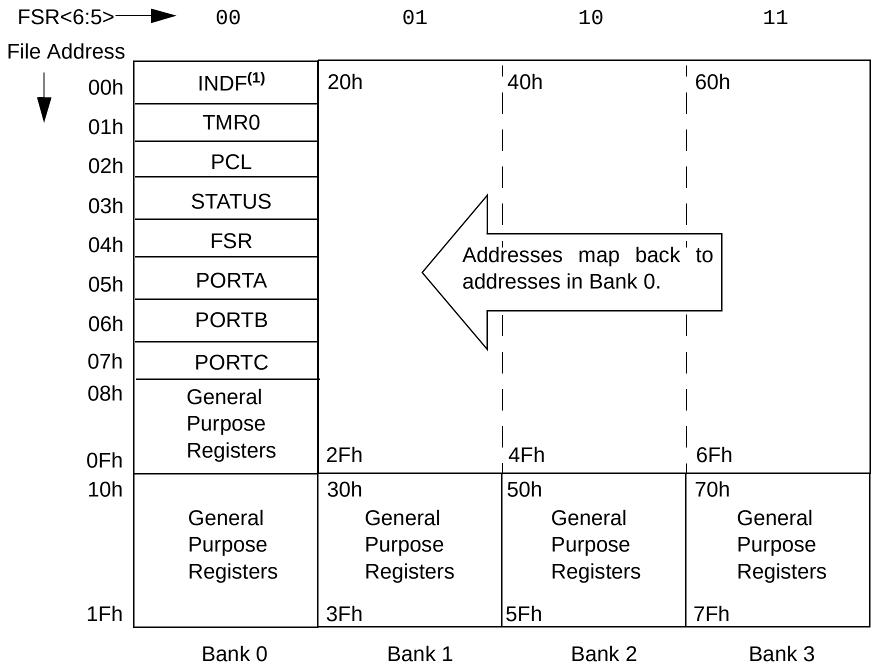
The register file is accessed either directly or indirectly through the File Select Register (FSR). The FSR Register is described in **Section 3.7 "Indirect Data Addressing; INDF and FSR Registers"**.

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"**.

3.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers (FSR) 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)								`1111 1111`	29
N/A	OPTION	Contains control bits to configure Timer0 and Timer0/WDT prescaler								`--11 1111`	17
00h	INDF	Uses Contents of FSR to Address Data Memory (not a physical register)								`xxxx xxxx`	19
01h	TMR0	Timer0 Module Register								`xxxx xxxx`	32
02h$^{(1)}$	PCL	Low order 8 bits of PC								`1111 1111`	18
03h	STATUS	PA2	PA1	PA0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	`0001 1xxx`	16
04h	FSR	Indirect Data Memory Address Pointer								`1xxx xxxx`$^{(1)}$	19
05h$^{(4)}$	PORTA	—	—	—	—	RA3	RA2	RA1	RA0	`---- xxxx`	29
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	`xxxx xxxx`	29
07h$^{(2)}$	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	`xxxx 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: These values are valid for PIC16F57. For the PIC16F54, the value on Reset is ‘`111x xxxx`’ and for $\overline{\text{MCLR}}$ and WDT Reset, the value is ‘`111u uuuu`’.

4: Unimplemented bits are read as ‘`0`’s.

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 $\overline{\text{TO}}$ and $\overline{\text{PD}}$ 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).

It is recommended, therefore, 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	$\overline{\text{TO}}$	$\overline{\text{PD}}$	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)
`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 since this may affect upward compatibility with future products.

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

bit 3 **$\overline{\text{PD}}$**: 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/$\overline{\text{borrow}}$ bit (for `ADDWF` and `SUBWF` instructions)
`ADDWF`**:**
`1` = A carry from the 4th low order bit of the result occurred
`0` = A carry from the 4th low order bit of the result did not occur
`SUBWF`**:**
`1` = A borrow from the 4th low order bit of the result did not occur
`0` = A borrow from the 4th low order bit of the result occurred

bit 0 **C**: Carry/$\overline{\text{borrow}}$ 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 Program Counter (PCL) is mapped to PC<7:0> (Figure 3-5 and Figure 3-6).

For the PIC16F57, 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 3-5 and Figure 3-6).

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 3-5 and Figure 3-6).

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

For the PIC16F57, 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 3-5 and Figure 3-6).

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

FIGURE 3-5: LOADING OF PC BRANCH INSTRUCTIONS – PIC16F54

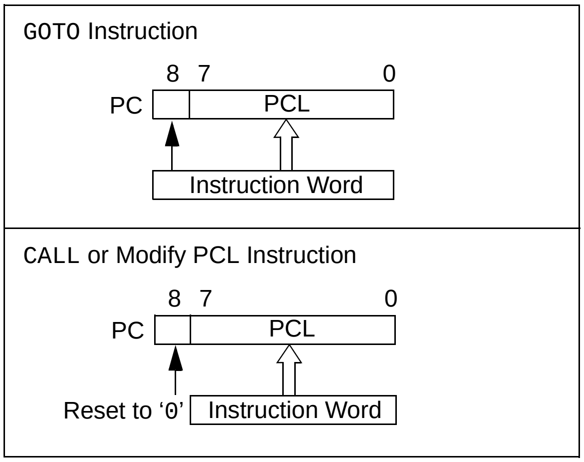
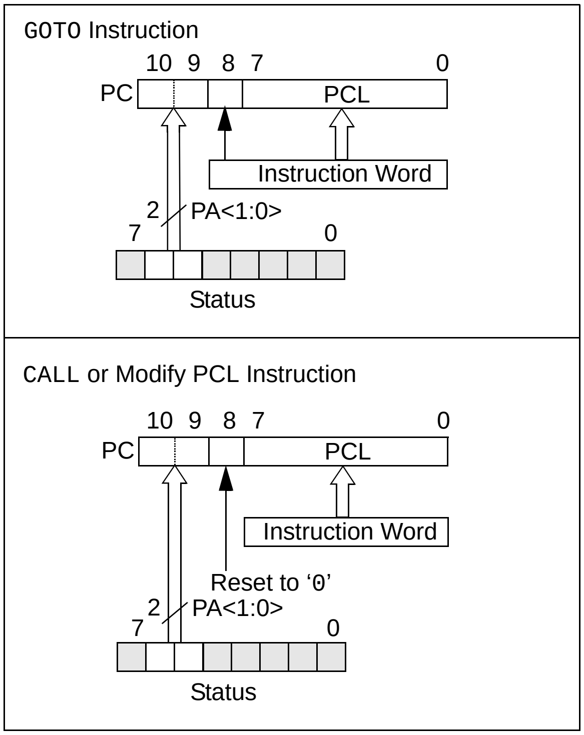


FIGURE 3-6: LOADING OF PC BRANCH INSTRUCTIONS – PIC16F57



3.5.1 PAGING CONSIDERATIONS – PIC16F57

If the Program Counter 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 program counter 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 and the PIC16F57 device has 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 `CALLs` are executed, only the most recent two return addresses are stored.

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

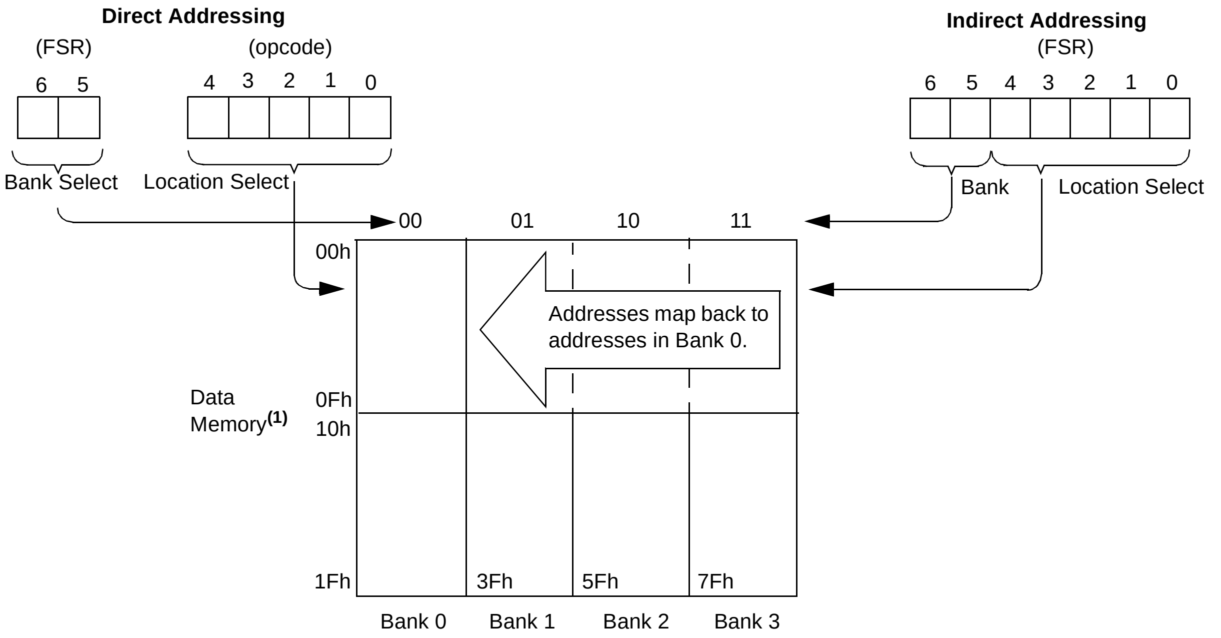
The FSR is either a 5-bit (PIC16F54) or 7-bit (PIC16F57) wide register. It is used in conjunction with the INDF register to indirectly address the data memory area.

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

PIC16F54: These do not use banking. FSR<6:5> bits are unimplemented and read as ‘1’s.

PIC16F57: 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).

FIGURE 3-7: DIRECT/INDIRECT ADDRESSING



Note 1: For register map detail, see **Section 3.2 “Data Memory Organization”**.

4.0 OSCILLATOR CONFIGURATIONS

4.1 Oscillator Types

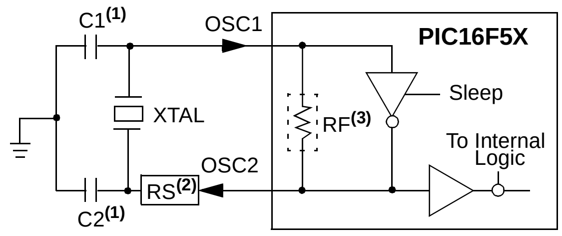
PIC16F5X devices can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1:FOSC0) to select one of these four modes:

- 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)

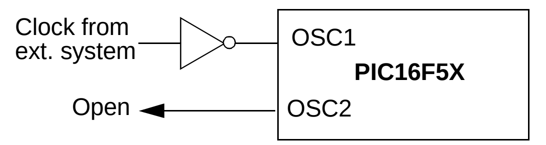


TABLE 4-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS – PIC16F5X

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 – PIC16F5X

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

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 specification. Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values of external components.

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

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

Figure 4-1 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: EXAMPLE OF EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT (USING XT, HS OR LP OSCILLATOR MODE)

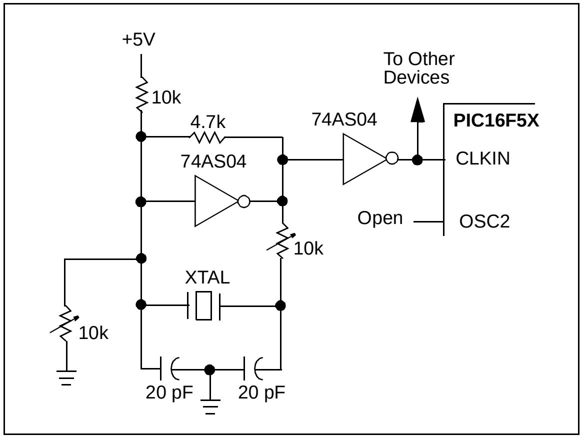
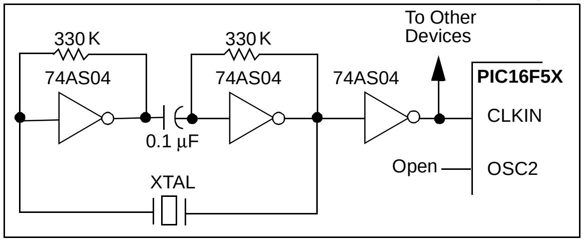


Figure 4-2 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverters perform a 360° phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

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



4.4 RC Oscillator

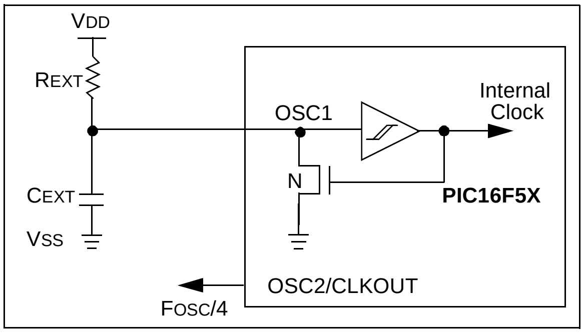
For applications where precise timing is not a requirement, the RC oscillator option is available. The operation and functionality of the RC oscillator is dependent upon a number of variables. The RC oscillator frequency is a function of:

- 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

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

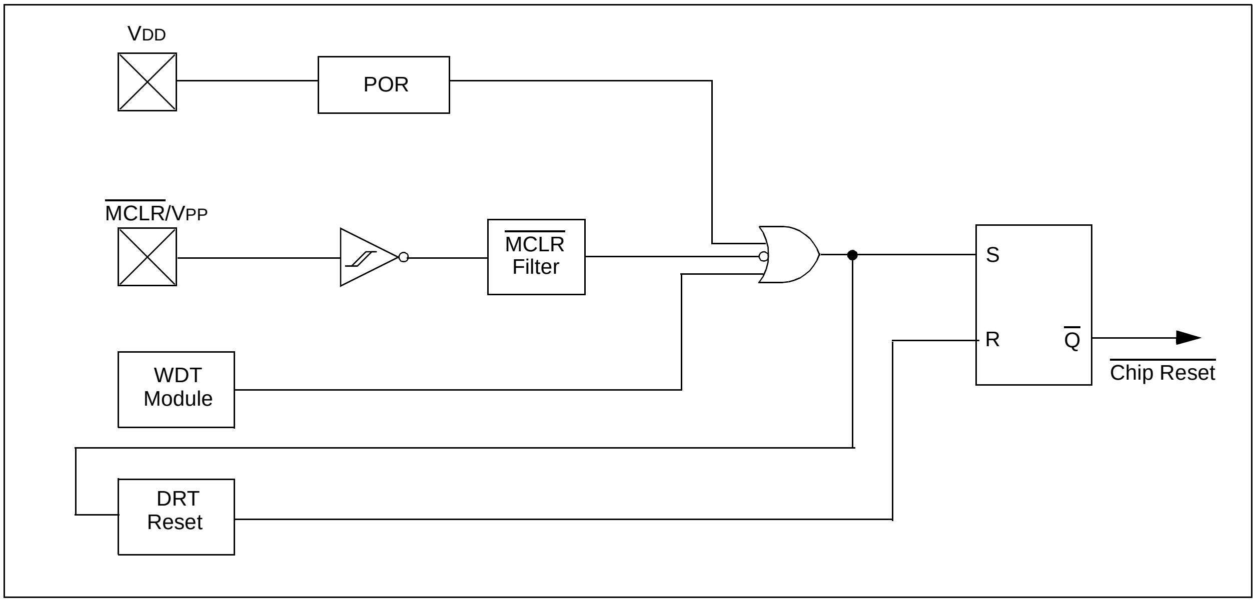
Register	Address	Power-on Reset	$\overline{\text{MCLR}}$ or WDT Reset
W	N/A	`xxxx xxxx`	`uuuu uuuu`
TRIS	N/A	`1111 1111`	`1111 1111`
OPTION	N/A	`--11 1111`	`--11 1111`
INDF	00h	`xxxx xxxx`	`uuuu uuuu`
TMR0	01h	`xxxx xxxx`	`uuuu uuuu`
PCL	02h	`1111 1111`	`1111 1111`
STATUS	03h	`0001 1xxx`	`000q quuu`
FSR(1)	04h	`1xxx xxxx`	`1uuu uuuu`
PORTA	05h	`---- xxxx`	`---- uuuu`
PORTB	06h	`xxxx xxxx`	`uuuu uuuu`
PORTC(2)	07h	`xxxx xxxx`	`uuuu uuuu`
General Purpose Register Files	08-7Fh	`xxxx xxxx`	`uuuu uuuu`

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

Note 1: These values are valid for PIC16F57. For the PIC16F54, the value on Reset is `111x xxxx` and for $\overline{\text{MCLR}}$ and WDT Reset, the value is `111u uuuu`.

2: General Purpose Register file on PIC16F54.

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



5.1 Power-on Reset (POR)

The PIC16F5X Family incorporates 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.

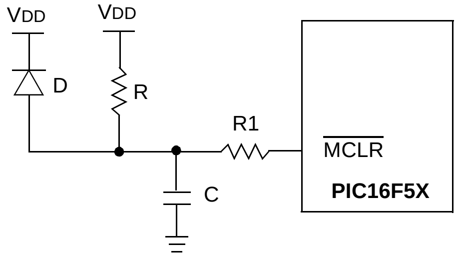
In Figure 5-4, the on-chip Power-on Reset feature is being used (MCLR and VDD are tied together). The VDD is stable before the start-up timer times out and there is no problem in getting a proper Reset. However, Figure 5-5 depicts a problem situation where VDD rises too slowly. The time between when the DRT senses a high on the MCLR/VPP pin, and when 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:	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.

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

The POR circuit does not produce an internal Reset when VDD declines.

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 ($\overline{\text{MCLR}}$ NOT TIED TO VDD)

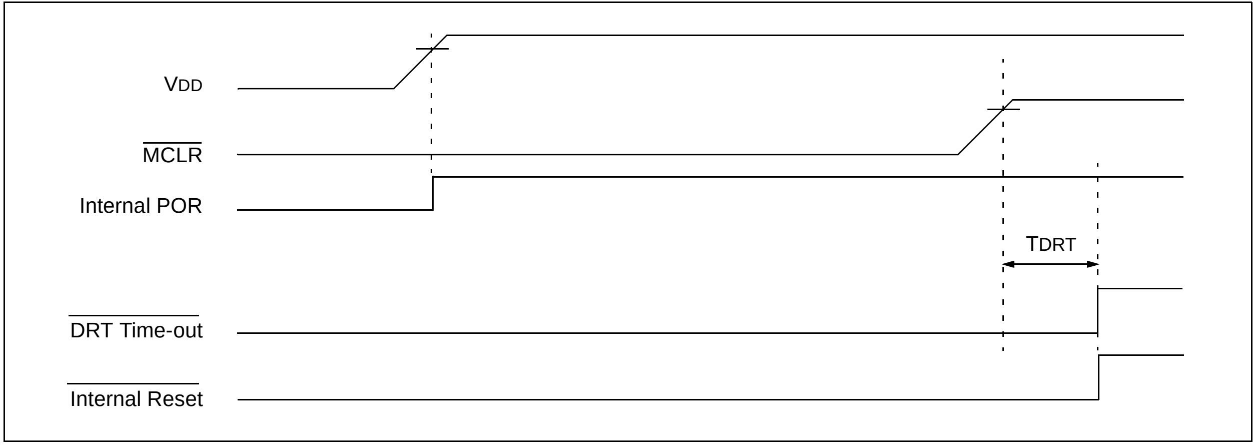


FIGURE 5-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO VDD): FAST VDD RISE TIME

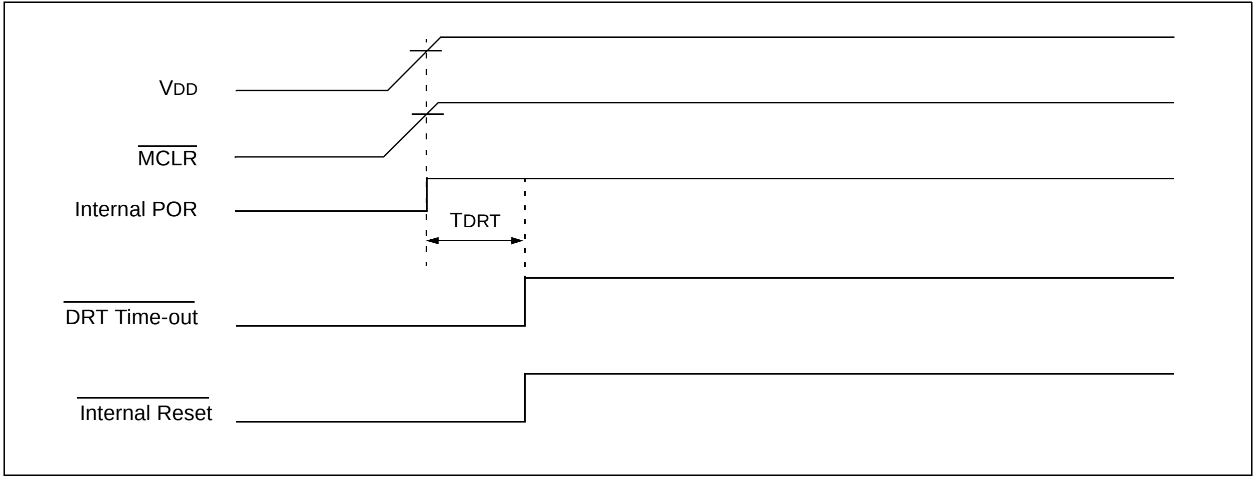
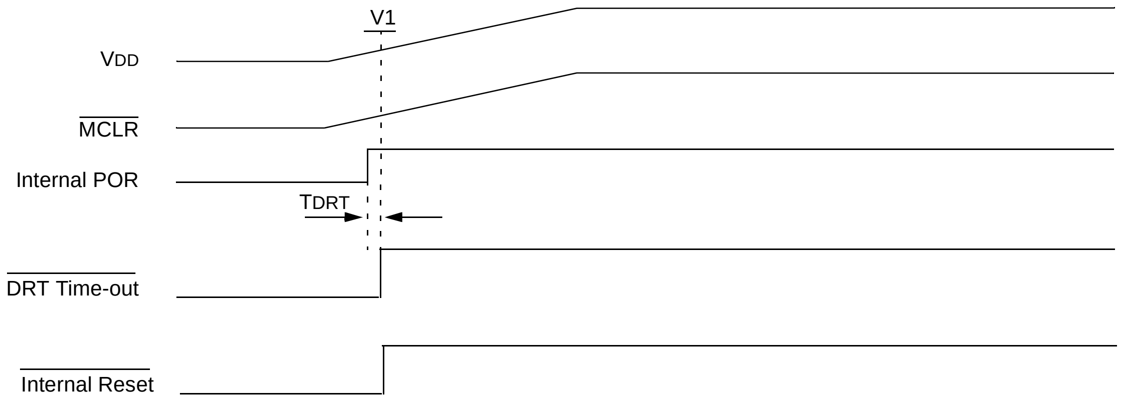


FIGURE 5-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO VDD): SLOW VDD RISE TIME



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

5.2 Device Reset Timer (DRT)

The Device Reset Timer (DRT) provides an 18 ms nominal time-out on Reset regardless of 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 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 $\overline{\text{MCLR}}$/V_{PP} pin has reached a logic high (V_{IH}) level. Thus, external RC networks connected to the $\overline{\text{MCLR}}$ 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.

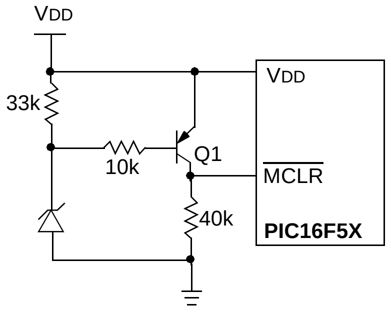
The DRT will also be triggered upon a Watchdog Timer time-out. This is particularly important for applications using the WDT to wake the PIC16F5X from Sleep mode automatically.

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.

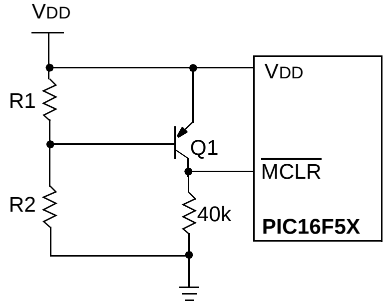
To reset PIC16F5X devices when a brown-out occurs, external brown-out protection circuits may be built, as shown in Figure 5-6, Figure 5-7 and Figure 5-8.

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



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

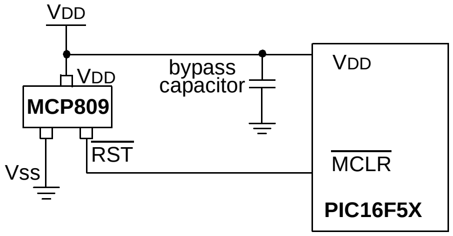
FIGURE 5-7: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2



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) are all set.

6.1 PORTA

PORTA is a 4-bit I/O register. Only the low order 4 bits are used (RA<3:0>). Bits 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.

PORTC is a general purpose register for the PIC16F54.

6.4 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.

6.5 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 and TRISC) 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

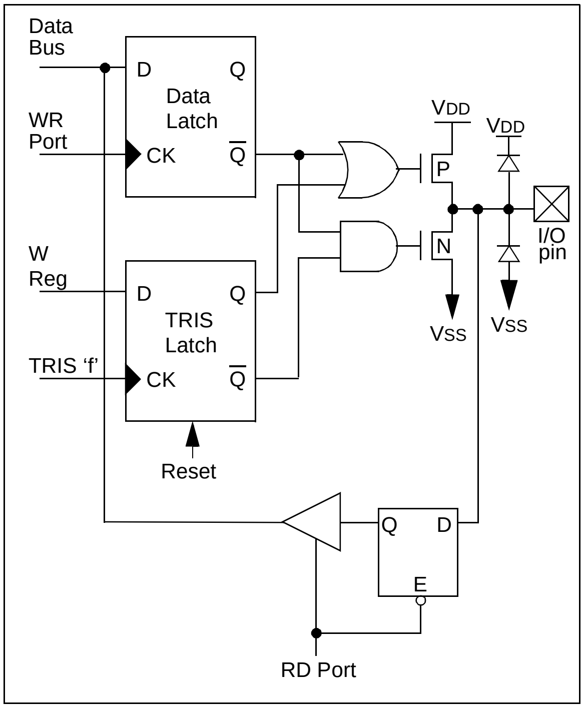


TABLE 6-1: SUMMARY OF PORT REGISTERS

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 $\overline{\text{MCLR}}$ and WDT Reset
N/A	TRIS	I/O Control Registers (TRISA, TRISB, TRISC)								`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`

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.

6.6 I/O Programming Considerations

6.6.1 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.

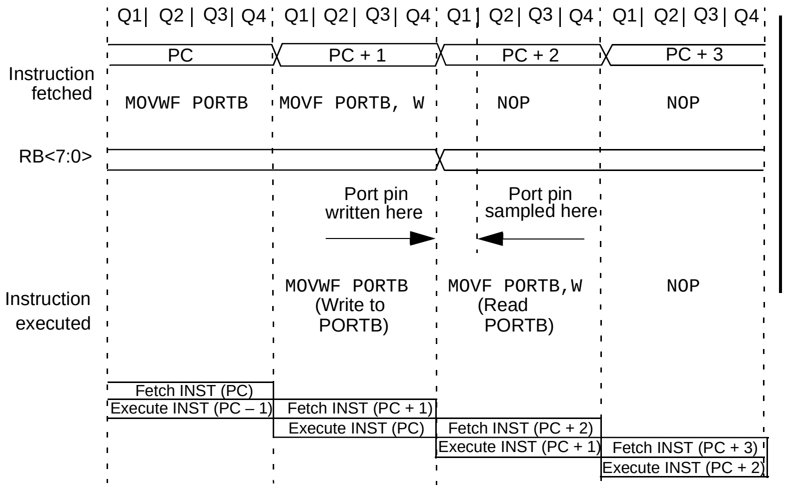
EXAMPLE 6-1: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
;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).
```

6.6.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 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



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

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.

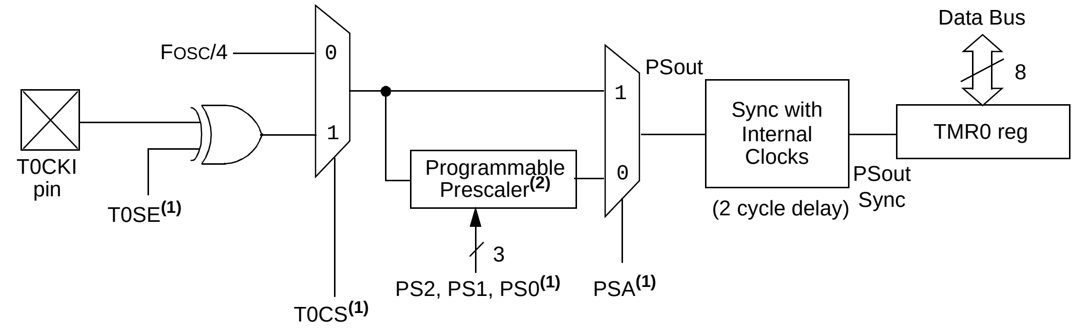
Counter mode is selected by setting the T0CS bit (Option<5>). In this mode, Timer0 will increment either on every rising or falling edge of pin T0CKI. The incrementing edge is determined by the source edge select bit T0SE (Option<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the 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

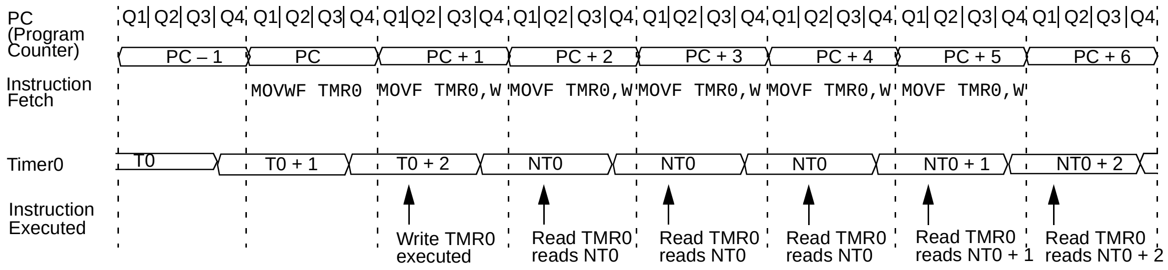


FIGURE 7-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALER 1:2

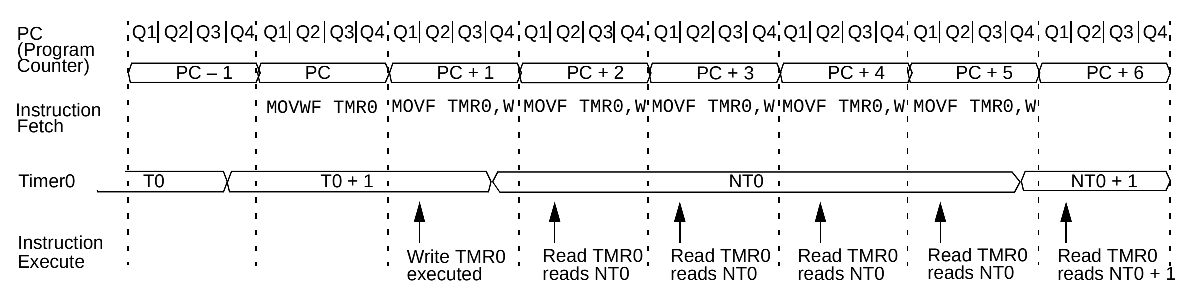


TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER0

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

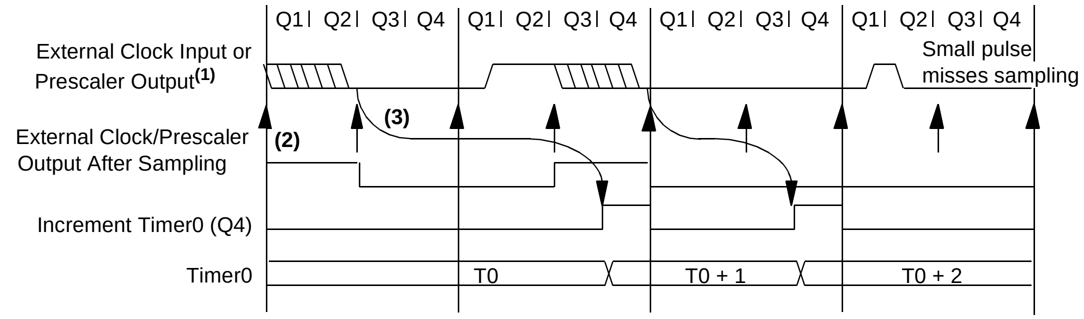
When no prescaler is used, the external clock is the Timer0 input. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 7-4). Therefore, it is necessary for T0CKI to be high for at least 2 TOSC (and a small RC delay of 20 ns) and low for at least 2 TOSC (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 4 TOSC (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 3 TOSC to 7 TOSC (duration of Q = TOSC). Therefore, the error in measuring the interval between two edges on Timer0 input = ± 4 TOSC max.

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
```

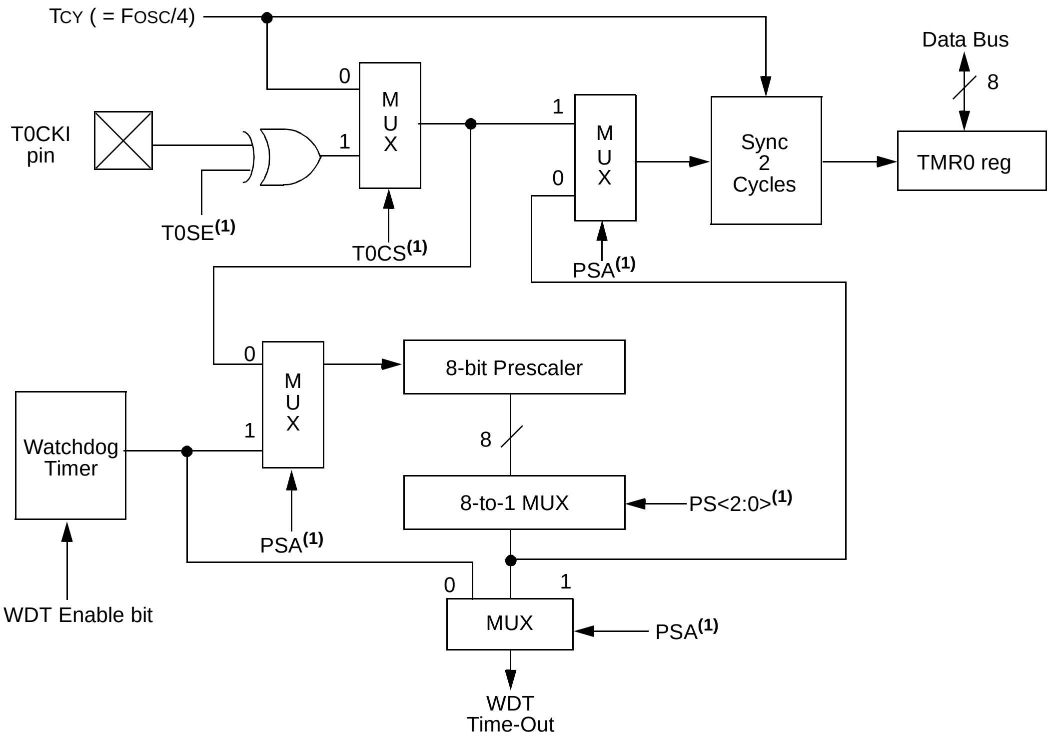
To change prescaler from the WDT to the Timer0 module, use the sequence shown in Example 7-2. This sequence must be used even if the WDT is disabled. A `CLRWDT` instruction should be executed before switching the prescaler.

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 (**Section 4.0 "Oscillator Configurations"**)
- Reset (**Section 5.0 "Reset"**)
- Power-on Reset (**Section 5.1 "Power-on Reset (POR)"**)
- Device Reset Timer (**Section 5.2 "Device Reset Timer (DRT)"**)
- Watchdog Timer (WDT) (**Section 8.2 "Watchdog Timer (WDT)"**)
- Sleep (**Section 8.3 "Power-Down Mode (Sleep)"**)
- Code protection (**Section 8.4 "Program Verification/Code Protection"**)
- User ID locations (**Section 8.5 "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 PIC16F54 and PIC16F57 devices (Register 8-1).

REGISTER 8-1: CONFIGURATION WORD FOR PIC16F54/57

—	—	—	—	—	—	—	—	$\overline{\text{CP}}$	WDTE	FOSC1	FOSC0
bit 11											bit 0

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

bit 3: **$\overline{\text{CP}}$:** 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 programming specifications "*PIC16F54 Programming Specification*" (DS41207) and "*PIC16F57 Programming Specification*" (DS41208) to determine how to access the Configuration Word register. 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).

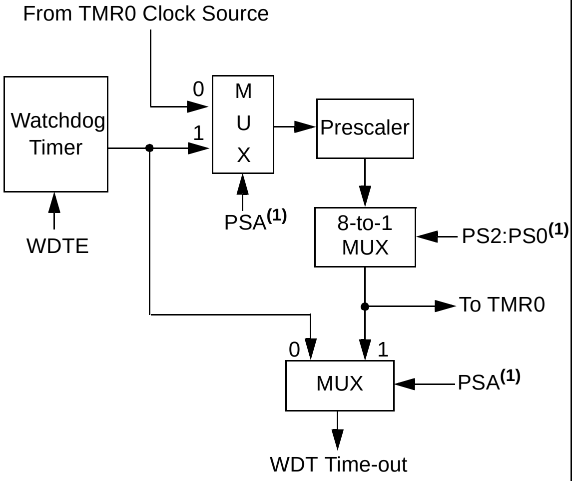
Under worst case conditions (V_{DD} = Min., Temperature = Max., WDT prescaler = 1:128), it may take several seconds before a WDT time-out occurs.

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, PS2:PS0 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 $\overline{MCLR}$ and WDT Reset
N/A	OPTION	—	—	TOCS	TOSE	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 $\overline{\text{TO}}$ bit (Status<4>) is set, the $\overline{\text{PD}}$ 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 $\overline{\text{MCLR}}$/V_{PP} pin low.

For lowest current consumption while powered down, the T0CKI input should be at V_{DD} or V_{SS} and the $\overline{\text{MCLR}}$/V_{PP} pin must be at a logic high level ($\overline{\text{MCLR}}$ = 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 $\overline{\text{MCLR}}$/V_{PP} pin.
2. A Watchdog Timer time-out Reset (if WDT was enabled).

Both of these events cause a device Reset. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits can be used to determine the cause of device Reset. The $\overline{\text{TO}}$ bit is cleared if a WDT time-out occurred (and caused wake-up). The $\overline{\text{PD}}$ 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^{SM} requests. This pattern number will be unique and traceable to the submitted code.

8.6 In-Circuit Serial Programming™ (ICSP™)

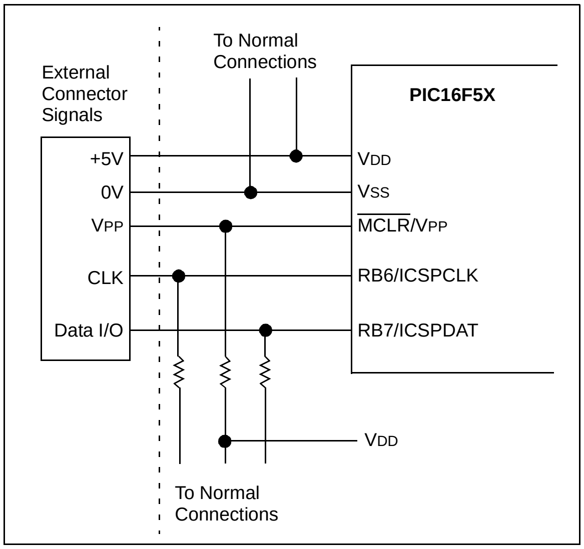
The PIC16F5X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program 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 $\overline{\text{MCLR}}$ (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 Programming Specification"* (DS41207) and "*PIC16F57 Programming Specification" (*DS41208).

A typical In-Circuit Serial Programming connection is shown in Figure 8-2.

FIGURE 8-2: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



NOTES:

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
$\overline{\mathtt{TO}}$	Time-out bit
$\overline{\mathtt{PD}}$	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

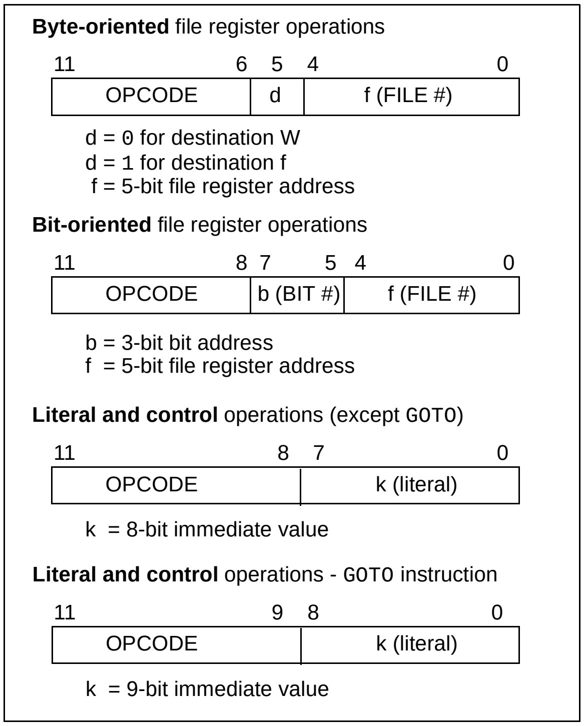


TABLE 9-2: INSTRUCTION SET SUMMARY

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	Call Subroutine	2	`1001`	`kkkk`	`kkkk`	None	**1**
CLRWDT	—	Clear Watchdog Timer	1	`0000`	`0000`	`0100`	$\overline{\text{TO}}$, $\overline{\text{PD}}$	
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	k	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`	$\overline{\text{TO}}$, $\overline{\text{PD}}$	
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`
<u>Example</u>:	`ADDWF TEMP_REG, 0`

Before Instruction
W = 0x17
TEMP_REG = 0xC2
After Instruction
W = 0xD9
TEMP_REG = 0xC2

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`
<u>Example</u>:	`ANDLW H’5F’`

Before Instruction
W = 0xA3
After Instruction
W = 0x03

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`
<u>Example</u>:	`ANDWF TEMP_REG, 1`

Before Instruction
W = 0x17
TEMP_REG = 0xC2
After Instruction
W = 0x17
TEMP_REG = 0x02

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`
<u>Example</u>:	`BCF FLAG_REG, 7`

Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BSF	**Bit Set f**
Syntax:	[*label*] BSF f,b
Operands:	$0 \le f \le 31$ $0 \le b \le 7$
Operation:	$1 \rightarrow (f<b>)$
Status Affected:	None
Encoding:	`0101` `bbbf` `ffff`
Description:	Bit ‘b’ in register ‘f’ is set.
Words:	1
Cycles:	1
<u>Example</u>:	`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:	$0 \le f \le 31$ $0 \le b \le 7$
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 2-cycle instruction.
Words:	1
Cycles:	1(2)
<u>Example</u>:	

```
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:	$0 \le f \le 31$ $0 \le b < 7$
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 2-cycle instruction.
Words:	1
Cycles:	1(2)
<u>Example</u>:	

```
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:	$0 \leq f \leq 31$ $d \in [0,1]$
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:	$0 \leq f \leq 31$ $d \in [0,1]$
Operation:	(f) – 1 $\rightarrow$ (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:	$0 \leq f \leq 31$ $d \in [0,1]$
Operation:	(f) – 1 $\rightarrow$ 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
<u>Example</u>:	`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
<u>Example</u>:	`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)
<u>Example</u>:	

```
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:	$0 \le k \le 255$
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:	$0 \le f \le 31$ $d \in [0,1]$
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:	$0 \le f \le 31$ $d \in [0,1]$
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:	$0 \le k \le 255$
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:	$0 \leq f \leq 31$
Operation:	(W) → (f)
Status Affected:	None
Encoding:	`0000` | `001f` | `ffff`
Description:	Move data from the W register to register ‘f’.
Words:	1
Cycles:	1
<u>Example</u>:	`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
<u>Example</u>:	`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
<u>Example</u>	`OPTION`

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

RETLW	**Return with Literal in W**
Syntax:	[*label*] RETLW k
Operands:	$0 \leq k \leq 255$
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

<u>Example</u>:

```
         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:	$0 \leq f \leq 31$ $d \in [0,1]$
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:	$0 \leq f \leq 31$ $d \in [0,1]$
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	**Enter Sleep 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 ≤ f ≤ 31 d ∈ [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 ≤ f ≤ 31 d ∈ [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 or 7
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 ≤ k ≤ 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 ≤ f ≤ 31
d ∈ [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 PICmicro® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
    - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
    - MPASM™ Assembler
    - MPLAB C17 and MPLAB C18 C Compilers
    - MPLINK™ Object Linker/ MPLIB™ Object Librarian
    - MPLAB C30 C Compiler
    - MPLAB ASM30 Assembler/Linker/Library
- Simulators
    - MPLAB SIM Software Simulator
    - MPLAB dsPIC30 Software Simulator
- Emulators
    - MPLAB ICE 2000 In-Circuit Emulator
    - MPLAB ICE 4000 In-Circuit Emulator
- In-Circuit Debugger
    - MPLAB ICD 2
- Device Programmers
    - PRO MATE® II Universal Device Programmer
    - PICSTART® Plus Development Programmer
    - MPLAB PM3 Device Programmer
- Low-Cost Demonstration Boards
    - PICDEM™ 1 Demonstration Board
    - PICDEM.net™ Demonstration Board
    - PICDEM 2 Plus Demonstration Board
    - PICDEM 3 Demonstration Board
    - PICDEM 4 Demonstration Board
    - PICDEM 17 Demonstration Board
    - PICDEM 18R Demonstration Board
    - PICDEM LIN Demonstration Board
    - PICDEM USB Demonstration Board
- Evaluation Kits
    - KEELOQ®
    - PICDEM MSC
    - microID®
    - CAN
    - PowerSmart®
    - Analog

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® based application that contains:

- An interface to 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
- Mouse over variable inspection
- Extensive on-line help

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PICmicro 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 increasing flexibility and power.

10.2 MPASM Assembler

The MPASM assembler is a full-featured, universal macro assembler for all PICmicro MCUs.

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

The MPASM assembler features include:

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

10.3 MPLAB C17 and MPLAB C18 C Compilers

The MPLAB C17 and MPLAB C18 Code Development Systems are complete ANSI C compilers for Microchip's PIC17CXXX and PIC18CXXX family of microcontrollers. 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 C17 and MPLAB C18 C compilers. 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 C30 C Compiler

The MPLAB C30 C compiler is a full-featured, ANSI compliant, optimizing compiler that translates standard ANSI C programs into dsPIC30F assembly language source. The compiler also supports many command line options and language extensions to take full advantage of the dsPIC30F device hardware capabilities and afford fine control of the compiler code generator.

MPLAB C30 is distributed with a complete ANSI C standard library. All library functions have been validated and conform to the ANSI C library standard. The library includes functions for string manipulation, dynamic memory allocation, data conversion, timekeeping and math functions (trigonometric, exponential and hyperbolic). The compiler provides symbolic information for high-level source debugging with the MPLAB IDE.

10.6 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 compiler uses the assembler to produce it's 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

10.7 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC hosted environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any pin. The execution can be performed in Single-Step, Execute Until Break or Trace mode.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and MPLAB C18 C Compilers, as well as the MPASM assembler. The software simulator offers the flexibility to develop and debug code outside of the laboratory environment, making it an excellent, economical software development tool.

10.8 MPLAB SIM30 Software Simulator

The MPLAB SIM30 software simulator allows code development in a PC hosted environment by simulating the dsPIC30F series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file, or user defined key press, to any of the pins.

The MPLAB SIM30 simulator fully supports symbolic debugging using the MPLAB C30 C Compiler and MPLAB ASM30 assembler. The simulator runs in either a Command Line mode for automated tasks, or from MPLAB IDE. This high-speed simulator is designed to debug, analyze and optimize time intensive DSP routines.

10.9 MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator

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

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

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

10.10 MPLAB ICE 4000 High-Performance Universal In-Circuit Emulator

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

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

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

10.11 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PICmicro MCUs and can be used to develop for these and other PICmicro microcontrollers. 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, CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real-time. MPLAB ICD 2 also serves as a development programmer for selected PICmicro devices.

10.12 PRO MATE II Universal Device Programmer

The PRO MATE II is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features an LCD display for instructions and error messages and a modular detachable socket assembly to support various package types. In Stand-Alone mode, the PRO MATE II device programmer can read, verify and program PICmicro devices without a PC connection. It can also set code protection in this mode.

10.13 MPLAB PM3 Device Programmer

The MPLAB PM3 is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 device programmer can read, verify and program PICmicro devices without a PC connection. It can also set code protection in this mode. MPLAB PM3 connects to the host PC via an RS-232 or USB cable. 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.

10.14 PICSTART Plus Development Programmer

The PICSTART Plus development programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus development programmer supports most PICmicro devices 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.

10.15 PICDEM 1 PICmicro Demonstration Board

The PICDEM 1 demonstration board demonstrates the capabilities of the PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The sample microcontrollers provided with the PICDEM 1 demonstration board can be programmed with a PRO MATE II device programmer or a PICSTART Plus development programmer. The PICDEM 1 demonstration board can be connected to the MPLAB ICE in-circuit emulator for testing. A prototype area extends the circuitry for additional application components. Features include an RS-232 interface, a potentiometer for simulated analog input, push button switches and eight LEDs.

10.16 PICDEM.net Internet/Ethernet Demonstration Board

The PICDEM.net demonstration board is an Internet/ Ethernet demonstration board using the PIC18F452 microcontroller and TCP/IP firmware. The board supports any 40-pin DIP device that conforms to the standard pinout used by the PIC16F877 or PIC18C452. This kit features a user friendly TCP/IP stack, web server with HTML, a 24L256 Serial EEPROM for Xmodem download to web pages into Serial EEPROM, ICSP/MPLAB ICD 2 interface connector, an Ethernet interface, RS-232 interface and a 16 x 2 LCD display. Also included is the book and CD-ROM *“TCP/IP Lean, Web Servers for Embedded Systems,”* by Jeremy Bentham

10.17 PICDEM 2 Plus Demonstration Board

The PICDEM 2 Plus demonstration board supports many 18, 28 and 40-pin microcontrollers, including PIC16F87X and PIC18FXX2 devices. All the necessary hardware and software is included to run the demonstration programs. The sample microcontrollers provided with the PICDEM 2 demonstration board can be programmed with a PRO MATE II device programmer, PICSTART Plus development programmer, or MPLAB ICD 2 with a Universal Programmer Adapter. The MPLAB ICD 2 and MPLAB ICE in-circuit emulators may also be used with the PICDEM 2 demonstration board to test firmware. A prototype area extends the circuitry for additional application components. Some of the features include an RS-232 interface, a 2 x 16 LCD display, a piezo speaker, an on-board temperature sensor, four LEDs and sample PIC18F452 and PIC16F877 Flash microcontrollers.

10.18 PICDEM 3 PIC16C92X Demonstration Board

The PICDEM 3 demonstration board supports the PIC16C923 and PIC16C924 in the PLCC package. All the necessary hardware and software is included to run the demonstration programs.

10.19 PICDEM 4 8/14/18-Pin Demonstration Board

The PICDEM 4 can be used to demonstrate the capabilities of the 8, 14 and 18-pin PIC16XXXX and PIC18XXXX MCUs, including the PIC16F818/819, PIC16F87/88, PIC16F62XA and the PIC18F1320 family of microcontrollers. PICDEM 4 is intended to showcase the many features of these low pin count parts, including LIN and Motor Control using ECCP. Special provisions are made for low-power operation with the supercapacitor circuit and jumpers allow on-board hardware to be disabled to eliminate current draw in this mode. Included on the demo board are provisions for Crystal, RC or Canned Oscillator modes, a five volt regulator for use with a nine volt wall adapter or battery, DB-9 RS-232 interface, ICD connector for programming via ICSP and development with MPLAB ICD 2, 2 x 16 liquid crystal display, PCB footprints for H-Bridge motor driver, LIN transceiver and EEPROM. Also included are: header for expansion, eight LEDs, four potentiometers, three push buttons and a prototyping area. Included with the kit is a PIC16F627A and a PIC18F1320. Tutorial firmware is included along with the User’s Guide.

10.20 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, PIC17C762 and PIC17C766. A programmed sample is included. The PRO MATE II device programmer, or the PICSTART Plus development programmer, can be used to reprogram the device for user tailored application development. The PICDEM 17 demonstration board supports program download and execution from external on-board Flash memory. A generous prototype area is available for user hardware expansion.

10.21 PICDEM 18R PIC18C601/801 Demonstration Board

The PICDEM 18R demonstration board serves to assist development of the PIC18C601/801 family of Microchip microcontrollers. It provides hardware implementation of both 8-bit Multiplexed/Demultiplexed and 16-bit Memory modes. The board includes 2 Mb external Flash memory and 128 Kb SRAM memory, as well as serial EEPROM, allowing access to the wide range of memory types supported by the PIC18C601/801.

10.22 PICDEM LIN PIC16C43X Demonstration Board

The powerful LIN hardware and software kit includes a series of boards and three PICmicro microcontrollers. The small footprint PIC16C432 and PIC16C433 are used as slaves in the LIN communication and feature on-board LIN transceivers. A PIC16F874 Flash microcontroller serves as the master. All three microcontrollers are programmed with firmware to provide LIN bus communication.

10.23 PICkit™ 1 Flash Starter Kit

A complete “development system in a box”, the PICkit Flash Starter Kit includes a convenient multi-section board for programming, evaluation and development of 8/14-pin Flash PIC® microcontrollers. Powered via USB, the board operates under a simple Windows GUI. The PICkit 1 Starter Kit includes the User’s Guide (on CD ROM), PICkit 1 tutorial software and code for various applications. Also included are MPLAB® IDE (Integrated Development Environment) software, software and hardware “Tips 'n Tricks for 8-pin Flash PIC® Microcontrollers” Handbook and a USB interface cable. Supports all current 8/14-pin Flash PIC microcontrollers, as well as many future planned devices.

10.24 PICDEM USB PIC16C7X5 Demonstration Board

The PICDEM USB Demonstration Board shows off the capabilities of the PIC16C745 and PIC16C765 USB microcontrollers. This board provides the basis for future USB products.

10.25 Evaluation and Programming Tools

In addition to the PICDEM series of circuits, Microchip has a line of evaluation kits and demonstration software for these products.

- KEELOQ evaluation and programming tools for Microchip’s HCS Secure Data Products
- CAN developers kit for automotive network applications
- Analog design boards and filter design software
- PowerSmart battery charging evaluation/ calibration kits
- IrDA® development kit
- microID development and rfLab™ development software
- SEEVAL® designer kit for memory evaluation and endurance calculations
- PICDEM MSC demo boards for Switching mode power supply, high-power IR driver, delta sigma ADC and flow rate sensor

Check the Microchip web page and the latest Product Selector Guide for the complete list of demonstration and evaluation kits.

NOTES:

11.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings(†)

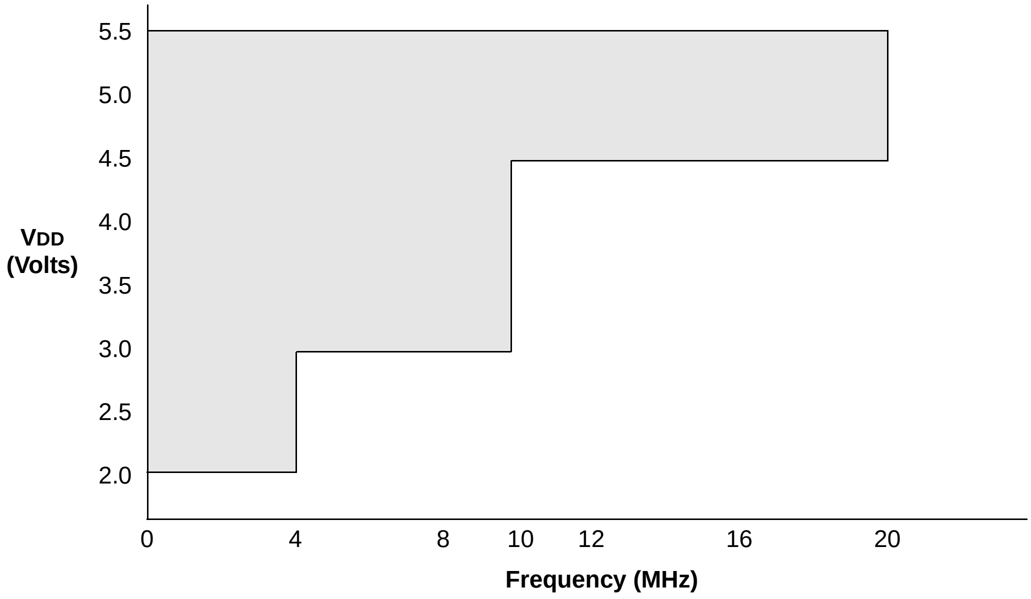
Ambient Temperature under bias	-40°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on V_{DD} with respect to V_{SS}	0V to +6.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS}(1)	0V to +13.5V
Voltage on all other pins with respect to V_{SS}	-0.6V to (V_{DD} + 0.6V)
Total power dissipation(2)	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.

FIGURE 11-1: PIC16F54/57 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: PIC16F54/57 (Industrial)

PIC16F54/57 (Industrial)			**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(3) mode
			—	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, $\overline{\text{MCLR}}$ = 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.

1. TSS

Uppercase letters and their meanings:

SS		
DR	Data Ram	
POR	Power-on Reset	
PD	Power-down	

11.2 DC Characteristics: PIC16F54/57 (Extended)

PIC16F54/57 (Extended)			**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	450	µA	FOSC = 4 MHz, VDD = 2.0V, XT or RC(3) mode
			—	0.4	1.0	mA	FOSC = 10 MHz, VDD = 3.0V, HS mode
			—	1.7	7.0	mA	FOSC = 20 MHz, VDD = 5.0V, HS mode
			—	15	40	µA	FOSC = 32 kHz, VDD = 2.0V, LP mode, WDT disabled
D020	IPD	**Power-down Current**(2)					
			—	1.0	15.0	µA	VDD = 2.0V, WDT enabled
			—	0.5	8.0	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, $\overline{\text{MCLR}}$ = 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: PIC16F54/57

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise specified)** Operating Temperature -40°C ≤ T_A ≤ +85°C for industrial -40°C ≤ T_A ≤ +125°C for extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
	V_{IL}	**Input Low Voltage**					
D030		I/O Ports	V_{SS}	—	0.8V	V	4.5V < V_{DD} ≤ 5.5V
		I/O Ports	V_{SS}	—	0.15 V_{DD}	V	V_{DD} ≤ 4.5V
		$\overline{\text{MCLR}}$ (Schmitt Trigger)	V_{SS}	—	0.15 V_{DD}	V	
		T0CKI (Schmitt Trigger)	V_{SS}	—	0.15 V_{DD}	V	
		OSC1 (Schmitt Trigger)	V_{SS}	—	0.15 V_{DD}	V	RC mode$^{(3)}$
		OSC1	V_{SS}	—	0.3 V_{DD}	V	HS mode
			V_{SS}	—	0.3	V	XT mode
				—	0.3	V	LP mode
	V_{IH}	**Input High Voltage**					
D040		I/O ports	2.0	—	V_{DD}	V	4.5V < V_{DD} ≤ 5.5V
		I/O ports	0.25 V_{DD} + 0.8	—	V_{DD}	V	V_{DD} ≤ 4.5V
		$\overline{\text{MCLR}}$ (Schmitt Trigger)	0.85 V_{DD}	—	V_{DD}	V	
		T0CKI (Schmitt Trigger)	0.85 V_{DD}	—	V_{DD}	V	
		OSC1 (Schmitt Trigger)	0.85 V_{DD}	—	V_{DD}	V	RC mode$^{(3)}$
		OSC1	0.7 V_{DD}	—	V_{DD}	V	HS mode
			1.6	—	V_{DD}	V	XT mode
			1.6	—	V_{DD}	V	LP mode
	I_{IL}	**Input Leakage Current$^{(1, 2)}$**					
D060		I/O ports	—	—	±1.0	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, pin at high-impedance
		$\overline{\text{MCLR}}$	—	—	±5.0	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}
		T0CKI	—	—	±5.0	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}
		OSC1	—	—	±5.0	μA	V_{SS} ≤ V_{PIN} ≤ V_{DD}, XT, HS and LP modes
	V_{OL}	**Output Low Voltage**					
D080		I/O ports	—	—	0.6	V	I_{OL} = 8.5 mA, V_{DD} = 4.5V
D083		OSC2/CLKOUT (RC mode)	—	—	0.6	V	I_{OL} = 1.6 mA, V_{DD} = 4.5V
	V_{OH}	**Output High Voltage$^{(2)}$**					
D090		I/O ports$^{(2)}$	V_{DD} – 0.7	—	—	V	I_{OH} = -3.0 mA, V_{DD} = 4.5V
D092		OSC2/CLKOUT (RC mode)	V_{DD} – 0.7	—	—	V	I_{OH} = -1.3 mA, V_{DD} = 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 $\overline{\text{MCLR}}$/V_{PP} 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 PIC16C5X 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

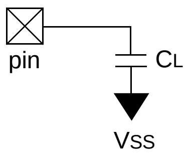
Lowercase letters (pp) and their meanings:

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 – PIC16F54/57



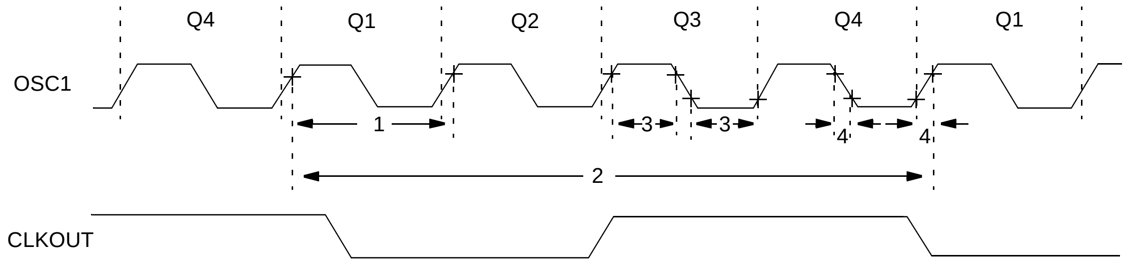
Legend:

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 – PIC16F54/57



Preliminary

TABLE 11-1: EXTERNAL CLOCK TIMING REQUIREMENTS PIC16F54/57

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 Oscillator mode
			DC	—	20	MHz	HS Oscillator mode
			DC	—	200	kHz	LP Oscillator mode
		Oscillator Frequency(1)	DC	—	4.0	MHz	RC Oscillator mode
			0.1	—	4.0	MHz	XT Oscillator mode
			4.0	—	20	MHz	HS Oscillator mode
			5.0	—	200	kHz	LP Oscillator mode
1	TOSC	External CLKIN Period(1)	250	—	—	ns	XT Oscillator mode
			50	—	—	ns	HS Oscillator mode
			5.0	—	—	μs	LP Oscillator mode
		Oscillator Period(1)	250	—	—	ns	RC Oscillator mode
			250	—	10,000	ns	XT Oscillator mode
			50	—	250	ns	HS Oscillator mode
			5.0	—	—	μs	LP Oscillator 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

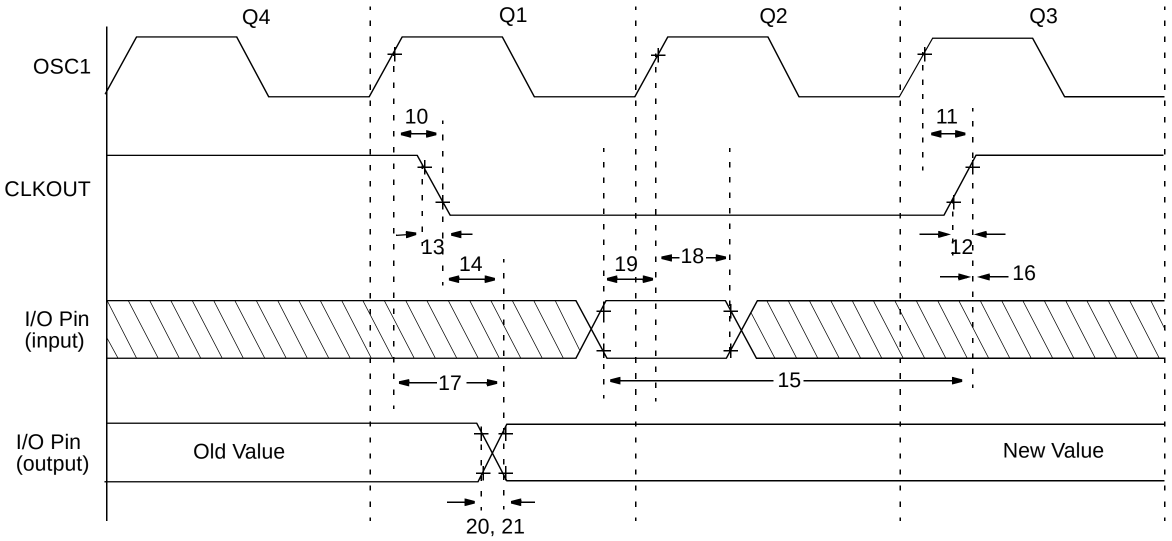
* These parameters are characterized but not tested.

† 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 – PIC16F54/57



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

TABLE 11-2: CLKOUT AND I/O TIMING REQUIREMENTS – PIC16F54/57

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)	—	10	25**	ns
21	TioF	Port output fall time(2)	—	10	25**	ns

* These parameters are characterized but not tested.

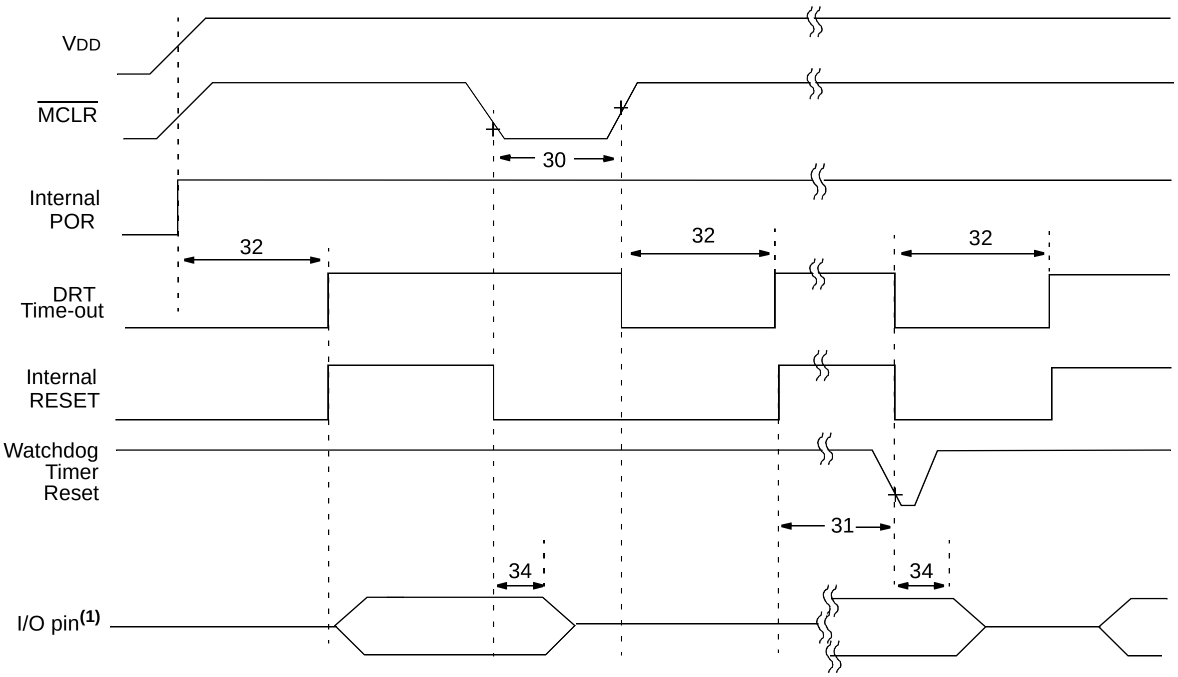
** These parameters are design targets and are not tested. No characterization data available at this time.

† 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.

FIGURE 11-5: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER TIMING – PIC16F54/57



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

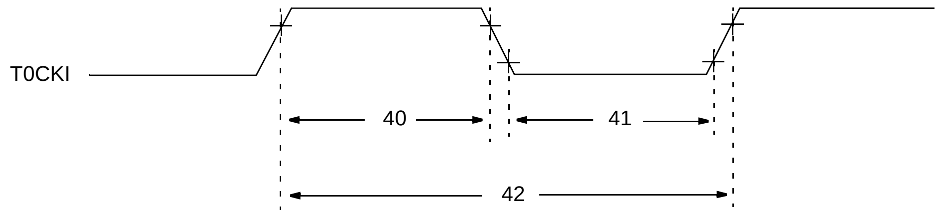
TABLE 11-3: RESET, WATCHDOG TIMER AND DEVICE RESET TIMER – PIC16F54/57

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 – PIC16F54/57



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

TABLE 11-4: TIMER0 CLOCK REQUIREMENTS – PIC16F54/57

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.

12.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

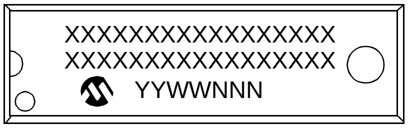
Graphs and Tables are not available at this time.

NOTES:

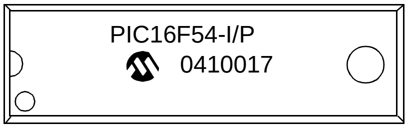
13.0 PACKAGING INFORMATION

13.1 Package Marking Information

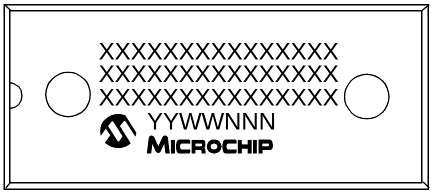
18-Lead PDIP



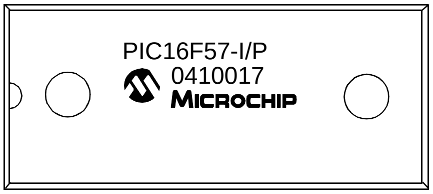
Example



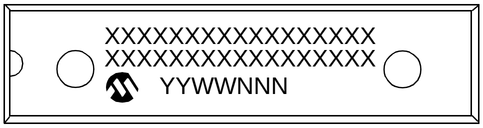
28-Lead PDIP



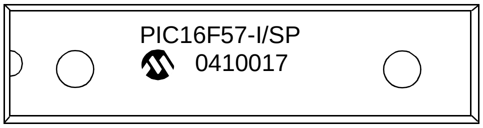
Example



28-Lead SPDIP



Example



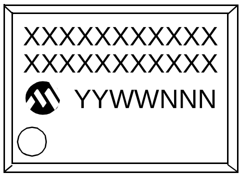
Legend:	XX...X	Customer specific information*
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code

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

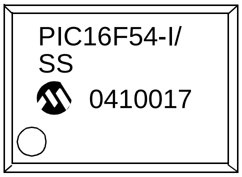
* Standard PICmicro device marking consists of Microchip part number, year code, week code, and traceability code. For PICmicro device marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

13.1 Package Marking Information (Continued)

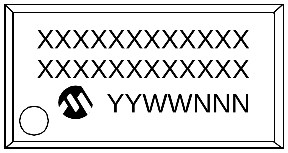
20-Lead SSOP



Example

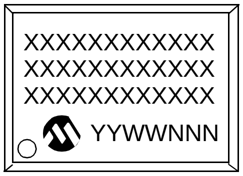


28-Lead SSOP

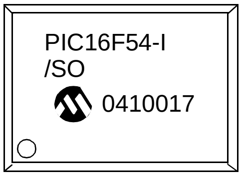


Example

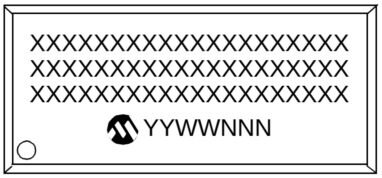
18-Lead SOIC



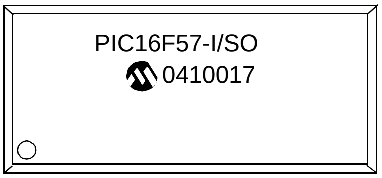
Example



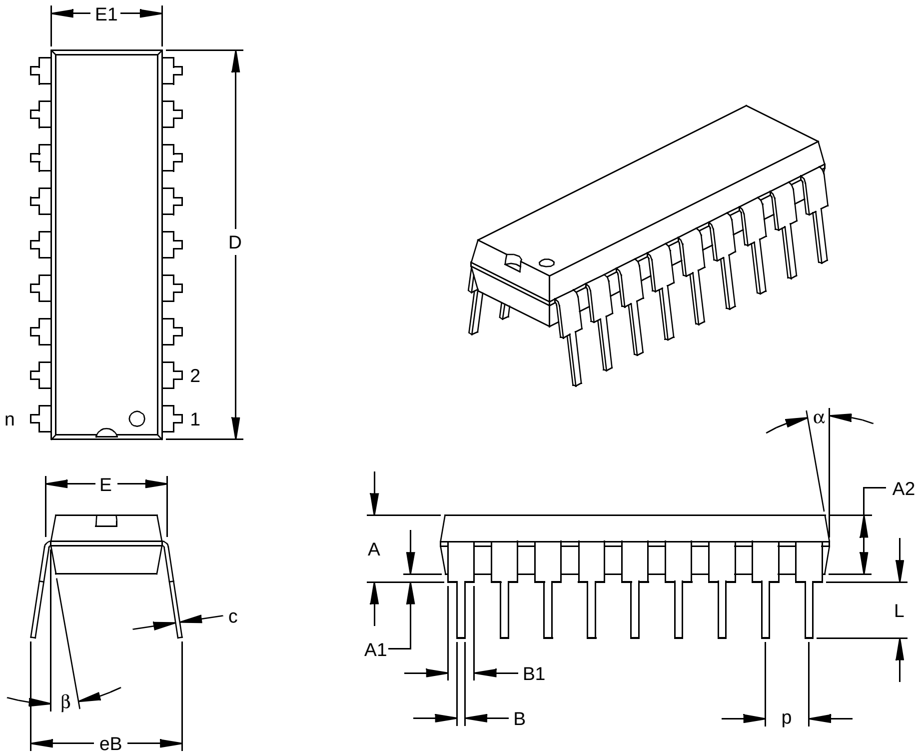
28-Lead SOIC



Example



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

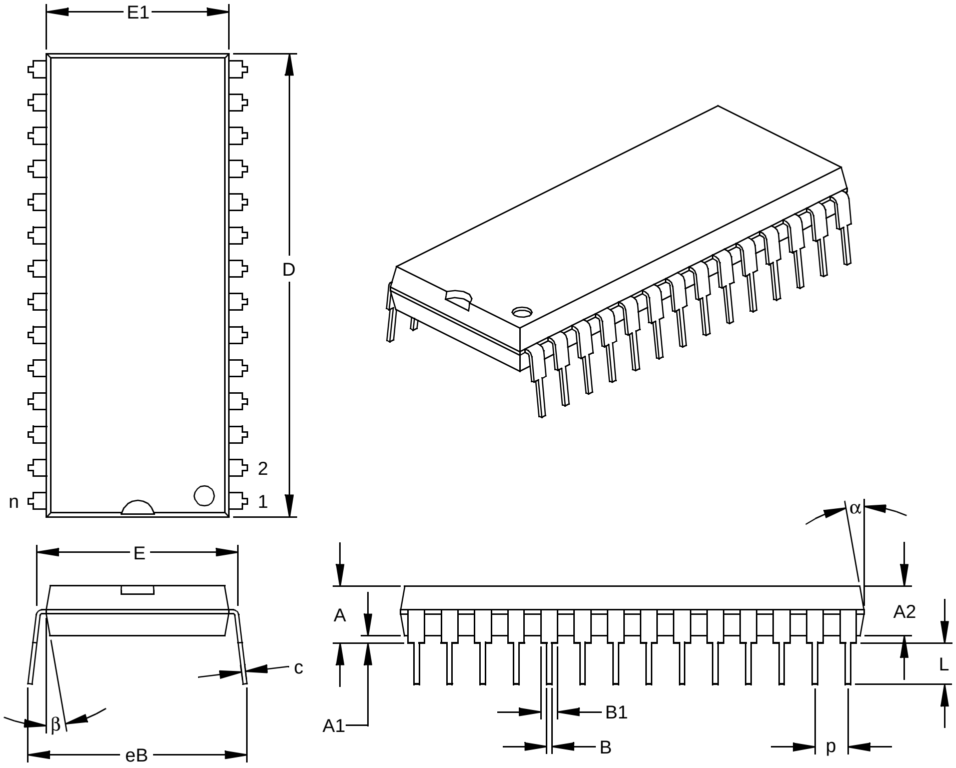


Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.890	.898	.905	22.61	22.80	22.99
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-001
Drawing No. C04-007

28-Lead Plastic Dual In-line (P) – 600 mil Body (PDIP)

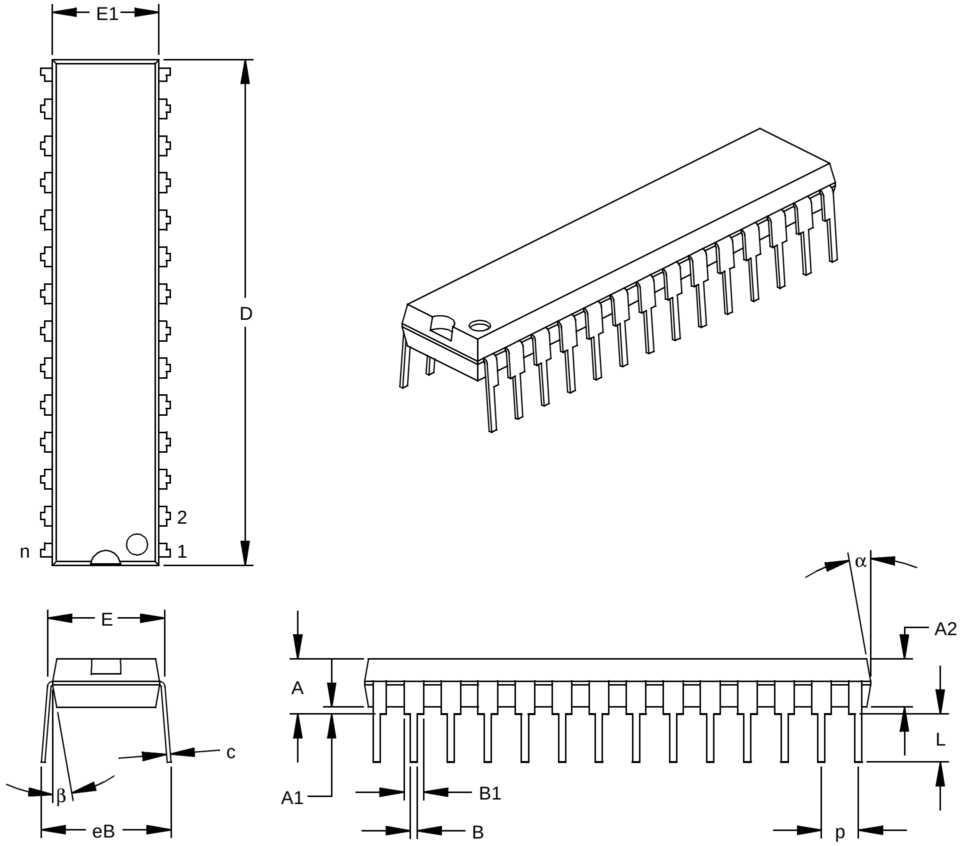


Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.160	.175	.190	4.06	4.45	4.83
Molded Package Thickness	A2	.140	.150	.160	3.56	3.81	4.06
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.595	.600	.625	15.11	15.24	15.88
Molded Package Width	E1	.505	.545	.560	12.83	13.84	14.22
Overall Length	D	1.395	1.430	1.465	35.43	36.32	37.21
Tip to Seating Plane	L	.120	.130	.135	3.05	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.030	.050	.070	0.76	1.27	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing §	eB	.620	.650	.680	15.75	16.51	17.27
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MO-011
Drawing No. C04-079

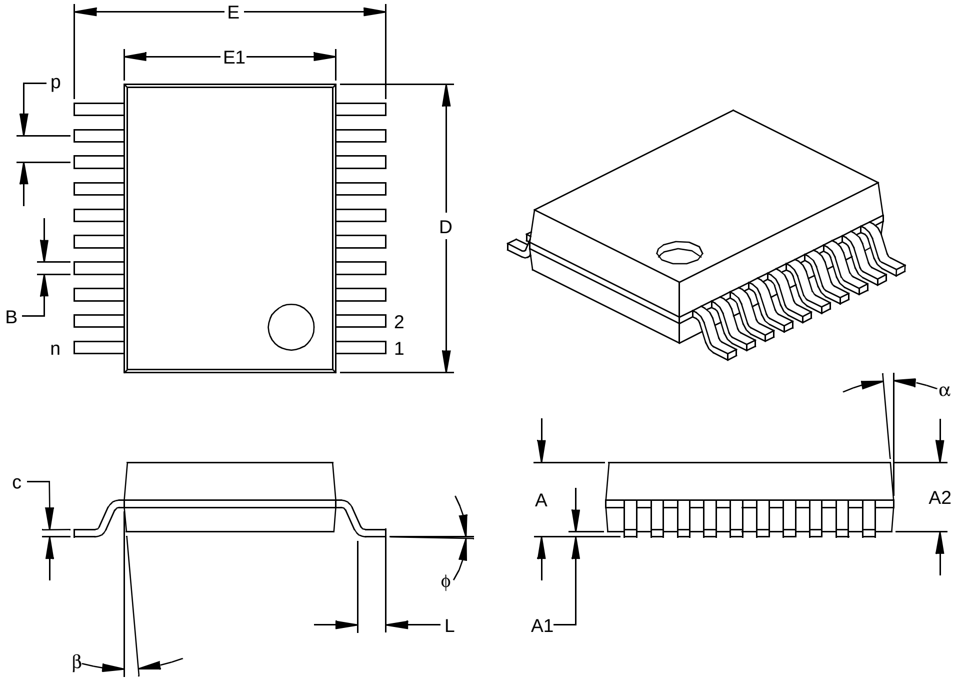
28-Lead Skinny Plastic Dual In-line (SP) – 300 mil Body (PDIP)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.150	.160	3.56	3.81	4.06
Molded Package Thickness	A2	.125	.130	.135	3.18	3.30	3.43
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.310	.325	7.62	7.87	8.26
Molded Package Width	E1	.275	.285	.295	6.99	7.24	7.49
Overall Length	D	1.345	1.365	1.385	34.16	34.67	35.18
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.040	.053	.065	1.02	1.33	1.65
Lower Lead Width	B	.016	.019	.022	0.41	0.48	0.56
Overall Row Spacing §	eB	.320	.350	.430	8.13	8.89	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter
§ Significant Characteristic
Notes:
Dimension D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MO-095
Drawing No. C04-070

20-Lead Plastic Shrink Small Outline (SS) – 209 mil, 5.30 mm (SSOP)

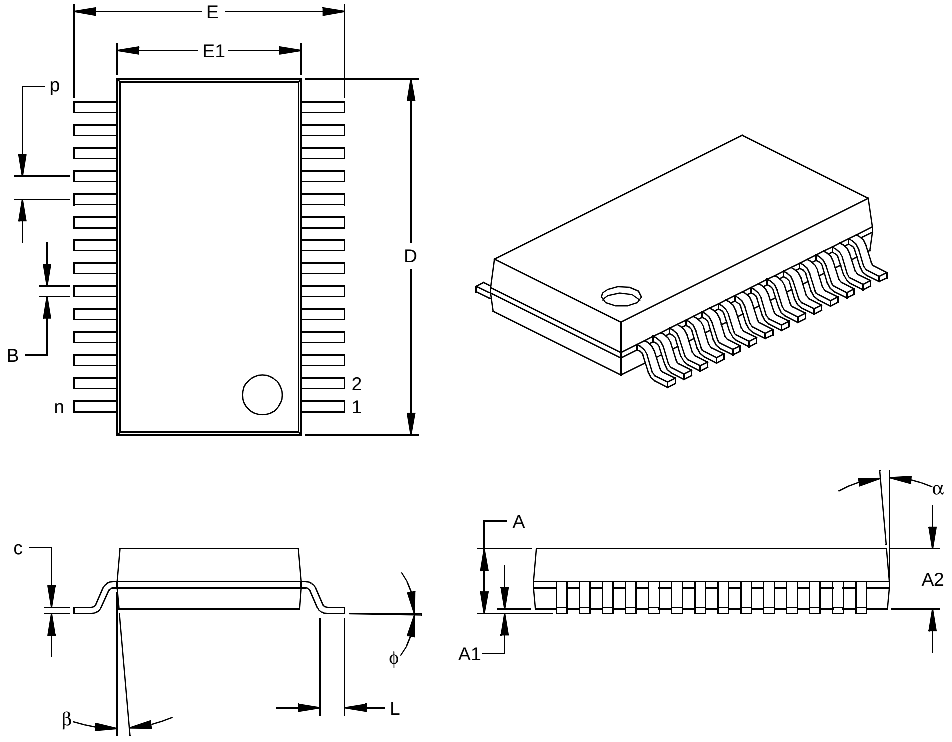


Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		20			20	
Pitch	p		.026			0.65	
Overall Height	A	.068	.073	.078	1.73	1.85	1.98
Molded Package Thickness	A2	.064	.068	.072	1.63	1.73	1.83
Standoff §	A1	.002	.006	.010	0.05	0.15	0.25
Overall Width	E	.299	.309	.322	7.59	7.85	8.18
Molded Package Width	E1	.201	.207	.212	5.11	5.25	5.38
Overall Length	D	.278	.284	.289	7.06	7.20	7.34
Foot Length	L	.022	.030	.037	0.56	0.75	0.94
Lead Thickness	c	.004	.007	.010	0.10	0.18	0.25
Foot Angle	φ	0	4	8	0.00	101.60	203.20
Lead Width	B	.010	.013	.015	0.25	0.32	0.38
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MO-150
Drawing No. C04-072

28-Lead Plastic Shrink Small Outline (SS) – 209 mil, 5.30 mm (SSOP)

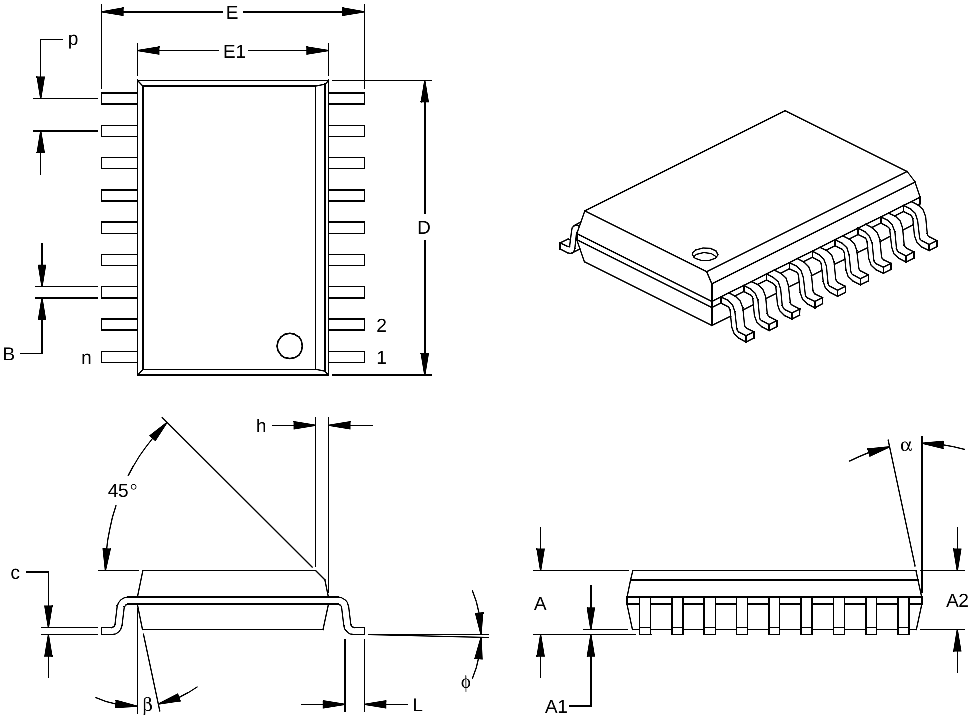


Units		INCHES			MILLIMETERS*		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p		.026			0.65	
Overall Height	A	.068	.073	.078	1.73	1.85	1.98
Molded Package Thickness	A2	.064	.068	.072	1.63	1.73	1.83
Standoff §	A1	.002	.006	.010	0.05	0.15	0.25
Overall Width	E	.299	.309	.319	7.59	7.85	8.10
Molded Package Width	E1	.201	.207	.212	5.11	5.25	5.38
Overall Length	D	.396	.402	.407	10.06	10.20	10.34
Foot Length	L	.022	.030	.037	0.56	0.75	0.94
Lead Thickness	c	.004	.007	.010	0.10	0.18	0.25
Foot Angle	φ	0	4	8	0.00	101.60	203.20
Lead Width	B	.010	.013	.015	0.25	0.32	0.38
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-150
Drawing No. C04-073

18-Lead Plastic Small Outline (SO) – Wide, 300 mil Body (SOIC)

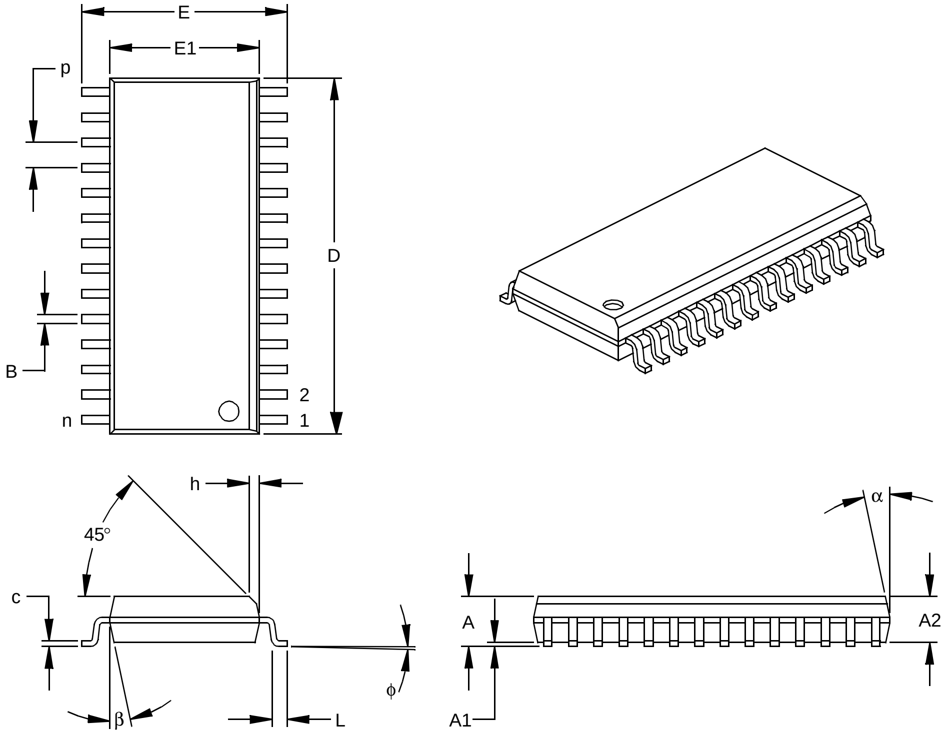


Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.050			1.27	
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff §	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.291	.295	.299	7.39	7.49	7.59
Overall Length	D	.446	.454	.462	11.33	11.53	11.73
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.012	0.23	0.27	0.30
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-013
Drawing No. C04-051

28-Lead Plastic Small Outline (SO) – Wide, 300 mil Body (SOIC)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		28			28	
Pitch	p		.050			1.27	
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff §	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.288	.295	.299	7.32	7.49	7.59
Overall Length	D	.695	.704	.712	17.65	17.87	18.08
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle Top	φ	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.013	0.23	0.28	0.33
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

* Controlling Parameter
§ Significant Characteristic

Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MS-013
Drawing No. C04-052

NOTES:

INDEX

A

- Absolute Maximum Ratings
  - PIC16F54/57 ... 57
- ADDWF ... 41
- ALU ... 7
- ANDLW ... 41
- ANDWF ... 41
- Applications ... 5
- Architectural Overview ... 7
- Assembler
  - MPASM Assembler ... 51

B

- Block Diagram
  - On-Chip Reset Circuit ... 24
  - PIC16F5X Series ... 8
  - Timer0 ... 31
  - TMR0/WDT Prescaler ... 34
  - Watchdog Timer ... 36
- Brown-Out Protection Circuit ... 27
- BSF ... 42
- BTFSC ... 42
- BTFSS ... 42

C

- C Compilers
  - MPLAB C17 ... 52
  - MPLAB C18 ... 52
  - MPLAB C30 ... 52
- CALL ... 18, 43
- Carry (C) bit ... 7, 16, 17
- Clocking Scheme ... 11
- CLRF ... 43
- CLRW ... 43
- CLRWDT ... 43
- Code Protection ... 35, 37
- COMF ... 44
- Configuration Bits ... 35

D

- DC and AC
  - Characteristics Graphs and Tables ... 67
- DC Characteristics
  - PIC16C54/57
    - Extended ... 60
    - Industrial ... 59
  - PIC16F54/57
    - Commercial ... 61
    - Industrial ... 61
- DECF ... 44
- DECFSZ ... 44
- Demonstration Boards
  - PICDEM 1 ... 54
  - PICDEM 17 ... 55
  - PICDEM 18R ... 55
  - PICDEM 2 Plus ... 54
  - PICDEM 3 ... 54
  - PICDEM 4 ... 54
  - PICDEM LIN ... 55
  - PICDEM USB ... 55
  - PICDEM.net Internet/Ethernet ... 54
- Development Support ... 51
- Device Reset Timer (DRT) ... 27
- Digit Carry (DC) bit ... 7, 16, 17
- DRT ... 27

E

- Electrical Specifications
  - PIC16F54/57 ... 57
- Errata ... 3
- Evaluation and Programming Tools ... 55
- External Power-On Reset Circuit ... 25

F

- FSR Register ... 19
  - Value on Reset ... 24

G

- General Purpose Registers
  - Value on Reset ... 24
- GOTO ... 18, 45

H

- High-Performance RISC CPU ... 1

I

- I/O Interfacing ... 29
- I/O Ports ... 29
- I/O Programming Considerations ... 30
- ID Locations ... 35, 37
- INCF ... 45
- INCFSZ ... 45
- INDF Register ... 19
  - Value on Reset ... 24
- Indirect Data Addressing ... 19
- Instruction Cycle ... 11
- Instruction Flow/Pipelining ... 11
- Instruction Set Summary ... 39
- IORLW ... 46
- IORWF ... 46

L

- Loading of PC ... 18

M

- $\overline{\text{MCLR}}$ Reset
  - Register values on ... 24
- Memory Map
  - PIC16F54 ... 13
  - PIC16F57 ... 13
- Memory Organization ... 13
- MOVF ... 46
- MOVLW ... 46
- MOVWF ... 47
- MPLAB ASM30 Assembler, Linker, Librarian ... 52
- MPLAB ICD 2 In-Circuit Debugger ... 53
- MPLAB ICE 2000 High-Performance Universal In-Circuit Emulator ... 53
- MPLAB ICE 4000 High-Performance Universal In-Circuit Emulator ... 53
- MPLAB Integrated Development Environment Software ... 51
- MPLAB PM3 Device Programmer ... 53
- MPLINK Object Linker/MPLIB Object Librarian ... 52

N

- NOP ... 47

O

- OPTION ... 47
- OPTION Register ... 17
  - Value on Reset ... 24
- Oscillator Configurations ... 21
- Oscillator Types
  - HS ... 21
  - LP ... 21
  - RC ... 21
  - XT ... 21

P

- PA0 bit ... 16, 17
- PA1 bit ... 16
- Paging ... 18
- PC ... 18
  - Value on Reset ... 24
- $\overline{\text{PD}}$ bit ... 16, 17, 23
- PICkit 1 Flash Starter Kit ... 55
- PICSTART Plus Development Programmer ... 54
- Pinout Description - PIC16F54 ... 9
- Pinout Description - PIC16F57 ... 10
- PORTA ... 29
  - Value on Reset ... 24
- PORTB ... 29
  - Value on Reset ... 24
- PORTC ... 29
  - Value on Reset ... 24
- Power-Down Mode ... 37
- Power-On Reset (POR) ... 25
- Power-on Reset (POR)
  - Register values on ... 24
- Prescaler ... 33
- PRO MATE II Universal Device Programmer ... 53
- Program Counter ... 18
- Program Memory Organization ... 13
- Program Verification/Code Protection ... 37
- PS0 bit ... 17
- PSA bit ... 17

Q

- Q cycles ... 11

R

- RC Oscillator ... 22
- Read-Modify-Write ... 30
- Register File Map
  - PIC16F54 ... 14
  - PIC16F57 ... 14
- Registers
  - Special Function ... 15
  - Value on Reset ... 24
- Reset ... 23
- Reset on Brown-Out ... 27
- RETLW ... 47
- RLF ... 48
- RRF ... 48

S

- SLEEP ... 48
- Sleep ... 35, 37
- Software Simulator (MPLAB SIM) ... 52
- Software Simulator (MPLAB SIM30) ... 52
- Special Features of the CPU ... 35
- Special Function Registers ... 15
- Stack ... 19
- Status Register
  - Value on Reset ... 24
- Status Register ... 7, 16
- SUBWF ... 49
- SWAPF ... 49

T

- Timer0
  - Switching Prescaler Assignment ... 34
  - Timer0 (TMR0) Module ... 31
  - TMR0 register - Value on Reset ... 24
  - TMR0 with External Clock ... 33
- Timing Diagrams and Specifications
  - PIC16F54/57 ... 62
- Timing Parameter Symbology and Load Conditions
  - PIC16F54/57 ... 62
- $\overline{\text{TO}}$ bit ... 16, 17, 23
- TRIS ... 49
- TRIS Registers ... 29
  - Value on Reset ... 24

W

- W Register
  - Value on Reset ... 24
- Wake-up from SLEEP ... 23
- Wake-up from Sleep ... 37
- Watchdog Timer (WDT) ... 35, 36
  - Period ... 36
  - Programming Considerations ... 36
  - Register values on Reset ... 24
- WWW, On-Line Support ... 3

X

- XORLW ... 50
- XORWF ... 50

Z

- Zero (Z) bit ... 7, 16, 17

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape® or Microsoft® Internet Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available at the following URL:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

ftp://ftp.microchip.com

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

SYSTEMS INFORMATION AND UPGRADE HOT LINE

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive the most current upgrade kits.The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and

1-480-792-7302 for the rest of the world.

042003

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: DS41213B

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 PIC16F54T(1) PIC16F57 PIC16F57T(1)	– V_{DD} range 2.0V to 5.5V – V_{DD} range 2.0V to 5.5V – V_{DD} range 2.0V to 5.5V – V_{DD} range 2.0V to 5.5V
Temperature Range	I E	= -40°C to +85°C (Industrial) = -40°C to +125°C (Extended)
Package	SO SS P SP SOG SSG PG SPG	= SOIC = SSOP = PDIP = Skinny Plastic DIP (SPDIP)(2) = SOIC (Pb-free) = SSOP (Pb-free) = PDIP (Pb-free) = Skinny Plastic DIP (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/SPG = 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

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: 480-792-7627
Web Address: www.microchip.com

Atlanta
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888
Fax: 949-263-1338

San Jose
1300 Terra Bella Avenue
Mountain View, CA 94043
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Unit 706B
Wan Tai Bei Hai Bldg.
No. 6 Chaoyangmen Bei Str.
Beijing, 100027, China
Tel: 86-10-85282100
Fax: 86-10-85282104

China - Chengdu
Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

China - Fuzhou
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

China - Hong Kong SAR
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

China - Shenzhen
Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

China - Shunde
Room 401, Hongjian Building, No. 2
Fengxiangnan Road, Ronggui Town, Shunde
District, Foshan City, Guangdong 528303, China
Tel: 86-757-28395507 Fax: 86-757-28395571

China - Qingdao
Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

India
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-22290061 Fax: 91-80-22290062

Japan
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or
82-2-558-5934

Singapore
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan
Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan
Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Austria
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

France
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy
Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands
Waegenburghtplein 4
NL-5152 JR, Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

United Kingdom
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

05/28/04