



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

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

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



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China





**PIC16F5X**  
**Data Sheet**

Flash-Based, 8-Bit CMOS  
Microcontroller Series

---

---

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

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

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

---

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

**Trademarks**

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, 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, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, PS logo, 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, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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

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

© 2007, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

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

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

## Flash-Based, 8-Bit CMOS Microcontroller Series

### High-Performance RISC CPU:

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

### Special Microcontroller Features:

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

### Low-Power Features:

- Operating Current:
  - 170  $\mu$ A @ 2V, 4 MHz, typical
  - 15  $\mu$ A @ 2V, 32 kHz, typical
- Standby Current:
  - 500 nA @ 2V, typical

### Peripheral Features:

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

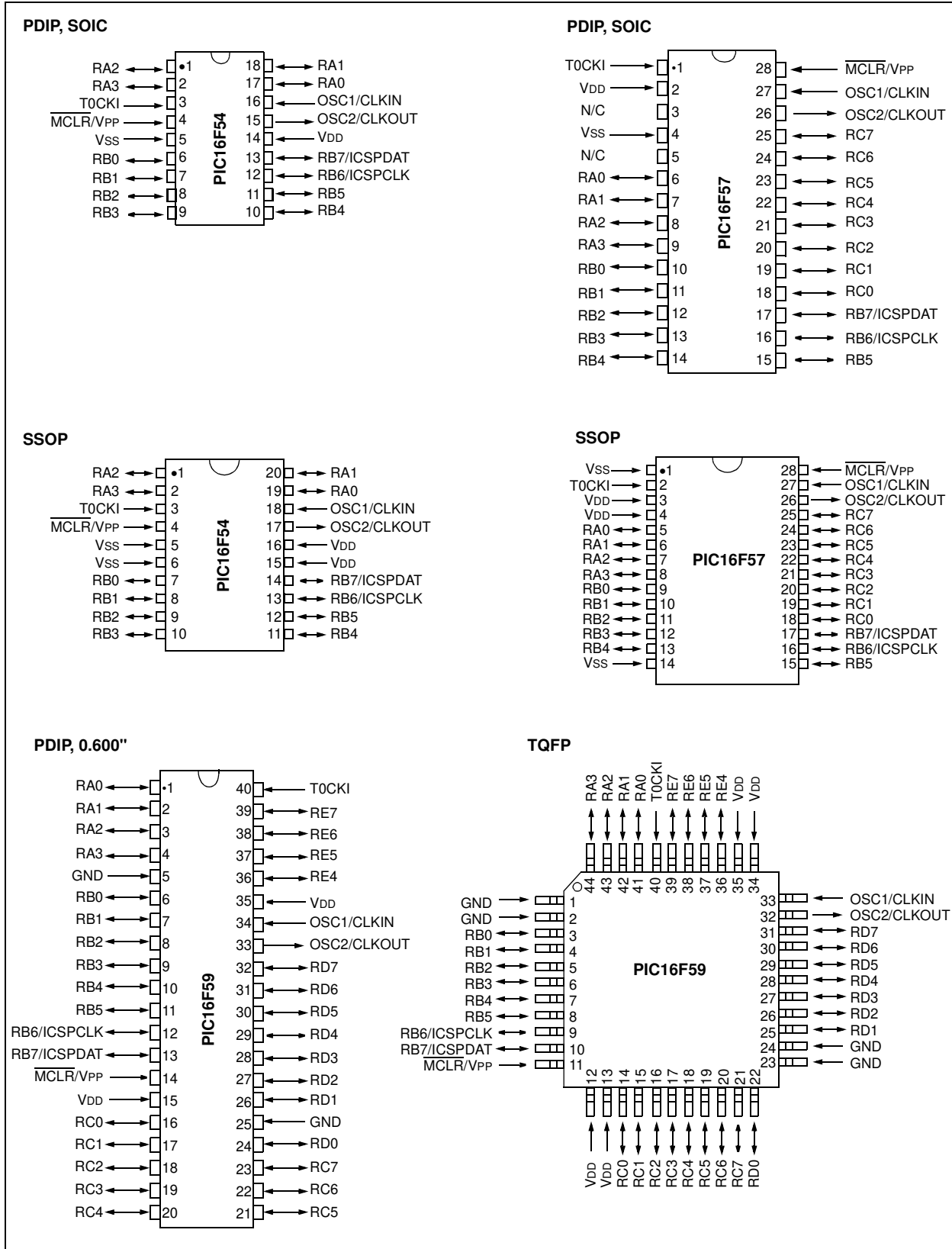
### CMOS Technology:

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

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

# PIC16F5X

## Pin Diagrams



## 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 .....	33
8.0	Special Features of the CPU.....	37
9.0	Instruction Set Summary .....	41
10.0	Development Support.....	53
11.0	Electrical Specifications for PIC16F54/57 .....	57
11.0	Electrical Specifications for PIC16F59 (continued) .....	58
12.0	Packaging Information.....	69
	The Microchip Web Site .....	83
	Customer Change Notification Service .....	83
	Customer Support.....	83
	Reader Response .....	84
	Product Identification System .....	85

## TO OUR VALUED CUSTOMERS

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

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

### Most Current Data Sheet

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

<http://www.microchip.com>

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

### Errata

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

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

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)

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

### Customer Notification System

Register on our web site at [www.microchip.com](http://www.microchip.com) to receive the most current information on all of our products.

# PIC16F5X

---

NOTES:



## 1.0 GENERAL DESCRIPTION

The PIC16F5X from Microchip Technology is a family of low-cost, high-performance, 8-bit, fully static, Flash-based CMOS microcontrollers. It employs a RISC architecture with only 33 single-word/single-cycle instructions. All instructions are single cycle except for program branches which take two cycles. The PIC16F5X delivers performance an order of magnitude higher than its competitors in the same price category. The 12-bit wide instructions are highly symmetrical resulting in 2:1 code compression over other 8-bit microcontrollers in its class. The easy-to-use and easy-to-remember instruction set reduces development time significantly.

The PIC16F5X products are equipped with special features that reduce system cost and power requirements. The Power-on Reset (POR) and Device Reset Timer (DRT) eliminate the need for external Reset circuitry. There are four oscillator configurations to choose from, including the power-saving LP (Low Power) oscillator and cost saving RC oscillator. Power-saving Sleep mode, Watchdog Timer and code protection features improve system cost, power and reliability.

The PIC16F5X products are supported by a full-featured macro assembler, a software simulator, a low-cost development programmer and a full featured programmer. All the tools are supported on IBM® PC and compatible machines.

## 1.1 Applications

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

**TABLE 1-1: PIC16F5X FAMILY OF DEVICES**

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

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



# PIC16F5X

---

NOTES:

## 2.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16F5X family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16F5X uses a Harvard architecture in which program and data are accessed on separate buses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched on the same bus. Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. Instruction opcodes are 12-bits wide, making it possible to have all single-word instructions. A 12-bit wide program memory access bus fetches a 12-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (33) execute in a single cycle except for program branches.

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

The PIC16F5X can directly or indirectly address its register files and data memory. All Special Function Registers (SFR), including the program counter, are mapped in the data memory. The PIC16F5X has a highly orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any Addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC16F5X simple, yet efficient. In addition, the learning curve is reduced significantly.

The PIC16F5X device contains an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

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

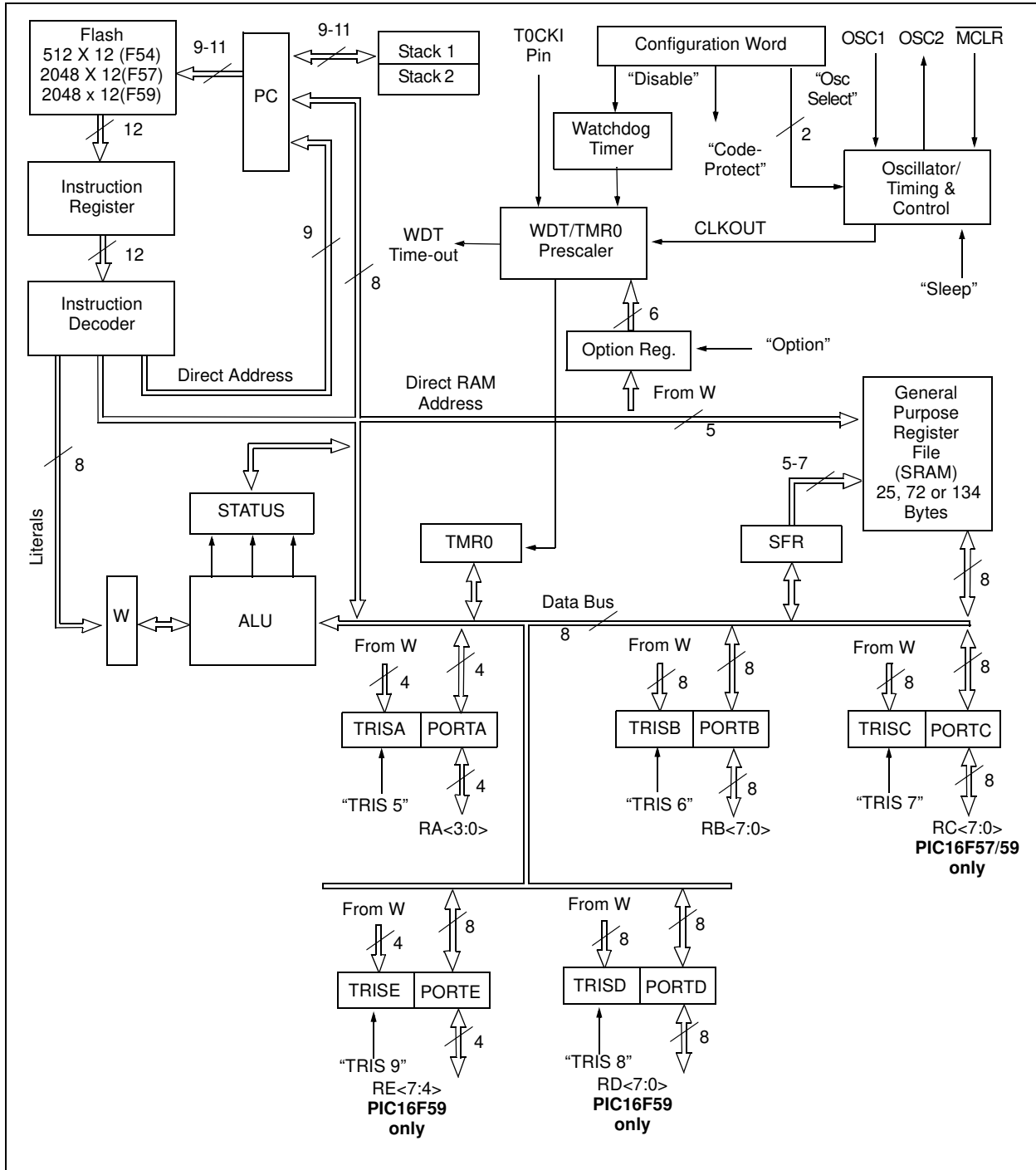
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

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

A simplified block diagram is shown in Figure 2-1 with the corresponding device pins described in Table 2-1 (for PIC16F54), Table 2-2 (for PIC16F57) and Table 2-3 (for PIC16F59).

# PIC16F5X

**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.
$\overline{\text{MCLR}}/\text{VPP}$	$\overline{\text{MCLR}}$	ST	—	Active-low Reset to device. Voltage on the $\overline{\text{MCLR}}/\text{VPP}$ 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    I/O = input/output    CMOS = CMOS output  
 O = output    — = Not Used    XTAL = Crystal input/output  
 ST = Schmitt Trigger input    TTL = TTL input    HV = High Voltage

# PIC16F5X

**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.
MCLR/VPP	MCLR	ST	—	Active-low Reset to device. Voltage on the 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    I/O = input/output    CMOS = CMOS output  
 O = output     — = Not Used     XTAL = Crystal input/output  
 ST = Schmitt Trigger input                                TTL = TTL input     HV = High Voltage

**TABLE 2-3: PIC16F59 PINOUT DESCRIPTION**

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

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

# PIC16F5X

## 2.1 Clocking Scheme/Instruction Cycle

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

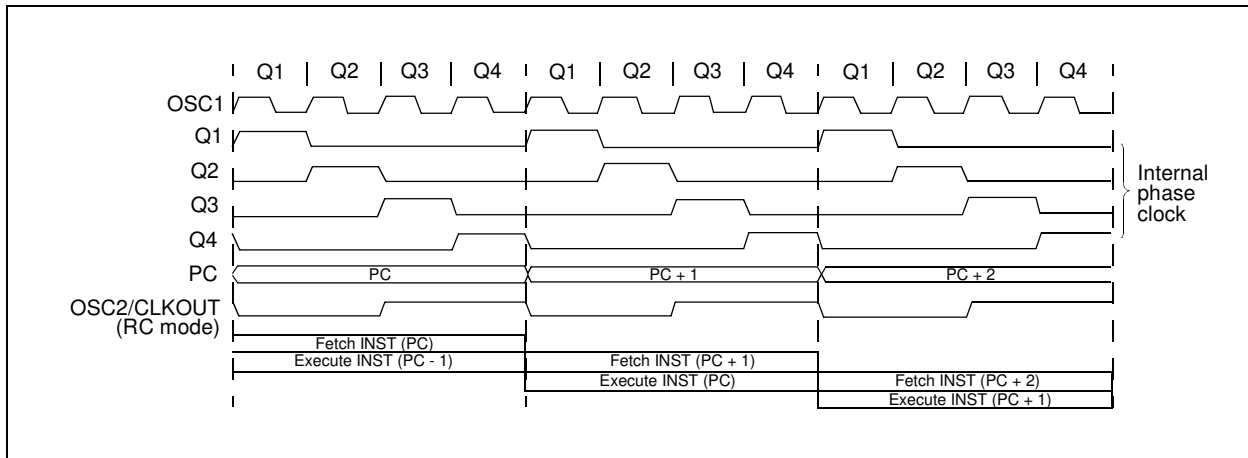
## 2.2 Instruction Flow/Pipelining

An instruction cycle consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle, while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the Program Counter to change (e.g., GOTO), then two cycles are required to complete the instruction (Example 2-1).

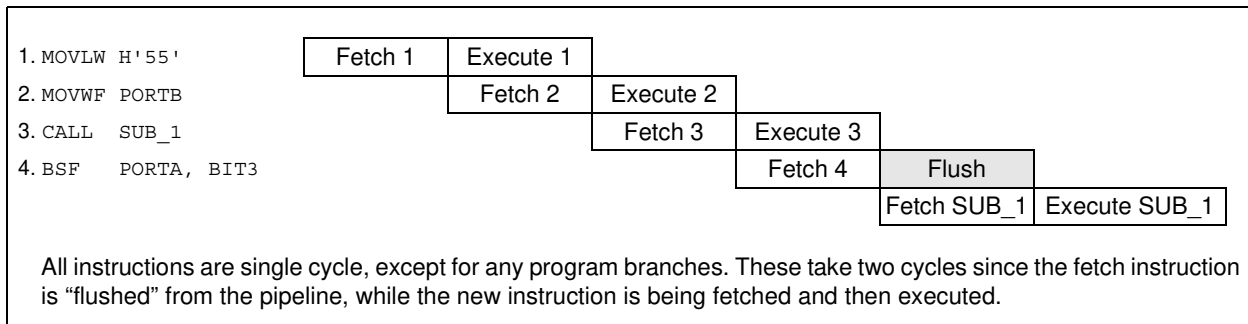
A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the instruction register in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

**FIGURE 2-2: CLOCK/INSTRUCTION CYCLE**



**EXAMPLE 2-1: INSTRUCTION PIPELINE FLOW**





## 3.0 MEMORY ORGANIZATION

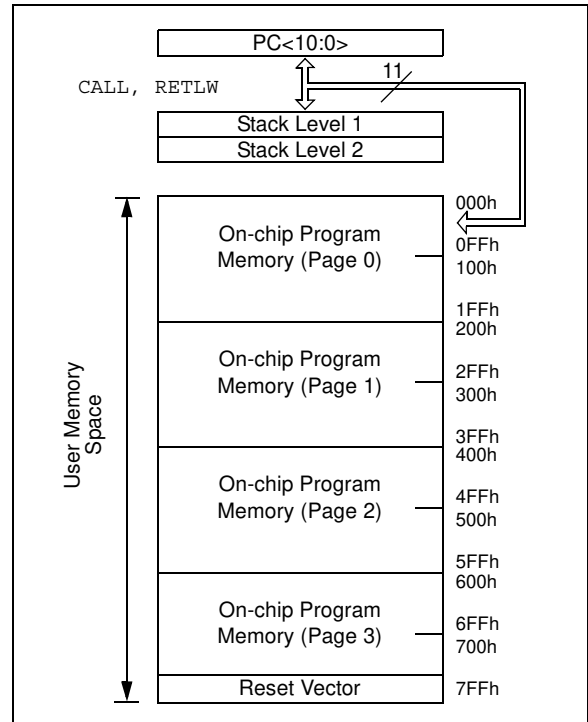
PIC16F5X memory is organized into program memory and data memory. For the PIC16F57 and PIC16F59, which have more than 512 words of program memory, a paging scheme is used. Program memory pages are accessed using one or two STATUS register bits. For the PIC16F57 and PIC16F59, which have a data memory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

### 3.1 Program Memory Organization

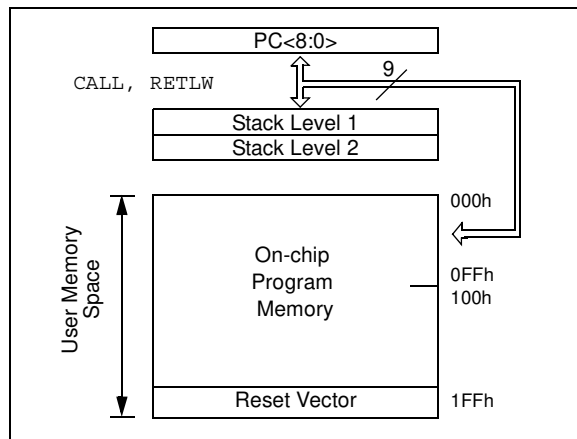
The PIC16F54 has a 9-bit Program Counter (PC) capable of addressing a 512 x 12 program memory space (Figure 3-1). The PIC16F57 and PIC16F59 have an 11-bit Program Counter capable of addressing a 2K x 12 program memory space (Figure 3-2). Accessing a location above the physically implemented address will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

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

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



**FIGURE 3-1: PIC16F54 PROGRAM MEMORY MAP AND STACK**



# PIC16F5X

## 3.2 Data Memory Organization

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

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

The General Purpose Registers are used for data and control information under command of the instructions.

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

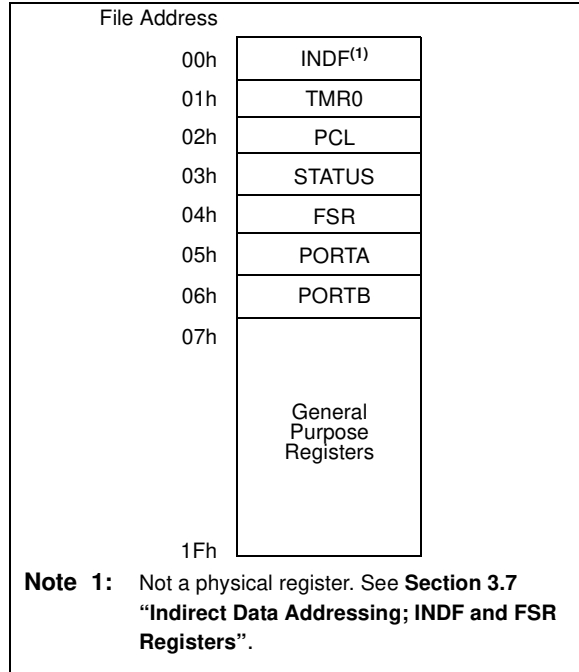
For the PIC16F57, the register file is composed of 8 Special Function Registers, 8 General Purpose Registers and 64 additional General Purpose Registers that may be addressed using a banking scheme (Figure 3-4).

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

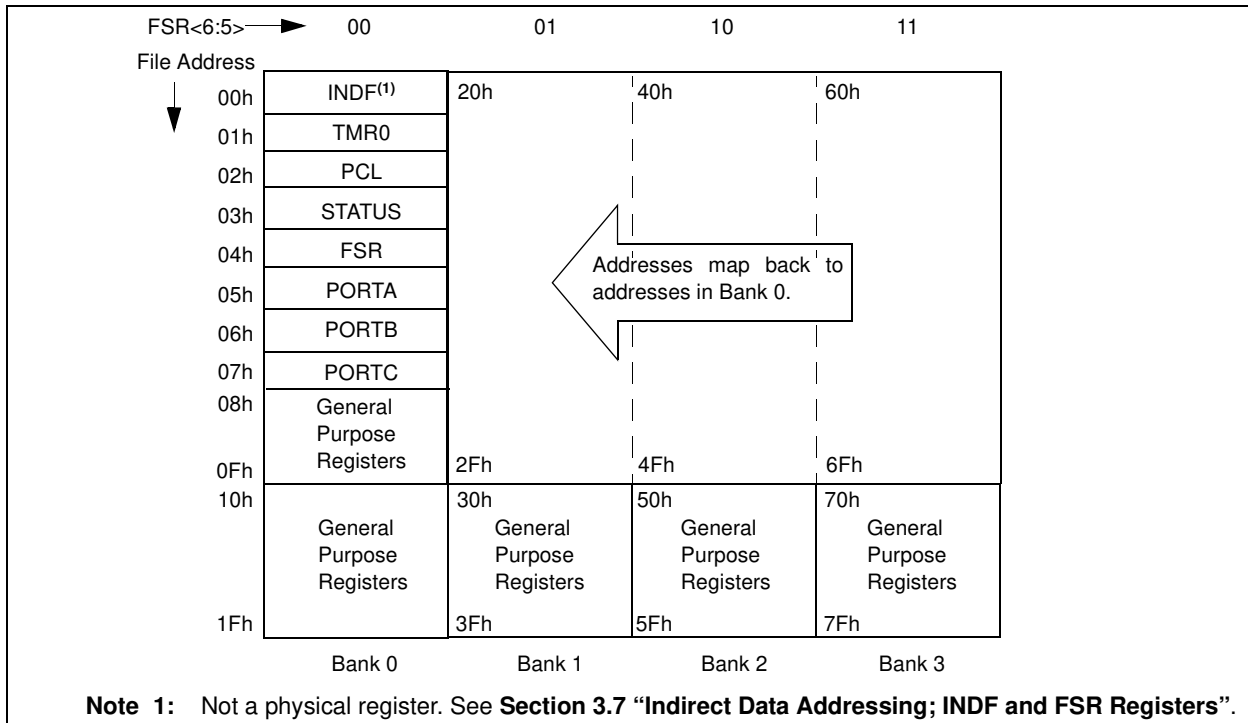
### 3.2.1 GENERAL PURPOSE REGISTER FILE

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

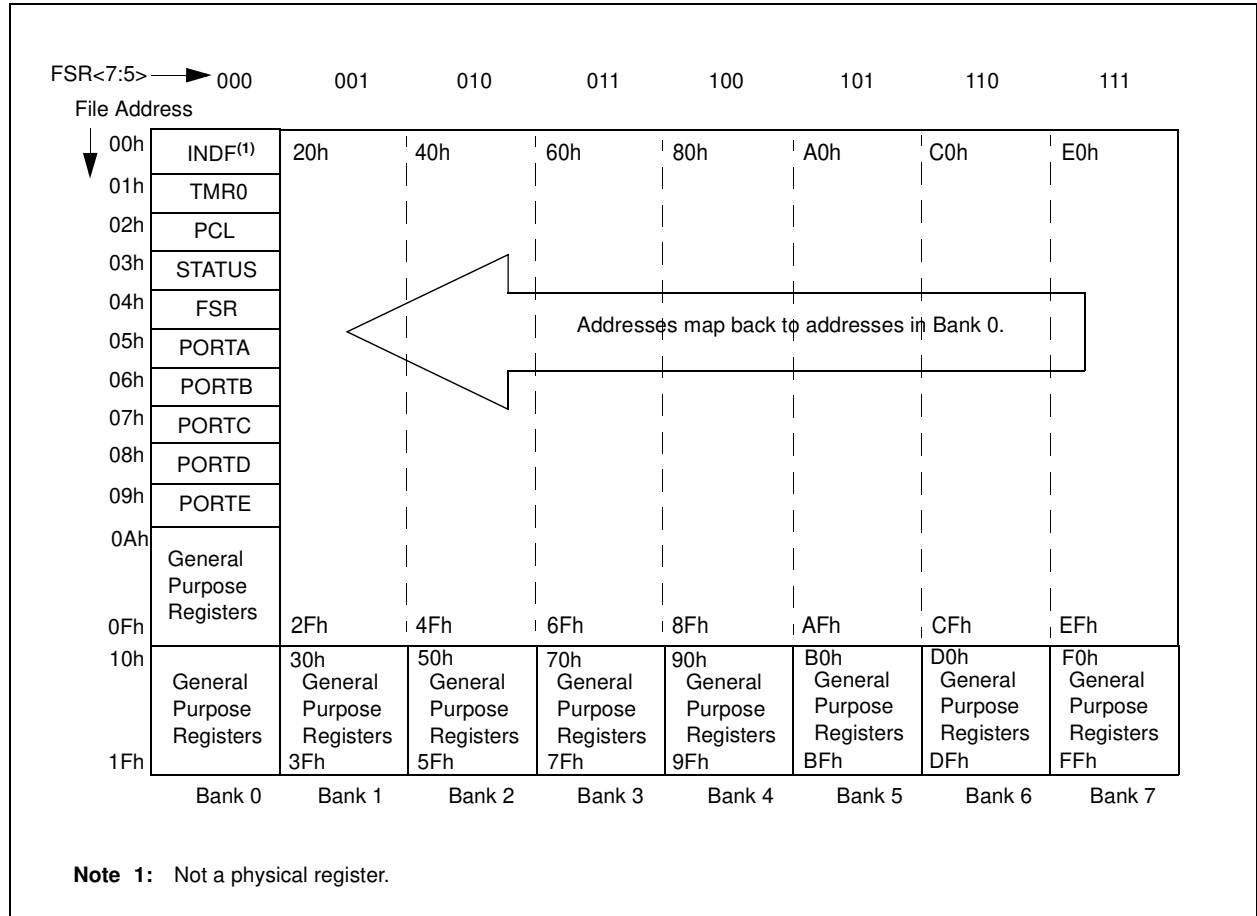
**FIGURE 3-3: PIC16F54 REGISTER FILE MAP**



**FIGURE 3-4: PIC16F57 REGISTER FILE MAP**



**FIGURE 3-5: PIC16F59 REGISTER FILE MAP**



# PIC16F5X

## 3.2.2 SPECIAL FUNCTION REGISTERS

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

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

**TABLE 3-1: SPECIAL FUNCTION REGISTER SUMMARY**

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

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

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

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

**3:** PIC16F54 only.

**4:** PIC16F57 only.

**5:** PIC16F59 only.

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

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

## 3.3 STATUS Register

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

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

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as `000u u1uu` (where `u` = unchanged).

Therefore, it is recommended that only `BCF`, `BSF`, `MOVWF` and `SWAPF` instructions be used to alter the STATUS register because these instructions do not affect the Z, DC or C bits from the STATUS register. For other instructions which do affect Status bits, see **Section 9.0 “Instruction Set Summary”**.

### REGISTER 3-1: STATUS REGISTER (ADDRESS: 03h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x	
PA2	PA1	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	
bit 7								bit 0

- bit 7 **PA2:** Reserved, do not use  
Use of the PA2 bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.
- bit 6-5 **PA<1:0>:** Program Page Preselect bits (PIC16F57/PIC16F59)  
00 = Page 0 (000h-1FFh)  
01 = Page 1 (200h-3FFh)  
10 = Page 2 (400h-5FFh)  
11 = Page 3 (600h-7FFh)  
Each page is 512 words. Using the PA<1:0> bits as general purpose read/write bits in devices which do not use them for program page preselect is not recommended. This may affect upward compatibility with future products.
- bit 4  **$\overline{TO}$ :** Time-Out bit  
1 = After power-up, `CLRWDT` instruction or `SLEEP` instruction  
0 = A WDT time-out occurred
- bit 3  **$\overline{PD}$ :** Power-Down bit  
1 = After power-up or by the `CLRWDT` instruction  
0 = By execution of the `SLEEP` instruction
- bit 2 **Z:** Zero bit  
1 = The result of an arithmetic or logic operation is zero  
0 = The result of an arithmetic or logic operation is not zero
- bit 1 **DC:** Digit Carry/Borrow bit (for `ADDWF` and `SUBWF` instructions)  
**ADDWF**  
1 = A carry to the 4th low order bit of the result occurred  
0 = A carry from the 4th low order bit of the result did not occur  
**SUBWF**  
1 = A borrow to the 4th low order bit of the result did not occur  
0 = A borrow from the 4th low order bit of the result occurred
- bit 0 **C:** Carry/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

# PIC16F5X

## 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

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

## 3.5 Program Counter

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

For a `GOTO` instruction, bits 8:0 of the PC are provided by the `GOTO` instruction word. The PC Latch (PCL) is mapped to PC<7:0> (Figure 3-6 and Figure 3-7).

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

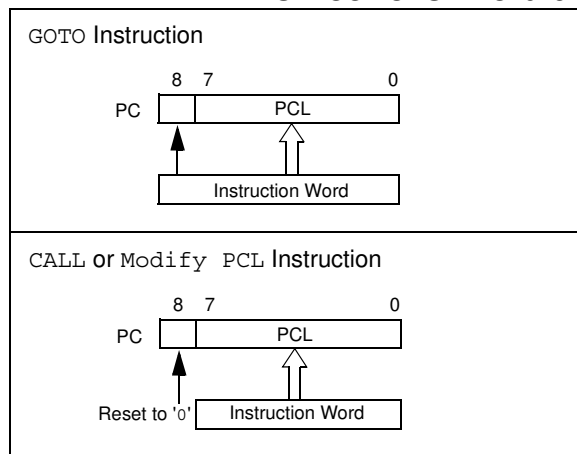
For a `CALL` instruction, or any instruction where the PCL is the destination, bits 7:0 of the PC again are provided by the instruction word. However, PC<8> does not come from the instruction word, but is always cleared (Figure 3-6 and Figure 3-7).

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

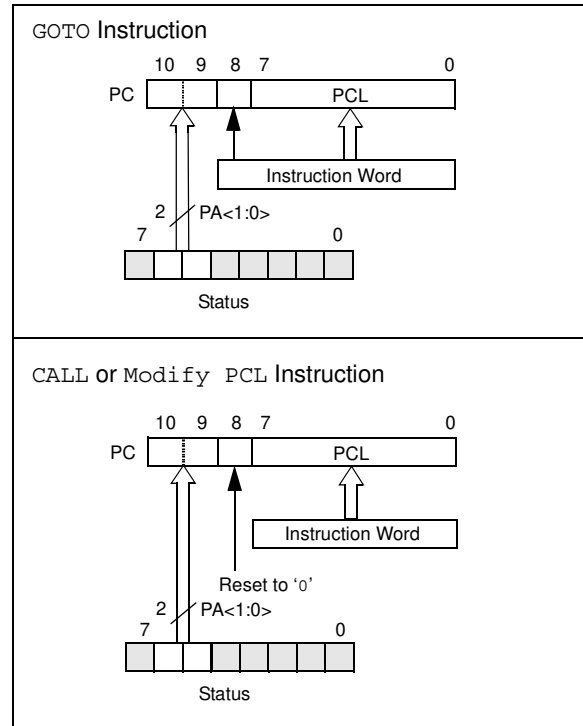
For the PIC16F57 and PIC16F59, a page number again must be supplied. Bit 5 and bit 6 of the STATUS register provide page information to bit 9 and bit 10 of the PC (Figure 3-6 and Figure 3-7).

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

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



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



### 3.5.1 PAGING CONSIDERATIONS PIC16F57 AND PIC16F59

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

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

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

### 3.5.2 EFFECTS OF RESET

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

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

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



# PIC16F5X

## 3.6 Stack

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

A `CALL` instruction will PUSH the current value of stack 1 into stack 2 and then PUSH the current program counter value, incremented by one, into stack level 1. If more than two sequential `CALL`'s are executed, only the most recent two return addresses are stored.

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

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

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

## 3.7 Indirect Data Addressing; INDF and FSR Registers

The INDF register is not a physical register. Addressing INDF actually addresses the register whose address is contained in the FSR Register (FSR is a *pointer*). This is indirect addressing.

### EXAMPLE 3-1: INDIRECT ADDRESSING

- Register file 08 contains the value 10h
- Register file 09 contains the value 0Ah
- Load the value 08 into the FSR register
- A read of the INDF register will return the value of 10h
- Increment the value of the FSR register by one (FSR = 09h)
- A read of the INDF register now will return the value of 0Ah.

Reading INDF itself indirectly (FSR = 0) will produce 00h. Writing to the INDF register indirectly results in a no-operation (although Status bits may be affected).

A simple program to clear RAM locations 10h-1Fh using indirect addressing is shown in Example 3-2.

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

```
        MOVLW  H'10'  ;initialize pointer
        MOVWF  FSR    ;to RAM
NEXT    CLRF   INDF   ;clear INDF Register
        INCF   FSR,F  ;inc pointer
        BTFSC  FSR,4  ;all done?
        GOTO   NEXT   ;NO, clear next
CONTINUE
        :            ;YES, continue
```

The FSR is either a 5-bit (PIC16F54), 7-bit (PIC16F57) or 8-bit (PIC16F59) wide register. It is used in conjunction with the INDF register to indirectly address the data memory area.

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

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

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

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

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

## 4.0 OSCILLATOR CONFIGURATIONS

### 4.1 Oscillator Types

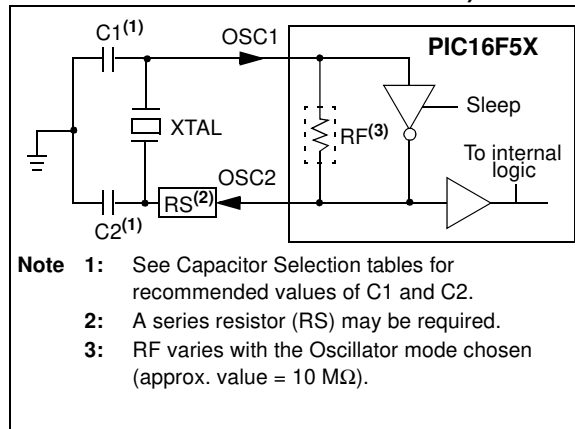
The PIC16F5X devices can be operated in four different oscillator modes. The user can program two Configuration bits (FOSC1:FOSC0) to select one of these four modes:

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

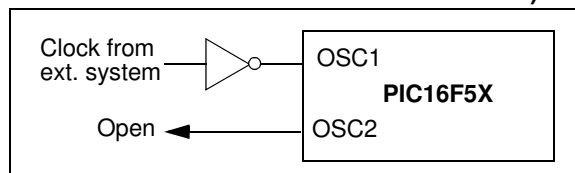
### 4.2 Crystal Oscillator/Ceramic Resonators

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

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



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



**TABLE 4-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS**

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

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

**TABLE 4-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR**

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

**Note 1:** For VDD > 4.5V, C1 = C2 ≈ 30 pF is recommended.

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

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

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

# PIC16F5X

## 4.3 External Crystal Oscillator Circuit

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

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

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

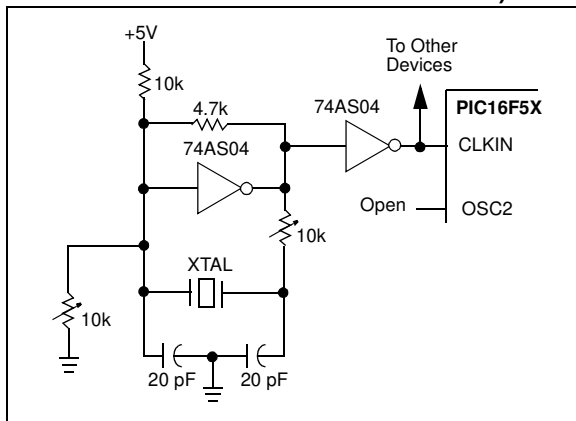
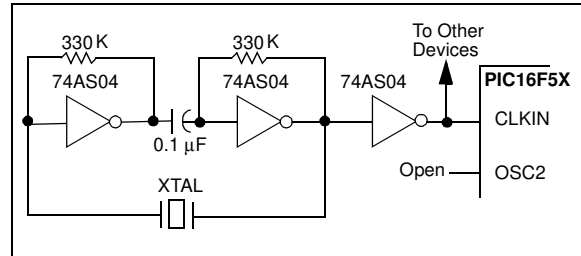


Figure 4-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverters perform a 360° phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear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

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



## 4.4 RC Oscillator

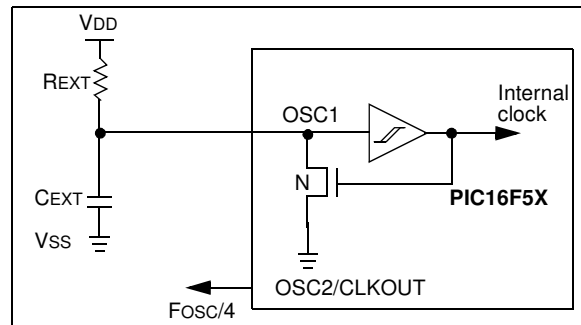
For applications where precise timing is not a requirement, the RC oscillator option is available. The operation and functionality of the RC oscillator is dependent upon a number of variables. The RC oscillator frequency is a function of:

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

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

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

**FIGURE 4-5: RC OSCILLATOR MODE**



## 5.0 RESET

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

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

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

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

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

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

**TABLE 5-1: STATUS BITS AND THEIR SIGNIFICANCE**

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

**Legend:** u = unchanged, x = unknown, — = unimplemented read as ‘0’.

**TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH RESET**

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

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